

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

VIDEO GÖRÜNTÜLERİNDEN İNSAN HAREKETLERİNİN
GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL
OTOMATLAR (GMACA) İLE TANINMASI

SERKAN PELDEK

KOCAELİ 2019

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

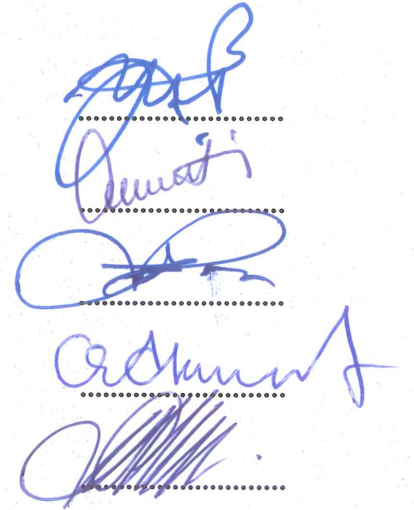
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

VIDEO GÖRÜNTÜLERİNDEN İNSAN HAREKETLERİNİN
GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL
OTOMATLAR (GMACA) İLE TANINMASI

SERKAN PELDEK

Prof. Dr. Yaşar BECERİKLİ
Danışman, Kocaeli Üniversitesi
Prof. Dr. Cemil ÖZ
Jüri Üyesi, Sakarya Üniversitesi
Doç. Dr. Ahmet SAYAR
Jüri Üyesi, Kocaeli Üniversitesi
Dr. Öğr. Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniversitesi
Dr. Öğr. Üyesi Osman Hilmi KOÇAL
Jüri Üyesi, Yalova Üniversitesi



Tezin Savunulduğu Tarih: 15.01.2019

ÖNSÖZ VE TEŞEKKÜR

Tez çalışmalarım süresince destek ve yardımlarını eksik etmeyen tez danışmanım Prof. Dr. Yaşar BECERİKLİ'ye ve tez izleme jürimde yer alan Prof. Dr. Cemil ÖZ ve Doç. Dr. Ahmet SAYAR'a yorum ve yönlendirmeleri için teşekkür ederim.

İyi ve kötü günde her zaman yanımda olan eşim Betül PELDEK'e çok teşekkür ederim. Kızlarım Amine Büşra ve Mahinur; sizlerin farkında olmadan yaptığınız moral desteği olmasaydı bu tez daha zor biterdi.

Mutluluk ve sıkıntılarımı benimle beraber yaşayan, çocukları için maddi manevi fedakârlıktan kaçınmayan annem Sevgi PELDEK ve babam İlhami PELDEK'e minnet ve saygılarımı sunarım.

Ocak – 2019

Serkan PELDEK

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	vi
TABLolar DİZİNİ	xii
SİMGELER VE KISALTMALAR DİZİNİ	xiv
ÖZET.....	xvi
ABSTRACT.....	xvii
GİRİŞ	1
1. HÜCRESEL OTOMATLAR	7
1.1. Hücresel Otomatların Tarihçesi	8
1.2. Hücresel Otomata Kural Nitelenmesi İçin Farklı Yaklaşımlar	14
1.2.1. Lineer / toplamsal hücresel otomatlar	15
1.2.2. Lineer olmayan hücresel otomatlar	16
1.3. Hücresel Otomata Temelleri	16
1.3.1. Toplamlı hücresel otomatlar(totalistic cellular automaton)	28
1.3.2. Sürekli hücresel otomatlar.....	32
1.3.3. Hücresel otomatlar neden “Yeni Bir Bilim Türü” olarak adlandırılıyor?	35
2. LİNEER OLMAYAN HÜCRESEL OTOMATLARIN NİTELENMESİ.....	40
2.1. CA Terminolojisi.....	42
2.2. Kural Min Terimi (Rule Min Term).....	44
2.3. Hücresel Otomata Cezbedicilerinin Nitelenmesi	46
2.3.1. Özellik 1	49
2.3.2. Hücresel otomata kurallarının gruplandırılması.....	50
2.3.3. Kural gruplarının gözlemlenmesi.....	51
2.3.4. Ters döndürülebilir ve ters döndürülemez hücresel otomatlar.....	54
2.4. Tek Uzunluk Döngülü Cezbedicilerin Nitelenmesi	55
2.4.1. Cezbedicilerin nitelenmesi için erişilebilirlik ağacı.....	56
2.4.2. İ'nci ve (i+1)'inci kuralların KMT'leri arasındaki ilişki	57
2.4.3. Erişilebilirlik ağacının oluşturulması	58
2.4.4. Cezbediciler için erişilebilirlik ağacı	63
2.4.5. Algoritma 1: kural vektörünün ürettiği tek uzunluk döngülü cezbedici durum sayısı	66
2.4.6. Algoritma 2: kapsamlı bitlerin belirlenmesi	67
2.4.7. Algoritma 3: istenen sayıda K bit için çoklu cezbedici hücresel otomata sentezi.....	70
2.4.8. KMT dizilimi	71
2.4.9. Özellik 2	74
2.4.10. Çok uzunluk döngülü cezbedicilerin oluşumu.....	76
2.4.11. Tek uzunluk döngülü cezbedici hücresel otomatların sentezi	77

2.4.12. Algoritma 4: belirli kapsayıcı bitlere sahip tek uzunluk döngülü cezbedici hücresel otomatların sentezi.....	81
2.4.13. Belirli kapsayıcı bitlere ve tek uzunluk döngülü cezbediciler için erişilebilirlik ağacı.....	88
2.5. GMACA'nın Sınıflandırıcı Olarak Kullanımı	89
2.6. Cezbedicilerin Sınıflandırıcı Olarak Kullanımı	90
2.7. Veri Setinin Hücresel Otomata Hesaplama Modeline Uygun Hale Getirilmesi.....	91
2.8. İkili Sınıflandırma İçin Deneysel Çalışma	93
2.9. Sonuç.....	98
2.10. GMACA'nın Çoklu Sınıflar İçin Kullanımı	98
2.10.1. Bire karşı hepsi (one-vs-all).....	99
2.10.2. Bire karşı bir (one-vs-one)	101
2.10.3. Cezbedicilerin doğal çoklu sınıf sınıflandırıcı olarak kullanımı.....	102
2.10.4. Çoklu sınıflandırma için deneysel çalışma	103
2.10.5. Görüntü verilerinin GMACA'ya uygun hale getirilmesi.....	103
2.10.6. Deneysel sonuçlar	104
2.10.6.1. Bire Karşı Bir ile elde edilen sonuçlar	105
2.10.6.2. Cezbedicilerin doğal çoklu sınıf sınıflandırıcı olarak kullanımı ile elde edilen sonuçlar	108
3. BİRLEŞTİR/BUL (UNION/FIND) VERİ YAPISI KULLANILARAK GMACA'LARIN KÜMELERE AYRIŞTIRILMASI.....	111
3.1. Birleştir/Bul (Union/Find) Veri Yapısı	112
3.1.1. Hızlı-bul	114
3.1.2. Hızlı-birleştir	114
3.1.3. Ağırlıklı-hızlı-birleştir.....	114
3.1.4. Yol-sıkıştırılmalı-ağırlıklı-hızlı-birleştir	115
3.2. Birleştir/Bul Veri Yapısının GMACA Sınıflarının Ayrıştırılmasında Kullanılması	115
3.3. Gerçek Veriler Üzerinde Test Edilmesi	117
3.4. Sonuç.....	120
4. GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL OTOMATLARIN GÖRÜNTÜ BÖLÜMLEMEDE KULLANILMASI	122
4.1. Motivasyon.....	123
4.2. Benzer Çalışmalar	124
4.3. Cezbedici Kap Üreten Çoklu Cezbedici Hücresel Otomatların Tasarımı.....	132
4.4. Görüntü Bölümleme	137
4.5. Cezbedici Kapların Görüntü Bölümlemede Kullanımı	138
4.5.1. Kullanılan veri seti	142
4.5.2. Deneysel sonuçlar	146
4.6. Yöntemin Performansı	155
4.7. Önerilen Yöntemin Karmaşıklığı	165
4.8. Sonuç.....	165
5. GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL OTOMATLARIN BOYUT AZALTMADA KULLANILMASI	167
5.1. Makine Öğrenme Modeli İçin Verilerin Hazırlanması	167

5.1.1.	Eksik verilerle başa çıkmak	168
5.1.2.	Kategori verilerini işleme.....	169
5.1.3.	Özellik vektöründeki değerlerin aynı aralığa çekilmesi.....	170
5.1.4.	Modelin kullanabileceği kaynak: “özellikler”	171
5.1.5.	Modelin basitleştirilmesi: düzenlileştirme(regularization)	173
5.1.6.	Alt küme seçimi	174
5.1.7.	Uzmanların bilgilerinden faydalanmak.....	176
5.1.8.	Veri setinin eğitim ve test veri setlerine ayrıştırılması.....	176
5.1.9.	Çapraz doğrulama (cross validation)	177
5.2.	Veri Boyutunun Azaltılması.....	178
5.2.1.	Temel bileşen analizi (principal components analysis-pca).....	179
5.3.	En Yakın Komşu (K Nearest Neighbour Knn) Algoritması	181
5.4.	GMACA İle Boyut Azaltma	184
5.5.	Kullanılan Veri tabanı	184
5.5.1.	Görüntü verilerinin GMACA’ya uygun hale getirilmesi.....	184
5.6.	Deneysel Sonuçlar.....	186
6.	GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL OTOMATLAR İLE VİDEO GÖRÜNTÜLERDE HAREKET TESPİTİ.....	189
6.1.	Arka Plan Çıkarma	189
6.2.	Ardışık Görüntülerin Karşılaştırılması.....	190
6.3.	Arka Plan Elde Edilmesi	191
6.4.	Önerilen Yöntem	191
6.4.1.	Ardışık görüntülerin karşılaştırılması	192
6.4.2.	Hareket tespitinden sonra yapılan görüntü işlemleri.....	197
6.4.2.1.	Morfolojik işlemler.....	197
6.5.	Deneysel Sonuçlar.....	200
6.6.	Sonuç	201
7.	İNSAN HAREKETLERİNİN VİDEO GÖRÜNTÜLERDEN TANINMASI ÜZERİNE YAPILAN LİTERATÜR ARAŞTIRMASI.....	203
7.1.	Önemi	204
7.2.	Literatür Değerlendirmesi	205
7.2.1.	Terim anlamları	205
7.2.2.	Hareket tanıma	207
7.2.3.	Görüntülerde hareketli nesne tespiti.....	216
7.2.4.	Hareketli nesnelerin takibi	217
7.2.5.	Özellik çıkarımı.....	218
7.2.6.	Hareketlerin temsili	219
7.2.7.	Hareket bölümlleme	223
7.2.8.	Hareket tanımadada kullanılan yöntemler	224
8.	VİDEO GÖRÜNTÜLERİNDEN İNSAN HAREKETLERİNİN TANINMASI.....	227
8.1.	Geliştirilen Hareket Tanıma Yaklaşımı.....	228
8.1.1.	Görüntülerin gri renge dönüştürülmesi	229
8.1.2.	Hareket tespiti	229
8.1.3.	HOG özelliklerinin çıkarılması	231
8.1.3.1.	Piksel eğilimlerinin hesaplanması	232
8.1.3.2.	Eğilim histogramının elde edilmesi.....	234
8.1.3.3.	Hücrelerden blok oluşturma	239
8.1.3.4.	Eğilim histogramının normalleştirilmesi.....	240

8.1.4.	Çapraz doğrulama (cross validation)	242
8.1.5.	SVM İle hareket tanıma	244
8.1.5.1.	En iyi ayrıştırma	244
8.2.	Deneysel Çalışma.....	247
8.2.1.	Karışıklık matrisi.....	247
8.2.2.	Karışıklık matrisinin oluşturulması.....	248
8.2.3.	Deneysel sonuçlar	250
8.2.4.	Sonuç.....	254
9.	SONUÇLAR VE ÖNERİLER	255
9.1.	Tezin Bilimsel Literatüre Katkıları	255
9.2.	Gelecekte Yapılacak Çalışmalar İçin Öneriler.....	258
	KAYNAKLAR	261
	KİŞİSEL YAYINLAR VE ESERLER	272
	ÖZGEÇMİŞ	274



ŞEKİLLER DİZİNİ

Şekil 1.1.	2B CA komşulukları. a) Neumann komşuluğu b) Moore komşuluğu	9
Şekil 1.2.	Kural 254'e göre hücrelerin bir sonraki durumu	17
Şekil 1.3.	Kural 254'ün 10 adım sonra ürettiği örüntü	19
Şekil 1.4.	Kural 250'ye göre hücrelerin bir sonraki durumu	19
Şekil 1.5.	Kural 250'nin 10 adım sonra ürettiği örüntü	20
Şekil 1.6.	Kural 30'un 50 adım sonra ürettiği örüntü	20
Şekil 1.7.	Kural 90'nın 100 adım sonra ürettiği örüntü	21
Şekil 1.8.	Kural 90'nın 1000 adım sonra ürettiği örüntü	22
Şekil 1.9.	Kural30'un 1500 adım sonra ürettiği örüntü	23
Şekil 1.10.	Kural 31'in 1500 adım sonra ürettiği örüntü	24
Şekil 1.11.	Kural 110'un 1000 adım sonra ürettiği örüntü	25
Şekil 1.12.	Hücrel otomata kurallarının oluşturduğu dört farklı örüntü grubuna ait görüntüler a) Tekrar (Kural250-16 adım sonra) b) İç içeli (Kural90-50 adım) c) Rasgele (Kural30-50 adım sonrası) d) Yerel yapılar (Kural110-150 sonrası)	26
Şekil 1.13.	0-127 arası hücrel otomata kurallarına ait örüntüler	27
Şekil 1.14.	128-255 arası hücrel otomata kurallarına ait örüntüler	28
Şekil 1.15.	3 durumlu tek boyutlu hücrel otomatlarda bütünsel kuralların elde edilmesi	30
Şekil 1.16.	Üçlük tabana göre kural numarası elde edilmesi	30
Şekil 1.17.	Üç renkten oluşan toplam hücrel otomata için örnek örüntüler.	31
Şekil 1.18.	Her hücrenin beyaz (0) ve siyah (1) arasında herhangi bir gri seviyeye sahip olabileceği sürekli bir hücrel otomat. Burada gösterilen kural, her bir hücrenin yeni gri seviyesini, kendi ve yakın komşularının gri değerlerinin ortalamaları olacak şekilde elde edilir.	32
Şekil 1.19.	Biraz daha karmaşık kurallara sahip sürekli bir hücrel otomat. Kural, her bir hücrenin yeni gri seviye değerini, hücrenin ve komşularının ortalama gri seviye değerinin kesirli kısmının $3/2$ ile çarpımından elde edilir. Şekildeki görüntü, tek bir siyah hücreden başlayarak, bu kuralın önemli karmaşıklık davranışları ortaya koyduğunu göstermektedir.	33
Şekil 1.20.	Kuralı bir hücre ve yakın komşularının ortalamasına $1/4$ sabit değerini ekleyen ve sonucun kesirli kısmını alan sürekli bir hücrel otomat. Arka plan her 4 basamağı basitçe tekrarlar, ancak ana desen karmaşıktır ve pek çok açıdan rastgeledir.	34
Şekil 1.21.	Yukarıdaki şekillerdeki resimlerde olduğu gibi aynı kurallara sahip, ancak farklı sabitler eklenerek oluşturulan sürekli hücrel otomatlar. Buradaki sabit değer iki durumlu otomatlarda davranış biçimini belirleyen ikili dizilerin çok küçük olduğunu unutmamak gerekir.	34

Şekil 2.1.	Periyodik sınırlı hücrel otomatin 1'inci ve sonuncu hücrelerinin komşulukları	41
Şekil 2.2.	Sıfır sınırlı hücrel otomatların 1'inci ve sonuncu hücrelerinin komşulukları.....	41
Şekil 2.3.	CA düzlemi flip-flop devrelerden oluşan bir düzlem olarak düşünülebilir.....	42
Şekil 2.4.	1011 örüntüsünün CA<142, 156, 244, 12> kural vektörüne göre bir sonraki durumunun elde edilmesi	45
Şekil 2.5.	CA<105, 129, 171, 65> kural vektörünün ürettiği durum geçiş diyagramları. Kural vektörü tek uzunluk döngülü ve çok uzunluk döngülü cezbediciler üretmektedir.	48
Şekil 2.6.	CA<140, 140, 140, 140> kural vektörünün ürettiği durum geçiş diyagramı	52
Şekil 2.7.	Grup7'deki kurallarda oluşan CA<200,205,206,220> kural vektörü sadece tek uzunluk döngülü cezbediciler üretir	52
Şekil 2.8.	Grup4'deki kurallarda oluşan CA<250,250,250,250> kural vektörü tek uzunluk döngülü cezbediciler ve çok uzunluk döngülü cezbediciler oluşturur	53
Şekil 2.9.	Kural 51'i içeren CA<200,205,206,51> kural vektörü sadece tek uzunluk döngülü cezbediciler ve çok uzunluk döngülü cezbediciler oluşturur	53
Şekil 2.10.	Ters döndürülebilir CA <195,105,105,195> durum geçiş diyagramı.....	54
Şekil 2.11.	CA<78, 142, 201, 92> kural vektörüne ait erişilebilirlik ağacı.....	59
Şekil 2.12.	CA<78, 142, 201, 92> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı.....	64
Şekil 2.13.	Algoritma 1: N uzunluklu CA kural vektörünün ürettiği tek uzunluk döngülü cezbedici durum sayısını veren algoritma	67
Şekil 2.14.	CA<10, 69, 204, 68> kural vektörüne ait durum geçiş diyagramı.....	68
Şekil 2.15.	Algoritma 2: N uzunluklu CA kural vektörünün ürettiği tek uzunluk döngülü cezbedici durum sayısını temsil edebilen belirleyici bitleri bulun algoritma (Das ve ark., 2009).....	68
Şekil 2.16.	CA< 10, 69, 204, 68 > kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı.....	69
Şekil 2.17.	Algoritma 3: İstenen sayıda kapsamlı bit üreten kural vektörünü çıktı olarak veren çoklu cezbedici hücrel otomata sentez algoritması	71
Şekil 2.18.	CA<165, 192, 216, 60> kural vektörünün durum geçiş diyagramı.....	80
Şekil 2.19.	CA<165, 192, 216, 60> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı.....	80
Şekil 2.20.	Algoritma 4: Çok uzunluk döngülü cezbedici oluşumuna izin vermeden, istenen sayıda kapsamlı bit üreten kural vektörünü çıktı olarak veren çoklu cezbedici hücrel otomata sentez algoritması.....	84
Şekil 2.21.	CA<8, 187, 255, 205, 0, 102> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı.....	87

Şekil 2.22.	CA<8, 187, 255, 205, 0, 102> kural vektörünün durum geçiş diyagramı	88
Şekil 2.23.	Cezbedicilerin sınıflandırıcı olarak gösterimi	90
Şekil 2.24.	Monks veri setindeki kategorik değerler ikilik tabana dönüştürülmüştür a)Orijinal veri seti b)Dönüştürülmüş veri	92
Şekil 2.25	Pima Indian veri setindeki kategorik ve normalleştirme işleminden sonra numerik değerler ikilik tabana dönüştürülmüştür a) Orijinal veri seti b) Dönüştürülmüş veri	93
Şekil 2.26.	Monk1 veri setine ait sınıflandırma raporu	94
Şekil 2.27.	Monk2 veri setine ait sınıflandırma raporu	94
Şekil 2.28.	Monk3 veri setine ait sınıflandırma raporu	95
Şekil 2.29.	Pima Indian veri setine ait sınıflandırma raporu	95
Şekil 2.30.	Bire Karşı Hepsi yaklaşımında her bir sınıfı diğer sınıflardan ayırtıran N tane ağırlık vektörü elde edilir.	99
Şekil 2.31.	Bire Karşı Bir yaklaşımında her bir sınıfı diğer başka sınıftan ayırtıran N(N-1)/2 tane ağırlık vektörleri elde edilir.	101
Şekil 2.32.	El yazması rakamlar veri setindeki orijinal görüntü örneği	103
Şekil 2.33.	Elyazması veri setinin GMACA'ya uygun hale getirilmesi. Her bir satır bir rakama aittir.	104
Şekil 2.34.	Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Bire Karşı Bir çoklu sınıflandırma yaklaşımı ile elde edilen karışıklık matrisi.....	106
Şekil 2.35.	Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Bire Karşı Bir çoklu sınıflandırma yaklaşımı ile elde edilen normalleştirilmiş karışıklık matrisi	106
Şekil 2.36.	Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Bire Karşı Bir çoklu sınıflandırma yaklaşımıyla elde edilen sınıflandırma raporu	107
Şekil 2.37.	Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Doğal çoklu sınıflandırma yaklaşımı ile elde edilen karışıklık matrisi	108
Şekil 2.38.	Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Doğal çoklu sınıflandırma yaklaşımı ile elde edilen normalleştirilmiş karışıklık matrisi	109
Şekil 2.39.	Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Doğal çoklu sınıflandırma yaklaşımıyla elde edilen sınıflandırma raporu.....	109
Şekil 3.1.	Birleştir/Bul veri yapısı ile GMACA kümelerinin ayrıştırılmasına ait sözde kod.	116
Şekil 3.2.	Yukarıdaki görüntülerin ilk resmi Yale Face Database 'den elde edilen orijinal görüntüdür ve takip eden görüntüler farklı kural vektörlerine göre elde edilen sonuçlardır. Her bir resmin altında uygulanan kural vektörü yer almaktadır	118
Şekil 3.3.	Farklı kural vektörlerine göre elde edilen sonuçlar. Her bir görüntünün altında uygulanan kural vektörü yer almaktadır.	118
Şekil 3.4.	Farklı kural vektörlerine göre elde edilen sonuçlar. Her bir görüntünün altında uygulanan kural vektörü yer almaktadır	119
Şekil 4.1.	Rosin'in simetri ve yansıyanı aynı olan kuralları çıkardıktan sonra elde ettiği 51 kural.	127

Şekil 4.2.	CA <105, 129, 171, 65> kural vektörünün ürettiği durum geçiş diyagramları	134
Şekil 4.3.	Algoritma 4: Çok uzunluk döngülü cezbedici oluşumuna izin vermeden, istenen sayıda kapsamlı bit üreten kural vektörünü çıkıktı olarak veren çoklu cezbedici hücresel otomata sentez algoritması	136
Şekil 4.4.	GMACA tabanlı görüntü bölümlenmenin iş akış diyagramı	139
Şekil 4.5.	GMACA tabanlı görüntü bölümlendirmeye ait sözde kod	140
Şekil 4.6.	Berkeley Bölümlendirme Veri Setinde yer alan 0-100 arası eğitim görüntüleri.	143
Şekil 4.7.	Berkeley Bölümlendirme Veri Setinde yer alan 100-200 arası eğitim görüntüleri.	144
Şekil 4.8.	Berkeley Bölümlendirme Veri setinden yer alan 200-300 arası test görüntüleri.....	145
Şekil 4.9.	Orijinal görüntü ve farklı kural vektörüne göre bölümlenmiş görüntüler	147
Şekil 4.10.	Orijinal görüntü ve farklı kural vektörüne göre bölümlenmiş görüntüler	148
Şekil 4.11.	16 cezbediciye göre elde edilen bölümlendirilmiş görüntüler	149
Şekil 4.12.	8 cezbediciye göre elde edilen bölümlendirilmiş görüntüler	150
Şekil 4.13.	4 cezbediciye göre elde edilen bölümlendirilmiş görüntüler	151
Şekil 4.14.	2 cezbediciye göre elde edilen bölümlendirilmiş görüntüler	152
Şekil 4.15.	2 cezbediciye göre elde edilen bölümlendirilmiş görüntülerin siyah beyaz renklerle görünümü	153
Şekil 4.16.	En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş görüntüler, en sağda geliştirilen yöntemle elde edilen sonuç yer almaktadır.	159
Şekil 4.17.	En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş görüntüler, en sağda geliştirilen yöntemle elde edilen sonuç yer almaktadır	160
Şekil 4.18.	En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş renkli görüntüler, en sağda geliştirilen yöntemle elde edilen renklendirilmiş sonuç yer almaktadır.....	161
Şekil 4.19.	En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş renkli görüntüler, en sağda geliştirilen yöntemle elde edilen renklendirilmiş sonuç yer almaktadır.....	162
Şekil 4.20.	En solda orijinal görüntü, devamında aynı görüntüler üzerinde elde edilen en iyi üç yönteme ait sonuçlar ve en solda geliştirilen yöntemle elde edilen sonuç yer almaktadır.	163
Şekil 4.21.	En solda orijinal görüntü, devamında aynı görüntüler üzerinde elde edilen en iyi üç yönteme ait sonuçlar ve geliştirilen yöntemle elde edilen sonuç yer almaktadır.	164
Şekil 5.1.	PCA temel bileşenleri.....	180
Şekil 5.2.	El yazması rakamlar veri setindeki görüntü örneği.....	185
Şekil 5.3.	64 olan özellik vektörünün 16'ya düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklere göre elde edilen tanıma oranları	187

Şekil 5.4.	64 olan özellik vektörünün 8'e düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklere göre elde edilen tanıma oranları	188
Şekil 6.1	Önerilen yöntemin blok diyagramı.....	192
Şekil 6.2.	Ardışık görüntü karelerinden t zamanındaki hareketin tespiti.	195
Şekil 6.3.	Orijinal görüntü ve, 5x5 elamanın erozyon uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Erozyon sonucu	198
Şekil 6.4	Orijinal görüntü ve 5x5 elamanın yamam uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Yayma sonucu	198
Şekil 6.5.	Orijinal görüntü ve 5x5 yapı elamanın açma uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Açma sonucu	199
Şekil 6.6.	Orijinal görüntü ve 5x5 elamanın kapama uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Kapama sonucu.....	199
Şekil 6.7.	Önerilen yöntemin KTH veri tabanı sonuçları: Kişi1	201
Şekil 6.8.	Önerilen yöntemin KTH veri tabanı sonuçları:Kişi2	201
Şekil 7.1.	Eylem tanıma sisteminin genel veri akış diyagramı gösterilmektedir	208
Şekil 7.2.	İnsan hareketleri tanıma sistemine genel bakış.	209
Şekil 7.3.	“Tekmeleme” eylemi. Aynı eylem kamera açısı, elbise, vücut şekilleri gibi farklılıklardan dolayı farklı görünmektedir.....	210
Şekil 7.4.	Durağan görüntü tabanlı hareket tanımaya birkaç örnek. İlgili eylemi belirlemek için tek bir görüntü yeterlidir.	211
Şekil 7.5.	Durağan görüntüler üzerine yapılan hareket tanıma yayınlarının yıllık ve birikimli eğilimi	213
Şekil 7.6.	UCM veri setinde ortam gözetleme sahnesi.....	214
Şekil 7.7.	Derinlik kamera örnekleri. (a) stereo kamera, (b) time-of-flight, (c) kodlanmış ışık (Microsoft kinect).	215
Şekil 7.8.	Johannson insan vücuduna bağlanan birkaç ışık noktasının hareketi gözlemleyen kişinin yapılan hareketi tanımasında yeterli olduğunu göstermiştir	220
Şekil 8.1.	KTH veri tabanında yer alan altı farklı hareket tespit görüntüleri	231
Şekil 8.2.	WEIZMANN veri tabanında yer alan on farklı hareket tespit görüntüleri	231
Şekil 8.3.	Görüntüye ait bir pikselin yatay ve dikey eksenindeki komşu piksellerinin değeri	233
Şekil 8.4.	Bir görüntüye ait Eğilim Büyüklüğü ve Eğilim Açısı matrisleri.....	234
Şekil 8.5.	Eğilim histogramında Eğilim Büyüklükleri ve Eğilim Açılarının temsili.....	236
Şekil 8.6.	Eğilim büyüklüklerinin eğilim açı merkezlerine dağılımı	237
Şekil 8.7.	Elde edilecek eğilim histogramının temsili görünümü.	238
Şekil 8.8.	64x128 ebatlarına sahip bir görüntü için hücre-blok yapısı	239
Şekil 8.9.	HOG yönteminin özet gösterimi	241
Şekil 8.10.	İki sınıfı birbirinden ayıran farklı doğrular	245
Şekil 8.11.	Sınıfların, sınıfları ayıran doğruya olan uzaklığını gösteren marjın bölgesi	245

Şekil 8.12.	KTH veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait karışıklık matrisi.....	251
Şekil 8.13.	KTH veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait normalleştirilmiş karışıklık matrisi.	251
Şekil 8.14.	KTH veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait sınıflandırma raporu.	252
Şekil 8.15.	WEIZMANN veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait karışıklık matrisi.	252
Şekil 8.16.	WEIZMANN veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait normalleştirilmiş karışıklık matrisi.	253
Şekil 8.17.	WEIZMANN veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait sınıflandırma raporu.....	253

TABLolar DİZİNİ

Tablo 1.1.	Toplamsal hücresele otomata kuralları	15
Tablo 2.1.	Kural 60, Kural 204 ve Kural 51'e göre orta hücrelerin alacağı bir sonraki durum	44
Tablo 2.2.	CA<142, 156, 244, 12> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları.....	46
Tablo 2.3.	CA<105, 129, 171, 65> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları.....	48
Tablo 2.4.	Kural204 ve Kural51 KMT'lerinin	49
Tablo 2.5.	Tek uzunluk cezbedici oluşumuna göre gruplandırılmış 256 hücresele otomata kuralı	51
Tablo 2.6.	Bir sonraki durum hesaplanması için i ve i+1 kurallarının KMT'leri arasındaki ilişki.	57
Tablo 2.7.	CA<78, 142, 201, 92> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları.....	59
Tablo 2.8.	CA<24, 74, 37, 41> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları.....	72
Tablo 2.9.	KMT $x_i \in KMT_0$ olduğunda KD_t ve KD_{t+1} KMT Dizilimleri arasındaki ilişki.	73
Tablo 2.10.	KMT $x_i \in KMT_1$ olduğunda KD_t ve KD_{t+1} KMT Dizilimleri arasındaki ilişki.	73
Tablo 2.11.	CA<165,192,216,60> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları.....	79
Tablo 2.12.	CA<8, 187, 255, 205, 0, 102> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları	87
Tablo 2.13.	MONKs ve Pima Indian Diabet veri setlerine ait bilgiler	92
Tablo 2.14.	Bilinen makine öğrenmesi modelleri kullanarak elde edilen başarı oranları	96
Tablo 2.15.	Monks ve Pima Indian elde edilen sonuçları hücresele otomata tabanlı ve diğer yöntemler ile karşılaştırılması	97
Tablo 2.16.	Bire Karşı Bir tüm kombinasyonlar için elde edilen başarı oranları	107
Tablo 2.17.	Bilinen makine öğrenme modelleriyle karşılaştırma	110
Tablo 3.1.	Farklı Birleştir/bul algoritmalarının zaman karmaşıklığı.....	113
Tablo 4.1.	CA<105, 129,171,65> kural vektörünün KMT'ye göre bir sonraki durum değerleri	135
Tablo 4.2.	Karışıklık matrisi hücrelerinin adları	155
Tablo 4.3.	Berkeley test veri setinde yer alan her bir görüntü içi elde edilen fl ve hassasiyet değerleri.....	156
Tablo 4.4.	Berkeley veri seti üzerinde yapılan çalışmaların başarı oranları.....	158
Tablo 5.1.	64 olan özellik vektörünün 16'ya düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklere göre elde edilen tanıma oranları	186

Tablo 5.2.	64 olan özellik vektörünün 8'e düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklere göre elde edilen tanıma oranları	187
Tablo 7.1.	Hareketlerin seviyeleri	206
Tablo 7.2.	Derinlik kameralarının karşılaştırılması	216
Tablo 8.1.	Örnek bir Karışıklık Matrisi	248
Tablo 8.2.	Elmalar sınıfının genelleştirilmiş karışıklık matrisi	249
Tablo 8.3.	Geliştirilen yönteme ve literatürde KTH veri seti üzerinde yapılan çalışmalara ait başarı oranları	254
Tablo 8.4.	Geliştirilen yönteme ve literatürde WEIZMANN veri seti üzerinde yapılan çalışmalara ait başarı oranları	254



SİMGELER VE KISALTMALAR DİZİNİ

b	: Bias
$gri_{yoğunluk}$	: Üç renk kanalının ortalamasına göre gri renge dönüştürme
$gri_{Luminance}$	: Renk kanallarını belirli ağırlıklara göre hesaplayıp gri renge dönüştürme
I	: Görüntü
$I(x,y,t)$	: Görüntüdeki x ve y konumundaki pikselin t zamanındaki değeri
D	: Hareket tespit görüntüsü
$D(x,y,t)$	: Hareket tespit görüntüsündeki x ve y konumundaki pikselin t zamanındaki değeri
D_g	: Gerçek hareket tespit görüntüsü
$D_g(x,y,t)$	: Gerçek hareket tespit görüntüsündeki x ve y konumundaki pikselin t zamanındaki değeri
S_i^t	: i 'ninci hücrenin t zamanındaki durumu
w	: Ağırlık vektörü
w^T	: Ağırlık vektörünün transpozu
X	: Özellik vektörü

Kısaltmalar

CA	: Cellular Automata(Hücresel Otomatlar)
CFG	: Content Free Grammar(İçerik Bağımsız Gramer)
DBA	: Dinamik Bayesian Ağ
DoG	: Difference of Gaussians (Gauss Farkı)
DoH	: Determinant of Hessian (Hessian Determinantı)
DTW	: Dynamic Time Warping (Dinamik Zaman Bükmesi)
EBR	: Edge-Based Regions(Kenar Tabanlı Bölgeler)
EDVAC	: Electronic Discrete Variable Computer(Elektronik Ayırık Değişken Bilgisayar)
ENIAC	: Electronic Numerical Integrator and Computer (Elektronik Sayısal Entegreli Bilgisayar)
GLOH	: Gradient Location-Orientation Histogram (Yönlü Eğilim Histogramı)
GMACA	: Generalized Multiple Attractor Cellular Automata(Genelleştirilmiş Çoklu Cezbedici Hücresel Otomatlar)
GMM	: Gizli Markov Modeli
FN	: False Negative (Yanlış Negatif)
FP	: False Positive (Yanlış Pozitif)
HOG	: Histogram of Gradient (Eğilim Histogramı)
IBR	: Intensity Extrema-based Region
KMT	: Kural Min Terimi
LESH	: Local Energy-Based Shape Histogram
LKT	: Lucas-Kanade-Tomasi

LoG	: Laplacian of Gaussian (Gauss'ın Laplace'ı)
MSER	: Maximally Stable Extremal Regions(Maksimum Düzeyde Kararlı Ekstremal Bölgeler)
NWFE	: Nonparametric Weighted Feature Extraction (Parametrik Olmayan Ağırlıklı Öznitelik Çıkarımı)
PCA-SIFT	: Principal Component Analysis- Scale-Invariant Feature Transform(Temel Bileşen Analizi - Ölçekten Bağımsız Öznitelik Dönüşümü)
PCBR	: Principal Curvature-Based Region Detector(Temel Eğrilik Tabanlı Bölge Dedektörü)
RVMs	: Relevance Vector Machines (İlişkili Vektör Makineleri)
SCFG	: Stocasting Context Free Grammer(Olasılıksak İçerik Bağımsız Gramer)
SIFT	: Scale-Invariant Feature Transform(Ölçekten Bağımsız Öznitelik Dönüşümü)
SURF	: Speeded Up Robust Features (Hızlandırılmış Gürbüz Öznitelikler)
SVM	: Support Vector Machine (Destek Vektör Makineleri)
TN	: True Negative (Doğru Negatif)
TP	: True Positive (Doğru Pozitif)
YSA	: Yapay Sinir Ağları

VIDEO GÖRÜNTÜLERİNDEN İNSAN HAREKETLERİNİN GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL OTOMATLAR (GMACA) İLE TANINMASI

ÖZET

HücreSEL otomatlar eş zamanlı programlamaya olanak sağlayan, ayrık uzay zaman yapısındaki hücreSEL ızgara düzleminde durum güncelleme mantığıyla çalışan hesaplama modeline sahiptir. Durum uzayında bazı durumların diğer durumları etrafında toplama yeteneklerinin araştırmacılar tarafından keşfedilmesi, hücreSEL otomatların örüntü tanıma potansiyelini ortaya çıkarmıştır. Örüntü tanıma da kullanılacak hücreSEL otomata kurallarının nitelenmesi için lineer ve lineer olmayan yöntemler önerilmiştir. Lineer yöntemler cebirSEL işlemler kullanırken lineer olmayan yöntemler genetik algoritma veya erişilebilirlik tabanlı yaklaşımlarla kural nitelenmesi yapmaktadırlar. Bu tez kapsamında hücreSEL otomata kurallarının nitelenmesi için erişilebilirlik ağacı tabanlı yaklaşımlar kullanılmıştır. Cezbedici durumlar, ikili sınıflandırmada başarıyla kullanılmıştır. Cezbedicilerin çoklu sınıflandırmada kullanılabilmesi için iki ayrı çoklu sınıf stratejileri test edilmiştir. Cezbedicilerin örüntü çekim yetenekleri farklı yapıdaki problemlerde kullanılmıştır. Birincisi: HücreSEL otomatların ürettiği cezbedici durumlar, görüntü bölümlendirmede kullanılmıştır. Önerilen yöntem, kümeleme görüntü bölümleme yöntemleri kapsamında değerlendirilebilir. Yöntemin temel varsayımı; bir bölgeye ait piksellerin yakın değere sahip olacak olmasıdır. İkincisi: Cezbedicilerin örüntü çekim merkezi formundaki davranışları, makine öğrenmesi modellerinin kullandığı özellik vektörlerinin boyutunun azaltılmasında kullanılmıştır. Cezbedicilerle birden fazla benzer özellik tek bir özellik tarafından temsil edilerek özellik vektörünün boyutu azaltılmıştır. Üçüncüsü: HücreSEL otomatların çekim merkezi formunda davranan cezbedici durumları görüntü tespitinde kullanılmıştır. Önerilen yöntem, ardışık görüntülerin farkına dayalı hareket tespit etme yöntemleri altında değerlendirilebilir. Önerilen yöntemin daha önceki yapılan çalışmalardan en önemli farkı hareket tespitinin cezbedici durumlar kullanılarak yapılmasıdır. Son olarak, cezbediciler kullanılarak elde edilen hareket tespit görüntülerinden insan hareketlerinin tanınması gerçekleştirilmiştir.

Anahtar Kelimeler: Boyutsal İndirgeme, Görüntü Bölütleme, Hareket Tanıma, Hareket Tespiti, HücreSEL Otomatlar.

RECOGNITION OF HUMAN ACTIONS FROM VIDEO IMAGES BY GENERALIZED MULTIPLE ATTRACTOR CELLULAR AUTOMATA(GMACA)

ABSTRACT

Cellular automata have a computational model that operates with update logic in the cellular grid plane in the discrete space time structure, which allows simultaneous programming. In the state of space, the discovery of the ability of some states to collect around other states by researchers revealed the potential of pattern recognition of cellular automata. Linear and nonlinear methods have been proposed to characterize the cellular automata rules to be used in pattern recognition. While linear methods use algebraic operations, nonlinear methods characterize the rules with genetic algorithm or reachability tree based approaches. Within the scope of this thesis, reachability tree based approaches were used to characterize the cellular automata rules. Attractor state have been successfully used in binary classification. Two different multi-class strategies have been tested in order to use attractor in multi-class classifications. Pattern attraction abilities of attractor have been used in different problems. First: Attractor states produced by cellular automata were used in image segmentation. The proposed method can be evaluated within the context of clustering image partitioning methods. The basic assumption of the method; that the pixels of a region will have a close value. Second: The behaviours of attractor in the form of pattern centres were used to reduce the size of the feature vectors used by machine learning models. The feature vector is reduced in size by representing one or more properties with attractors. Third: Attractor state were used for motion detection. The proposed method can be evaluated under the methods of motion detection based on the difference of consecutive images. The most important difference of the proposed method from previous studies is that motion detection is made using attractor generated by rule vector. Finally, recognition of human actions from motion detection images using attractors were carried out.

Keywords: Dimensional Reduction, Image Segmentation, Action Recognition, Motion Detection, Cellular Automata.

GİRİŞ

Hücresele Otomatlar(Cellular Automata(CA)), mevcut matematiksel yaklaşımlardan farklı hesaplama modellerine ilgi duyan araştırmacıların dikkatlerini üzerine çekmeyi başaran ayrık bir hesaplama modelidir. Bu hesaplama modelinin yapı taşı, etrafındaki hücrelerle etkileşim halinde olan hücredir. Hücresel otomata hesaplamada hücrelerin durumuna göre bir sonraki duruma geçmesi, hücresele otomata hesaplama modelinin temelini oluşturmaktadır.

1687’de Isaac Newton’un “Doğa Felsefesinin Matematiksel Temelleri (Mathematical Principles of Natural Philosophy)”, orijinal adıyla Prensipler(Principia) kitabında ortaya koyduğu prensipler, mevcut matematiksel yaklaşımların temelleri olduğu kabul edilir (Wolfram, 2002; Das, 2006). Bu yaklaşım, doğal dünyanın matematiksel denklemlere dayanan kurallarla açıklanabileceği fikrini ortaya çıkarmıştır. Basit iterasyonlarla gerçekleştirilebilen, daha genel türde kurallar bulabilme potansiyeline sahip hesaplama yapısı, hücresele otomatların doğayı farklı bir şekilde açıklayabileceği fikrini vermiştir.

Hücresele otomatlar ilk olarak Von Neumann tarafından canlıların kendini yenileme yeteneklerini modellemek için önerilmiştir (Neumann, 1966). Neumann, ızgara düzlemindeki hücresele otomata hücrelerinin uygun kurallarla birbirleriyle etkileşimi ayarlandığında biyolojik canlıların büyüme, kendini yenileme yeteneklerinin modellenebileceğini göstermiştir. Hücresel otomatların potansiyelinin daha fazla kişi tarafından fark edilmesi, Conway’in hücresele otomata mantığına dayanan ‘Yaşam Oyunu (Gardner, 1970; Gardner, 1971) sayesinde olmuştur. Hücresele otomata alanındaki en önemli gelişme Wolfram’ın çalışmalarıyla yaşanmıştır (Wolfram S., 2002).

Hücresele otomata kurallarının nitelenmesi için önerilen ilk çalışmalar lineer cebir tabanlı yaklaşımlardır. Bu yaklaşımlar XOR ve XNOR mantıksal operatörlerine göre durum güncellemesi yapan kurallar üzerinde işlem yapabiliyorlardı. Daha sonraları, tüm kurallar üzerinde işlem yapabilen lineer olmayan yöntemler önerildi. Lineer

yöntemler kuralların nitelenmesi için genetik algoritmaları veya erişilebilirlik ağacı kullanır. Erişilebilirlik ağacı tabanlı yöntemler hem zaman karmaşıklığı hem de hafıza kullanımı açısından verimli çözümler sunma potansiyeline sahiptir.

Hücrel otomatların ilk önerildiğindeki karmaşık yapısının, evrilerek daha basit yapıya kavuşması mühendislik uygulamalarında kullanılma potansiyelini ortaya çıkarmıştır. Bu potansiyelin yanında, ızgara düzlemindeki çalışma uzayının görüntü piksellerinin matris yapısına benzemesi görüntü işleme uygulamalarında kullanılmasına ilham vermiştir.

Tezin Amacı:

Hücrel otomata kullanılarak yapılan görüntü işleme uygulamalarında çoğunlukla basit görüntü işleme görevleri gerçekleştirilmiştir. Yapılan çalışmalarda kuralların nitelenmesi genelde sezgisel olarak belirlenmiştir. Bu yaklaşımlar, görüntü piksellerinin işlenmesini hücrelerin birbirleriyle olan etkileşimi bağlamında ele alarak gerçekleştirmişlerdir.

Hücrel otomatların örüntü çekim gücü, örüntü sınıflandırma ve tanıma uygulamalarında kullanılmıştır. Örüntü çekim gücü olan hücrel otomatlar kural yapısının nitelenmesine göre Çoklu Cezbedici Hücrel Otomatlar(Multiple Attractor Cellular Automata(MACA)) ve Genelleştirilmiş Çoklu Cezbedici Hücrel Otomatlar(Generalized Multiple Attractor Cellular Automata(GMACA)) olarak adlandırılmıştır.

Hücrel otomatların ayırık ve eş zamanlı programlamaya imkân veren yapısı GMACA'ların örüntü sınıflandırma ve başka mühendislik uygulamalarında kullanılmasını cazip hale getirmiştir. GMACA'lar birçok mühendislik uygulamasında kullanılmasına rağmen görüntü işlemede ve bilgisayarlı görüde kullanılmamıştır.

Tezin temel amaçları:

1. Birincisi: Genelleştirilmiş Çoklu Cezbedici Hücrel Otomatların örüntü çekim davranışlarının nitelenmesi için gerekli teorik alt yapının araştırılmasıdır.
2. İkincisi: Hücrel otomatların örüntü çekim gücünün örüntü tanıma ve sınıflandırmada kullanılmasıdır.

3. Üçüncüsü: Gerekli teorik alt yapı elde edildikten sonra, hücresele otomatları ileri görüntü işleme uygulamalarında kullanmaktır.
4. Dördüncüsü: Hücresele otomata kullanılarak yapılan görüntü işleme sonuçlarını ileri bir bilgisayarlı görü problemi olan hareket tanımda kullanmaktır.

Tezin Organizasyonu:

Bu tez çalışmasının amacının gerçekleştirilmesi için yapılan araştırmalar, çalışmalar ve elde edilen sonuçlar sekiz ana başlık altında yazılmıştır:

1. Hücresele Otomatlar
2. Lineer Olmayan Hücresele Otomatların Nitelenmesi
3. Birleştir/Bul Veri Yapısı Kullanılarak GMACA'ların Kümelere Ayırıştırılması
4. Genelleştirilmiş Çoklu Cezbedici Hücresele Otomatların Görüntü Bölümlendirmede Kullanılması
5. Genelleştirilmiş Çoklu Cezbedici Hücresele Otomatların Boyut Azaltmada Kullanılması
6. Genelleştirilmiş Çoklu Cezbedici Hücresele Otomatlar İle Video Görüntülerde Hareket Tespiti
7. İnsan Hareketlerinin Video Görüntülerden Tanınması Üzerine Yapılan Literatür Araştırması
8. Video Görüntülerden İnsan Hareketlerinin Tanınması

“Hücresele Otomatlar” başlıklı bölümde hücresele otomatlar hakkında kapsamlı literatür araştırması yapılmıştır. Literatür araştırmasında hücresele otomatların tarihçesi, hesaplama modelindeki değişiklikler, bilim insanları tarafından yapılan önemli çalışmalara ve hücresele otomatların uygulama alanlarına değinilmiştir. Hücresele otomatların durum geçişlerinin nitelenmesi için kullanılan lineer ve lineer olmayan yöntemler ayrı başlıklar altında ele alınmıştır. Bu bölümde ayrıca hücresele otomatların temel hesaplama modelinin anlatımı ve farklı hücresele otomata türleri anlatılmıştır. Son olarak, Wolfram'ın hücresele otomatları neden yeni bir bilim türü olarak tanımladığı önermelerine yer verilmiştir.

“Lineer Olmayan Hücresele Otomatların Nitelenmesi” başlıklı bölüm, tez çalışmasının teorik alt yapısının oluşturulduğu bölümdür. Öncelikle durum geçişlerinde önemli bir yeri olan Kural Min Terim kavramı ve hücresele otomata terminolojileri açıklanmıştır.

Hücresel otomata durum geçişlerinde özel bir karakteristiği olan cezbedici durumlar ele alınmıştır. İki farklı cezbedici durum olan; tek uzunluk döngülü cezbediciler ve çok uzunluk döngülü cezbedicilerin analizi yapılmıştır. Tezde yapılan uygulamalarda kullanılan tek uzunluk döngülü cezbedicilerin nitelenmesi anlatılmıştır. Cezbedici nitelenmesinde kullanılan erişilebilirlik ağacı ve bu ağaç kullanılarak yapılan; cezbedici durum belirleme, ulaşılabilir ve ulaşılmaz durumların belirlenmesi ve kapsamlı bitlerin tanımlanması için geliştirilen algoritmalara yer verilmiştir. Ayrıca çok uzunluk döngülü cezbedici oluşumunun nedenleri ve istenen sayıda kapsayıcı bite sahip ve sadece tek uzunluk döngülü cezbedici üreten kural vektörünün elde edilmesi açıklanmıştır. Cezbedici durumların örüntü tanımada kullanılması farklı veri setleri üzerinde gerçekleştirilmiştir. Cezbedicilerin örüntü tanımada kullanılması için veri setinin hücresel otomata hesaplama modeline uygun hale getirilmesi gerekir. İki farklı veri setinin hücresel otomata hesaplama modeline uygun hale getirilmesi için gerekli ön işlemler anlatılmıştır. Literatür araştırmasında hücresel otomata kullanılarak yapılan örüntü tanıma çalışmaları iki sınıf içeren veri setleri kullanılarak yapılmıştır. Bu bölümde ilk defa, hücresel otomatları çoklu sınıf içeren veri setlerinde kullanılabilme potansiyeli ortaya konmuştur. Elde edilen sınıflandırma sonuçları karışıklık matrisi ve sınıflandırma raporlarıyla beraber verilmiştir.

“Birleştir/Bul Veri Yapısı Kullanılarak GMACA’ların Kümelere Ayırıştırılması” başlıklı bölümde, hücresel otomata uygulamalarının gerçekleştirilmesinde kullanılabilen, hafıza kullanımı ve zaman karmaşıklığı açısından verimli bir yöntem olan birleştir/bul yöntemleri anlatılmıştır. Durum geçiş diyagramlarının ayırıştırılmasında kullanılan; hızlı-bul, hızlı-birleştir, ağırlıklı-hızlı-birleştir ve yol-sıkıştırılmalı-ağırlıklı-birleştir gibi farklı birleştir/bul yöntemleri analiz edilmiştir. Birleştir/bul yöntemlerinin durum geçiş diyagramlarının ayırıştırılmasında kullanılması için yapılanması gerekli olan adımlar içeren sözde kod verilmiştir. Durum geçişlerinin birleştir/bul yöntemleri kullanılarak ayırıştırılması gerçek veriler üzerinde test edilmiştir. Test için görüntü verileri kullanılmıştır.

“Genelleştirilmiş Çoklu Cezbedici Hücresel Otomatların Görüntü Bölümlendirmede Kullanılması” başlıklı bölümde GMACA’ların örüntü çekim gücü, görüntü bölümlendirmede kullanılmıştır. Temel hücresel otomatlar, daha önceki bilimsel çalışmalarda görüntülerde kenar bulma, gürültü giderme, morfolojik işlemler vb.

görevlerde kullanılmışlar. Ancak GMACA'ların örüntü çekim yetenekleri görüntü işleminde kullanılmamıştır. Bu görev ilk defa bu tez kapsamında gerçekleştirilmiştir. Bunu için GMACA'ların ürettiği tek uzunluk döngülü cezbediciler, görüntülerde görüntünün bir bölümünü temsil edecek şekilde tasarlanmıştır. Geliştirilen görüntü bölümlendirme yöntemi, Berkeley görüntü bölümlendirme veri seti üzerinde test edilmiştir. Elde edilen görsel ve istatistiksel sonuçlar aynı veri seti üzerinde yapılan çalışmalarla karşılaştırılmıştır.

“Genelleştirilmiş Çoklu Cezbedici Hücresel Otomatların Boyut Azaltmada Kullanılması” başlıklı bölümde GMACA'lar boyut azaltmada kullanılmıştır. GMACA'ların ürettiği cezbediciler, birden fazla örüntüyü temsil edebilen temsil yeteneği güçlü örüntüler olarak ele alınması, geliştirilen boyut azaltma yönteminin temel mantığıdır. Geliştirilen yöntem, 64 uzunluğundaki özellik vektörlerinden oluşan, el yazması rakamların olduğu MNIST veri seti üzerinde test edilmiştir. Özellik vektörü 8, 16 ve 32 boyutlarına düşürülmüştür. Düşürülen özellik vektörleri kullanılarak yapılan sınıflandırma işlemlerinde elde edilen başarı sonuçlarına yer verilmiştir. Elde edilen başarı sonuçları, GMACA kullanılarak gerçekleştirilen boyut azaltmanın gerçek veriler üzerinde kullanılabileceğini göstermektedir.

“Genelleştirilmiş Çoklu Cezbedici Hücresel Otomatlar İle Video Görüntülerde Hareket Tespiti” başlıklı bölümde video görüntülerde insan hareketlerinin tespiti GMACA kullanılarak gerçekleştirilmiştir. Bu bölümde yapılan çalışma, GMACA kullanılarak gerçekleştirilen ilk hareket tespit çalışmasıdır. Hücresel otomatların durum değişim mantığı, daha önceki bilimsel çalışmalarda hareket tespitinde kullanılmıştır. Hücresel otomata kullanılarak yapılan daha önceki hareket tespit çalışmaları ikili görüntüler üzerinde işleme yapabilme yeteneğine sahiptir. Bu nedenle günümüzde yaygın olarak kullanılan yoğunluk görüntüleri üzerinde kullanılması mümkün değildir. GMACA kullanılarak gerçekleştirilen hareket tespit çalışması, yoğunluk görüntüleri üzerinde hareket algılama yeteneğine sahiptir. Geliştirilen yöntem ardışık görüntülerin durum değişimlerine göre hareket tespiti yapmaktadır. Geliştirilen hareket tespit yöntemi iki farklı veri seti üzerinde test edilmiştir. Elde edilen sonuçlar, GMACA'ların video görüntülerde hareket tespitinde kullanılabileceğini göstermektedir.

“İnsan Hareketlerinin Video Görüntülerden Tanınması Üzerine Yapılan Literatür Araştırması” başlıklı bölümde video görüntülerde insan hareketlerinin tanınması üzerine yapılan bilimsel çalışmalara yer verilmiştir. İnsan hareketlerinin tanınması bir sonraki bölümde gerçekleştirilmesine rağmen, kapsamlı bir literatür araştırması yapıldığı için ayrı ana başlık altında ele alınmıştır. Literatür araştırması kapsamında ilk önce: Hareket tanıma çalışmalarında farklı hareketlerin adlandırılması için yapılan tanımlara yer verilmiştir. Farklı hareket seviyelerine göre yapılan hareket tanıma çalışmalarına değinilmiştir. Hareket tanımanın gerçekleştirilmesi için yapılan görevler açıklanmıştır. Hareketli nesne tespiti, hareketli nesnenin takibi, özellik çıkarımı, hareketlerin teşmil edilmesi, hareketlerin bölümlendirilmesi ve hareket tanıma kullanılan algoritmalar ayrı başlıklar altında değerlendirilmiştir.

“Video Görüntülerden İnsan Hareketlerinin Tanınması” başlıklı bölümde GMACA kullanılarak yapılan hareket tespit görüntülerinden insan hareketlerinin tanınması gerçekleştirilmiştir. İnsan hareketlerinin tanınmasında KTH ve WEIZMANN veri setleri kullanılmıştır. İnsan hareketlerinin tanınması beş aşamada gerçekleştirilmiştir. Öncelikle görüntüler gri seviyeye dönüştürülmüştür. Sonra, gri video görüntülerinden GMACA kullanılarak hareket tespiti yapılmıştır. Bir sonraki aşamada hareket tespit görüntülerinden HOG özellikleri çıkartılmıştır. Dördüncü aşamada çapraz doğrulama yöntemine göre veri seti oluşturulmuştur. Son olarak oluşturulan veri setleri üzerinde SVM kullanılarak hareket tanıma işlemi gerçekleştirilmiştir. Gerçekleştirilen hareket tanıma sonuçları, istatistiksel ölçüm metrikleri; karışıklık matrisi ve sınıflandırma raporuyla verilmiştir. Elde edilen başarı oranları literatürde aynı veri setleri üzerinde yapılan çalışmalarla karşılaştırılmıştır. Elde edilen sonuçlar GMACA’lar kullanılarak gerçekleştirilen hareket tespit görüntülerinin, ileri bilgisayarlı görü uygulamalarında kullanılabileceğini göstermektedir.

1. HÜCRESEL OTOMATLAR

Bilgisayar biliminin temel amacı bilgisayarların hesaplama yeteneklerine uygun soyut modeller ortaya koymaktır. Önerilen modelin verimliliği ve doğruluğunun ortaya konması gerekir (Das, 2006). Hücresel otomatlar bilgisayarların hesaplama yeteneğine uygun bir hesaplama modeli olarak John von Neumann ve Stan Ulam tarafından önerilmiştir (Neumann, 1966). Neumann günümüz bilgisayarların hesaplama yeteneğini en iyi bilecek kişilerin başında gelir. Çünkü günümüz bilgisayarları Neumann'ın önerdiği hesaplama modeline göre işlem yapmaktadır ve Neumann mimarisi olarak adlandırılır (Nabiyev, 2013).

Neumann onluk tabanda işlem yapan ENIAC (Electronic Numerical Integrator and Computer) bilgisayar sisteminden sonra önerilen EDVAC (Electronic Discrete Variable Computer) projesinin yürütücüsüdür. EDVAC projesi raporunda günümüz bilgisayarlarının çalışma prensipleri yer almaktadır. Bu prensipler yukarıda belirttiğimiz gibi günümüzde Neumann mimarisi olarak adlandırılmaktadır. Bu mimarinin ana fikri hesaplamaların ayrık yapıda olması ve ikili sayı sistemleri kullanılarak gerçekleştirilmesine dayanmaktadır. CA da Neumann tarafından bilgisayarların hesaplama yeteneğine uygun geliştirilmiş soyut bir hesaplama modelidir.

Hücresel otomata, bir takım ayrık bileşenlerden (hücreler) oluşan, uzamsal olarak genişletilmiş merkezi olmayan bir sistemin basit bir modeli olarak görülebilir. Oluşturulan hücreler arasındaki iletişim, yerel etkileşimle sınırlıdır. Her bir hücre, yerel komşularının durumlarına bağlı olarak zamanla değişen belirli bir duruma geçer. Genel yapı bir paralel işlem cihazı olarak görülebilir. Bununla birlikte, bu basit yapı birkaç kez yinelendiğinde karmaşık desenler üretir ve farklı sofistike doğal fenomenleri benzetim etme potansiyeli gösterir (Ganguly ve ark., 2003).

Turing makineleri, matematikçi Alan Turing tarafından önerilen ve her türlü hesaplamayı yapma kapasitesine sahip varsayımsak bir makinedir (Turing, 1937). Turing makinesi bilgisayar bilimlerinde çok önemli bir yere sahiptir. Das, CA'nın daha

etkili bir yapıya sahip olduğunu belirtmiştir. Bunun nedenin; CA'nın ayırık hesaplama yapısı ile paralel programlamaya imkân sunması ile açıklamıştır (Das, 2006).

CA'nın ayırık ve yerel etkileşimle sonuç üreten hesaplama yapısı farklı alanlarda araştırma yapan bilim insanlarının dikkatini çekmiştir. Bu sayede CA farklı alanlardaki uygulamalarda ve araştırmalarda kullanılmıştır. CA fizik (Osborne, 2001) ve matematik (Chaudhuri, 1997) başta olmak üzere, coğrafya (Ku, 2016), sosyal bilimler (Dabbaghian ve ark., 2011), biyoloji (Green, 1990), biçimsel diller (Çalıkoğlu, 1973) ve görüntü işleme (Rosin ve ark., 2014) gibi farklı disiplinlerde yoğun bir şekilde kullanılmıştır

1.1. Hücresel Otomatların Tarihçesi

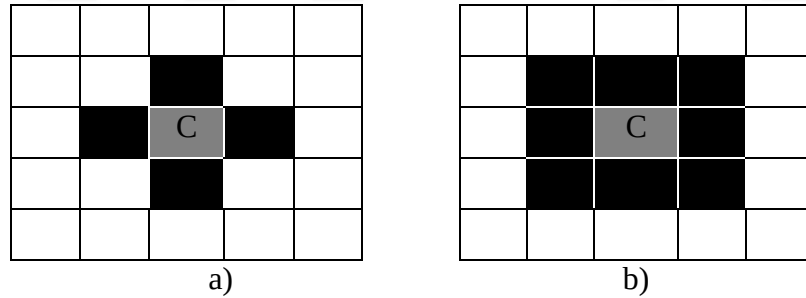
Jon von Neumann biyolojik canlıların kendini yenilemesini modellemek için bilgisayarların hesaplama yeteneğine uygun bir hesaplama modeli olan CA kavramını önerdi. Stan Ulam Neumann'ın yarım kalan çalışmasına bazı ilaveler getirerek tamamlayıp bilim camiasına hediye etmiştir (Neumann, 1966). Neumann biyolojik canlıların kendini üretme yeteneğinin mantıksal bir organizasyonla modellenmesi ile ilgileniyordu.

Neumann kendini üretme yeteneğinin genetik ve biyokimya bağlamında değil soyut bir mantıksal örgü ile modellenmesi ile ilgileniyordu (Langton, 1984). Langton'a göre Neumann'ın temel düşüncesi şöyledir: Kendini üretmesi biyokimyasal makineler tarafından gerçekleştirilen bir makinenin kendini üretme adımları mantıksal adımlarla tanımlanabilirse bu bir algoritma olarak düşünülebilir. Turing makinesi her türlü algoritmayı gerçekleştirebilecek kapasitede tasarlandığına göre kendi üretme yeteneği olan bir Turing makinesi tasarlanabilir.

Önerilen ilk CA 29 durum ve 5 komşuluktan oluşuyordu. Daha sonra yapılan çalışmalarda durum sayısını 8'e azaltıldı (Codd, 1968). Codd kendini üretme yeteneğine sahip Neumann makinesini daha basit bir hale getirdi. Langton yaptığı çalışmada Neumann ve Codd geliştirdikleri soyut makineyi daha basit hale getirilebileceğini gösterdi (Langton, 1984).

Wolfram, Von Neumann'ın, önerdiği kendini üretme yeteneğini modelleyebilen hücresel otomatların çok karmaşık olduğunu belirtir. Bunu sebebinin Von Neumann'ın muhtemelen kısmen de olsa, gerçek biyolojik organizmaların ve elektronik bilgisayarların karmaşıklığını görerek – bu seviyede bir karmaşıklığın, bir sistemin kendi kendini yeniden üretme gibi sofistike yetenekleri sergilemesi için gerekli olacağına inanmış görünüyor olmasına bağlar. Kendi kitabında (Wolfram, 2002) bunun kesinlikle doğru olmadığını, ancak var olan matematik ve mühendislik sezgileriyle von Neumann'ın bunu asla hayal etmediğini belirtir.

Buraya kadar bahsedilen CA çalışmalarının hepsi Neumann komşuluğuna dayanır. Neumann komşuluğu hücrenin kendisi ve iki yatay(sol ve sağ) ve iki dikey(alt ve üst) komşusundan oluşur. Moore ilk defa farklı bir komşuluk tanımladı (Moore, 1962). Bu komşuluk hücrenin kendisi ve etrafındaki sekiz hücrenin komşuluklarından oluşuyordu. Aşağıdaki Şekil 1.1'de Neumann ve Moore komşuluğunun gösterimi yer almaktadır. Moore komşuluğuna sahip iki durumlu CA'nın evrensel hesaplama gücüne sahip olduğu Moore komşuluğunun önerildiği çalışmada (Moore, 1962) gösterilmiştir.



Şekil 1.1. 2B CA komşulukları. a) Neumann komşuluğu b) Moore komşuluğu

CA, ızgara düzleminde yer alan hücrelerin bir durumdan başka bir duruma geçişini belirleyen kurallardan oluşur. Hücrenin etkileşimde olduğu hücre sayısı ile orantılı durum sayısı vardır. Kurallar belirlenen sayıda hücreye uygulandığında hücreler bir durumdan başka bir duruma geçiş yapar veya aynı durumunda kalır.

Moore, hücrelerin durum geçişlerini gözlemlediğinde bazı durumların atası olmadığını ve bazı durumların ise birden fazla ataya sahip olduğunu fark etmiştir (Moore, 1962). Eğer bir durumun birden fazla atası varsa bu hücresel otomatta bazı durumların atası

yok demektir. Böyle bir otomata “Cennet Bahçesi” olarak adlandırılan bir durum geçiş döngüsüne sahiptir. Yani atası olmayan durumlara bu durumlardan geçiş olmayacaktır ve bu durum döngüsünden çıkış olmayacaktır. Moore döngüden çıkış olmamasını cennete girdikten sonra çıkış olmamasına atfen Cennet Bahçesi olarak adlandırmıştır. Myhill daha sonra yaptığı çalışmada Cennet Bahçesi döngüsünün oluşması için bir durumun birden fazla ataya sahip olmasının yeterli olduğunu göstermiştir (Myhill, 1963).

Cennet Bahçesi teoreminin CA’nın örüntü tanıma ve örüntü sınıflandırma çalışmalarına ilham verdiği söylenebilir. Çünkü bazı durumların diğer durumları etrafından toplanması veya beraber aynı döngü içerisinde bulunması CA’nın örüntü tanımada kullanılmasını sağlamıştır (Das ve ark.,2009; Ganguly ve ark., 2002).

1950'lerin sonuna gelindiğinde hücresel otomata paralel bilgisayarlar olarak görülebiliyordu. 1960'larda Turing makineleriyle benzerlik gösteren gittikçe artan ayrıntılı ve teknik teoremler CA’nın hesaplama yeteneklerini ispatlamıştı. Ve 1960'larda hücresel otomatlarla ilgili çalışmalar çoğunlukla soyut matematiksel tartışmalar oluyordu (Wolfram, 2002). Bilgisayar biliminde bile tezelayon otomatu, hücresel alanlar, iteratif otomata, homojen yapılar ve evrensel alanlar gibi hücresel otomata bağlamında çeşitli isimler kullanılmıştır.

CA, 1970'lere kadar sınırlı sayıda bilim insanı tarafında üzerinde çalışılan bir fenomendi (Wolfram, 2002). Tanınırlığını büyük ölçüde Conway’in geliştirdiği Hayat Oyunu(Game of Life) ile beraber kazandığı çoğu araştırmada vurgulanmıştır (Sarkar, 2000; Das, 2006; Wolfram, 2002). Conway’in hayat oyununun bilim çevreleri tarafından bilinmesi Gardner’in sayesinde olmuştur (Gardner, 1970; Gardner, 1971). Hayat Oyunu, canlı organizmalardan oluşan bir topluluğun yükselişi, düşüşü ve dönüşümleri ile olan benzerliklerinden ötürü, gerçek hayattaki süreçleri andıran "simülasyon oyunları" olarak adlandırılan oyun sınıflardan birine aittir (Gardner, 1970).

Hayat Oyunundaki temel fikir, ızgara düzlemindeki hücrelerin basit bir ilk durum ile başlaması, daha sonra doğumlar, ölümler ve hayatta kalmalar için Conway'in "genetik kanunları" nın uygulanmasından oluşur (Gardner, 1970). Conway, oyundaki üç durum olan hayatta kalma, ölüm ve dogma için kurallar belirlemiştir. Hayat Oyunu, her biri

canlı ya da ölü olan iki muhtemel durumdan birinde olan sonsuz kare hücrelerden oluşan ızgara düzleminde gerçekleşir. Her hücre, doğrudan yatay, dikey veya çapraz bitişik hücreler olan sekiz komşusu ile etkileşime girer. Zaman içindeki her adımda, aşağıdaki kurallarla gerçekleşir (URL-3, 2017):

1. İki den daha az yaşayan komşusu olan canlı hücre ölür (az nüfus veya yalnızlığa maruz kalma olarak anılır).
2. Üçten fazla canlı komşu olan canlı hücre ölür (aşırı nüfus veya aşırı kalabalık olarak anılır).
3. İki veya üç yaşayan komşusu olan herhangi bir canlı hücre, değişmeden, bir sonraki kuşağa geçer.
4. Tam olarak üç canlı komşusu olan ölü bir hücre hayat bulur

Izgara düzleminin başlangıç örüntüsü, sistemin 'tohumunu' oluşturur. İlk nesil yukarıdaki kuralları tohumdaki her hücreye eşzamanlı olarak uygulayarak oluşturulur; doğumlar ve ölümler aynı anda gerçekleşir ve bunun gerçekleştiği ayrık ana, yineleme adı verilir. Kurallar yeni nesiller oluşturmak için art arda uygulanmaya devam eder (LURL-3, 2017).

Yaşam Oyunu CA'nın vadettikleri ve mantığını anlama açısından oldukça faydalı olmuştur. Basit kuralların kompleks davranışları sergileyebilmesi, hücre, komşuluk ilişkisi gibi kavramlar somut bir örnekle gösterilmiştir. Yukarıda verdiğimiz dört basit kuralla canlı bir organizma topluluğunun benzetiminin yapılabilmesi oyunun en dikkat çeken özelliğidir. Yaşam Oyununda kullanılan komşuluk Moore tarafından önerilen komşuluktur.

Hayat Oyununun CA'nın araştırmacıların dikkatini çekmesi açısından önemli bir gelişmeydi. Bununla birlikte, hücresel otomata tarihindeki devrim 1980'lerin başında oldu. Stephen Wolfram karmaşık davranışları benzetim edebilen basit bir boyutlu hücresel otomata türünü ayrıntılı olarak inceledi. Önerilen CA yapısı, 3 komşuya (kendi, sol ve sağ komşular) sahip ve iki durumdan oluşuyordu. Bu yapı, farklı alanlarda çalışan araştırmacıların büyük bir ilgi gördü. Lineer ve Toplamsal (Linear / Additive) CA (Chaudhuri ve ark., 1997) olarak adlandırılan 1 boyutlu CA'nın özel bir sınıfı daha fazla dikkat çekmiştir (Das, 2006).

Wolfram 1980'lerde yaptığı çalışmalar CA çalışmaları için yeni bir evre olarak tanımlanır ve modern CA çalışmalarının başlangıcı kabul edilir (Sarkar, 2000). Wolfram basit kuralların çok karmaşık sistemlerin benzetimini yapabileceğini düşünmekteydi. Bu düşünce, Wolfram'ın CA'yı daha basit bir yapıya dönüştürmesinde etkili olmuştur. Wolfram'ın çalışmalarıyla CA artık, Neumman tarafından tasarlandığı gündeki kompleks yapısından oldukça sade bir yapıya kavuşmuştur.

Wolfram ilk olarak, hücresel otomatların kendi kendini organize eden istatistiksel sistemler için matematiksel modeller olarak incelenmesini gerçekleştirdi. Pratikte uygulanması basit olmasına rağmen, CA'nın çok karmaşık davranış sergileyebileceğini ortaya koydu (Wolfram, 1983). Daha sonra hesaplama teorisinin hücresel otomata genel analizine uygulanmasında bazı ön çalışmaları yaptı (Wolfram, 1984a). Wolfram tek boyutlu hücresel otomatları dört ayrı evrensel sınıfına ayırdığını gösterdi (Wolfram, 1984b). Sınıfların adlandırmasını Sınıf I, Sınıf II, Sınıf III ve Sınıf IV sıralamasıyla yaptı. Bu sınıflar artan karmaşıklığa göre sırayla numaralandırılmıştır ve her biri belirgin özelliklere sahiptir. Bu sınıflarda üretilen yapıların niteliklerini inceledi. Her bir sınıfın özet açıklaması aşağıdadır (Wolfram, 2002).

Sınıf I: Davranış çok basittir ve neredeyse tüm başlangıç koşulları aynı son duruma geçmektedir. Kural0, Kural8 ve Kural136 bu kurallardan bazılarıdır.

Sınıf II: Olası birçok son durum vardır, ancak bunların tamamı ya sadece sonsuza kadar aynı kalıyor ya da her birkaç adımda tekrar eden belirli bir dizi basit yapıdan oluşuyor.

Sınıf III: Davranışları daha karmaşıktır. Üçgenler ve diğer küçük ölçekli yapılar birçok açıdan rasgele görünüm sergiler.

Sınıf IV: Düzen ve rastgele bir karışım içerir. Kendi başına oldukça basit olan lokal yapılar üretilir, ancak bu yapılar birbirlerine dolaşmakta ve çok karmaşık biçimde birbirleriyle etkileşime girmektedir.

Hücresel otomatlar, biçimsel olarak belirlenemeyen ve hesaplama zorluklarına neden olan birçok problemin benzetimi için kullanılır. Wolfram bu tür problemlerin fiziksel

modellerde yaygın olduđu ve CA'nın kullanılabileceğini önermiştir (Wolfram, 1985). Wolfram CA'nın modelleme yeteneğini rasgele dizi üretiminde kullanmıştır. Geliştirdiğı yöntemle ürettiğı dizileri, çeşitli ampirik, kombinasyonel, istatistiksel, dinamik sistemler teorisi ve hesaplama teorisi yöntemleri ile analiz ederek etkili bir rasgele dizi üretici önerisinde bulunduğunu belirtmiştir (Wolfram, 1986).

Wolfram 1980'lerin başında CA üzerine yapmaya başladığı araştırmaların şekillendirdiğı ünlü kitabını (Wolfram, 2002) "A New Kind of Science" yayımladı. Kitaba iddialı bir isim olan "Yenir Bir Bilim Dalı" ismini vermiştir. CA üzerine yaptığı çalışmaları yeni bir bilim dalının temelleri olarak görüyordu. Bu bilim dalının daha önce neden keşfedilemediğini şöyle açıklıyor:

"CA'nın anlamlı sonuçlar ortaya koyması zahmetli tekrar işlemleri ile elde ediliyor. Daha önceki zamanlarda tekrarlı işlemleri otomatik olarak hızlı yapacak mekanizmalara sahip olunmadığı için CA'nın potansiyeli bilim insanları tarafından fark edilememiştir" (Wolfram, 2002).

Kitabında ortaya koyduğu basit kuralların karmaşık sistemleri modelleme yeteneğinin daha önce fark edilmesi o zamana kadarki bilimsel önsezilerin engel olduğu belirtmiştir.

"Üç yüz yıl önce bilim, doğal dünyayı tanımlamak için matematik denklemlerine dayanan kuralların kullanılabileceğı fikrini benimsedi. Bu kitabımdaki amacım, benzer bir dönüşümü başlatmak ve basit bilgisayar programlarında somutlaşabilen daha genel kural türlerine dayanan yeni bir bilim türünü tanıtmaktır " (Wolfram, 2002).

Wolfram bu bilim türünün Einstein'nın fikirlerinden daha önemli olduğunu ve Newton'nun keşiflerinden sonra en önemli gelişme olduğu belirtmektedir (Naiditch, 2003). Wolfram, yeni bilim türünün, mevcut bilim alanlarının çok azı hariç her yerine dokunduğunu belirtiyor. Önsözün ilerleyen kısımlarında daha iddialı bir şekilde şunu söylüyor:

"Zaman içerisinde, bu kitabın fikirlerinin nüfuz etmesini bekliyorum; sadece bilim ve teknolojiye değil aynı zamanda birçok genel düşünce alanına. Matematik günümüzde eğitimin bir parçası olduğu gibi geliştirilen yöntemler, eğitimin standart bir parçası haline gelecektir" (Wolfram, 2002).

Wolfram'ın iddialarını dile getirirken takındığı emin tavır ve kendinden önceki araştırmacılara yeterli atıfta bulunmaması eleştiri konusu olmuştur. Wolfram kitabının esasına bakılmaksızın, birçok okuyucu Wolfram'ın iddialarını son derece rahatsız edici

buluyor (Naiditch, 2003). Bu konudaki düşüncemiz Wolfram'ın fazla eleştirildiği yönündedir. Bilime yaptığı katkı egosunun önünde olduğu söylenebilir.

İlk önerildiği günden (Neumann, 1966) Wolfram 'New Kind of Science' (Wolfram, 2002) adlı kitabına ve günümüze kadar, CA'nın basit yapısı çeşitli disiplinlerden araştırmacıların ilgisini çekmiştir. Son elli yılda titiz matematiksel ve fiziksel analizlere tabi tutulan CA'nın uygulamaları, hem fiziksel hem de sosyal olmak üzere farklı bilim dallarında önerildi. CA üzerine her yıl çok sayıda araştırma makalesi yayımlanmakta ve son yıllarda çeşitli dergiler CA konusunda uzmanlaşmış konferanslar organize etmektedir. Birkaç üniversitede hücrel otomatlarla ilgili dersler verilmektedir (Ganguly ve ark., 2003).

Hücrel otomatların araştırmacıların dikkatini çekme nedeni; çalışma mantığındaki basitliğine rağmen karmaşık sistemleri modellemedeki muazzam potansiyelinden kaynaklanmaktadır.

1.2. Hücrel Otomata Kural Nitelenmesi İçin Farklı Yaklaşımlar

Hücrel otomatlar Von Neumann tarafından ilk geliştirildiğinde iki boyutlu ve 29 durumdan oluşuyordu. Daha sonraki yapılan çalışmalarda durum sayısı 8'e (Codd, 1968) ve 2'ye (Moore, 1970) azaltıldı. Wolfram çalışmalarında CA'nın daha sade bir formu olan tek boyutlu iki durumlu hücrel otomatlar üzerine yoğunlaştı. Bu hücrel otomatlar üç komşuya sahiptir(sol, kendi ve sağ). Wolfram'dan sonra yapılan çalışmaların çoğu bu tür hücrel otomatlar üzerinedir. Bu tez kapsamında üç komşuya ve iki duruma sahip hücrel otomatlar üzerine çalışılmıştır.

Bir boyutlu ve iki durumlu hücrel otomata üzerine çok sayıda araştırma vardır. Bunu iki nedeni olabilir: Birincisi bir boyutlu ve iki durumlu hücrel otomatlar basit bir yapıya sahiptir. İkincisi bilim çevreleri tarafından büyük ilgi gören Wolfram'ın çalışmaları bir boyutlu hücrel otomalar üzerinedir. Bir boyutlu hücrel otomatlarda genelde komşu sayısı üçtür. Ancak komşu sayısını yarıçap katsayısıyla belirleyen (Li ve ark., 1990) çalışma olduğu gibi sadece yedi (Mitchell ve ark., 1993) komşuya sahip tek boyutlu hücrel otomatların olduğu farklı çalışmalar vardır. Hücrel otomatlardaki durum sayısı Galois Field $GF(q)$ biçiminde gösterilir. Parantez

içerisindeki q durum sayısını ifade eder. İki durumlu hücresel otomatlar $GF(2)$ biçiminde gösterilir.

CA davranışlarının nitelenmesi üzerine yapılan yöntemler ikiye ayrılır (Das, 2006): Lineer / Toplamsal (Linear/Additive) CA ve Lineer Olmayan(Nonlinear CA). Lineer CA davranışlarının nitelendirilmesi yapılırken lineer cebirsel işlemler kullanılabilirken Lineer olmayan CA'da lineer cebirsel işlemler kullanılamaz. Lineer olmayan CA davranışlarının nitelendirilmesi üzerine birçok çalışma (Das, 2006; Naskar, 2009; Peldek, 2012; Das ve ark., 2009) yapılmıştır. Bu tez çalışmasında lineer olmayan CA kullanılmıştır.

1.2.1. Lineer / toplamsal hücresel otomatlar

Hücresel otomatının lineer olabilmesi için kullanılan kuralların lineer veya toplamsal olması gerekir. Bir kuralın lineer olması için bir sonraki durumun belirlenmesinin XOR mantıksal operatörüyle yapılabilmesi gerekir. Toplamsal otomata içinde benzer şeyler söylenebilir: Toplamsal otomata bir sonraki durumun XOR veya XNOR mantıksal operatörü ile gerçekleştirilmesine olanak sağlayan kurallardan oluşur. Aşağıdaki Tablo 1.1'de toplamsal kurallar verilmiştir.

Tablo 1.1. Toplamsal hücresel otomata kuralları (Ganguly, 2003)

Lineer kurallar (XOR mantıksal operatörü ile)	Lineer kuralların tamlayanı (XNOR mantıksal operatörü ile)
Kural 60 : $q_i^{t+1} = q_{i-1}^t \oplus q_i^t$	Kural 195 : $q_i^{t+1} = \overline{q_{i-1}^t \oplus q_i^t}$
Kural 90 : $q_i^{t+1} = q_{i-1}^t \oplus q_{i+1}^t$	Kural 165 : $q_i^{t+1} = \overline{q_{i-1}^t \oplus q_{i+1}^t}$
Kural 102 : $q_i^{t+1} = q_i^t \oplus q_{i+1}^t$	Kural 153 : $q_i^{t+1} = \overline{q_i^t \oplus q_{i+1}^t}$
Kural 150 : $q_i^{t+1} = q_{i-1}^t \oplus q_i^t \oplus q_{i+1}^t$	Kural 105 : $q_i^{t+1} = \overline{q_{i-1}^t \oplus q_i^t \oplus q_{i+1}^t}$
Kural 170 : $q_i^{t+1} = q_{i+1}^t$	Kural 85 : $q_i^{t+1} = \overline{q_{i+1}^t}$
Kural 204 : $q_i^{t+1} = q_i^t$	Kural 51 : $q_i^{t+1} = \overline{q_i^t}$
Kural 240 : $q_i^{t+1} = q_{i-1}^t$	Kural 15 : $q_i^{t+1} = \overline{q_{i-1}^t}$

Yukarıdaki tabloda görülebileceği gibi lineer kurallar şöyledir; 60, 90, 102, 150, 170, 204 ve 240. Tablodaki kural denklemlerinde yer alan q_i^{t+1} i'ninci hücrenin $t+1$ zamanındaki durumunu(değerini) ifade etmektedir. Kural 60 için düşünülecek olursa; i'ninci hücrenin $t+1$ zamanındaki değeri, solundaki ve kendisinin t zamandaki değerlerinin XOR işlemine tabi tutulmasıyla elde edilir. Tablo incelenirken Kural 60 için yapılan açıklama dikkate alınırsa diğer kuralların XOR ve XNOR işlemine göre

nasıl bir sonraki durumu belirlediği anlaşılabilir. Toplamsal CA'nın lineer cebirsel araçları kullanımı ile alakalı birçok teorik çalışma yapılmıştır (Martin ve ark., 1984; Chaudhuri ve ark.,1997).

1.2.2. Lineer olmayan hücresel otomatlar

Lineer/Toplamsal hücresel otomatlar cebirsel işlemlere imkan tanımasına karşın sınırlı sayıda kuralla işlem yapmaktadır. Tablo 2.1'de görüldüğü gibi Lineer/Toplamsal otomatlar 14 farklı kural kullanmaktadır. İki durum ve üç komşuya sahip hücresel otomatlar, hücrelerin bir sonraki durumunu 256 farklı kurala göre belirleyebilir.

Lineer olmayan hücresel otomatlar, iki durum ve üç komşuya sahip hücresel otomatların sahip olabileceği 256 farklı kuralın hepsini kullanmak amacı ile geliştirilmiştir. Lineer olmayan hücresel otomata yaklaşımları genelde ikiye ayrılmaktadır; erişilebilirlik ağacı tabanlı yöntemler (Das ve ark., 2009) ve genetik algoritma tabanlı yöntemler (Ganguly ve ark., 2002). Bu tez çalışması kapsamında kural nitelemesi için erişilebilirlik ağacı tabanlı yöntemler kullanılmıştır.

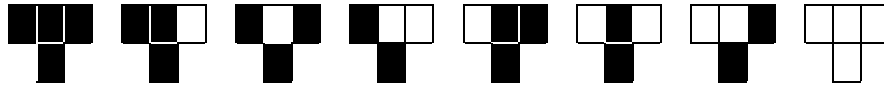
1.3. Hücresel Otomata Temelleri

Buraya kadar olan bölümlerde hücresel otomatların tarihçesinden ve gelişiminden bahsedilmiştir. Bu bölümde hücresel otomatların çalışma mantığı detaylı bir şekilde ele alınmıştır.

Farklı boyutlarda hücresel otomatlar araştırmacılar tarafından tasarlanmıştır. Hücresel otomatlar ilk Neuman tarafından önerildiğinde iki boyutlu olarak tasarlanmıştır. Daha sonra bir boyutlu ve üç boyutlu otomata türleri önerilmiştir. Bu tez çalışmasında bir boyutlu hücresel otomatlar üzerinde çalışılmıştır.

Wolfram ilk önerilen hücresel otomata yapısının karışık olduğunu belirtir (Wolfram, 2002). Bunu nedenini ise Neumann'ın biyolojik canlıların kendini yenileme yeteneğinin karmaşık bir yapıyla modellenebileceğini düşündüğünü ileri sürer. Kendi yaptığı çalışmalarda bunun aslında yanlış bir varsayım olduğunu ve karmaşık sistemlerin basit yapılar ile modellenbildiğini gösterdiğini belirtir. Wolfram bir boyutlu hücresel otomatlar üzerinde çalışmıştır (Wolfram, 2002). Yaptığı çalışmalarda basit kuralların karmaşık sistemleri modelleyebileceğini göstermiştir.

Hücresel otomatların önemli bir özelliği davranışlarının görsel bir yolla gösterilebiliyor olmasıdır. Hücresel otomatlar, renkleri siyah veya beyaz olan hücrelerden oluşan ızgara düzleminde çalışır. Hücrelerin durumu, renk bağlamında siyah veya beyaz renklerine, sayısal değer bağlamında 0 veya 1 değerlerine karşılık gelir. Her bir hücre bir sonraki durumu sol komşusu, sağ komşusu ve kendi durumuna göre belirlenir. Hücrelerin durum kavramını daha iyi anlamak için Şekil 1.2'yi incelemek faydalı olacaktır.



Şekil 1.2. Kural 254'e göre hücrelerin bir sonraki durumu

Yukarıdaki şekil, sekiz ayrı küçük şekilden oluşmaktadır. Her bir küçük şekil, üste üç hücre ve onların altında bir hücreden oluşmaktadır. İki durumlu hücresel otomatlar için üç hücrenin alabileceği sekiz farklı durum vardır. Şekillerde yer alan üstteki üç hücre incelendiğinde sekiz farklı renk dizilişi görülür. Üç hücrenin altında yer alan tek hücre, üç hücrenin ortasındaki hücrenin bir sonraki durumunu ifade eder.

Yukarıdaki şekiller için ortadaki hücrenin bir sonraki durumunun elde edilmesini basitçe şöyle açıklanabilir: Eğer üç hücreden birinin durumu siyah ise ortadaki hücrenin bir sonraki durumu siyah olur. Ancak üç komşu ve iki durumdan oluşan hücresel otomatlar 256 farklı kurala göre bir sonraki durumunu belirleyebilir. Her bir durum geçişi için bir tanım yapılması olanaksızdır. Bunun için daha iyi bir yöntem vardır. Her bir farklı durum geçişi için bir kural numarası verilir.

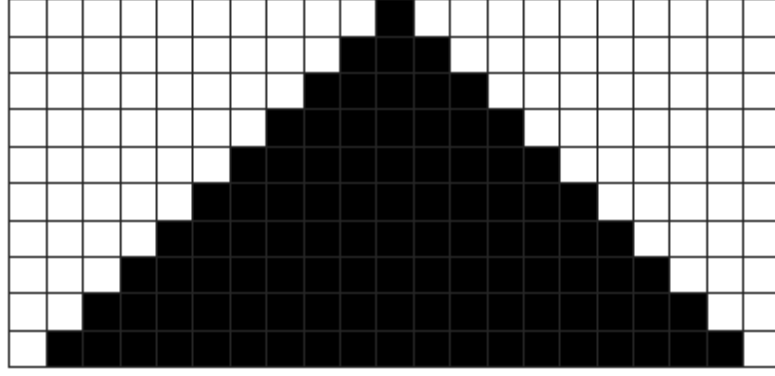
Kural numarasını yukarıdaki şekilde yer alan alttaki sekiz hücrenin durumuna göre belirlenir. Hücresel otomatlarda siyah hücreler 1 ve beyaz hücreler 0 rakamına karşılık gelir. Yukarıdaki şekilde alttaki tek hücrelerin rakam karşılıklarını yazarsak 11111110 değerini elde edebilir. Bu elde edilen sayısal değer ikilik tabanda bir sayı olarak kabul edilir. Bunun nedeni hücrelerin iki durumdan oluşuyor olmasıdır. İkilik tabandaki 11111110 sayısının onluk tabanda karşılığı 254'tür. Elde edilen bu sayı uygulanan kuralın numarasıdır. Örneğin; yukarıdaki Şekil 1.2'de ortadaki hücrenin bir sonraki durumu Kural 254'e göre elde edilmiştir

Üç hücre sekiz farklı durum alabildiği için kural değerinin ikili tabanda sekiz bitten oluşabilir. Bu, $2^8=256$ farklı kural uygulanabileceği anlamına gelir. Uygulanabilecek kural sayısı ile komşu sayısı arasında üstel bir bağıntı vardır. Komşu sayısına k diyecek olursak kural sayısı 2^k olacaktır. Örneğin; kendisi, sol, sağ, üst ve alt hücre komşuluklarından oluşan bir hücresel otomata için uygulanabilecek kural $2^5=4294967296$ sayısı kadardır.

Hücrelerin bir sonraki durumunun elde edilmesi tekrarlı işlemlerle devam eder. Bu noktada, zaman kavramı devreye girer. Yukarıdaki Şekil 1.2’de yer alan küçük şekillerdeki üstteki üç hücrenin mevcut durumunu zamansal olarak t ile ifade edilirse, alttaki hücre, orta hücrenin $t+1$ zamanındaki alacağı durum olacaktır. Her bir hücrenin $t+1$ zamanındaki değeri t zamanındaki kendi, sol ve sağ komşularının değerine göre belirlenir. i ’nci bir hücrenin t zamanındaki durumunu S_i^t olsun, hücrenin $t+1$ zamanındaki değeri $S_i^{t+1}=f_i(S_{i-1}^t, S_i^t, S_{i+1}^t)$ eşitliği ile elde edilir. f_i fonksiyonu, üç hücreye uygulanacak kurala göre sonuç verir. Şekil 1.2’de f_i fonksiyonun kullandığı kural 254’tür.

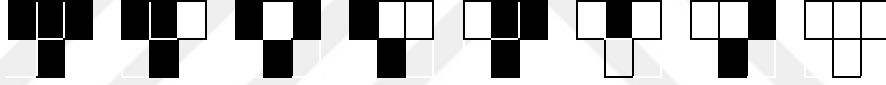
CA çalışmalarında hücrenin bir sonraki durumunun belirlenmesini “fonksiyon” yerine “kural” sözcüğü ile ifade edilir. Bu çalışmada bir hücrenin bir sonraki durumunun belirlenmesini ifade etmek için “kural” sözcüğü kullanılmıştır. Örneğin bir sonraki durum için 60 değerinin ikili karşılığı kullanılacaksa “Kural 60 göre bir sonraki durum” şeklinde ifade edilecektir.

Hücresel otomatların gizemi kuralların tekrarlı bir şekilde uygulanmasında saklıdır. Yukarıdaki şekilde verilen Kural254’ün 10 adım sonra alacağı şekil aşağıda Şekil 1.3’te verilmiştir.



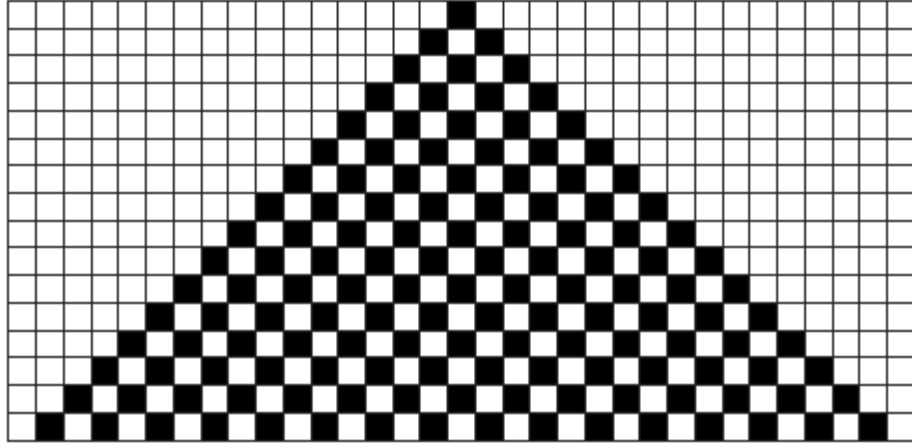
Şekil 1.3. Kural 254'ün 10 adım sonra ürettiği örüntü

Yukarıdaki şeklin başlangıç durumu tek bir siyah hücreden oluşur. Bu siyah hücre en üstteki satırda ortadaki hücrede yer alır. Bu hücresel düzleme Kural 254 uygulandığında yukarıdaki örüntü elde edilir. Şimdi başka kuralları bu hücresel düzleme uygulayarak neler olduğunu görelim. Kural 250'yi hücresel düzleme uygulayalım. Kural 250'nin bir sonraki durumun elde edilmesini Şekil 1.4'teki gibi gösterilebilir.



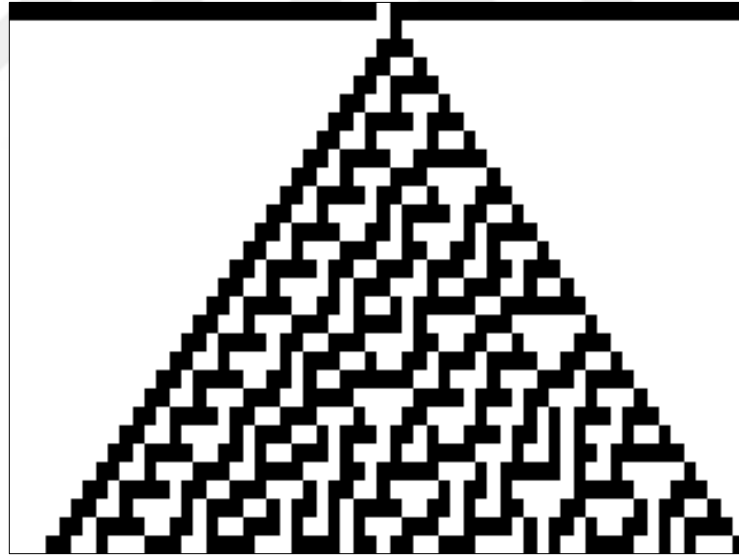
Şekil 1.4. Kural 250'ye göre hücrelerin bir sonraki durumu

Hücresel otomata hücrelerinin sahip oldukları renklerin bir rakama karşılık geldiğini yukarıda belirtilmişti. Siyah hücreler 1 ve beyaz hücreler 0 rakamına karşılık gelir. Yukarıdaki şekilde, ortadaki hücrenin sekiz farklı duruma göre bir sonraki durumu verilmiştir. Bu durumlar rakamlarla gösterilecek olursa; 11111010 biçiminde olacaktır. Bu ikilik tabanda bir sayıdır. 11111010 sayısının onluk tabanda karşılığı 250'dir. Bu nedenle yukarıdaki durum geçişi Kural 250'ye göre yapılmıştır denilir. Kural 250'nin hücresel düzleme on defa uygulanması ile elde edilen örüntü aşağıdaki Şekil 1.5'teki gibi olacaktır.



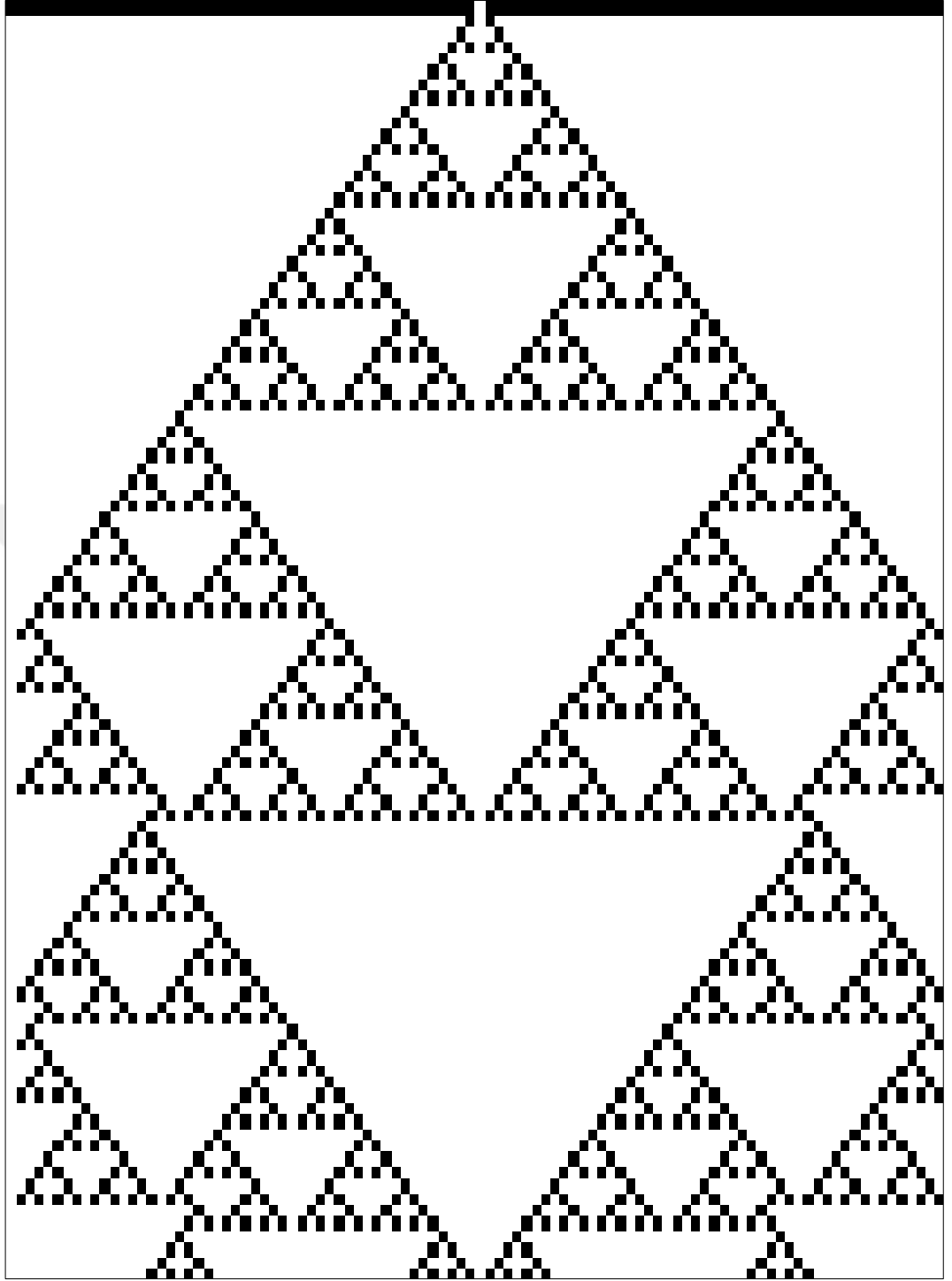
Şekil 1.5. Kural 250'nin 10 adım sonra ürettiği örüntü

Yukarıdaki şekilde Kural 250'nin hücresel düzleme on defa uygulanması sonucu elde edilen örüntü yer almaktadır. Şimdiye kadar elde edilen örüntülerin basit formlarda olduğu düşünülebilir. Aşağıda Şekil 1.6'da Kural 30'a göre elde edilen örüntüye yer verilmiştir.



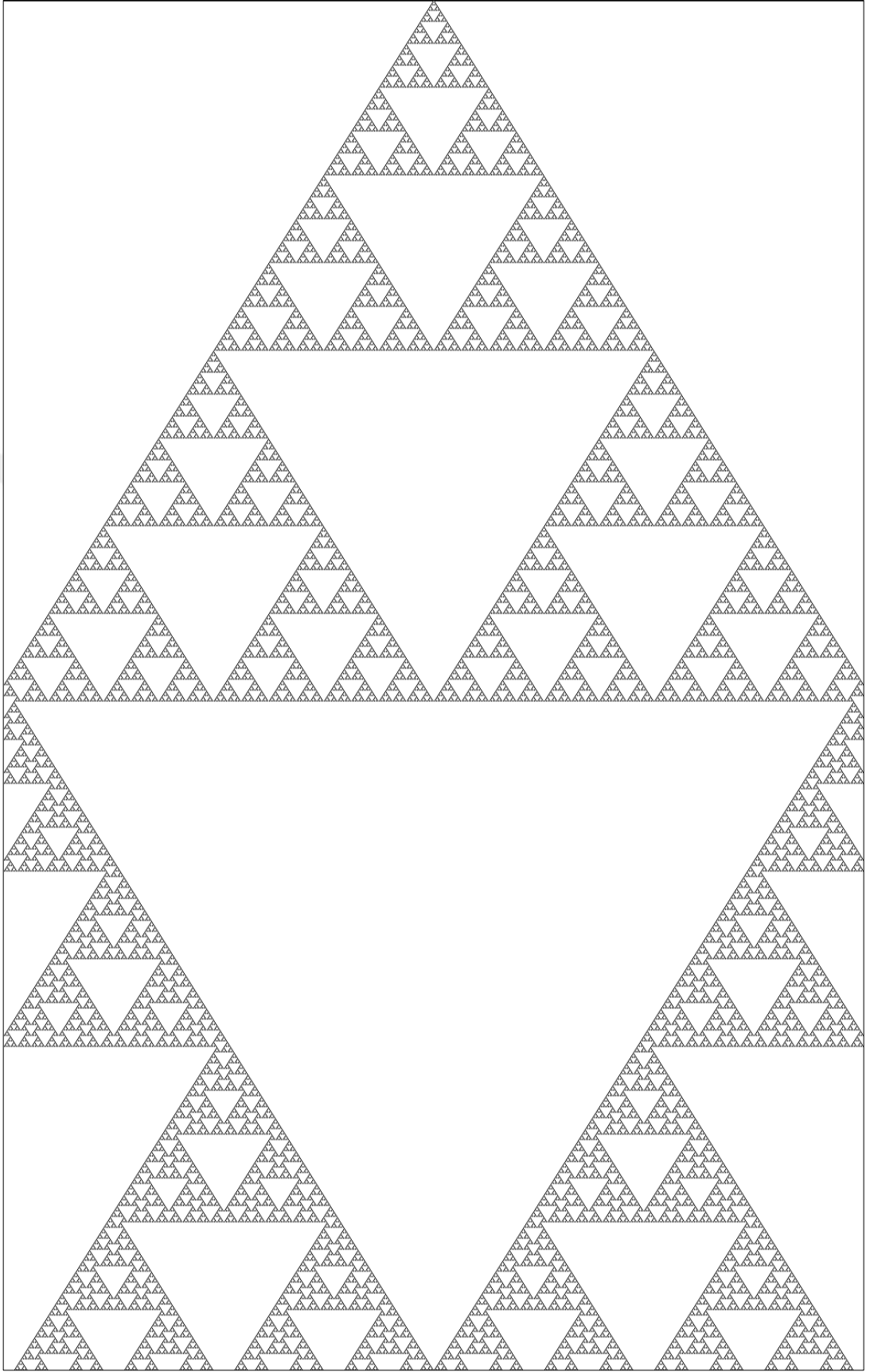
Şekil 1.6. Kural 30'un 50 adım sonra ürettiği örüntü

Yukarıdaki şekilde Kural 30'un hücresel düzleme otuz defa uygulanması sonucu elde edilen örüntü yer almaktadır. Tekrar sayısı arttıkça elde edilen örüntüler daha dikkat çekici hale gelmektedir. Aşağıda Şekil 1.7'de Kural 90'a göre elde edilen örüntüye yer verilmiştir.



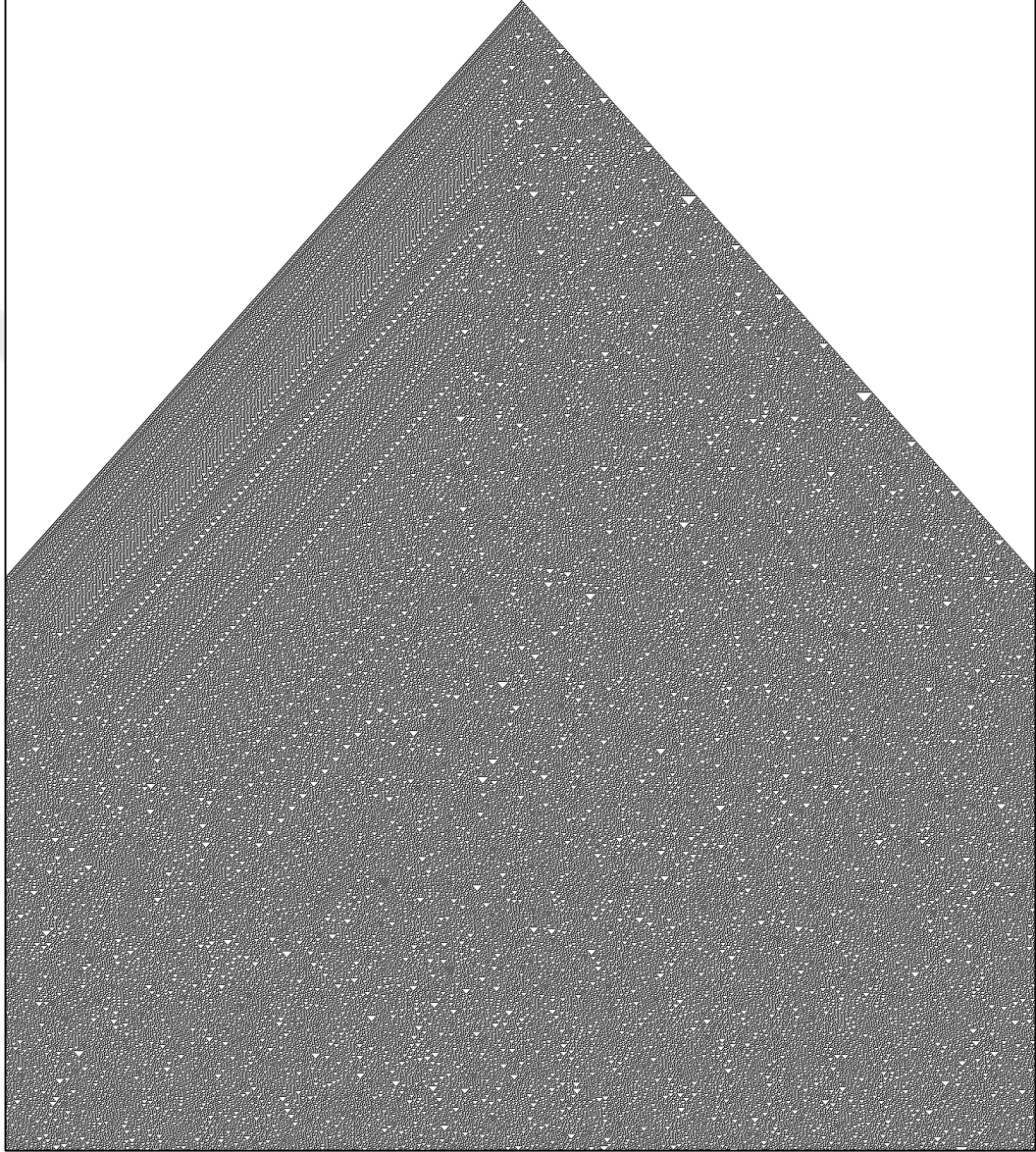
Şekil 1.7. Kural 90'nın 100 adım sonra ürettiği örüntü

Yukarıdaki şekilde Kural 90'nın hücresel düzleme yüz defa uygulanması sonucu elde edilen örüntü yer almaktadır. Daha önceki örüntülerden farklı olarak belirgin iç içe yapıya sahiptir. Tekrar sayısı ne kadar fazla olursa olsun iç içe yapı bozulmamaktadır. Aşağıdaki Şekil 1.8'de Kural 90'nın bin adım sonra ortaya koyduğu örüntü yer almaktadır.



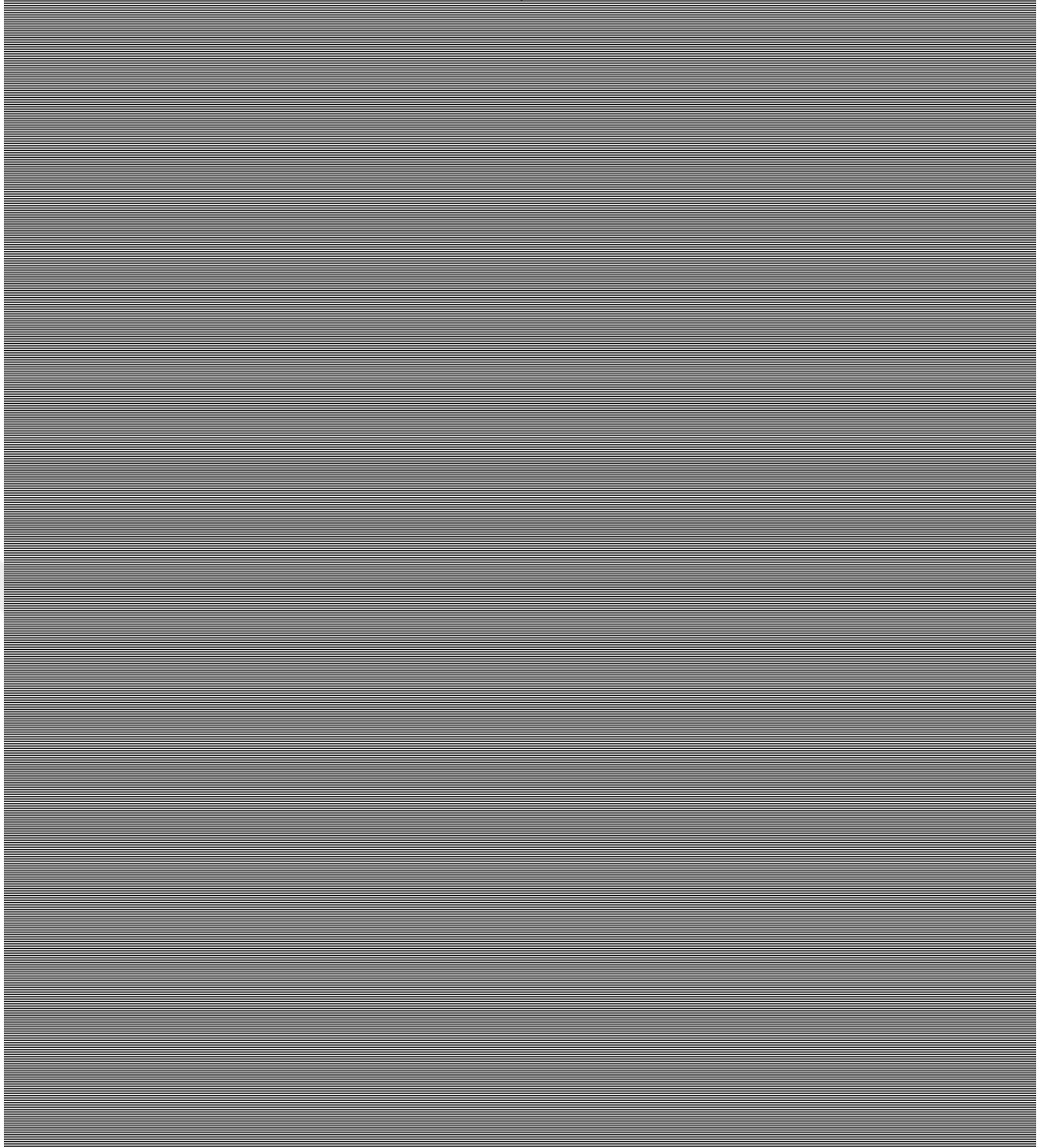
Şekil 1.8. Kural 90'nın 1000 adım sonra ürettiği örüntü

Yukarıdaki şekilde Kural 90'ın hücresel düzleme bin defa uygulanması sonucu elde edilen örüntü yer almaktadır. Wolfram kurallarının ürettiği örüntüler içerisinde kendisini en çok etkileyen kuralın Kural 30 olduğunu belirtir. Bu kural düzensizlik içerisinde bir düzen barındırır. Aşağıdaki Şekil 1.9'da Kural 30'nu ürettiği örüntü yer almaktadır.



Şekil 1.9. Kural30'un 1500 adım sonra ürettiği örüntü

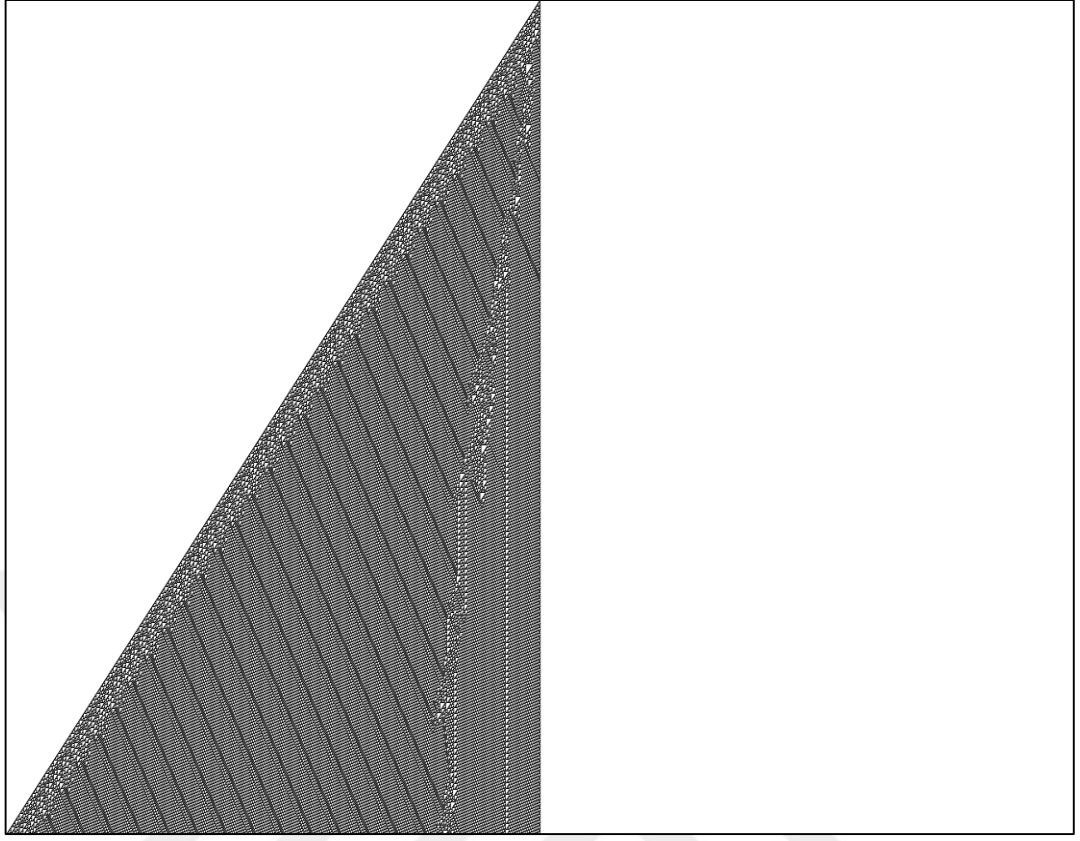
Yukarıdaki şekilde Kural 30'un hücresel düzleme 1500 defa uygulanması sonucu elde edilen örüntü yer almaktadır. Kural yapılarındaki az bir değişim kuralın ürettiği örüntü formunu önemli derecede etkileyebilmektedir. Örneğin Kural 30'un kural numarası değerini bir artırdığımızda Kural 31'i elde ederiz. . Aşağıdaki Şekil 1.10'da Kural 31'in ürettiği örüntü yer almaktadır.



Şekil 1.10. Kural 31'in 1500 adım sonra ürettiği örüntü

Yukarıdaki şekilde Kural 31'in hücresel düzleme 1500 defa uygulanması sonucu elde edilen örüntü yer almaktadır. Kural 30 ve Kural 31'in ürettiği örüntüler birbirinden çok farklıdır. Bu, kurallardaki basit değişimlerin üretilen örüntünün formuna önemli derecede etki ettiğini göstermektedir.

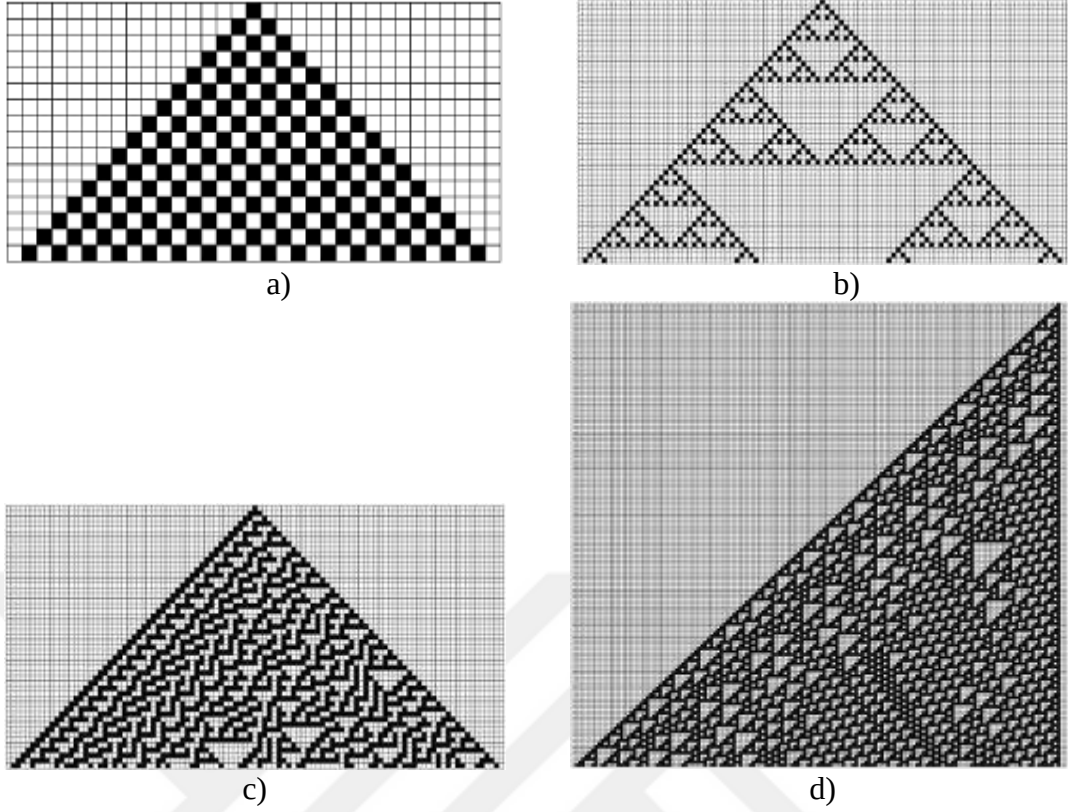
Kurallardan bazıları düzen ve düzensizliği aynı anda içeren örüntüler içerebilmektedir. Örneğin Kural 110 düzen ve düzensizliği içeren örüntüler üretmektedir. Aşağıdaki Şekil 1.11'de Kural 110'un ürettiği örüntü yer almaktadır.



Şekil 1.11. Kural 110'un 1000 adım sonra ürettiği örüntü

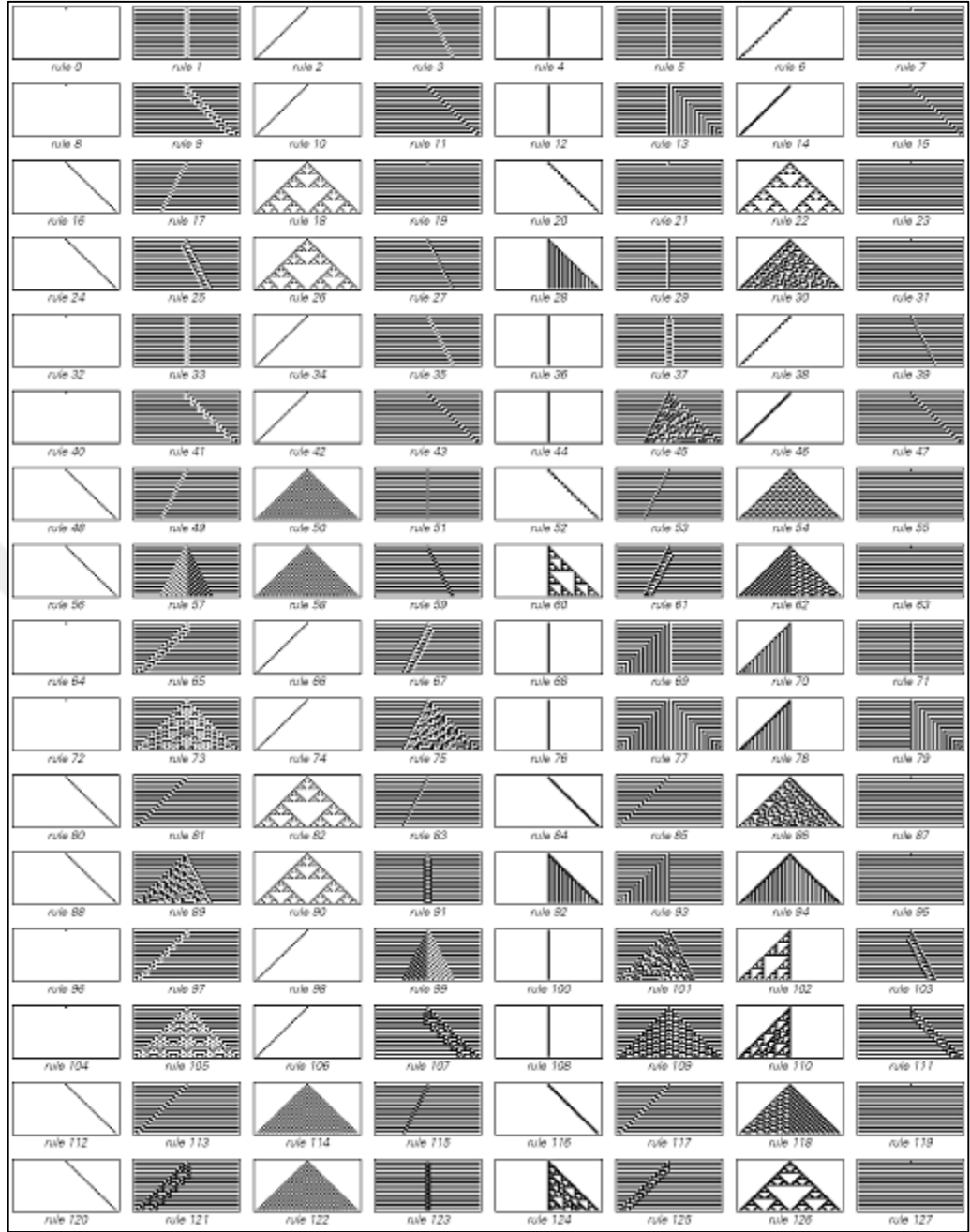
Yukarıdaki şekilde Kural 110'un hücresel düzleme 1000 defa uygulanması sonucu elde edilen örüntü yer almaktadır. Kural tarafından üretilen örüntü incelendiğinde düzen ve düzensizliğin iç içe olduğu bir yapı söz konusu olduğu görülür.

Wolfram iki durumlu ve üç komşudan oluşan hücresel otomatları 256 farklı kurala göre elde ettiği örüntüleri dört farklı gruba ayırmıştır. Birinci grupta örüntüler tekrar eden yapılara sahiptir. Bir iki tekrardan sonra örüntüler aynı şekilde devam etmektedir. İkinci gruptaki örüntüler iç içe geçmiş örüntü formundadır. Bu örüntü formları fraktal yapılar gibi düşünülebilir. Üçüncü gruptaki örüntüler düzen ve düzensizliği içinde barındıran örüntüler üretmektedir. Örüntüler belirli süre düzenli örüntüler oluşturduktan sonra adım sayısı arttıkça ortaya çıkan örüntüleri tahmin etmek imkansızlaşır. Wolfram en çok dikkatini çeken örüntü formlarının üçüncü grupta yer alan örüntüler olduğunu belirtir (Wolfram, 2002). Dördüncü gruptaki örüntüler yerel etkileşime göre örüntü formları ortaya koymaktadır. Aşağıdaki Şekil 1.12'de hücresel otomata kuralları ile elde edilen dört farklı örüntü grubuna ait görüntüler yer almaktadır.

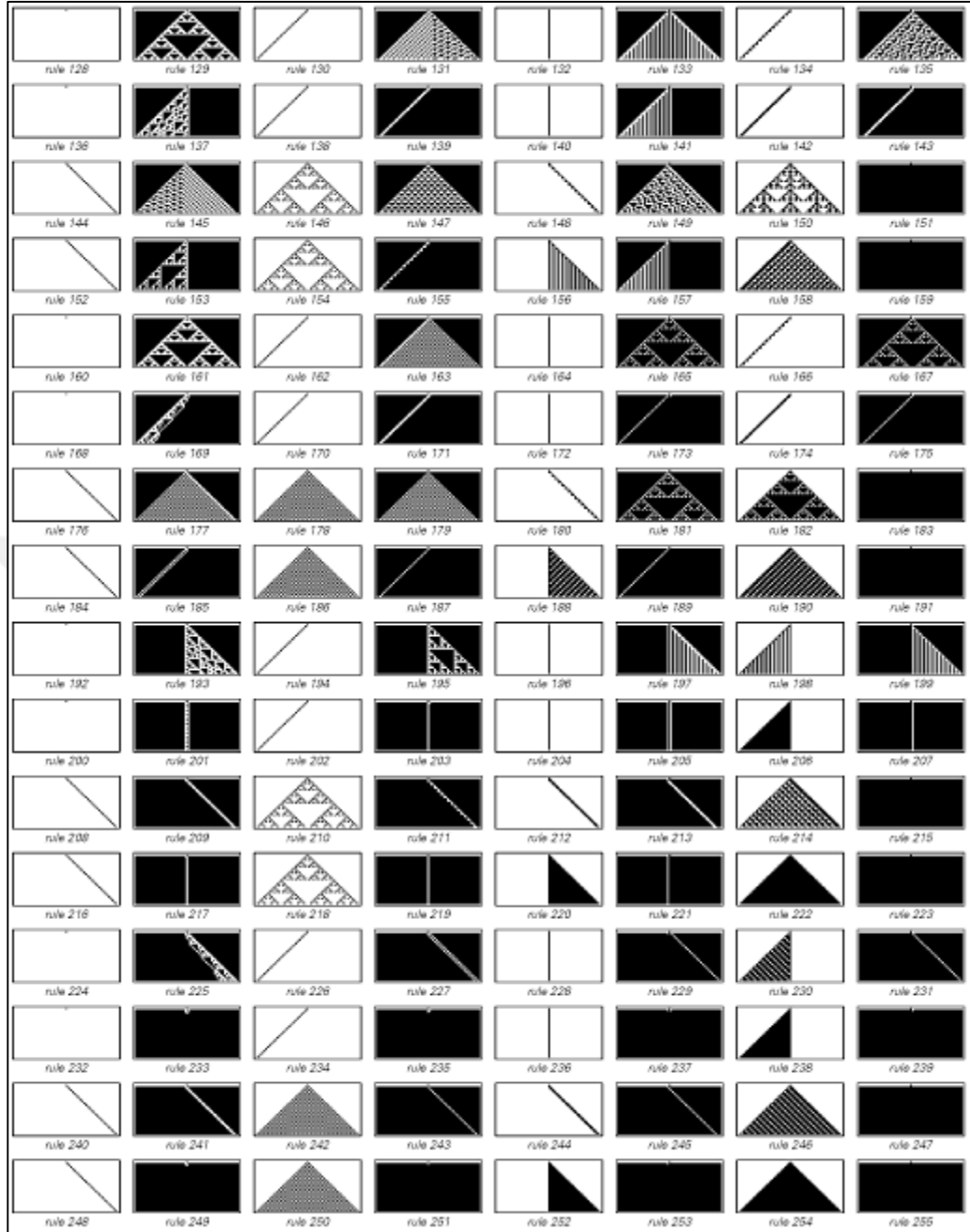


Şekil 1.12. Hücresel otomata kurallarının oluşturduğu dört farklı örüntü grubuna ait görüntüler (Wolfram, 2002) a) Tekrar (Kural250-16 adım sonra) b) İç içeli (Kural90-50 adım sonrası) c) Rasgele (Kural30-50 adım sonrası) d) Yerel yapılar (Kural110-150 adım sonrası)

Yukarıda bazı kurallara ait örüntülere yer vermiştir. İki durumlu ve üç komşuya sahip hücresel otomat kuralları ile elde edilen tüm örüntüler aşağıdaki Şekil 1.13 ve Şekil 1.14'te yer almaktadır. Şekil 1.13'te 0-127 arası kurallara ait örüntüler, Şekil 1.14'te 128-255 arası kurallara ait örüntüler yer almaktadır.



Şekil 1.13. 0-127 arası hücresel otomata kurallarına ait örüntüler (Wolfram, 2002)



Şekil 1.14. 128-255 arası hücresel otomata kurallarına ait örüntüler (Wolfram, 2002)

1.3.1. Toplamlı hücresel otomatlar(totalistic cellular automaton)

Bu kısma kadar iki durumlu hücresel otomatlardan bahsettik. Bu otomatların alabileceği kural sayısı 256'dır. Üç hücre sekiz farklı durum alabildiği için kural değeri ikili tabanda sekiz bitten oluşabilir. Bu, $2^8=256$ farklı kural uygulanabileceği anlamına gelir. Uygulanabilecek kural sayısı ile komşu sayısı arasında üstel bir bağıntı vardır.

Bu bağıntıyı formüle edecek olursak. Komşu sayısına k ve durum sayısına n diyelim kural sayısı n^k olacaktır. $2^3=256$ sonucunu verir.

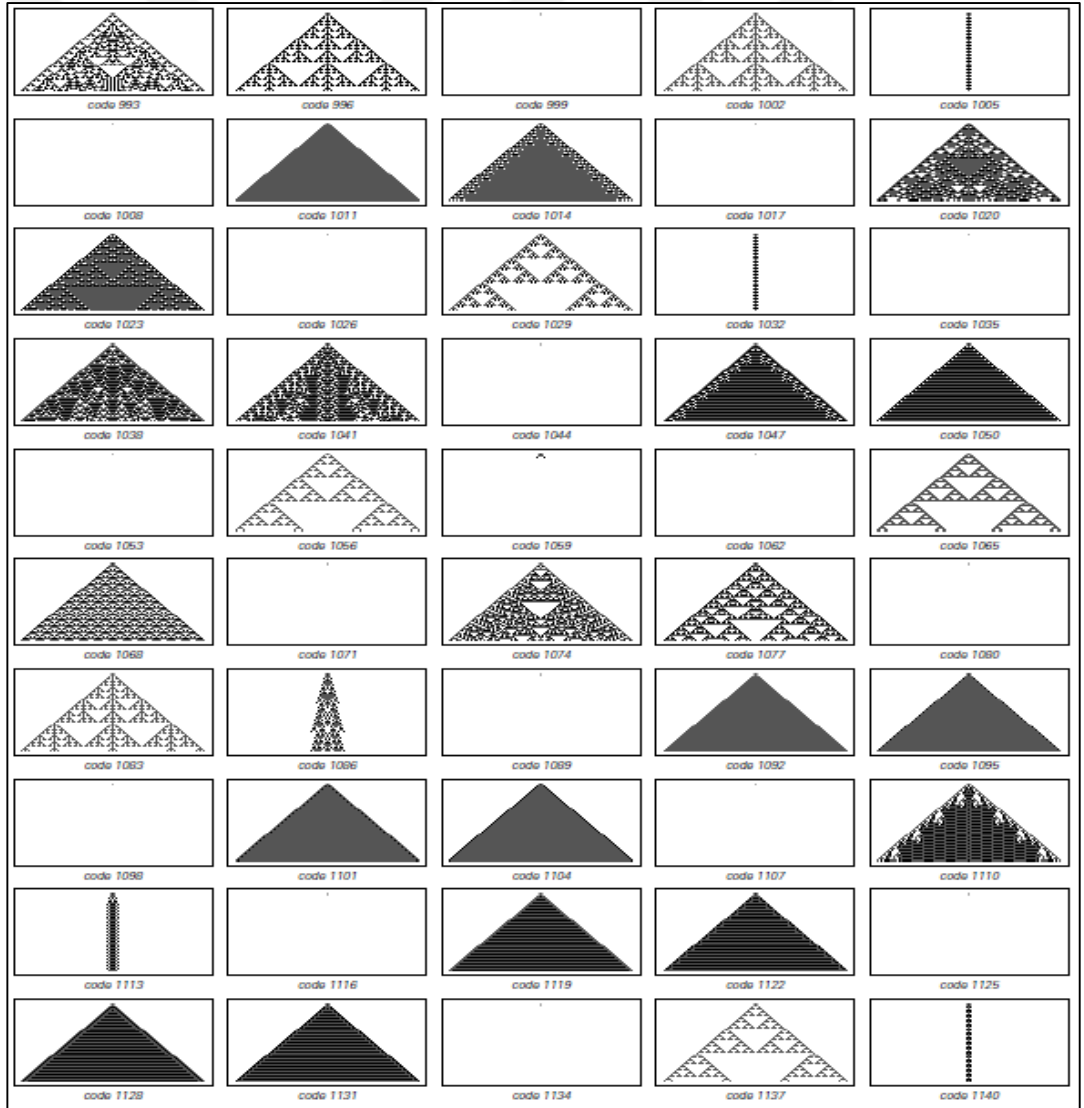
Wolfram bu tür hücresel otomatları “Temel Hücresel Otomat(Elementary Cellular Automaton)” olarak adlandırmıştır (Wolfram, 2002). Eğer durum sayısını üçe çıkarırsa sonuç ne olur? Bu durumda kural sayısı $3^3=7\ 625\ 597\ 484\ 987$ olacaktır. Bu kadar fazla kurala sahip olmak uygulamayı imkânsız hale getirir. Bunun için belirli şartlardaki kuralları alarak bu sayı azaltılabilir. Wolfram bunun için toplamı bir kural belirlemiştir.

Üç renkten oluşan hücrelerde 0 beyaz, 1 gri ve 2 siyah renkleri temsil etmiş olsun. Hücrenin bir sonraki durumu bu üç hücrenin toplamına göre elde edilsin. Buna göre kural sayısı 2187 olacaktır. Kural sayısını nasıl elde ettiğimiz biraz daha açalım. Tek boyutlu bir hücresel otomat kendi, sol ve sağ hücreler olmak üzere üç komşuya sahiptir. Bu hücreler k sayıda değer sahip olabilirse toplamı otomata için $3k-2$ farklı değere sahip olabilir. Böyle bir durum sayısı için k^{3k-2} farklı kural elde edilir. Eğer renk sayımız 3 tane ise elde edeceğimiz kural sayısı $3^{3 \times 3 - 2} = 2187$ olur. Kural sayısını nasıl elde ettiğimizi daha iyi anlamak için aşağıdaki Şekil 1.15’i incelemek faydalı olacaktır.

Aşağıdaki şekilde üç farklı renkten(durum) oluşan, üç komşulu tek boyutlu bir hücresel otomatın toplamı kurallara göre alabileceği değerler gösterilmektedir. Üç duruma sahip üç hücrenin alabileceği toplam 7 farklı diziliş vardır. Ancak üç hücrenin toplamı 0 ile 6 arasında değerler almaktadır. Yani 7 farklı değer almaktadır. Bunun için üçlük tabanda oluşan kurallar en fazla 7 basamaklı olur. Bu nedenle üç durumlu toplamı hücresel otomata göre $3^7=2187$ kural vardır. Aşağıdaki Şekil 1.15 tipik bir toplamı kuralın üçlük tabana göre nasıl elde edildiğini göstermektedir.

Üstteki satır hücrelerin alabileceği değerleri(renkleri) göstermektedir. Beyaz 0 değerini, gri 1 değerini ve siyah 2 değerini temsil etmektedir. Orta satırda hücrelerin toplamına göre bir sonraki durumda alacağı değerler gösterilmektedir.

Yukarıdaki şekilde verilen kuralı inceleyelim. Eğer komşu hücrelerin toplamı 0, 3, 4 veya 5 ise hücrenin bir sonraki durumu 0'dır. Eğer komşu hücrelerin toplamı 1 ise bir sonraki durum 2'dir. Eğer komşu hücrelerin toplamı 2 veya 6 ise bir sonraki 1'dir. Böyle bir kuralın kural numarası $(1 \times 729) + (1 \times 9) + (2 \times 3) = 744$ denklemi ile hesaplanır. Wolfram toplamalı kurallar için "code" ile başlayan numaralandırma vermiştir. Bu nedenle yukardaki şekilde verilen kural code744 olarak isimlendirilir. Aşağıdaki Şekil 1.17'de örnek toplamalı kurallar yer almaktadır.



Şekil 1.17. Üç renkten oluşan toplamalı hücresel otomata için örnek örüntüler (Wolfram, 2002)

Wolfram toplamli hücresele otomata ile elde edilen örüntülerin iki durumlu temel otomata ile elde edilen kurallardan biraz daha karmaşık davranışlar sergilediğini. Ancak kural sayısına göre beklenen karmaşıklığı göstermediğini belirtmektedir (Wolfram, 2002).

1.3.2. Sürekli hücresele otomatlar

Buraya kadar olan bölümlerde bahsettiğimiz hücresele otomatların tamamı ayırık bir yapıya sahipti. Siyah ve beyaz renklerinde olduğu gibi sınırlı sayıda forma sahipti. Sürekli hücresele otomatlar herhangi bir aralık için istenen sayıda sürekli değere sahip olabilir (Wolfram, 2002). Bu hücresele otomatların kullanımına esneklik sağlar.

Sürekli hücresele otomatlar mantığı hücre komşularının değerlerinin ortalamasının alınmasına dayanır. Tek boyutlu CA için üç komşu hücrenin değerlerinin ortalamalarının alınmasıyla elde edilir. Diğer hücresele otomatların anlatımında olduğu gibi başlangıçta tek bir siyah hücre vardır. Siyah hücrenin altındaki değer sol, sağ komşuları ve kendi değerinin toplamının ortalamalarının alınmasıyla elde edilir. Yani başlangıçtaki siyah hücrenin altındaki hücrenin değeri; $(0+1+0)/3=0,333$ ile elde edilir. Aşağıdaki Şekil 1.18’de sürekli hücresele otomatlar ile hücrelerin aldığı değerler görülmektedir. Hücrelerin sahip oldukları değerler istenen aralıktaki değerlere eşitlenebilir. Görüntüler için gri seviye değerler 0 ile 255 arasındadır. Elde edilen 0,333 değerinin gri seviye eşitlenmesi $0,333 \times 255 = 84,9$ işlemi ile elde edilebilir.

0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0,333	0,333	0,333	0	0	0	0
0	0	0	0,111	0,222	0,333	0,222	0,111	0	0	0
0	0	0,037	0,111	0,222	0,259	0,222	0,111	0,037	0	0
0	0,012	0,049	0,123	0,198	0,235	0,198	0,123	0,049	0,012	0
0,004	0,021	0,062	0,123	0,185	0,21	0,185	0,123	0,062	0,021	0,004

Şekil 1.18. Her hücrenin beyaz (0) ve siyah (1) arasında herhangi bir gri seviyeye sahip olabileceği sürekli bir hücresele otomat. Burada gösterilen kural, her bir hücrenin yeni gri seviyesini, kendi ve yakın komşularının gri değerlerinin ortalamaları olacak şekilde elde edilir (Wolfram, 2002)

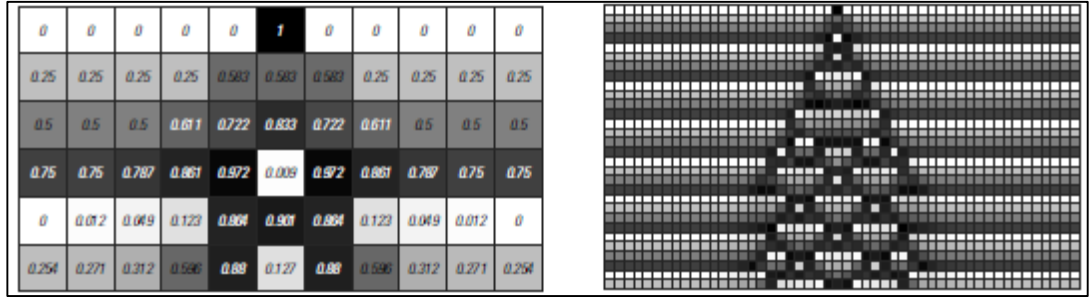
Yukarıdaki şekildeki sürekli hücresel otomatın kolay bir kuralı var. Bunun yerine biraz daha karmaşık bir kural belirleyebilir. Örneğin; elde edilen ortalama $3/2$ ile çarpılarak sonuç elde edilebilir. Eğer sonuç 1'den büyük ise noktadan sonraki kısmı alınır. Aşağıdaki Şekil 1.19'da bu kuralla elde edilmiş hücre değerleri görülmektedir.

0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0,5	0,5	0,5	0	0	0	0
0	0	0	0,25	0,5	0,75	0,5	0,25	0	0	0
0	0	0,125	0,375	0,75	0,85	0,75	0,375	0,125	0	0
0	0,063	0,25	0,625	0	0,188	0	0,625	0,25	0,063	0
0,031	0,156	0,469	0,438	0,406	0,094	0,406	0,438	0,469	0,156	0,031

Şekil 1.19. Biraz daha karmaşık kurallara sahip sürekli bir hücresel otomat. Kural, her bir hücrenin yeni gri seviye değerini, hücrenin ve komşularının ortalama gri seviye değerinin kesirli kısmının $3/2$ ile çarpımından elde edilir. Şekildeki görüntü, tek bir siyah hücreden başlayarak, bu kuralın önemli karmaşıklık davranışları ortaya koyduğunu göstermektedir (Wolfram, 2002)

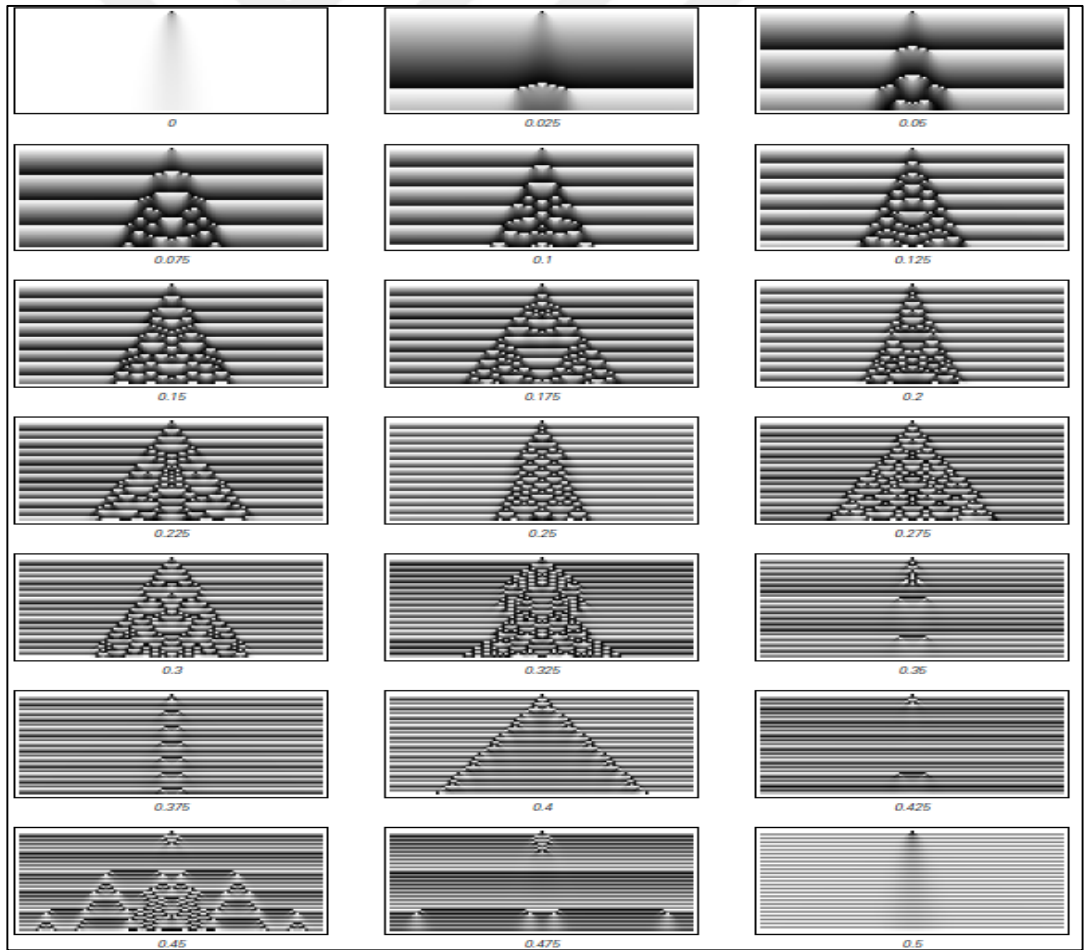
Yukarıdaki şekil bize sürekli hücresel otomatlar ile daha önceki iki durumlu hücresel otomatlarla gördüğümüz karmaşık davranışları elde edebileceğimizi göstermektedir.

Aslında, sürekli hücresel otomatlar ile oldukça karmaşık bir davranış üretmek için yalnızca çok basit kuralların yeterli olduğu ortaya çıkıyor. Aşağıda Şekil 1.20.'de yer alan örnek, bir hücre için yeni gri seviyeyi belirleyen bir kuralı göstermektedir; bu, hücrenin ve yakın komşularının ortalama gri seviyesine sabit ekleyerek ve sonucun kesirli kısmını alarak gerçekleşir.



Şekil 1.20. Kuralı bir hücre ve yakın komşularının ortalamasına $\frac{1}{4}$ sabit değerini ekleyen ve sonucun kesirli kısmını alan sürekli bir hücresel otomat. Arka plan her 4 basamağı basitçe tekrarlar, ancak ana desen karmaşıktır ve pek çok açıdan rastgele biçimdedir (Wolfram, 2002)

Aşağıdaki Şekil 1.21’de yer verilen sürekli hücresel otomatlarda dikkate değer bir davranış çeşitliliği görülür. Bazen davranış tamamen tekrarlanır, ancak çoğunlukla etkili rastgele görüntü sergilerler.



Şekil 1.21. Yukarıdaki şekillerdeki resimlerde olduğu gibi aynı kurallara sahip, ancak farklı sabitler eklenerek oluşturulan sürekli hücresel otomatlar. Buradaki sabit değerlerin iki durumlulu otomatlarda davranış biçimini belirleyen ikili dizilerin çok küçük olduğunu unutmamak gerekir (Wolfram, 2002).

1.3.3. Hücresel otomatlar neden “Yeni Bir Bilim Türü” olarak adlandırılıyor?

Hücresel otomata kurallarının ürettiği yukarıda verdiğimiz şekiller basit kuralların karmaşık davranışlar sergileyebileceğini göstermektedir. Wolfram hücresel otomatlar üzerine yaptığı çeyrek yüz yıla yakın çalışmaları sonucu çok önemli bir iddiada bulunmaktadır. Wolfram hücresel otomata temelli çalışmalarını Yeni Bir Bilim Türü(A New Kind of Science) olarak adlandırmaktadır (Wolfram, 2002). Aşağıdaki paragraflarda Wolfram’ın iddiası için yaptığı yorumlara yer verilmiştir.

Günlük tecrübelerimiz, karmaşık görünen bir nesnenin karmaşık bir şekilde oluşturulmuş olmasını düşünmemizi sağlamıştır. Mesela, karmaşık bir mekanik cihaz görürsek, cihazın yapılması için kullanılan planların da bir şekilde karmaşık olması gerektiğini varsayıyoruz.

Yukarıdaki şekillerde elde edilen sonuçlar, en azından böyle bir varsayımın tamamen yanlış olabileceğini göstermektedir. Gördüğümüz örüntüler gerçekte çok basit kurallara göre oluşturulmuştur - tek bir siyah hücreden başlamanız ve daha sonra basit bir hücresel otomat kuralını uygulamamız yeterlidir. Bununla birlikte, bu kuralların ortaya koyduğu şey karmaşıklığın muazzam bir seviyesini göstermektedir.

Normal sezgilerimizi başarısız kılan şey nedir? En önemli nokta, çoğunlukla yapı oluşturma ve mühendislik yapma tecrübelerinden kaynaklanmaktadır. Bu noktada, yukarıda elde edilen örüntülerin basit kurallarla elde edilemeyeceğini düşünmek mümkündür.

Normalde, elde etmek istediğimiz davranışları üretebilecek bir sistem tasarlamaya çalışıyoruz. Ancak bunu güvenilir bir şekilde yapmak için, kendimizi davranışını kolaylıkla anlayabildiğimiz ve tahmin edebildiğimiz sistemlerle sınırlamak zorundayız - çünkü bir sistemin nasıl davranacağını öngöremedikçe, sistemin istediğimizi yapacağından emin olamayız.

Ancak mühendisliğin aksine, doğa için böyle bir kısıtlama yoktur. Dolayısıyla, yukarıda verdiğimiz sistemlerin ortaya çıkmasını engelleyecek hiçbir şey yoktur. Aslında bu kitabın(A New Kind of Science) önemli sonuçlarından biri, bu tür sistemlerin gerçekten doğada çok yaygın olduğunu göstermesidir.

Bununla birlikte, temel kurallardan ve detaylardan haberdar olduğumuz tek durum, bir şeyler inşa ettiğimiz veya mühendislik yaptığımız durumlardır. Yukarıda yer alan sistemlere ilişkin hiçbir sezgiye günlük yaşantımızda sahip olamayız.

Peki, o zaman bu sistemlerde oluşan fenomen hakkında bize bir ipucu verecek deneyimlere sahip miyiz? Akla gelen en yakın pratik bilgisayarların hesaplama yetenekleridir. Çünkü bilgisayarların çok karmaşık görevleri gerçekleştirebileceklerini biliyoruz. Temel donanım düzeyinde, tipik bir bilgisayar, basit mantıksal, aritmetik ve diğer komutların sadece birkaç türünü gerçekleştirebilir. Ve bir dereceye kadar, bu türden talimatların çok sayıda yürütülmesiyle, her tür karmaşık işlem gerçekleştirilir.

Fakat önemli bir farkı belirtmek gerekir. Bilgisayar tarafından yürütülen tek tek makine komutları oldukça basit olabilir, bir program tarafından tanımlanan komutların sırası uzun ve karmaşık olabilir. Gerçekten de -en az mühendisliğin

diğer alanlarında olduđu gibi- yazılım geliřtirmede karmařık bir grevdir. Bilgisayara zor grevler yaptırmak iin karmařık bir program kodlamak gerekebilir.

Hücresele otomata gibi bir sistemin temelini oluřturan kurallar, bilgisayarların alıřmasının temellerini oluřturan makine komutlarının bir benzeri olarak düşünülebilir ve hücresele otomatların bařlangı durumu bir program olarak düşünülebilir. Yukarıda gördüğümüz hücresele otomatlarda temel kuralların basit olabilmeksinin yanında bařlangı kořulları da basit olabilir. Sadece tek bir siyah hücreyle üretilen davranıř ok yüksek karmařıklıđa sahip olabilir. Hücresele otomatları aıklamak kolaydır. Ancak kuralları tekrarlı uyguladığımızda ortaya ıkan öüntüler basitliğin tam zıttı gibi görünüyor.

Hücresele otomata olgusunu anlamak iin bazı mevcut sezgi türlerini kullanmanın mümkün olacağını düşünüyor olabiliriz. Fakat aslında, ihtiya duyulan şeyleri sađlayan gündelik tecrübe türü yok gibi görünüyor. Ve bu yüzden yepyeni bir sezgi geliřtirmeye alıřmaktan bařka řansımız yok.

Wolfram mevcut tecrübelerimizi ve kabullerimizi hücresele otomatlar hesaplama yeteneklerini anlamamızı engellediğini belirtiyor. Karmařık sistemlerin ancak karmařık yöntemlerle modellenebileceğini düşünmemiz bunun en önemli nedeni olduğunu belirtiyor. Hücresele otomatların hesaplama yöntemlerinin daha önce keřfedilememesini birçok neden bađlıyor. Ařağıdaki paragraflarda hücresele otomatların hesaplama yeteneklerini řimdiye kadar keřfedilememesinin nedenleri iin yaptığı yorumlara yer verilmiştir.

Bilim tarihinde yeni teknolojilerin eninde sonunda temel bilimin yeni alanlarını geliřtiren şey olduđu kanısı oldukça yaygındır. Örneğin, modern astronomiye teleskop teknolojisi ve modern biyolojiye mikroskop teknolojisi yol amıştır. Aynı řekilde, bu kitapta(A New Kind of Scinece) anlatılan yeni bilim türüne yol aan bilgisayar teknolojisidir.

Gerçekten de, yukarıda elde edilen öüntülerin bir kısmı, bir bakıma bilgisayarları kullanarak yapılabilecek en basit deneylerin bir sonucu olarak görülebilir. Ama neden bu tür basit denemeler daha önce hiç yapılmadı? Bunun bir nedeni, mevcut herhangi bir bilim veya matematik alanının ana akımında bu konuların bulunmamasıdır. Fakat daha önemli bir neden, geleneksel bilimdeki standart sezgilerin, sonuçlarının ilgin olacağını düşünmek iin herhangi bir neden vermemesidir.

Gerçekten deđerli oldukları bilinseydi, yukarıda yapılan deneylerin birođu bilgisayarlar var olmadan ok önce yapılabilirdi. Biraz sıkıcı olabilse de, hücresele otomat gibi bir şeyin davranışını elle özmek kesinlikle mümkündür. Ve aslında bunun iin, matematiğin veya bařka hiçbir sofistike fikir gerekmemektedir: Tek gereken, basit kuralların tekrar tekrar nasıl uygulanacağı konusunun anlaşılmadır.

Süsleme sanatının tarihsel örneklerine baktığımızda birçok hücresele otomata ait davranışın yüzyıllar hatta bin yıl önce özölemeyeceğini düşünmek iin ok az neden var gibi görünüyor. Ve belki de bir gün, Kural 30 ile oluřturulmuş bazı Babil eserleri ortaya ıkar. Ancak bu konuda o kadar emin değılim. ünkü sanırım Kural 30'un ortaya koyduğu öüntüler eski zamanlarda göröldüyse, bilim, aslında getiği yollardan ok farklı bir yol izlemiş olurdu.

Antik çağın erken döneminde bile, basit soyut kuralların doğal sistemlerin davranışını yeniden üretip üretemeyeceğini görmek için yapılan girişimler olabilir. Ancak şimdiye kadar denenen kural türlerinin standart geometri ve aritmetik ile ilişkili olduğunu söyleyebiliriz. Bu tür kurallar kullanıldığında astronomide gözlemlenen düzensizliklerin açıklamak için oldukça basit davranışlar elde edilebilir, ancak doğanın başka yerlerinde görülen şeylerin çoğunu yakalayamazlar.

Belki de bundan dolayı, doğal dünyanın birçok yönünün basitçe insan anlayışının ötesinde olduğu varsayılmaya başlandı. Ancak nihayet 1600'lü yılların sonlarında hesaplamalara dayanan başarılar bu inancı devirmeye başladı. Çünkü hesaplamayla, insan düşüncesinin yarattığı soyut kurallar kullanılarak doğal dünyadaki her türlü fenomeni yeniden üretme konusunda gerçek bir başarı elde edildi.

Fakat işe yarayan kurallar, belirli türdeki matematik denklemlerine dayanan oldukça karmaşık kurallardı. Bu kuralların gelişmişliği, doğal sistemlerin davranışını yeniden üretmede daha basit kuralların kullanışlı olmasının neredeyse mümkün olmadığı inancını bilinçaltında geliştirmeye başladı.

1700'lü ve 1800'lü yıllarda fiziksel fenomenleri analiz etmek için matematik denklemlerine dayalı kurallar kullanmada giderek artan bir başarı elde edildi. 1900'lü yılların başında fizikte elde edilen olağanüstü sonuçlar matematik denklemler sayesinde olunca doğal dünyanın her alanının böyle denklemlerle açıklanacağına dair neredeyse evrensel bir inanç ortaya çıktı. Bu yaklaşımın işe yaramadığı birçok fenomen vardı. Fakat genel kanı, gerekli hesaplamalar yapılabilirse, matematik denklemleri ile ilgili bir açıklama sonunda bulunabilirdi.

1940'lı yıllardan başlayarak elektronik bilgisayarların geliştirilmesi, yapılabilecek hesaplama aralıklarını büyük ölçüde genişletti. Ancak hayal kırıklığı yaratan bir şekilde, denenen gerçek hesaplamaların çoğunda temelde yeni bilgiler mevcut değildi. Sonuç olarak, birçok insan bilgisayarların temel bilim konularına gerçekten katkıda bulunamayacağına inanmaya başladı.

Ancak kaçırılan önemli nokta, bilgisayarların yeteneklerinin matematik denklemlerinin sonuçlarını incelemekle sınırlı olmadığıdır. Gerçekten de, hücresel otomat kurallarının ortaya koyduğu örüntüler çok basit programlarla elde edilebilir.

Geriye doğru bakıldığında, doğal sistemlerin modellenmesi için basit programlar kullanma fikri, bilgi işlemenin ilk günlerinde görülmemesi ironidir. Hücresel otomatlar gibi sistemler, matematiksel denklemlerden önce bilgisayarlarda ele alınması daha kolay olurdu. Fakat mesele, bilgisayarların ilk kullanıldığı zamanlarda pahalı bir meta olmasıydı ve bu nedenle, herhangi bir şey denemektense, köklü matematiksel modeller denemek düşünülmüştü. Ancak 1970'lerin sonlarına gelindiğinde durum değişti ve büyük miktarda bilgisayar kolaylıkla erişilebilir hale geldi. Ve bu, hücresel otomata ilişkin deneylerime başlamama imkan verdi.

Yukarıda değindiğim gibi, prensipte hücresel otomata çalışmak için bir bilgisayar kullanmayı gerektiren hiçbir şey yoktur. Fakat pratik bir mesele olarak, çağdaş insanların hücresel otomata örüntülerini elle üretme konusunda yeterli sabra sahip olduklarını hayal etmek zordur.

Belirli bir olgunun temel özellikleri hakkında açık bir fikriniz varsa, oldukça ayrıntılı deneyler yaparak daha fazla ayrıntı elde edebilirsiniz. Ancak benim tecrübelerime göre, önceden bilinmeyen olguları bulmanın tek yolu; çok sistematik ve genel deneyler yapmak ve sonuçlara olabildiğince az önyargı ile bakmaktır. Hücresel otomata örüntüleri üretmek oldukça basit bir bilgisayar

teknolojisi gerektirse de, büyük çaplı deneyler yapmak daha gelişmiş teknolojiler gerektirir.

Hücresele otomata ile elde edilen sonuçların, bilgisayar teknolojisinin 1980'lerde geldiği noktaya gelmeden çok önce keşfedilmesinin zor olmasının iki farklı nedeni vardır: Birincisi, gerekli bilgisayar deneylerinin yapılmasının kolay olmayacağını düşünülmesidir. İkincisi, bilgisayar hesaplamaları ile ilgili olan sezgilerin varsayımdır. Bu varsayıma göre bilgisayar hesaplamaları yoğun bir pratik hesaplama sonucu elde edilebilir.

İki boyutlu hücresele otomata versiyonları 1950'lerin başında biyolojik sistemler için mümkün olan en ideal modeller olarak kabul edildiği görülüyor. Ancak, 1980'li yıllardaki çalışmalarımıza kadar, hücresele otomatlar üzerine gerçekleştirilen ciddi araştırmalar, esasen oldukça basit davranışlara neden olan karmaşık kurallar dizilerinden oluşuyordu.

Hücresele otomatlar kullanılarak karmaşık davranışların oluşup oluşamayacağı sorusu zaman zaman gündeme gelmiştir. Ancak, mühendislikten gelen sezgilerin temelinde, büyük bir karmaşıklık elde etmek için, temel karmaşık kurallara sahip olmak gerektiği genel olarak kabul edilir. Sonuç olarak, basit kurallarla hücresele otomata çalışmak fikri asla ortaya çıkmadı ve sonuç olarak anlatılan deneyler gibi hiçbir şey yapılmadı.

Genel olarak, ana akım bilim bağlamında geliştirilen standart sezgi, hücresele otomata örüntüleri elde etmek için kullandığımız basit bilgisayar programlarının davranışlarını incelemeye değer olduğunu hayal etmeyi zorlaştırdı. Fakat ana akım bilim dışında bazı çalışmalar yapıldı. Ve örneğin, 1960'lardaki, erken bilgisayar tutkunları, çeşitli basit programları çalıştırmayı denediler ve bazı durumlarda iç içe geçmiş desen üretmede bu programlarla başarılabilceğini keşfettiler.

1970'lerin başında, Hayat Oyunu olarak bilinen, iki boyutlu hücresele otomata temelli eğlence amaçlı hesaplama oyunu geliştirildi. Hayat Oyununun davranışları, bazı açılardan, bu bölümde tartışılan 110 hücresele otomat ile benzerlik göstermektedir. Hayat Oyununun çalışma mantığından mühendislik için kullanılabilecek kadar basit ve öngörülebilir yapılar bulmak için büyük bir çaba harcandı. Hayat Oyununda karmaşık davranışlar gözlemlenmesine rağmen, bu kaçınılması gereken bir sıkıntı olarak kabul edildi. Hayat Oyunu üzerine birçok şey yapılabilecekken şaşırtıcı bir şekilde hücresele otomata çalışmalarımıza kadar ciddi çalışmalar yapılmamıştır. Temel bilime olan bağlantının olmaması kısmen bir sebep olarak düşünülebilir.

Sebepler ne olursa olsun, aslında, birkaç yüzyıl boyunca pek çok ipucuna rağmen, hücresele otomata ile ortaya koyduğum temel fenomen daha önce keşfedilmemiştir.

Bilim tarihinde, yeni fenomen tanımlandıktan sonra daha önce çoktan kanıt bulunduğunu görebilirsiniz. Fakat önemli olan nokta şu ki, genel fenomeni tanımlayan çerçeve olmadan, bu kanıtların göz ardı edilmesi kaçınılmazdır.

Bilimin ilerlemesinde yaşanan ironilerden biride şudur ki; beklenmedik sonuçlara ulaşıldığında daha önce birçok ipucunun ulaşılan gerçeğe açıkça işaret ettiğini görürüz. Hücresele otomata ile elde edilen sonuçları yirmi yıldan fazla yaşayan biri olarak işlerin başka bir şekilde çalışabileceğini hayal etmem güçtür. Ancak, bu bölümde özetlediğim tarih-birçok bilimsel keşfin tarihi gibi – ortaya çıkan ve herkes tarafından kabul edilen bilimsel gerçeklerin daha önce anlaşılamamalarının ne kadar kolay olduğunu gösteren örneklerle doludur.

Yukarıda verilen paragraflarda Wolfram'ın hücresele otomata üzerine yaptığı çalışmaları neden yeni bir bilim türü olarak adlandırdığını anlatan açıklamalar yer

almaktadır. Wolfram’ın bu konudaki idealarına en iyi cevabı tarih verecektir. Ancak şunu söylemek gerekir ki: Wolfram’ın hücresele otomatlar üzerine yaptığı çalışmalar ve bu çalışmaları ile yazdığı “A New Kind of Science” adlı kitap birçok açıdan değerlidir. Öncelikle hücresele otomatların ilk önerildiği zamanlarındaki karmaşık yapısında çok daha basit bir hücresele otomatın biyolojik canlıların kendini üretme özelliklerini modelleyebileceğini göstermiştir. İkincisi mevcut bilimsel yaklaşımları sorgulayan cürekâr bir iddiası vardır. Bu iddia bilimsel çalışmalara ve mühendislik çözümlerine yeni bir bakış açısı kazandırma potansiyeline sahiptir.



2. LİNEER OLMAYAN HÜCRESEL OTOMATLARIN NİTELENMESİ

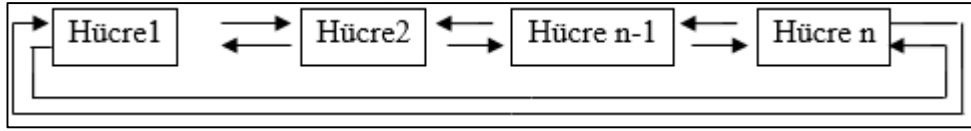
Bir önceki bölümde hücresel otomatların kural yapısına göre ikiye ayrıldığını belirtmiştik; lineer hücresel otomatlar ve lineer olmayan hücresel otomatalar. Lineer hücresel otomatlarda ortadaki hücrenin bir sonraki durumu XOR ve XNOR mantıksal operatörleri ile gerçekleştirilir. XOR ve XNOR mantıksal operatörleri ile bir sonraki durumu belirlenmesi cebirsel işlemlerle analiz yapmaya imkân verir (Chaudhuri, 1997). İki durum ve üç komşuya sahip hücresel otomatların ilk uygulamaları bu otomata türü ile gerçekleştirilmiştir. Lineer otomatlar ile kullanılabilen kural sayısı kısıtlı idi. Bu nedenle araştırmacılar lineer olmayan kuralların nitelenmesi üzerine odaklandılar (Ganguly, 2003; Das, 2006). Lineer olmayan hücresel otomatlar daha fazla kural kullanmaya olanak sağlar. Ancak lineer olmayan hücresel otomatların nitelenmesi için kullanılacak yöntemler, cebirsel işlemler kullanamaz. Bu nedenle araştırmacılar lineer olmayan hücresel otomatların kural seçimi için yöntemler geliştirmişlerdir. Bu tez çalışmasında lineer olmayan hücresel otomatlar kullanılmıştır.

Hücresel otomatlar ızgara yapısında yer alan birbiri ile bağlantılı hücrelerden oluşur. Her bir hücre sınırlı sayıda değer tutabilir. İki durumlu hücresel otomatlar için bu değer sayısı ikidir. Hücresel otomatlar zaman ve uzay ayrık yapıya sahiptir. Hücrelerin tuttuğu değer hücrenin durumunu temsil eder. t zamandaki bir hücrenin $t+1$ zamandaki durumu kendisi, sol ve sağ komşularının durumuna göre belirlenir. Hücrenin bir sonraki adımda durumunun belirlenmesi matematiksel bir formülle ifade edebilir; $S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t)$. Her hangi bir hücrenin t zamandaki durumu S_i^t olsun. S_i^t gösteriminde t zamanı, i ise hücrenin konumunu verir. i 'nci hücrenin $t+1$ zamanındaki durumu sol(S_{i-1}^t), kendisi(S_i^t) ve sağ(S_{i+1}^t) hücrelerinin durumuna göre belirlenir. Durumun belirlenmesinin kurallara göre yapıldığını bir önceki bölümde belirtmiştik. Kullanılan kuralı bir fonksiyon olarak düşünülecek olursa; i konumundaki hücrenin $t+1$ zamandaki durumu $S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t)$ şeklinde formüle edilebilir.

Bir hücreli otomata düzleminin t zamandaki durumu S^t olsun. N tane hücreden oluşan bir hücreli düzlem; $S^t=(S_1^t,S_2^t,...,S_n^t)$ biçiminde formüle edilebilir. Bir hücreli düzlemin bir sonraki durumu aşağıdaki gibi formüle edebilir.

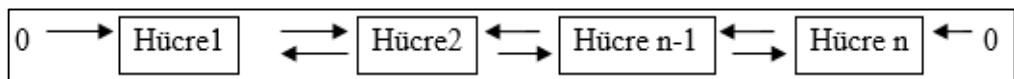
$$S^{t+1}=(f_1 (S_0^t, S_1^t, S_2^t),f_2 (S_1^t, S_2^t, S_3^t), ...,f_n (S_{n-1}^t, S_n^t, S_{n+1}^t))$$

Yukarıdaki formülde bir hücreli düzlemin bir sonraki durumunun elde edilmesi verilmektedir. Hücre numaraları 1’den n ’e kadardır. Yukarıda dikkat edilmesi gereken bir detay vardır. 1’inci hücre ve sonuncu hücrenin bir sonraki durumları elde edilirken olmayan bir hücrenin komşuluğu kullanılmıştır. Yani 1’inci hücrenin sol komşusu S_0^t sembolü ile temsil ediliyor. Aynı şekilde sonuncu hücrenin sağ komşusu S_{n+1}^t sembolü ile temsil edilmiştir. Bu iki hücrenin değerleri iki farklı şekilde belirlenir. Birinci yöntemde periyodik komşular kullanılır. Yani 1’inci hücrenin sol komşusu sonuncu hücre olur. Aynı şekilde sonuncu hücrenin sağ komşusu birinci hücre olur. Diğer bir değişle $S_0^t=S_n^t$ ve $S_{n+1}^t=S_1^t$ olacaktır. Bu hücreli otomata periyodik sınırlı hücreli otomata olarak adlandırılır. Aşağıdaki Şekil 2.1’de periyodik sınırlı hücreli otomata gösterilmektedir.



Şekil 2.1. Periyodik sınırlı hücreli otomatin 1’inci ve sonuncu hücrelerinin komşulukları

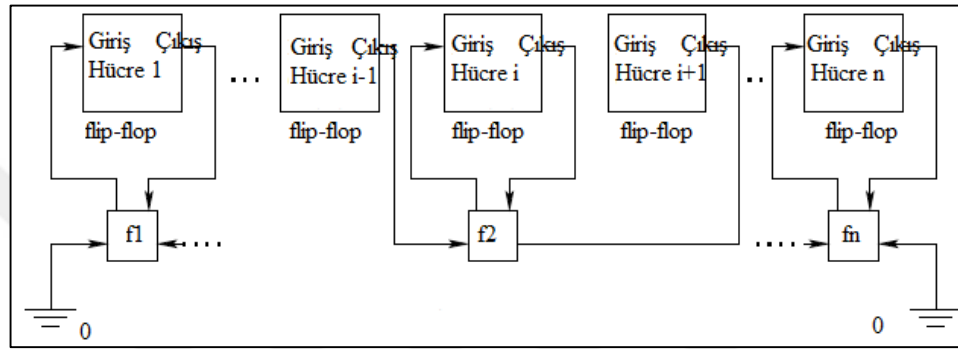
En sol ve en sağdaki hücrelerin olmayan komşularının değerlerinin belirlenmesinde kullanılan diğer bir yöntem daha vardır. Bu yöntemde göre 1’inci hücrenin sol komşusunun ve sonuncu hücrenin sağ komşusunun değerleri 0 olarak atanır. Diğer bir değişle $S_0^t=0$ ve $S_{n+1}^t=0$ olacaktır. Bu tür hücreli otomatlar sıfır sınırlı hücreli otomatlar olarak adlandırılır. Aşağıdaki Şekil 2.2’de olmayan komşular için 0 değerinin kullanıldığı sıfır sınırlı hücreli otomata gösterilmektedir.



Şekil 2.2. Sıfır sınırlı hücreli otomatların 1’inci ve sonuncu hücrelerinin komşulukları

Bu tez çalışmasında sıfır sınırlı hücreli otomata kullanılmıştır. Yani; en soldaki ve en sağdaki hücrelerin olmayan koşullarının değeri varsayılan olarak 0 belirlenmiştir.

CA'ların durum değişimini flip-flop devrelerin durum değişmesi şeklinde de gösterilebilir. Her bir zaman adımında flip-flop devrenin durumu soldaki, kendi ve sağdaki devrelerin durumuna göre değişecektir. N tane flip-flop devreden oluşan bir yapıyı CA düzlemi olarak düşünebilir. Şekil 2.3'te CA'nın flip-flop devre şeklinde gösterimi yer almaktadır.



Şekil 2.3. CA düzlemi flip-flop devrelerden oluşan bir düzlem olarak düşünülebilir

2.1. CA Terminolojisi

CA'da kullanılan terimlerin açıklanması, CA mantığının anlaşılması için faydalı olacaktır. Aşağıda hücreli otomatlara ait terimlerin açıklamaları yer almaktadır.

Kural vektörü: $R = \langle R_1, \dots, R_i, \dots, R_n \rangle$ örnek; $R = \langle 90, 100, 33, 74 \rangle$. Kural vektörü iki durum ve üç komşuya sahip hücreli otomat için 256 farklı kural içerebilir. Kullanılacak kural sayısı uygulamanın ihtiyacına göre belirlenir.

Lineer ve toplamsal kurallar: Kural yapısı XOR mantıksal operatörüne göre işlem yapıyorsa lineer kural, XOR veya NXOR mantıksal operatörüne göre işlem yapıyorsa toplamsal kural olarak adlandırılır. Toplamsal kurallar lineer kuralları içerisinde barındırır.

Dengeli Kural: Kuralın ikilik tabandaki karşılığında eşit sayıda 1 ve 0 varsa bu kural dengeli kural olarak adlandırılır. Dengeli kuralda 1 ve 0'ların sayısı 4 olmalıdır. Örnek: 10010110

Dengesiz Kural: Kuralın ikilik tabandaki karşılığında eşit sayıda 1 ve 0 yoksa bu kural dengesiz kural olarak adlandırılır. Dengesiz kuralda 1 veya 0'ların sayısı 4 olamaz. Örnek: 10000110.

Tekdüze CA: Kural vektöründeki tüm kural aynı ise tekdüze hücresel otomata olarak adlandırılır.

Melez CA: Kural vektöründeki kural birbirinden farklı kural içeriyorsa melez hücresel otomata olarak adlandırılır.

Durum geçiş diyagramı: n hücreye n uzunluklu bir kural vektörü uygulandıktan sonra hücre durumlarının geçişi gösteren diyagram.

Ulaşılabilir ve ulaşılamaz durumlar: Durum geçiş diyagramında atası olmayan durumlar ulaşılamaz durumlar ve atası olan durumlar ulaşılabilir durumlar olarak adlandırılır.

Eş zamanlı ve eş zamanlı olmayan hücresel otomatlar: Hücrelerin durum güncellemesi eş zamanlı olarak yapılıyorsa eş zamanlı hücresel otomatlar ve eş zamanlı olarak yapılmıyorsa eş zamanlı olmayan hücresel otomatlar olarak adlandırılır.

Lineer CA: Kural vektöründeki tüm kuralların lineer kural olması gerekir

Toplamsal CA: Kural vektöründeki tüm kuralların toplamsal kural olması gerekir

Lineer olmayan CA: Kural vektöründeki kurallar toplamsal olmayan kural içerirse bu CA lineer olmayan CA olarak adlandırılır.

Sıfır sınırlı ve periyodik sınırlı hücresel otomatlar: CA'da bir sonraki durum için üç hücreye ihtiyaç duyar. En soldaki hücrenin sol komşusu ve en sağdaki hücrenin sağ komşusu yoktur. Bunun üstesinden gelmek için iki yol vardır; sıfır sınırlı ve periyodik sınırlı komşuluk. Sıfır sınırlı komşulukta sınırdaki hücrelerin sıfır olarak kabul edilir, periyodik komşulukta ise en soldaki hücrenin komşusu en sağdaki hücre ve en sağdaki hücrenin komşusu en soldaki hücre kabul edilir. Örnek: sıfır sınırlı komşuluklar 0 1110 0 ve periyodik sınırlı komşuluklar 0 1110 1

Kural vektörü uzunluğu: Kural vektörü uzunluğu kural vektöründeki kural sayısı kadardır. Örneğin kural sayısı beş ise, beş bit uzunluklu CA olarak ifade edilir.

Bu tez kapsamında lineer olmayan hücresel otomatlar üzerine çalışılmıştır. Sınır komşuluğu olarak sıfır sınır komşuluğu kullanılmıştır. Hücrelerin durum değişimi eş zamanlı gerçekleştirilmiştir.

2.2. Kural Min Terimi (Rule Min Term)

Bir sonraki durumun, uygulanan kuralda hangi basamaktaki bit değerini alacağını Kural Min Terimi(rule min term) ile ifade edilir. Üç komşu ve iki durumlu CA kuralları için sekiz KMT vardır. KMT'ler 0'dan 7'ye kadar numara alabilirler ve KMT0, KMT1,...,KMT7 şeklinde ifade edilirler. Tablo 2.1'de kural KMT'leri ve kurallar arasındaki ilişki gösterilmiştir.

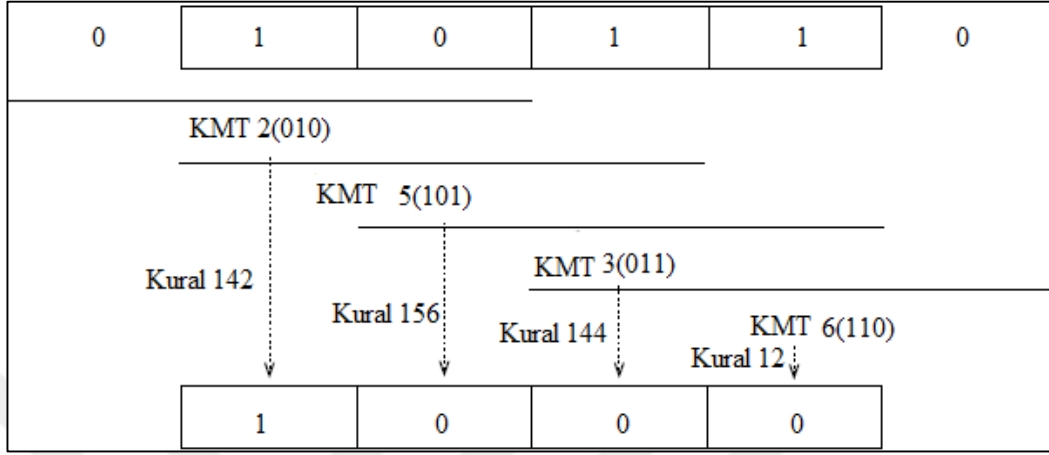
Tablo 2.1. Kural 60, Kural 204 ve Kural 51'e göre orta hücrelerin alacağı bir sonraki durum

t zamanındaki durum:	111	110	101	100	011	010	001	000
KMT:	7	6	5	4	3	2	1	0
Kural 60'a göre orta hücrenin t+1 zamandaki durumu	0	0	1	1	1	1	0	0
Kural 204'e göre orta hücrenin t+1 zamandaki durumu	1	1	0	0	1	1	0	0
Kural 51'e göre orta hücrenin t+1 zamandaki durumu	0	0	1	1	0	0	1	1

Kendi değeri 0 sol komşusunun değeri 1 ve sağ komşusunun değeri 1 olan bir durumun KMT'si 101(5) tir ve Tablo 2.1.'deki kurallara göre orta hücrenin bir sonraki durumları Kural 60 için 1, Kural 204 için 0 ve Kural 51 için 1 değerleridir. Tablo 2.1.'de koyu arka planla gösterilen yer, KMT5'in kurallara göre bir sonraki durumunu göstermektedir.

Kural KMT'lerine göre bir sonraki durumun elde edilmesini birazda daha açıklamak faydalı olacaktır. Dört hücreden oluşan bir hücresel otomatın bir sonraki durumunu belirlemek için dört bitlik bir kural vektörüne ihtiyaç duyulur. Bu kural vektörü

CA<142,156, 244, 12> olsun. Bu kural vektörüne göre 1011 değerinin bir sonraki durumu elde edilsin. 1011 değerinin bir sonraki durumunun elde edilmesini aşağıdaki Şekil 2.4 incelenerek daha iyi anlaşılabilir



Şekil 2.4. 1011 örüntüsünün CA<142, 156, 244, 12> kural vektörüne göre bir sonraki durumunun elde edilmesi

Dikkat edilirse 1011 değerinin sol ve sağına 0 eklenmiştir. Bunun nedeni kullanılan hücresel otomatın sıfır sınırlı hücresel otomata olmasından kaynaklanmaktadır. En soldaki hücrenin bir sonraki durumu Kural 142’inin KMT2 değerine göre belirlenir. Soldan ikinci hücrenin bir sonraki durumu Kural 156’nın KMT5 değerine göre belirlenir. Soldan üçüncü hücrenin bir sonraki durumu Kural 144’ün KMT3 değerine göre belirlenir. En sağdaki hücrenin bir sonraki durumu Kural 12’nin KMT6 değerine göre belirlenir. Kuralların kullanılacak KMT’leri ilgili hücrenin kendisi, sol ve sağ komşularının değerine göre belirlenir. Sonuç olarak 1011 örüntüsünün CA<142, 156, 244, 12> kural vektörüne göre bir sonraki durumu 1000 değeridir.

Tablo 2.2’de CA<142, 156, 244, 12> kural vektörüne ait kuralların KMT’lerine karşılık gelen ikilik tabandaki değerleri verilmiştir. Tabloda 142 kuralına ve 12 kuralına ait satırlarda kuralların ikilik tabandaki karşılığının bazı basamaklarında “d” harfi yer almaktadır. ‘d’ harfi “don’t care” anlamında kullanılmaktadır (Das ve ark., 2009). Bu, “d” harfinin olduğu KMT’lerin hiçbir zaman dikkate alınmayacağını göstermektedir. Bunun sebebi kullanılan hücresel otomatın sıfır sınırlı olmasından kaynaklanmaktadır.

Tablo 2.2. CA<142, 156, 244, 12> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları

Kural	KMT							
	111	110	101	100	011	010	001	000
	KMT (7)	KMT (6)	KMT (5)	KMT (4)	KMT (3)	KMT (2)	KMT (1)	KMT (0)
142	d	d	d	d	1	1	1	0
156	1	0	0	1	1	1	0	0
244	1	1	1	1	0	1	0	0
12	d	0	d	0	d	1	d	0

Dikkate alınmayan KMT'ler en soldaki ve en sağdaki hücrelerin kurallarına aittir. En soldaki hücrenin sol komşusu her zaman 0 olacağından bu hücrenin bir sonraki durumunu belirlemek için kullanılacak KMT numarası en fazla 011(KMT3) değerine sahip olabilir. Bu nedenle en soldaki kural olan Kural 142'ye ait KMT7, KMT6, KMT5 ve KMT4 KMT'leri hiçbir zaman kullanılmayacaktır. En sağdaki hücrenin sağındaki hücrenin değeri her zaman 0 olacağından bu hücrenin bir sonraki durumunu belirlemek için kullanılacak KMT numarası çift değere sahip olacaktır. . Bu nedenle en sağdaki kural olan Kural 12'ye ait KMT7, KMT5, KMT3 ve KMT1 KMT'leri hiçbir zaman kullanılmayacaktır.

Hücrel otomata kurallarının nitelenmesinde KMT'ler önemli bir yere sahiptir. Bir sonraki bölümde KMT'lerin nitelenmesinde kullanımı ve yanaşık hücrelere ait kuralların KMT etkileşimine yer verilecektir.

2.3. Hücrel Otomata Cezbedicilerinin Nitelenmesi

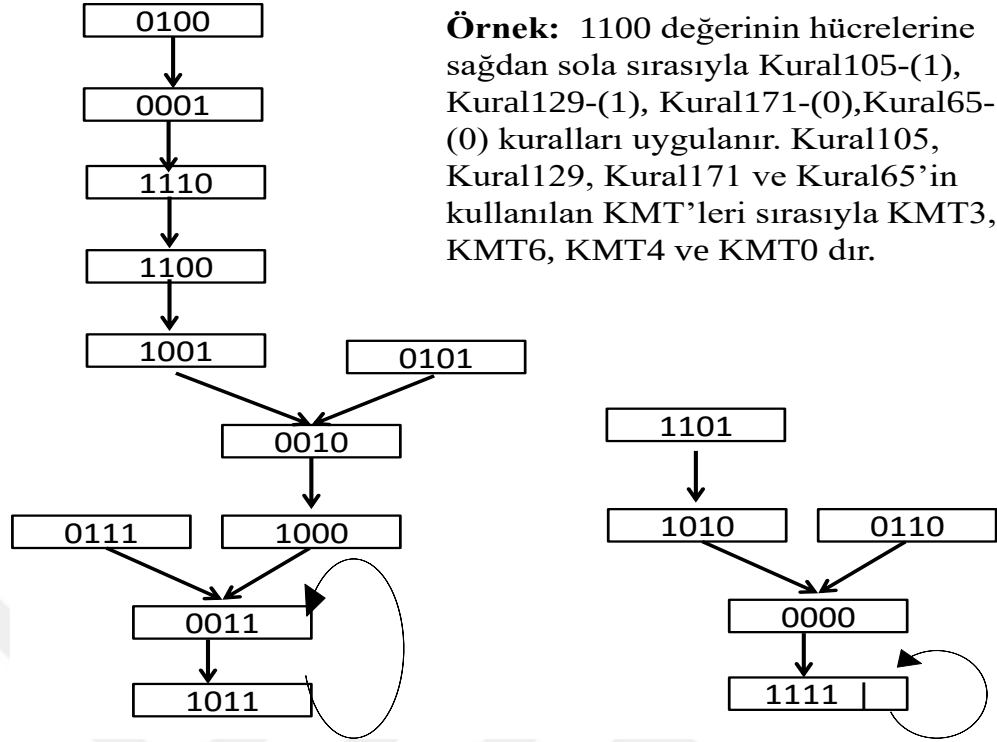
Bu bölümde ters döndürülemez hücrel otomatların tek döngülü cezbedicilerinin nitelenmesi anlatılacaktır. Bu kısım Das ve arkadaşlarının yaptıkları çalışmaya dayanmaktadır (Das ve ark., 2009). Das ve arkadaşları yaptıkları çalışmada lineer olmayan hücrel otomatların nitelenmesini ikili ağaç kullanarak yapışlardır (Das ve ark., 2004; Das ve ark., 2005; Das ve Sikdar, 2006). Tasarlanan bu ağaca Erişilebilirlik Ağacı(Reachability Tree) adını vermişlerdir.

KMT kavramı kuralların nitelenmesinde çok önemli bir yere sahiptir. Eğer KMT kavramı anlaşılmadıysa KMT'nin anlatıldığı kısmı tekrardan incelemek faydalı olacaktır.

Cezbedici durumlar diğer durumlar ile arasında dinamik bir ilişki olan durumlardır. Bu durumlar çekim merkezi gibi davranmaktadırlar. Araştırmacılar çekim merkezi gibi davranan bu durumları incelemişlerdir. Bu durumların nitelenmesi için lineer ve lineer olmayan kurallar üzerinde çalışmışlardır. Lineer kuralların nitelenmesi için cebirsel işlemler kullanılabilirken (Chaudhuri ve ark., 1997), lineer olmayan kurallar için cebirsel işlemler kullanılmaz. Bu nedenle lineer olmayan kuralların nitelenmesi için yeni algoritmalar önerilmiştir (Das ve ark., 2009; Ganguly ve ark., 2002).

Cezbedici durumlar kendine dönen durumlardır. Cezbedici durumun kendine dönmesi bir sonraki durumda olabildiği gibi, birden fazla durumdan sonrada kendine dönebilir. Eğer cezbedici durum bir sonraki durumda kendine dönüyorsa bu tek uzunluk döngülü cezbedici(single length cycle attractor) olarak adlandırılır. Eğer cezbedici durum birkaç durumdan sonra kendine dönüyorsa bu cezbedici çok uzunluk döngülü cezbedici(multi length cycle attractor) olarak adlandırılır. Aşağıdaki Şekil 2.5'te tek uzunluk döngülü cezbedici ve çok uzunluk döngülü cezbedici üreten kural vektörüne ait durum geçiş diyagramı yer almaktadır.

Aşağıdaki şekilde durumların değerleri ikili tabanda verilmiştir. Durumların ikilik tabandaki sayısal karşılığı dört basamaktan oluşmaktadır. Bu nedenle bir sonraki durumun elde edilmesi için dört uzunluklu bir kural vektörüne ihtiyaç duyulur. Uygulanan kural vektörü 105, 129, 171 ve 65 olmak üzere dört kuraldan oluşmaktadır. Kuralların ikili değerleri ve KMT karşılıkları Tablo 2.3'te verilmiştir.



Şekil 2.5. CA<105, 129, 171, 65> kural vektörünün ürettiği durum geçiş diyagramları. Kural vektörü tek uzunluk döngülü ve çok uzunluk döngülü cezbediciler üretmektedir.

Tablo 2.3. CA<105, 129, 171, 65> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları

Kural	KMT							
	111	110	101	100	011	010	001	000
	KMT (7)	KMT (6)	KMT (5)	KMT (4)	KMT (3)	KMT (2)	KMT (1)	KMT (0)
105	0	1	1	0	1	0	0	1
129	1	0	0	0	0	0	0	1
171	1	0	1	0	1	0	1	1
65	0	1	0	0	0	0	0	1

Kural vektörüne göre bir sonraki durumların nasıl elde edildiğini birkaç örnek vererek inceleyelim. Mevcut durum 0100(4) olsun, sıfır sınırlı CA olduğu için en soldaki hücreden an sağdaki hücreye kadar bir sonraki durumu belirlemek için bakmamız gereken komşuluklar sırasıyla 001, 010, 100 ve 000'dır. Bu dört komşuluk için uygulanacak kurallar kural vektöründeki sıraya göre olacak. 001 komşuluğu için 105 kuralının KMT1 değerine bakmamız gerekir. 105 kuralının KTM1 değeri 0'dır. 010, 100 ve 000 komşulukları için 129, 171 ve 65 kurallarına karşılık gelen KMT2, KMT4

ve KMT0 değerleri sırasıyla 0, 0 ve 1'dir. Yani 0100(4) durumunun bir sonraki durumu 0001(1) durumudur. 0011 durumu için bir sonraki durumu bulalım. 0011 durumu için kural vektöründeki kuralların soldan sağa olacak şekilde KMT0, KMT1, KMT3 ve KMT6 değerleri 1, 0, 1 ve 1'dir. 0011(3) durumunun bir sonraki durumu $CA<105, 129, 171, 65>$ kural vektörüne göre 1011(11) dir. Son olarak 1011 durumunu ele alalım. 1011 durumu için kural vektöründeki kuralların soldan sağa olacak şekilde KMT2, KMT5, KMT3 ve KMT6 değerleri 0, 0, 1 ve 1'dir. 1011(11) durumunun bir sonraki durumu $CA<105, 129, 171, 65>$ kural vektörüne göre 0011(3) tür.

2.3.1. Özellik 1

Cezbedici durumlar diğer durumların bir sonraki durumu kendisindir. Bu noktada kural vektörünün yapısı önemli bir rol oynamaktadır. Kurallar cezbedici durum oluşturma potansiyeline göre farklılık gösterirler. Kuralların cezbedici durum oluşturmaya katkısı “özellik 1” adlandırması kullanılarak tanımlanmıştır. “Özellik 1” tanımı aşağıda yer almaktadır.

Özellik 1: Bir kuralın cezbedici durum oluşturmaya katkısı kuralın KMT'lerine karşılık gelen bitlerin değerine göre belirlenir. KMT0, KMT1, KMT4 ve KMT5 bitlerinin 0 olması ve KMT2, KMT3, KMT6 ve KMT7 bitlerinin 1 olması cezbedici oluşumuna katkı sağlar.

Aşağıdaki tabloyu inceleyerek “özellik 1” tanımını daha iyi anlaşılabilir. Aşağıdaki Tablo 2.4'te tüm KMT'leri ile cezbedici oluşumuna katkı sağlayan Kural 204 ve hiçbir KMT'sinde cezbedici oluşumuna katkı sağlamayan Kural 51'e yer verilmiştir.

Tablo 2.4. Kural204 ve Kural51 KMT'lerinin

	111	110	101	100	011	010	001	000	
KMT'ler	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
	1	1	0	0	1	1	0	0	Kural 204
	0	0	1	1	0	0	1	1	Kural51

Yukarıdaki tabloyu inceleyelim. KMT'lerin ikilik tabandaki değerleri en üst satırda ve onluk tabandaki değerleri ikinci satırda yer almaktadır. KMT'lerin ikilik tabandaki değerleri üç bitten oluşmaktadır. Bu üç bit sol, orta ve sağ hücrelerini temsil etmektedir. Orta basamakta yer al değerlerin bir sonraki değeri; sol, kendi ve sağdaki

basamakta yer alan değerlere ve bir sonraki durum için kullanılan kurala göre belirlenir.

Yukarıdaki tabloda verilen Kural 204 ve Kural 51'i inceleyelim. Kural 204'e göre 000(KMT0), 001(KMT1), 100(KMT4) ve 101(KMT5) KMT'leri için ortadaki hücrenin bir sonraki durumu 0'dır ve 010(KMT2), 011(KMT3), 110(KMT6) ve 111(KMT7) KMT'leri için ortadaki hücrenin bir sonraki durumu 1'dir. Dikkat edilirse ortadaki hücrenin değeri Kural 204'e göre değişmemektedir. Yani Kural 204 yukarıda verilen "özellik 1" tanımını tüm KMT'lerinde taşımaktadır.

Kural 51'e göre 000(KMT0), 001(KMT1), 100(KMT4) ve 101(KMT5) KMT'leri için ortadaki hücrenin bir sonraki durumu 1'dir ve 010(KMT2), 011(KMT3), 110(KMT6) ve 111(KMT7) KMT'leri için ortadaki hücrenin bir sonraki durumu 0'dır. Dikkat edilirse ortadaki hücrenin değeri Kural 51'e göre hep değişmektedir. Yani Kural 51 yukarıda verilen "özellik 1" tanımını hiçbir KMT'sinde taşımamaktadır.

Kural 204 tek uzunluk döngülü cezbedici oluşumuna tüm KMT'lerinde katkı sağlarken, Kural 51 tek uzunluk döngülü cezbedici oluşumuna hiçbir KMT'sinde katkı sağlamamaktadır.

2.3.2. Hücresel otomata kurallarının gruplandırılması

İki durumdan ve üç komşudan oluşan hücresel otomatların bir sonraki durumu 256 farklı kurala göre belirlenir. Bu kurallar 0 ile 255 arasındaki değerlere sahip olabilir. Bu kurallar tek uzunluk döngülü cezbedici oluşumuna katkılarına göre dokuz farklı guruba ayrılmıştır (Naskar, 2009). Grup numaraları 0 ile 8 arasındaki değerlerden biri olabilir. Örneğin; Kural 204 sekiz KMT'sinde tek uzunluk döngülü cezbedici oluşumuna katkı sağladığı için Grup 8'de yer almaktadır. Kural 51 hiçbir KMT'sinde tek uzunluk döngülü cezbedici oluşumuna katkı sağlamadığı için Grup 0'da yer almıştır. Grup numarası arttıkça tek uzunluk döngülü cezbedici oluşumuna olan katkıları da artmaktadır. Tablo 2.5'te tek uzunluk döngülü cezbedici oluşumuna katkılarına göre gruplandırılmış 256 hücresel otomata kuralı yer almaktadır.

Tablo 2.5. Tek uzunluk cezbedici oluşumuna göre gruplandırılmış 256 hücresel otomata kuralı

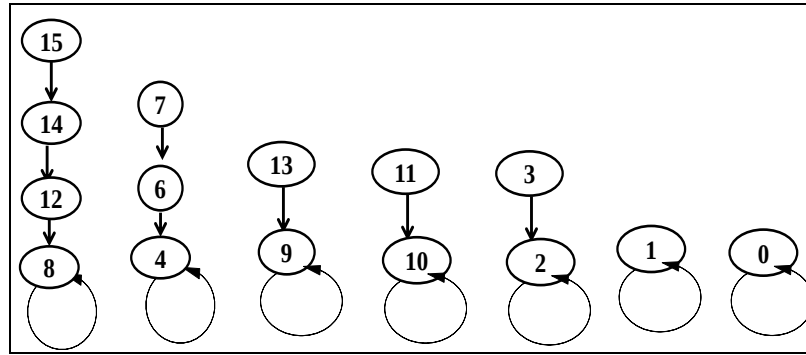
Grup Numarası	Kurallar
Grup0	51
Grup1	19, 35, 49, 50, 55, 59, 115, 179
Grup2	3, 17, 18, 23, 27, 33, 34, 39, 43, 48, 53, 54, 57, 58, 63, 83, 99, 113, 114, 119, 123, 147, 163, 177, 178, 183, 187, 243
Grup3	1, 2, 7, 11, 16, 21, 22, 25, 26, 31, 32, 37, 38, 41, 42, 47, 52, 56, 61, 62, 67, 81, 82, 87, 91, 97, 98, 103, 107, 112, 117, 118, 121, 122, 127, 131, 145, 146, 151, 155, 161, 162, 167, 171, 176, 181, 182, 185, 186, 191, 211, 227, 241, 242, 247, 251
Grup4	0, 5, 6, 9, 10, 15, 20, 24, 29, 30, 36, 40, 45, 46, 60, 65, 66, 71, 75, 80, 85, 86, 89, 90, 95, 96, 101, 102, 105, 106, 111, 116, 120, 125, 126, 129, 130, 135, 139, 144, 149, 150, 153, 154, 159, 160, 165, 166, 169, 170, 175, 180, 184, 189, 190, 195, 209, 210, 215, 219, 225, 226, 231, 235, 240, 245, 246, 249, 250, 255
Grup5	4, 8, 13, 14, 28, 44, 64, 69, 70, 73, 74, 79, 84, 88, 93, 94, 100, 104, 109, 110, 124, 128, 133, 134, 137, 138, 143, 148, 152, 157, 158, 164, 168, 173, 174, 188, 193, 194, 199, 203, 208, 213, 214, 217, 218, 223, 224, 229, 230, 233, 234, 239, 244, 248, 253, 254
Grup6	12, 68, 72, 77, 78, 92, 108, 132, 136, 141, 142, 156, 172, 192, 197, 198, 201, 202, 207, 212, 216, 221, 222, 228, 232, 237, 238, 252
Grup7	76, 140, 196, 200, 205, 206, 220, 236
Grup8	204

Bir sonraki kısımda farklı gruptaki kurallar hücresel otomata durum geçiş diyagramlarına etkisine yer verilmiştir. Çoklu cezbedici hücresel otomata üzerine yeni çalışan araştırmacılar için bu gözlemler faydalı olacaktır.

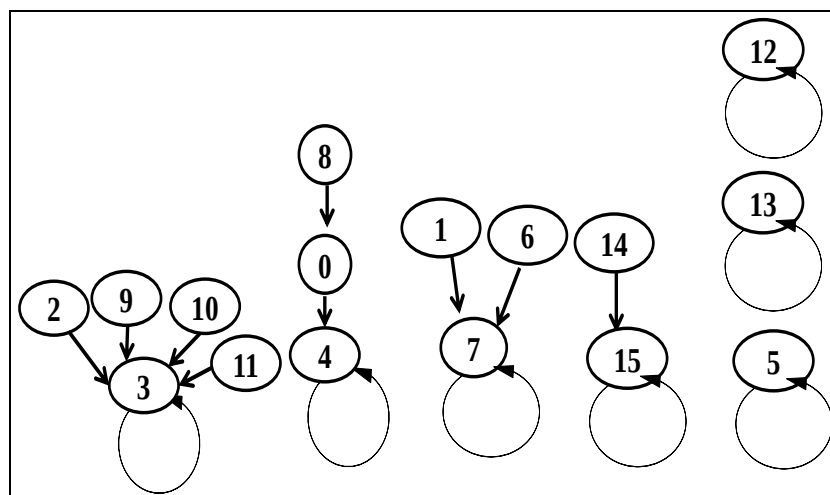
2.3.3. Kural gruplarının gözlemlenmesi

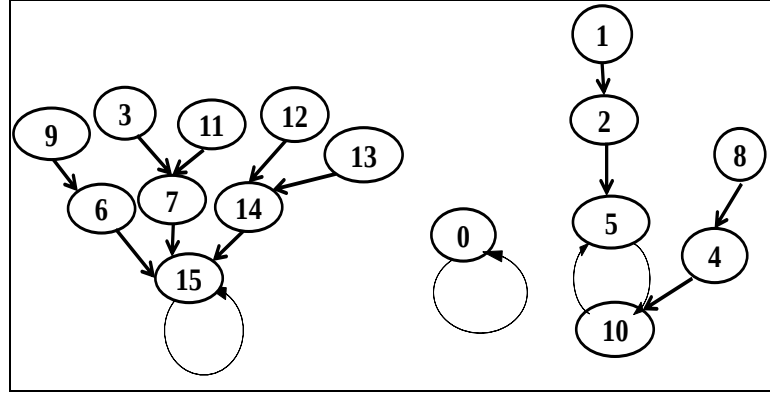
Naskar yüksek lisans tez çalışmasında farklı gruptaki hücresel otomata kurallarının cezbedici oluşumuna etkisi üzerinde gözlemde bulunmuştur (Naskar, 2009). Kuralların ait oldukları gruba göre ürettiği cezbedici formlarını inceleyelim. Aşağıdaki şekillerde hücresel otomata kural vektörlerinin uygulanmasıyla elde edilen durum geçiş diyagramlarına yer verilmiştir. Durum geçiş diyagramları incelenerek kural vektörlerinin ürettiği cezbedici formlarının farklılıkları görülebilir.

Yedinci gruptaki kurallardan oluşan kural vektörü sadece tek uzunluk döngülü cezbediciler üretmektedir. Aşağıdaki Şekil 2.6'da dört bitten oluşan $CA < 140, 140, 140, 140 >$ tek düze kural vektörünün durum geçiş diyagramı yer almaktadır.



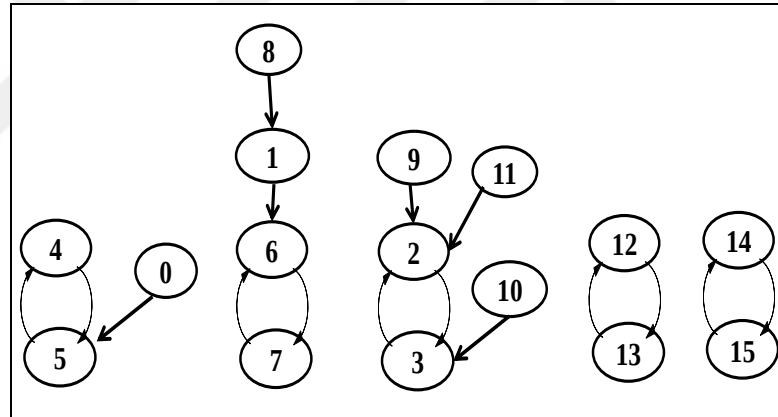
CA<140, 140, 140, 140> kural vektörü çok sayıda tek uzunluk döngülü cezbedici üretmiştir. Kural vektörü çok sayıda ulaşılmayan durum vardır ve durum geçiş diyagramlarının yüksekliği en fazla dörttür.





Şekil 2.8. Grup 4'teki kurallarda oluşan $CA\langle 250,250,250,250 \rangle$ kural vektörü tek uzunluk döngülü cezbediciler ve çok uzunluk döngülü cezbediciler oluşturur

Eğer bir kural vektörü Grup 0 ait 51 kuralını içeriyorsa bu kural vektörü tek uzunluk döngülü cezbediciler üretemez. Şekil 2.9'da kural vektörü Kural 51'i içeren kural vektörünün durum geçiş diyagramı yer almaktadır.



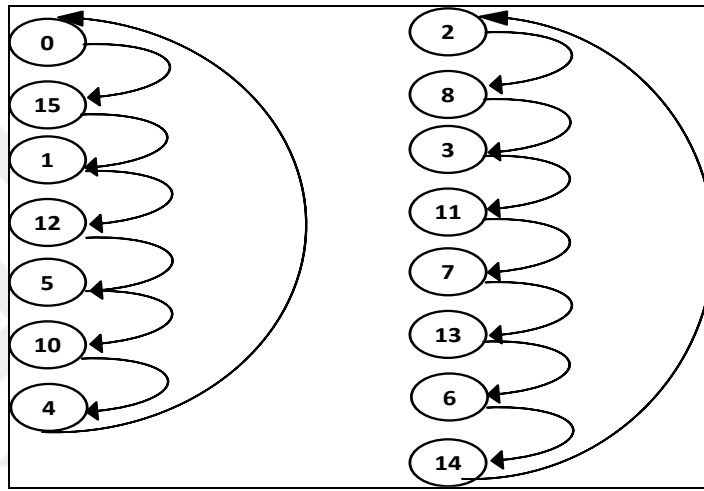
Şekil 2.9. Kural 51'i içeren $CA\langle 200,205,206,51 \rangle$ kural vektörü sadece tek uzunluk döngülü cezbediciler ve çok uzunluk döngülü cezbediciler oluşturur

Yukarıdaki şekillerde de görüldüğü gibi kurallar durum geçiş diyagramlarını ve cezbedici oluşumunu doğrudan etkilemektedir. Sadece grup numaralarına dikkat ederek tek uzunluk döngülü cezbediciler üreten kural vektörleri oluşturulabilir. Ancak bu kural vektörü kontrolsüz sayıda tek uzunluk döngülü cezbedici üretecektir ve fazla hafıza kullanımına neden olacaktır. İleriki bölümlerde kontrollü sayıda cezbedici üreten kural vektörünün nitelenmesi ele alınacaktır.

2.3.4. Ters döndürülebilir ve ters döndürülemez hücresel otomatlar

Kural vektöründeki kuralların yapısına göre elde edilen durum geçiş diyagramı farklı özelliklere sahip olabilir. Hücresel otomatlar durum geçiş diyagramları temelde ikiye ayrılmıştır; ters döndürülebilir ve ters döndürülemez hücresel otomatlar.

Ters döndürülebilir hücresel otomatlarda her bir durumun bir atası vardır ve her bir duruma başka bir durumdan ulaşılabilir. Şekil 2.10'da ters döndürülebilir hücresel otomata örneği verilmiştir.



Şekil 2.10. Ters döndürülebilir CA <195,105,105,195> durum geçiş diyagramı (Peldek, 2012)

Ters döndürülemez hücresel otomatlarda durumların birden fazla atası olabilir veya hiç atası olmaya bilir. Bu tip hücresel otomatlar kural vektörünün yapısına göre ikiye ayrılır; Çoklu Cezbedici Hücresel Otomatlar(Multiple Attractor Cellular Automata(MACA)) ve Genelleştirilmiş Çoklu Cezbedici Hücresel Otomatlar(Generalized Multiple Attractor Cellular Automata(GMACA)). MACA'lar sadece toplamsal kurallardan oluşurken, GMACA toplamsal olmayan kurallardan oluşabilir. MACA'ların sadece toplamsal kurallar içermesi üzerinde cebirsel işlemlerin yapılmasına imkân vermektedir. GMACA'lar toplamsal olmayan kurallar içerdiği için üzerinde cebirsel işlemler yapılamaz. MACA hakkında detaylı bilgiler (Chaudhuri, 1997; Ganguly, 2003) kaynaklarından bulunabilir. MACA ve GMACA isimlendirmesi hücreler üzerinde uygulanan kuralların yapısına bağlı olarak bir sonraki durumu elde etmedeki işlemlerin farklılıklarından kaynaklanmaktadır.

GMACA'lar kural vektörünün yapısına göre farklı geçiş diyagramlarına sahip olurlar. GMACA geçiş diyagramlarında bazı durumlar diğer durumları çekerler ve bu tip diğer durumları çeken durumlar cezbedici(attractor) durum olarak adlandırılır. Ters döndürülemez hücresele otomatlar olarak adlandırılan GMACA'lar diğer durumları çeken cezbedici durumun bir sonraki durumda kendine veya başka bir duruma dönmesine göre iki ayrılırlar; Tek Uzunluklu Döngülü Cezbedici(Single Length Cycle Attractor(SLCA)) ve Çok Uzunluklu Döngülü Cezbedici(Multi Length Cycle Attractor(MLCA)).

Bir sonraki bölümde tek uzunluk döngülü cezbedicilerin elde edilmesi için önerilen erişilebilirlik ağacı tabanlı yöntem (Das ve ark., 2009) yer verilmiştir.

2.4. Tek Uzunluk Döngülü Cezbedicilerin Nitelenmesi

Bu kısımda tek uzunluk döngülü cezbedicilerin elde edilmesi için geliştirilen yöntemlere yer verilecektir. Das ve arkadaşları tek uzunluk döngülü cezbedicilerin nitelenmesi için ikili ağaç tabanlı algoritmalar geliştirmişlerdir (Das ve ark., 2009). Kullandıkları ikili ağaca Erişilebilirlik Ağacı(Reachability Tree) adını vermişlerdir (Das ve Sikdar, 2006; Das ve ark, 2004).

Tek uzunluk döngülü cezbedicinin bir sonraki durumu kendisi olduğu için, tek uzunluk döngülü cezbedici üretecek kural vektöründe yer alan kurallardan her birinin en az bir KMT'sinin durumunu değiştirmemesi gerekir (Peldek ve Becerikli, 2019). Bunun nedeni; cezbedicinin durumunu değiştirmemesidir.

Kurala ait KMT'ler özellik1'i taşıdıkları ölçüde tek uzunluk döngülü cezbedici oluşumuna katkı sağlarlar. Özellik1 hakkında bilgi için aynı başlıkta yer alan anlatımlara bakılabilir. Kural 204 özellik1'i tüm KMT'lerinde taşımaktadır. Kural 204'ün KMT0, KMT1, KMT4 ve KMT5 KMT'lerine karşılık gelen değerleri 0'dır. Benzer şekilde Kural 204'ün KMT2, KMT3, KMT6 ve KMT7 KMT'lerine karşılık gelen değerleri 1'dir. Kural 51 özellik1'i hiçbir KMT'sinde taşımamaktadır. Kural 51'in KMT0, KMT1, KMT4 ve KMT5 KMT'lerine karşılık gelen değerleri 1'dir. Benzer şekilde Kural51'in KMT2, KMT3, KMT6 ve KMT7 KMT'lerine karşılık gelen değerleri 0'dır. Hücresele otomata kuralları KMT'lerinde özellik1'i karşılama

sayılarına göre gruplara ayrılmıştır. 256 farklı hücresel otomata kuralının gruplandırılması 2.3.3. numaralı kısımda yer almaktadır.

2.4.1. Cezbedicilerin nitelenmesi için erişilebilirlik ağacı

Das ve arkadaşları erişilebilir durumları temsil eden Erişilebilirlik Ağacı önermişlerdir (Das ve ark., 2005; Das ve Sikdar, 2006; Das ve ark., 2004). Erişilebilirlik ağacı hücresel otomata kurallarından oluşan kural vektörünün ürettiği durum geçiş diyagramında erişilebilir durumları bulmak için önerilen ikilik bir ağaçtır.

Erişilebilirlik ağacı kural vektörü kullanılarak oluşturulur ve ağacın düğümleri kural vektöründe yer alan kuralların KMT'leri ile oluşturulur. Her bir düğüm en fazla sola ve sağa dallana bilir. Sola dallanan kenar “sol kenar” ve sağa dallanan kenar “sağ kenar” olarak adlandırılır. Sol kenar 0 değerine ve sağ kenar 1 değerine sahiptir. Eğer kural vektörü n tane kuraldan oluşursa ağacın seviyesi $n+1$ olabilir. Seviyelerin numaralandırması 0 ile n arasındadır. Ağacın kök düğümü seviye 0'da ve yaprak düğümleri seviye n 'de yer alır. Erişilebilirlik ağacındaki yaprak düğümler erişilebilir durumları temsil eder.

Erişilebilirlik ağacı inşa edilirken kural vektörü soldan sağa doğru taranarak kurallar değerlendirilir. Örneğin $CA < 78, 142, 201, 92 >$ kural vektörünün 1'inci kuralı 78 ve sonuncu kuralı 92'dir. Erişilebilirlik ağacının i 'ninci seviyesi kural vektöründeki $i+1$ sırasındaki kurala göre oluşturulur. Kök düğümün seviye 0'da olduğu için, kök düğümün oluşturulması için 1'inci kural kullanılır.

Erişilebilirlik ağacındaki yaprak düğümlerin hangi durumu ait olduğu kök düğümlerle yaprak düğüm arasındaki kenarlar belirler. Sol düğüm 0 ve sağ düğüm 1 değerlerini temsil eder. Kök düğümünden yaprak düğüme ulaşmak için kullanılan sol ve sağ kenarlar ulaşılabilir durumun sayısal değerini verir. Erişilebilirlik ağacının oluşturulmasına geçmeden önce yanaşık iki hücre kurallarının KMT'leri arasındaki ilişkinin anlaşılması gerekir. Bir sonraki bölümde i 'ninci kural ve sağındaki kural olan $(i+1)$ 'inci kuralın KMT'leri arasındaki ilişki anlatılmaktadır.

2.4.2. İ'ninci ve (i+1)'inci kuralların KMT'leri arasındaki ilişki

Hücrel otomata durum değişimi kural vektörü kullanılarak yapılır. Kural vektöründe yanaşık olan hücre kurallarının KMT'leri bir sonraki durumun elde edilmesinde ilişkilidir (Das ve ark., 2009). Daha formel bir deyişle ifade edilecek olursa; R_i ve R_{i+1} kurallarının bir sonraki durumu belirlemek için kullanılacakları KMT'ler birbiriyle ilişkilidir.

Bir hücrenin bir sonraki durumu belirlenirken ilgili kuralın KMT'sine karşılık gelen bit değerine bakılır. Her bir hücrel otomata kuralının sekiz tane KMT'si vardır. Aşağıdaki Tablo 2.6'da yanaşık iki hücre kurallarının ilişkili olduğu KMT numaraları verilmektedir.

Tablo 2.6. Bir sonraki durum hesaplanması için i ve i+1 kurallarının KMT'leri arasındaki ilişki (Das ve ark., 2009).

İ'ninci kuralda kullanılan KMT	(i+1)'inci kuralda kullanılabilecek KMT'ler
KMT0(000)	KMT0(000), KMT1(001)
KMT1(001)	KMT2(010), KMT3(011)
KMT2(010)	KMT4(100), KMT5(101)
KMT3(011)	KMT6(110), KMT7(111)
KMT4(100)	KMT0(000), KMT1(001)
KMT5(101)	KMT2(010), KMT3(011)
KMT6(110)	KMT4(100), KMT5(101)
KMT7(111)	KMT6(110), KMT7(111)

Yukarıdaki tabloda iki yanaşık hücre kuralının bir sonraki durumun belirlenmesinde kullanılacak KMT'leri arasındaki ilişki verilmektedir. Sol sütünde i'ninci hücre kuralı için kullanılan KMT ve sağ sütünde (i+1)'inci hücre kuralı için kullanılabilecek KMT'ler yer almaktadır.

Örnek vererek KMT ilişkisinin daha iyi anlaşılmasına yardımcı olalım. İ'ninci kuralın bir sonraki durumunu belirlemek için kullanılacak KMT'si 000(KMT0) olsun. (i+1)'inci hücrenin bir sonraki durumunu belirlemek için kullanılacak KMT'si

000(KMT0) veya 001(KMT1) olabilir. Bir KMT üç bitten oluşur. Bu bitler $b_0b_1b_2$ olsun. KMT0 için kullanılan bitler $b_0^0b_1^0b_2^0$ biçiminde temsil edilebilir. Aynı şekilde KMT1 için kullanılan bitler $b_0^1b_1^1b_2^1$ biçiminde temsil edilebilir. Eğer i 'ninci kuralın kullanılacak KMT'si KMT0 ise $b_0^0b_1^0b_2^0=000$ olacaktır. Bu durumda $(i+1)$ 'inci kuralın kullanılacak KMT'si ya KMT0 veya KMT1'dir. $(i+1)$ 'inci kuralının kullanılacak KMT bitleri KMT0 ve KMT1 için sırasıyla $b_0^1b_1^1b_2^1=000$ veya $b_0^1b_1^1b_2^1=001$ olacaktır. Dikkate edilirse i 'ninci ve $(i+1)$ 'inci kurallarının $b_1^0b_2^0=b_1^1b_2^1=00$ bitleri her iki durum için aynıdır. $(i+1)$ 'inci kural için kullanılacak KMT'nin ilk iki bit değeri i 'ninci kural için kullanılan KMT'nin son iki bit değeri ile aynıdır. $(i+1)$ 'inci kural için kullanılacak KMT'nin ilk iki bit değeri $b_0^1b_1^1=00$ olduğunda, b_2^1 bit değeri ya 0 yada 1'dir. Bu durumda $(i+1)$ kuralı için kullanılacak KMT'nin ya KMT0($b_0^1b_1^1b_2^1=000$) veya KMT1($b_0^1b_1^1b_2^1=001$) olabileceğini göstermektedir.

Yukarıdaki tabloda ikinci satırda i 'ninci hücre kuralı için kullanılacak KMT, KMT0 ise $(i+1)$ 'inci hücre kuralı için kullanılacak KMT'lerin KMT0 ve KMT1 olması bu ilişkiden kaynaklanmaktadır. Yukarıdaki tabloda ikinci sütundaki tüm KMT numaraları yukarıda anlatılan ilişkiye göre elde edilmiştir.

Yukarıdaki tabloda kırmızı ve kalın fontla verilen KMT bitleri i 'ninci ve $(i+1)$ 'inci kurallar için kullanılacak KMT'lerin ortak bitlerini göstermektedir. Yukarıdaki tabloya dikkate edilirse; ikinci sütundaki ilk dört satır ile ikinci sütundaki son dört satır aynıdır. Bunun nedeni KMT0(000), KMT1(001), KMT2(010) ve KMT3(011) KMT'lerinin son iki bitinin sırasıyla KMT4(100), KMT5(101), KMT6(110) ve KMT7(111) KMT'lerinin son iki bitiyle aynı olmasından kaynaklanmaktadır.

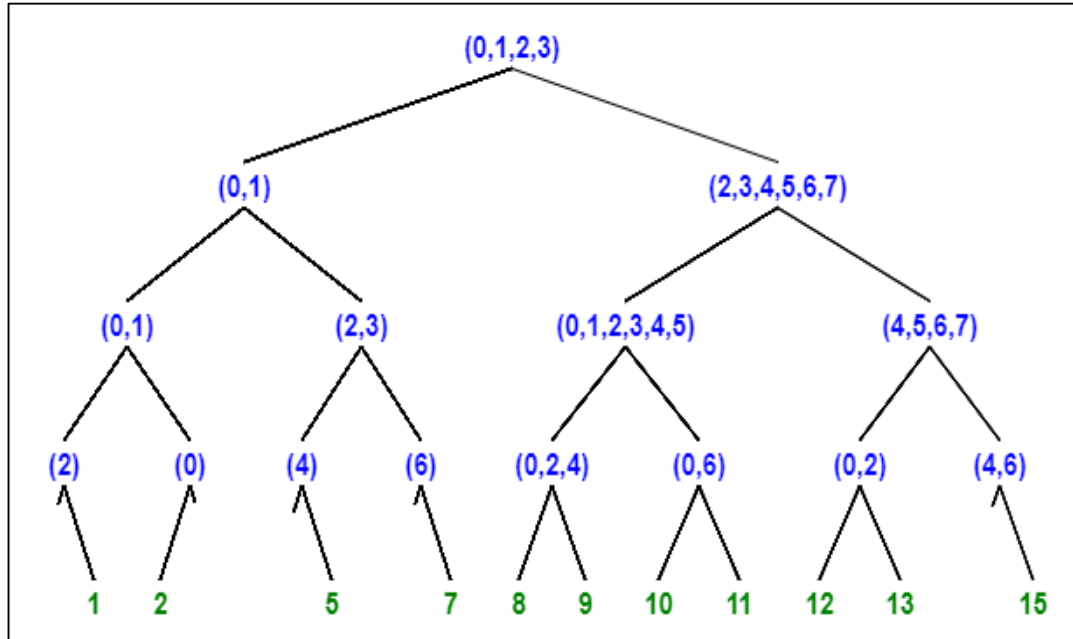
2.4.3. Erişilebilirlik ağacının oluşturulması

Bu bölümde erişilebilirlik ağacının oluşturulmasını örnek bir kural vektörü kullanılarak gerçekleştirilecektir. Erişilebilirlik ağacı oluşturmak için kullanılacak kural vektörü $CA < 78, 142, 201, 92 >$ olsun. Bu kural vektörüne ait kuralların ikilik tabandaki değerleri ve kuralların KMT numaraları aşağıdaki Tablo 2.7'de verilmiştir.

Tablo 2.7. CA<78, 142, 201, 92> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları

Kural	KMT							
	111	110	101	100	011	010	001	000
	KMT (7)	KMT (6)	KMT (5)	KMT (4)	KMT (3)	KMT (2)	KMT (1)	KMT (0)
78	d	d	d	d	1	1	1	0
142	1	0	0	0	1	1	1	0
201	1	1	0	0	1	0	0	1
92	d	1	d	1	d	1	d	0

Aşağıdaki şekilde CA<78, 142, 201, 92> kural vektörüne ait erişilebilirlik ağacı yer almaktadır. Ağaç yüksekliği beş olduğundan dolayı ağaçta beş seviye vardır. Kök düğümü seviye0'da, yaprak düğümler seviye4'te yer almaktadır. Her bir düğümde yer alan numaralar kullanılan KMT'leri belirtmektedir. Her bir seviyedeki düğümlerin oluşturulması kural vektöründeki bir kuralın kullanılmasıyla gerçekleştirilmiştir. Kök düğümün oluşturulması kural vektöründeki birinci kural olan 78'in kullanılmasıyla gerçekleştirilmiştir. Seviye1, seviye2 ve seviye3'te yer alan düğümlerin oluşturulması sırasıyla 142, 201 ve 92 kurallarının kullanılmasıyla gerçekleştirilir. Aşağıdaki Şekil 2.11'de <78, 142, 201, 92> kural vektörüne ait erişilebilirlik ağacı yer almaktadır.



Şekil 2.11. CA<78, 142, 201, 92> kural vektörüne ait erişilebilirlik ağacı

Şimdi erişilebilirlik ağacındaki her bir düğümün nasıl oluşturulduğunu açıklayalım. Ağaçtaki düğümlerde ilgili kuralın kullanacağı KMT'ler yer almaktadır. Kök düğümü oluşturmak için Kural 78 kullanılacaktır. Kök düğümün kullanılabileceği KMT'ler KMT0, KMT1, KMT2 ve KMT3'tür. Sıfır sınırı hücresele otomata kullanıldığı için KMT3'ten sonraki KMT'ler kök düğüm için kullanılmamaktadır.

Kök düğümünden sonraki her seviyedeki düğümün oluşturulması ebeveyn düğümündeki KMT'ler kullanılarak gerçekleştirilir. Seviye1'de yer alan düğümlerdeki KMT numaraları kök düğümde yer alan KMT numaraları kullanılarak elde edilmiştir. Bunu genelleştirecek olursak; $(i+1)$ 'inci seviyedeki düğümlere ait KMT numaralarının elde edilmesini i 'inci seviyede yer alan ebeveyn düğümde yer alan KMT'ler kullanılarak gerçekleştirilir. Bunun için "İ'inci ve $(i+1)$ 'inci Kuralların KMT'leri Arasındaki İlişki" başlıklı kısımda verilen tablo kullanılacaktır.

Erişilebilirlik ağacındaki düğümlerin sol kenarının 0 ve sağ kenarının 1 olduğu daha önce belirtilmişti. Kök düğümde kullanılan KMT numaralarının (0, 1, 2, 3) olduğu için, bu KMT'lerin Kural 78'in ikilik tabandaki değerine göre sahip olacakları bit değeri sırasıyla 1, 1, 1, 0'dır. Bu değerler bir sonraki seviyedeki düğümlerde kullanılacak KMT'lerin belirlenmesine yarayacaktır. 0 değerine sahip KMT'ler sol kenara bağlı çocuk düğümün oluşturulmasında, 1 değerine sahip KMT'ler sağ kenara bağlı çocuk düğümün oluşturulmasında kullanılacaktır.

Kural 78'in KMT0'ı sol kenarda ve KMT1, KMT2, KMT3'ü sağ kenarda yer alacaktır. Diğer bir deyişle sol kenarda yer alan çocuk düğüm için KMT0'ı ve sağ kenarda yer alan çocuk düğüm için KMT1, KMT2 ve KMT3'ü kullanılacaktır.

Seviye1'de yer alan (0,1) KMT numaralarına sahip düğümü nasıl elde edildiğini anlamaya çalışalım. Düğümde yer alan (0,1) KMT numaraları kök düğümün solunda yer alan KMT0'dan elde edilmiştir. Seviye1'de yer alan (2, 3, 4, 5, 6, 7) KMT numaralarına sahip düğümün KMT numaraları kök düğümün sağında yer alan KMT1, KMT2 ve KMT3'ten türetilmiştir.

Seviye2'de yer alan düğümlerinin KMT değerlerinin elde edilmesini inceleyelim. Seviye2'de dört tane düğüm yer almaktadır. Bu düğümler sırasıyla (0,1), (2,3), (0,1,2,3,4,5) ve (4,5,6,7) KMT numaralarına sahiptirler. (0,1) KMT numaralarına

sahip düğümün KMT değerleri, ebeveyn düğümün sol kenarında yer alan KMT0 kullanılarak türetilmiştir. (2,3) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sağ kenarında yer alan KMT1 kullanılarak türetilmiştir. Üçüncü düğüm olan ve (0,1,2,3,4,5) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sol kenarında yer alan KMT4, KMT5, KMT6 kullanılarak türetilmiştir. (4,5,6,7) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sağ kenarında yer alan KMT2, KMT3, KMT7 kullanılarak türetilmiştir.

Seviye3'te yer alan düğümlerin KMT değerlerinin elde edilmesini inceleyelim. Seviye3'te sekiz tane düğüm yer almaktadır. Bu düğümler sırasıyla (2), (0), (4), (6), (0,2,4), (0,6), (0,2) ve (4,6) KMT numaralarına sahiptirler. (2) KMT numarasına sahip düğümün KMT değerleri, ebeveyn düğümün sol kenarında yer alan KMT1 kullanılarak türetilmiştir. Bu düğümde (2,3) yazılması gerekirken sadece (2) yazılmıştır. Bunun nedeni son kuralda tek numaralı KMT değerlerinin kullanılamamasından kaynaklanmaktadır. Tek numaralı KMT'leri kullanılmamasının sebebi; sıfır sınırlı hücreli otomata kullanımından kaynaklanıyor. Seviye3'te düğümlerinde hiçbirinde tek KMT numarası yer almamaktadır.

Seviye3'te (0) KMT numarasına sahip düğümün KMT değerleri, ebeveyn düğümün sağ kenarında yer alan KMT0 kullanılarak türetilmiştir. Üçüncü düğüm olan ve (4) KMT numarasına sahip düğümün KMT değerleri, ebeveyn düğümün sol kenarında yer alan KMT2 kullanılarak türetilmiştir. (6) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sağ kenarında yer alan KMT3 kullanılarak türetilmiştir. (0,2,4) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sol kenarında yer alan KMT1, KMT2, KMT4, KMT5 kullanılarak türetilmiştir. (0,6) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sağ kenarında yer alan KMT1, KMT3 kullanılarak türetilmiştir. (0,2) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sol kenarında yer alan KMT4, KMT5 kullanılarak türetilmiştir. (4,6) KMT numaralarına sahip düğümün KMT değerleri, ebeveyn düğümün sağ kenarında yer alan KMT6, KMT7 kullanılarak türetilmiştir.

Son olarak yaprak düğümlerin elde edilmesini inceleyelim. Yaprak düğümler seviye 3'te yer alan düğümlerin sahip oldukları KMT değerlerine göre belirlenecektir. Yaprak düğümlerin elde edilmesi yukarıda anlatılan diğer düğümleri elde etmeden biraz

farklıdır. Yaprak düğümleri elde ederken düğümlerin KMT numaralarının bulunmasına gerek yoktur. Bunu yerine yaprak düğümün yer alacağı kenarın belirlenmesi yeterli olacaktır.

Seviye 3'te sekiz tane düğüm yer almaktadır. Bu sekiz düğüm kullanılarak yaprak düğümler belirlenecektir. Yaprak düğümlerin belirlenmesi Kural 92'nin KMT'leri kullanılarak gerçekleştirilecektir. Kural 92 son kural olduğu için KMT numarası tek olan KMT'leri kullanılamamaktadır. Kural 92'nin kullanılabilecek KMT'leri KMT0, KMT2, KMT4 ve KMT6'dır.

Seviye 3'te yer alan düğümlerin sol ve sağ kenarlarının olup olmaması Kural 92'nin KMT'leri KMT0, KMT2, KMT4 ve KMT6'nın sahip oldukları değere göre belirlenir. KMT0 sol kenarın oluşması için KMT2, KMT4 ve KMT6 sağ kenarın oluşması için gerekli olacaktır.

Seviye 3'te yer alan düğümlerin sahip oldukları KMT numaralarına göre sol ve sağ kenarlarının olup olmamasına bakalım:

- (2) değerine sahip düğüm sadece sağ kenara sahip olacaktır.
- (0) değerine sahip düğüm sadece sol kenara sahip olacaktır.
- (4) değerine sahip düğüm sadece sağ kenara sahip olacaktır.
- (6) değerine sahip düğüm sadece sağ kenara sahip olacaktır.
- (0,2,4) değerine sahip düğüm sol ve sağ kenarlara sahip olacaktır.
- (0,6) değerine sahip düğüm sol ve sağ kenarlara sahip olacaktır.
- (0,2) değerine sahip düğüm sol ve sağ kenarlara sahip olacaktır.
- (4,6) değerine sahip düğüm sadece sağ kenara sahip olacaktır.

Son olarak yaprak düğümlerin sahip olacağı değerleri hesaplayalım. Erişilebilirlik ağacının sol kenarı 0 ve sağ kenarı 1 değerini temsil etmektedir. Kök düğümünden yaprak düğüme ulaşmak için kullanılacak kenarlar yaprak düğümü belirleyecektir. Yaprak düğümlerin değerlerini sol baştaki düğümünden başlayarak belirleyelim:

- Kök düğümünden sol/sol/sol/sağ kullanılarak ulaşılan düğüm değeri 0001=1
- Kök düğümünden sol/sol/sağ/sol kullanılarak ulaşılan düğüm değeri 0010=2
- Kök düğümünden sol/sağ/sol/sağ kullanılarak ulaşılan düğüm değeri 0101=5
- Kök düğümünden sol/sağ/sağ/sağ kullanılarak ulaşılan düğüm değeri 0111=7

- Kök düğümünden sağ/sol/sol/sol kullanılarak ulaşılan düğüm değeri $1000=8$
- Kök düğümünden sağ/sol/sol/sağ kullanılarak ulaşılan düğüm değeri $1001=9$
- Kök düğümünden sağ/sol/sağ/sol kullanılarak ulaşılan düğüm değeri $1010=10$
- Kök düğümünden sağ/sol/sağ/sağ kullanılarak ulaşılan düğüm değeri $1011=11$
- Kök düğümünden sağ/sağ/sol/sol kullanılarak ulaşılan düğüm değeri $1100=12$
- Kök düğümünden sağ/sağ/sol/sağ kullanılarak ulaşılan düğüm değeri $1101=13$
- Kök düğümünden sağ/sağ/sağ/sağ kullanılarak ulaşılan düğüm değeri $1111=15$

Yukarıdaki erişilebilirlik ağacını özetli: $CA<78, 142, 201, 92>$ kural vektörü kullanılarak oluşturulan ağaç yüksekliği beştir. Ağacın sahip olduğu yaprak sayısı 11'dir. Bu $CA<78, 142, 201, 92>$ kural vektörünün ürettiği durum geçiş diyagramlarındaki ulaşılabilir durum sayısını vermektedir. Kural vektörünün ürettiği durum geçiş diyagramındaki ulaşılabilir durumlar erişilebilirlik ağacının yapraklarında yer alan değerlerdir. $CA<78, 142, 201, 92>$ kural vektörünün ürettiği durum geçiş diyagramlarına ait ulaşılabilir durumlar; 1, 2, 5, 7, 8, 9, 10, 11, 12, 13 ve 15 durumlarıdır.

Bir sonraki bölümde tek uzunluk döngülü cezbediciler için erişilebilirlik ağacı anlatılmıştır. Tek uzunluk döngülü cezbedici üreten kural vektörünün nitelenmesi, cezbediciler için oluşturulan erişilebilirlik ağacı kullanılarak gerçekleştirilir.

2.4.4. Cezbediciler için erişilebilirlik ağacı

Bir önceki bölümde erişilebilir durumlar için "Erişilebilirlik Ağacı" oluşturulması anlatılmıştır. Erişilebilirlik ağacında değişiklik yaparak sadece cezbedici durumları içeren erişilebilirlik ağacı oluşturulabilir.

Bir hücreye ait kuralın tüm KMT'leri cezbedici oluşumuna katkı sağlamıyor olabilir. Kural KMT'sinin cezbedici oluşumuna katkı sağlaması için "özellik 1" tanımına uyması gerekir. Bu konu hakkındaki detaylı bilgiyi "özellik 1" başlıklı bölümde yer almaktadır.

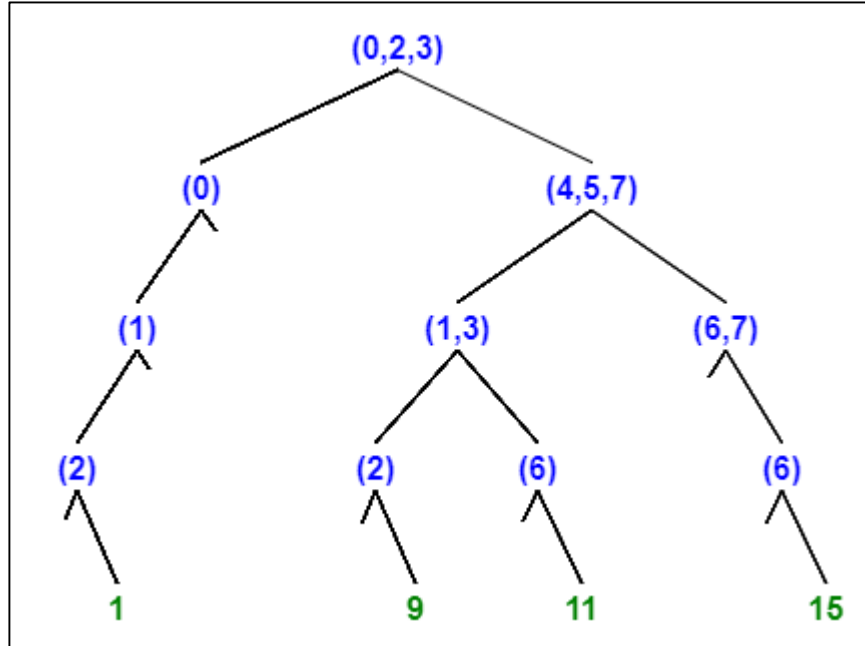
$CA<78, 142, 201, 92>$ kural vektörünün birinci kuralı olan kural 78'in bir sonraki durumun hesaplanmasında kullanılan KMT numaraları; 0, 1, 2, 3. Bu KMT'lerden hepsi tek uzunluk döngülü cezbedici oluşumuna katkı sağlamamaktadır. Kural 78

KMT'lerinden cezbedici oluşumuna katkı sağlayan KMT numaraları; 0, 2, 3. Kural 78 KMT'lerinden 1 numaralı KMT'ler cezbedici oluşumuna katkı sağlamadığından dolayı bir sonraki düğümün KMT hesaplamasında kullanılmayacaktır.

Kural 142'nin bir sonraki durum hesaplanmasında kullanılan KMT numaraları; 0, 1, 2, 3, 4, 5, 6, 7. Bu KMT'lerden hepsi cezbedici oluşumuna katkı sağlamamaktadır. Kural 142 KMT'lerinden cezbedici oluşumuna katkı sağlayan KMT numaraları; 0, 2, 3, 4, 5, 7. Kural 142 KMT'lerinden 1 ve 6 numaralı KMT'ler cezbedici oluşumuna katkı sağlamadığından dolayı bir sonraki düğümün KMT hesaplamasında kullanılmayacaktır.

Kural 201 ve 92 KMT'lerinden cezbedici oluşumuna katkı sağlayan KMT numaraları sırasıyla (1, 3, 4, 5, 6, 7) ve (0, 2, 6). Seviye 3'teki düğümlerin KMT'lerinin hesaplanmasında (1, 3, 4, 5, 6, 7) numaralı KMT'ler ve seviye 4'teki yani yaprak düğümlerin belirlenmesinde (0, 2, 6) numaralı KMT'ler kullanılacaktır.

Cezbediciler için erişilebilirlik ağacı oluşturulurken sadece cezbedici oluşumuna katkı sağlayan kural KMT'leri kullanılacaktır. CA<78, 142, 201, 92> kural vektörüne ait erişilebilirlik ağacı aşağıdaki Şekil 2.12'de yer almaktadır.



Şekil 2.12. CA<78, 142, 201, 92> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı

Yukarıdaki şekilde CA<78, 142, 201, 92> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı yer almaktadır. Her bir seviyede yer alan düğümlerde cezbedici oluşumuna katkı sağlayan KMT'ler vardır. Kök düğümündeki (0, 2, 3) KMT numaraları kural 78'e aittir. Seviye 1'deki düğümlerde yer alan (0) ve (4,5,7) KMT numaraları kural 142'ye aittir. Seviye 2'deki düğümlerde yer alan (1), (1,3) ve (6,7) KMT numaraları kural 201'e aittir. Seviye 3'teki düğümlerde yer alan (2), (2), (6) ve (6) KMT numaraları kural 92'ye aittir.

Seviye 4'te yer alan düğümler yaprak düğümledir. Bu düğümler CA<78, 142, 201, 92> kural vektörünün ürettiği cezbedici durumların değerlerini tutmaktadır. CA<78, 142, 201, 92> kural vektörünün ürettiği cezbedici durum sayısı dördüttür. Bu cezbedici durumlar 1, 9, 11 ve 15 durumlardır.

Erişilebilirlik ağacının sol kenarı 0 ve sağ kenarı 1 değerini temsil etmektedir. Kök düğümden yaprak düğüme ulaşmak için kullanılacak kenarlar yaprak düğümü belirleyecektir. Yaprak düğümlerin değerlerini sol baştaki düğümden başlayarak belirleyelim:

- Kök düğümden sol/sol/sol/sağ kullanılarak ulaşılan düğüm değeri 0001=1
- Kök düğümden sağ/sol/sol/sağ kullanılarak ulaşılan düğüm değeri 1001=9
- Kök düğümden sağ/sol/sağ/sağ kullanılarak ulaşılan düğüm değeri 1011=11
- Kök düğümden sağ/sağ/sağ/sağ kullanılarak ulaşılan düğüm değeri 1111=15

Buraya kadar erişilebilirlik ağacının oluşturulması ve erişilebilirlik ağacının cezbediciler için oluşturulması açıklanmıştır. Erişilebilirlik ağacı hücrel otomata kural vektörünün nitelenmesinde önemli bir yere sahiptir. Das ve arkadaşları erişilebilirlik ağacını hücrel otomata kural vektörünün nitelenmesinde kullanmışlardır (Das ark.,2009). Bir sonraki kısımlarda Das ve arkadaşlarının erişilebilirlik ağacı kullanarak önerdikleri algoritmalar yer verilecektir. Algoritmalar tez uzunluk döngülü cezbedicilerin elde edilmesini adım adıma anlataktadır.

2.4.5. Algoritma 1: kural vektörünün ürettiği tek uzunluk döngülü cezbedici durum sayısı

Önceki bölümlerde erişilebilirlik ağacının oluşturulmasını ve erişilebilirlik ağacının cezbedici durumlar için kullanılması incelenmiştir. Bu bölümde cezbedici durumların elde edilmesini algoritma adımlarıyla açıklanacaktır.

Hücresele otomata kural vektörünün ürettiği cezbedici durumları bulmak için kural vektörünü soldan sağa doğru tarayarak her bir kural ele alınmalıdır. Diğer yandan erişilebilirlik ağacının oluşumu dikkate alınmalıdır. Cezbediciler için erişilebilirlik ağacının yaprak sayısı, kural vektörünün ürettiği cezbedici sayısını verir.

Girdi: n uzunluklu kural vektörü $CA \langle R_1, \dots, R_i, \dots, R_n \rangle$
Çıktı: Kural vektörünün ürettiği cezbedici sayısı
Adım 1: Eğer kural vektöründeki her hangi bir kural özellik 1'i taşıyorsa cezbedici durum sayısını 0 olarak belirle ve algoritmayı sonlandır
Adım 2: S_0 ve S_1 birinci kuralın cezbedici oluşumuna katkı sağlayan iki KMT kümesi olsun. S_0 KMT'leri 0 ve S_1 KMT'leri 1'dir.
Eğer $S_0 \neq 0$ ise sol alt ağacın cezbedici durum üretme kapasitesini 2^{n-1} olarak ata.
Eğer $S_1 \neq 0$ ise sağ alt ağacın cezbedici durum üretme kapasitesini 2^{n-1} olarak ata.
Adım 3: For $i=2$ 'den n 'e
Her bir KMT kümesi için {
Erişilebilirlik ağacındaki bir sonraki seviyede yer alan her bir düğüm için cezbedici durum üretme kapasitesine sahip KMT kümelerini ebeveyn düğümün sahip olduğu kuralı dikkate alarak tanımla.
Elde edilen KMT'lerini S'_0 ve S'_1 kümelerine ayır. S'_0 KMT'leri 0 ve S'_1 KMT'leri 1'dir.
Eğer $S'_0 \neq 0$ ise sol alt ağacın cezbedici durum üretme kapasitesini 2^{n-i} olarak ata.
Eğer $S'_1 \neq 0$ ise sağ alt ağacın cezbedici durum üretme kapasitesini 2^{n-i} olarak ata.
}
Yinelenen kümeler (alt bölüm ağaçları) varsa, yalnızca birini düşünün ve ağırlığını tüm yinelenen kümeler toplamı olarak atayın.
Adım 4: Yaprak düğüme sahip olan ağaçların ağırlıkları toplanır ve cezbedici sayısı elde edilir.

Şekil 2.13. Algoritma 1: N uzunluklu CA kural vektörünün ürettiği tek uzunluk döngülü cezbedici durum sayısını veren algoritma (Das ve ark., 2009)

Erişilebilirlik ağacının alt ağaçları cezbedici duruma sahip olabilme kapasitesini belirler. Cezbedici durum üretme kapasitesi alt ağacı olmayan ağaçlar(yaprak düğüme sahip ağaçlar) için hesaplanır. Son olarak her yaprak düğüme sahip olmayan ağaçların

ürettiği cezbedici sayısı toplanarak kural vektörünün ürettiği cezbedici sayısı elde edilir. Şekil 2.13'te yer alan algoritmada kural vektörünü ürettiği cezbedici sayısının elde eden algoritmaya yer verilmiştir.

Algoritmanın karmaşıklığı kural vektörünün uzunluğuna ve RMT sayısına bağlıdır. Kural vektörü uzunluğuna n diyecek olursak zaman karmaşıklığı $T(n)=8n=O(n)$ olacaktır. RMT sayısı her zaman 8 olduğu için zaman karmaşıklığı ön kötü durum için $O(n)$ kabul edilir.

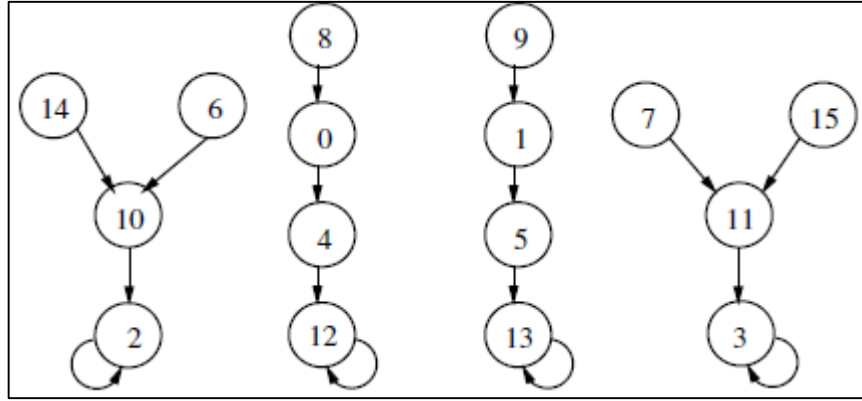
Cezbediciler için erişilebilirlik ağacı oluşturulurken sadece cezbedici oluşumuna katkı sağlayan kural KMT'leri kullanılır $CA<78, 142, 201, 92>$ kural vektörüne ait erişilebilirlik ağacı Şekil 2.12'de yer almaktadır.

Bir sonraki kısımda kapsamlı bitlerin tanımına ve erişilebilirlik ağacı kullanarak kapsamlı bitlerin belirlenmesine yer verilmiştir.

2.4.6. Algoritma 2: kapsamlı bitlerin belirlenmesi

Bir önceki kısımda erişilebilirlik ağacı kullanarak cezbedici durum sayısını elde etme açıklanmıştı. Bu bölümde cezbedici durumları daha az sayıda bitle temsil etme kapasitesine sahip kapsamlı bitlerin elde edilmesi anlatılmıştır. Öncelikle kapsamlı bitlerin ne olduğunu anlamaya çalışalım.

Kapsamlı Bitler (Pseudo-Exhaustive Bits): Bir m bit kümesi, n -hücreli bir CA'nın 2^m cezbedicisini benzersiz şekilde tanımlayabilir; burada $m \leq n$. Bu bitler, 2^m tane cezbedici içerisinde görünebilirler ve Kapsamlı Bitler(KB) olarak adlandırılırlar. Aşağıdaki Şekil 2.14'te $CA<10, 69, 204, 68>$ kural vektörünün ürettiği cezbedici durumlar yer almaktadır; 2(0010), 3(0011), 12(1100) ve 13(1101). Cezbedici durumların ikilik tabandaki bit değerlerine dikkate edelim. Cezbedicilerin sağdan son iki bitleri sırasıyla 10, 11, 00 ve 01 bitlerinden oluşmaktadır. Bu bitlerin her biri benzersizdir. Sadece bu bitleri kullanarak cezbedici durumları temsil edebiliriz.



Şekil 2.14. CA<10, 69, 204, 68> kural vektörüne ait durum geçiş diyagramı (Das ve ark., 2009).

Cezbediciler n bittten oluşuyorsa, K tane bit 2^K kadar cezbedici durumu temsil edebilir, burada $K \leq n$. Şekil 2.15'te erişilebilirlik ağacı kullanarak cezbedicileri temsil edecek kapsamlı bitleri belirleyen algoritma yer almaktadır. Algoritma erişilebilirlik ağacının seviyelerini dolaşmakta ve eğer seviyelerdeki düğümlerde sol ve sağ çocuğu olan düğüme denk gelirse o seviye için kullanılan biti kapsamlı bit olarak atamaktadır.

Girdi: n uzunluklu kural vektörü CA< $R_1, \dots, R_i, \dots, R_n$ >

Çıktı: Belirleyici bitler

Adım 1: Eğer kural vektöründeki her hangi bir kural özellik 1'i taşıyorsa belirleyici bit sayısını 0 olarak belirle ve algoritmayı sonlandır

Adım 2: S_0 ve S_1 birinci kuralın cezbedici oluşumuna katkı sağlayan iki KMT kümesi olsun. S_0 KMT'leri 0 ve S_1 KMT'leri 1'dir.

Eğer $S_0 \neq S_1$ birinci biti belirleyici bit olarak ata.

Adım 3: For $i=2$ 'den n 'e

Her bir KMT kümesi için {

Erişilebilirlik ağacındaki bir sonraki seviyede yer alan her bir düğüm için cezbedici durum üretme kapasitesine sahip KMT kümelerini ebeveyn düğümün sahip olduğu kuralı dikkate alarak tanımla.

Elde edilen KMT'lerini S'_0 ve S'_1 kümelerine ayır. S'_0 KMT'leri 0 ve S'_1 KMT'leri 1'dir.

Eğer $S'_0 \neq S'_1$ i'ninci biti belirleyici bit olarak ata

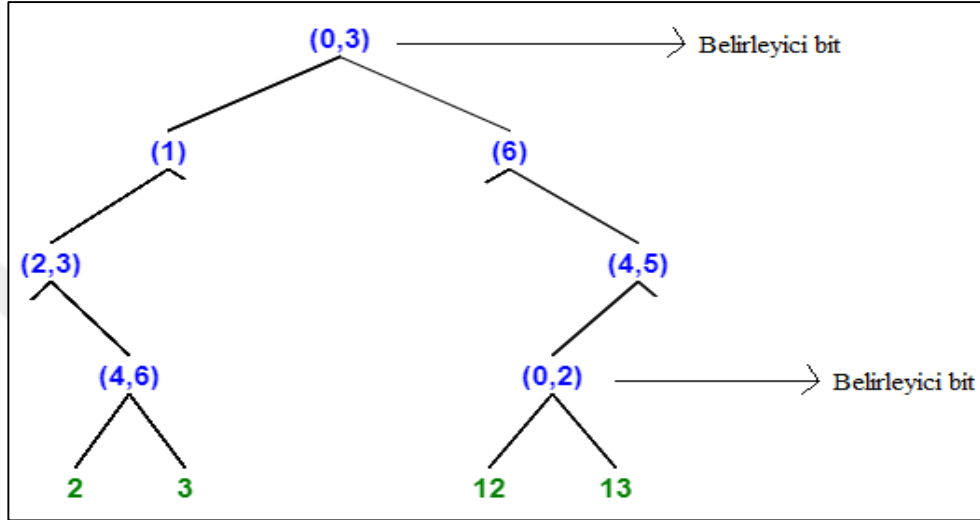
}

Adım 4: Belirleyici bit olarak atanan bitleri göster

Şekil 2.15. Algoritma 2: N uzunluklu CA kural vektörünün ürettiği tek uzunluk döngülü cezbedici durum sayısını temsil edebilen belirleyici bitleri bulun algoritma (Das ve ark., 2009)

Algoritma 2'nin karmaşıklığı kural vektörünün uzunluğuna ve KMT sayısına bağlıdır. N tane kurala sahip kural vektörü ve 8 KMT değeri için zaman karmaşıklığı şöyledir; $T(n)=8n=O(n)$.

Aşağıda Şekil 2.16'da yer alan erişilebilirlik ağacını incelenerek Algoritma 2 daha iyi anlaşılabilir.



Şekil 2.16. CA < 10, 69, 204, 68 > kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı

Yukarıdaki şekilde CA < 10, 69, 204, 68 > kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı yer almaktadır. Ağaçtaki her bir seviye bir biti temsil etmektedir. Kök düğüm soldan birinci biti ve yapraklardan bir önceki seviye sonuncu biti temsil etmektedir. Belirleyici bitler bulmak için ağacın her bir seviyesi sırayla kontrol edilir. Eğer seviyede yer alan düğümlerden birisi iki tane çocuğa sahip ise bu seviyeye karşılık gelen bit pozisyonu belirleyici bit olarak atanır. Yukarıdaki şekilde yer alan erişilebilirlik ağacında kök düğüm sol ve sağ çocuklara sahip olduğu için birinci bit belirleyici bittir. Seviye 3'te yer alan düğümlerden iki çocuğa sahip iki tane düğüm olduğu için 4'üncü bitte belirleyici bit olarak atanır. Bu bitler tüm cezbedici durumlarda görüldüğü için kapsayıcı bitlerdir.

Bu kısımda belirleyici bitlerin bulunması ve kapsayıcı bit tanımlamasını yapılmıştır. Bir sonraki kısımda bu bölümde anlatılan teorik alt yapıya dayanan çoklu cezbedici hücresel otomatların sentez şeması sunulacaktır.

2.4.7. Algoritma 3: istenen sayıda K bit için çoklu cezbedici hücresel otomata sentezi

Bu kısımda, bir önceki kısımda yer alan teorik alt yapıya dayalı çoklu cezbedici hücresel otomatların sentez edilmesinin çözüm adımları vardır. Aşağıdaki Şekil 2.17’de yer alan Algoritma 3’te kural vektörü elde etmek için kullanılan sentez yönteminin algoritması yer almaktadır. Algoritma girdi olarak kural vektörü uzunluğunu ve kapsamlı bit sayısını almakta ve çıktı olarak istenen uzunluğu sahip kural vektörü vermektedir.

Girdi: n (Kural vektörü uzunluğu) ve K (Kapsamlı bit sayısı)
Çıktı: Kural Vektörü
Adım 1: Kapsamlı bit yeteneğine sahip K tane biti rasgele belirle
Adım 2: S_0 ve S_1 birinci kuralın cezbedici oluşumuna katkı sağlayan iki KMT kümesi olsun. S_0 KMT’leri 0 ve S_1 KMT’leri 1’dir.
Eğer birinci bit kapsamlı bit ise, $S_0 \neq \emptyset \neq S_1$ olacak şekilde rasgele ata.
Değilse, KMT kümelerini $S_0 \neq \emptyset$ & $S_1 = \emptyset$ veya $S_0 = \emptyset$ & $S_1 \neq \emptyset$ olacak şekilde ata
Adım 3: For $i=2$ ’den n ’e
Her bir KMT kümesi için {
Erişilebilirlik ağacındaki bir sonraki seviyede yer alan her bir düğüm KMT kümelerini ebeveyn düğümün sahip olduğu kuralı dikkate alarak tanımla.
Eğer i ’ninci bit kapsamlı bit ise, KMT kümelerini $S'_0 \neq \emptyset \neq S'_1$ olacak şekilde rasgele ata.
Değilse, KMT kümelerini $S'_0 \neq \emptyset$ & $S'_1 = \emptyset$ veya $S'_0 = \emptyset$ & $S'_1 \neq \emptyset$ olacak şekilde ata
}
Adım 4: Eğer varsa boş kalan KMT’leri doldur, atama yaparken ekstradan kapsayıcı bit oluşmamasına dikkat et.
Adım 5: K tane kapsamlı bite sahip kural vektörünü rapor et.

Şekil 2.17. Algoritma 3: İstenen sayıda kapsamlı bit üreten kural vektörünü çıktı olarak veren çoklu cezbedici hücresel otomata sentez algoritması (Das ve ark., 2009)

Algoritma 3’ün karmaşıklığı kural vektörünün uzunluğuna ve KMT sayısına bağlıdır. N tane kurala sahip kural vektörü ve 8 KMT değeri için zaman karmaşıklığı şöyledir; $T(n)=8n=O(n)$.

Yukarıdaki algoritma girdi olarak kural vektörü uzunluğunu ve kapsamlı bit sayısını almakta ve çıktı olarak istenen uzunluğu sahip kural vektörü vermektedir. Yukarıdaki algoritmaya açıklayalım: Birinci adımda k tane kapsamlı bit rasgele atanır. İkinci adımda, soldan birinci bittin kapsamlı bit olarak atanıp atanmadığı kontrol edilir. S_0

ve S_1 birinci kuralın cezbedici oluşumuna katkı sağlayan iki KMT kümesi olsun. S_0 KMT'leri 0 ve S_1 KMT'leri 1'dir. Eğer soldan birinci bit kapsamlı bit olarak atanmış ise $S_0 \neq \emptyset \neq S_1$ olacak şekilde KMT'ler rasgele atanır. Eğer soldan birinci bit kapsamlı bit olarak atanmamışsa KMT kümelerini $S_0 \neq \emptyset \& S_1 = \emptyset$ veya $S_0 = \emptyset \& S_1 \neq \emptyset$ olacak şekilde rasgele atanır.

Üçüncü adımda diğer bitlerin KMT atamaları birinci bitte uygulanan kriterler dikkate alınarak yapılır. Dördüncü adımda, kapsamlı bitlerin KMT atamaları bittikten sonra hala atanmamış KMT'lerin ataması yapılır. KMT ataması yapılırken ekstra kapsamlı bit oluşmamasına dikkat ederek KMT atamaları yapılır.

Bu bölümde istenen sayıda kapsamlı bit üreten kural vektörünün elde edilmesi anlatılmıştır. Bu bölümdeki algoritmanın eksikliği kural vektörünün cezbedici uzunluğu birden fazla olan durumları dikkate almamasıdır. Algoritma 3 ile istenen sayıda kapsamlı bit elde edilmesine odaklanılmıştır. Algoritma 3'ün çıktı olarak verdiği kural vektörü cezbedici uzunluğu birden fazla olan durumlar üretebilir. Bir sonraki bölümde KMT dizilimi kavramı açıklanmıştır. KMT dizilimi kural vektörünün ürettiği cezbedicilerin döngü uzunluklarını nitelemede önemli bir yere sahiptir.

2.4.8. KMT dizilimi

Bu bölümde kural vektörünün ürettiği cezbedicilerin döngü uzunluklarını nitelemede önemli bir yere sahip olan KMT Dizilimi(KD) açıklanmıştır. Sonraki kısımlarda yer verilecek algoritmaların anlaşılması için KMT diziliminin anlaşılması gerekiyor.

Bir ulaşılabilir durum için erişilebilirlik ağacında gezinirken kullanılan KMT'lere KMT dizilimi adı verilir ve şöyle gösterilir: $KD\langle x_1 x_2 \dots x_n \rangle$. $KD\langle x_1 x_2 \dots x_n \rangle$ ulaşılabilir durumlar için kullanılan KMT'leri içerir (Das ve ark., 2009).

Örneğin $CA\langle 24, 74, 37, 41 \rangle$ kural vektörü kullanarak 0110 durumunun bir sonraki durumunu bulalım. 0110 örüntüsünün bir sonraki durumu için kullanılacak KMT'ler sırasıyla KMT1, KMT3, KMT6 ve KMT4'tür. Bu KMT'ler sırasıyla birinci, ikinci, üçüncü ve dördüncü kurallara aittir. Diğer bir ifadeyle bu KMT'ler sırasıyla 24, 74, 37 ve 41 kurallarına aittir. $CA\langle 24, 74, 37, 41 \rangle$ kural vektörüne ait kuralların ikilik tabandaki değerleri ve kuralların KMT numaraları aşağıdaki Tablo 2.8'de verilmiştir.

Tablo 2.8. CA<24, 74, 37, 41> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları

Kural	KMT							
	111	110	101	100	011	010	001	000
	KMT (7)	KMT (6)	KMT (5)	KMT (4)	KMT (3)	KMT (2)	KMT (1)	KMT (0)
24	d	d	d	d	1	0	0	0
74	0	1	0	0	1	0	0	0
37	0	0	1	0	0	1	0	1
41	d	0	d	0	d	0	d	1

0110 örüntüsünün CA<24, 74, 37, 41> kural vektörüne göre bir sonraki durumu şöyledir; 0100. 0110 örüntüsünün bir sonraki durumunu elde etmek için kullanılan KMT dizilimini şöyle gösterilir; KD<1364>.

Ulaşılabilir bir durum ve bu durumun bir sonraki durumunun KD'leri arasında ilişki vardır. Bu ilişkiyi anlamak için 8 KMT iki farklı gruba ayrılmıştır; $KMT_0 = \{0, 1, 4, 5\}$ ve $KMT_1 = \{2, 3, 6, 7\}$. KMT_0 ve KMT_1 kümelerinde yer alan KMT'ler tek uzunluk döngülü cezbedici oluşumuna katkı sağlamalarına göre ayrılmıştır. KMT_0 kümesinde yer alan KMT'lerinin orta hücre değeri 0'dır ve KMT_1 kümesinde yer alan KMT'lerin orta hücre değeri 1'dir.

$\langle x_1 x_2 \dots x_n \rangle$, n hücreli bir hücresel otomata için bir sonraki durumu elde etmek için kullanılan KMT dizilimi olsun. i'ninci kuralın KMT'si x_i olsun. Şimdi KMT x_i özellik 1'i sağlamasın ve $x_i \in KMT_0$ olsun. Bu şartlarda KMT x_i 'nin bir sonraki durum değeri 1 olacaktır. KMT x_{i-1} ve KMT x_{i+1} değerleri 0 veya 1 olabilir. $\langle y_1 y_2 \dots y_n \rangle$, bir sonraki durumun KMT dizilimi olsun. Bu şartlarda KMT y_i 010(KMT1), 011(KMT3), 110(KMT6) veya 111(KMT7) olabilir. Şimdi bir önceki şartlardan farklı olarak KMT x_i özellik 1'i sağlasın ve $x_i \in KMT_0$ olsun. Bu şartlarda KMT x_i 'nin bir sonraki durum değeri 0 olacaktır. KMT x_{i-1} ve KMT x_{i+1} değerleri 0 veya 1 olabilir. $\langle y_1 y_2 \dots y_n \rangle$, bir sonraki durumun KMT dizilimi olsun. Bu şartlarda KMT y_i 000(KMT0), 001(KMT1), 100(KMT4) veya 101(KMT5) olabilir. İki KD arasındaki ilişki aşağıdaki Tablo 2.9 ve Tablo 2.10'da verilmiştir.

Tablo 2.9. $KMT x_i \in KMT_0$ olduğunda KD_t ve KD_{t+1} KMT Dizilimleri arasındaki ilişki (Das ve ark., 2009).

KD_t			KD_{t+1}
$KMT x_{i-1}$	$KMT x_i = \{0,1,4,5\}$	$KMT x_{i+1}$	$KMT y_i$
0	1	0	2
0	1	1	3
1	1	0	6
1	1	1	7
0	0	0	0
0	0	1	1
1	0	0	4
1	0	1	5

Tablo 2.10. $KMT x_i \in KMT_1$ olduğunda KD_t ve KD_{t+1} KMT Dizilimleri arasındaki ilişki (Das ve ark., 2009).

KD_t			KD_{t+1}
$KMT x_{i-1}$	$KMT x_i = \{2,3,6,7\}$	$KMT x_{i+1}$	$KMT y_i$
0	0	0	0
0	0	1	1
1	0	0	4
1	0	1	5
0	1	0	2
0	1	1	3
1	1	0	6
1	1	1	7

Bu tez kapsamında gerçekleştirilen CA tabanlı uygulamaların çoğu tek uzunluk döngülü cezbedicileri kullanmaktadır. Tek uzunluk döngülü cezbedicilerin nitelenmesinin anlaşılması için çok uzunluk döngülü cezbedici oluşumuna neden olan sebeplerin iyi anlaşılması gerekiyor. Bir sonraki kısımda çok uzunluk döngülü cezbedicilerin oluşumuna neden olan sebepler incelenmiştir.

2.4.9. Özellik 2

Bu bölümde çok uzunluk döngülü cezbedici oluşumu incelenmiştir. Tek uzunluk döngülü cezbedicilerin nitelenmesinin yapılabilmesi için çok uzunluk döngülü cezbedici oluşumunun anlaşılması gerekir. Mukherjee, bunu için yeni bir tanım yapmıştır. Bu tanım “Özellik 2” olarak adlandırılır ve 3.3. kısmında yer alan “Özellik 1” den yararlanır.

Özellik 2: n hücreli bir hücresel otomata için KMT dizilimi tanımlansın: $KD_t < x_1 x_2 \dots x_n >$. Bu KMT diziliminde x_i , kural R_i 'ye ait KMT olsun ve özellik 1'i karşılasın. Bir sonraki durumu inşa etmek için kullanılan KMT dizilimi $KD_{t+1} < y_1 y_2 \dots y_n >$ olsun. KD_{t+1} yer alan y_1 ile y_{i-2} arasındaki KMT'ler ile KD_t yer alan x ile x_{i-2} arasındaki KMT'ler aynıdır ve aynı şey $(i+2)$ 'den n kadar olan KMT'ler için de geçerlidir (Mukherjee, 2009).

Bir örnek vererek yukarıda tanımlanan özellik 2'yi açıklayalım. 10 hücreli bir hücresel otomata için 1101110011 örüntüsünün bir sonraki durumunu hesaplamak için kullanacağımız KMT dizilimi şöyledir; $KD_t = < 3653764136 >$. Bu KMT diziliminde beşinci bitte yer alan KMT7'nin özellik 1'i karşılamadığını varsayalım. 1101110011 örüntüsünün bir sonraki durumu 1101010011 olacaktır Bir sonraki durum için KMT dizilimi şöyledir; $KD_{t+1} = < 365\mathbf{252}4136 >$. Bir sonraki durumu ait KMT diziliminde üç farklı KMT vardır. Bunlar dördüncü, beşinci ve altıncı pozisyonlarda yer alan KMT'lerdir. Mevcut ile bir sonraki durum KMT dizilimleri arasındaki fark mevcut durumda özellik 1'e uymayan KMT'nin sol, kendi ve sağ KMT'lerinde gerçekleşmiştir. Bu, özellik 2'de yer verilen tanımla açıklıyor. Özellik 2'deki tanıma göre değişmeyen KMT'leri hesaplayalım. Beşinci hücre KMT'si özellik 1'e uymadığı için bir sonraki durumun 1 ile $(5-2=3)$ ve $(5+2=7)$ ile 10 arasındaki KMT'leri ile mevcut durumun aynı pozisyonundaki KMT'leri aynıdır.

Mukherjee, özellik 2'yi kullanarak cezbedici döngü uzunluğu 3 ve 4 olan durumların tanımını yapmıştır (Mukherjee, 2009). Döngü uzunluğu 3 olan cezbedicilerin oluşumunu inceleyelim. Bunun için üç ayrı KMT kümesi tanımlanacaktır. Bu kümelerin isimleri M, N ve V olsun ve bu kümelerde yer alan KMT'ler özellik 1'i karşılasın. Bu üç KMT kümesi sırasıyla KD_t , KD_{t+1} ve KD_{t+2} KMT dizilimlerine

aittir KMT'lerden oluşmaktadır. KD_t , KD_{t+1} ve KD_{t+2} KMT dizilimleri aşağıdaki gibi tanımlı olsun;

$$M=KD_t=<x_1x_2x_3\dots x_n> \quad (2.1)$$

$$N=KD_{t+1}=<y_1y_2y_3\dots y_n> \quad (2.2)$$

$$V=KD_{t+2}=<z_1z_2z_3\dots z_n> \quad (2.3)$$

Cezbedici döngü uzunluğu 3 olan bir durum oluşumu için M, N ve V KMT'lerinin seçimi şöyle yapılmalıdır;

$$V=(M \cup N)-(M \cap N) \quad (2.4)$$

Yukarıda verilen eşitliği örnek vererek açıklayalım. KD_t dizilimine ait $M=\{3,i\}$ olsun. Bu KMT'ler özellik 1'i karşılamamaktadır. KD_{t+1} dizilimine ait N kümesini M kümesini kullanarak elde edilecektir. N kümesini oluştururken seçilebilecek KMT'ler bir önceki bölümde tanımlı verilen özellik 2'ye göre belirlenir. Özellik 2'ye göre KD_{t+1} diziliminden seçilebilecek KMT'ler şöyledir; $y_2, y_3, y_4, y_{i-1}, y_i, y_{i+2}$. N kümesini seçerken dikkat edilmesi gereken bir diğer durum; $(M \cup N) \neq (M \cap N)$. Eğer $(M \cup N) = (M \cap N)$ olursa V kümesi boş küme olacaktır. Bu cezbedici döngü uzunluğu 3 olan durumlar elde edilmesini engelleyecektir.

KD_{t+1} diziliminden oluşturacağımız N kümesi şöyle olsun $N=\{3, 4, i+2\}$. KD_{t+2} dizilimine ait N ve M KMT kümelerini kullanarak elde edebiliriz. V kümesi, yukarıda verilen Denklem (2.4)'e göre elde edilir ve sonuç şöyledir; $(\{3,i\} \cup \{3, 4, i+2\}) - (\{3,i\} \cap \{3, 4, i+2\}) = \{4, i, i+2\}$.

Bu kısımda çok uzunlun döngülü cezbedicilerin oluşumuna etki eden faktörlerden bahsedilmiştir. Bunun için özellik 2 tanımlı kullanılmıştır. Özellik 2 tanımlı kullanarak döngü uzunluğu 3 olan cezbedicilerin oluşumunu nitelenmesini gerçekleştirilmiştir. Bir sonraki bölümde çok uzunluk döngülü cezbedicilerin oluşumuna neden olan faktörlerin genelleştirmesi anlatılmıştır.

2.4.10. Çok uzunluk döngülü cezbedicilerin oluşumu

Bir önceki kısımda döngü uzunluğu üç olan cezbedicilerin oluşumu incelenmiştir. Bu bölümde döngü uzunluğu birden fazla olan cezbedicilerin oluşumunun genel çerçevesi çizilmiştir.

Bu tez kapsamında tek uzunluk döngülü cezbediciye sahip lineer olmayan hücresel otomatların örüntü çekme yeteneği mühendislik uygulamalarında kullanımı ele alınmıştır. Tek uzunluk döngülü cezbedicilerin nitelenmesi için çok uzunluk döngülü cezbedici oluşumuna neden olan teorik alt yapının bilinmesi gerekir.

Çok uzunluk döngülü cezbedici oluşumunun nedenleri, araştırmacılar tarafından KMT dizilimi kavramı kullanılarak açıklanmıştır (Das ve ark., 2009; Mukherjee, 2009; Naskar, 2009).

Çok uzunluk döngülü cezbedici oluşumunda ardışık KMT dizilimlerinde yer alan KMT'lerinin incelenmesi gerekir. N tane kuraldan oluşan bir hücresel otomata için KD_t KMT diziliminin i 'ninci KMT'si x_i olsun ve özellik 1'i karşılamasın. KMT x_i ait olduğu küme S_0 (veya S_1) ise, başka bir KMT diziliminde KD_{t+j} i 'ninci pozisyonda bir KMT özellik 1'i karşılamıyor ve S_1 (veya S_0) kümesine ait ise çok uzunluk döngülü cezbedici oluşumuna sebebiyet verir (Mukherjee, 2009).

Mukherjee çok uzunluk döngülü cezbedicilerin oluşumunu açıklayan bir teorem geliştirmiştir (Mukherjee, 2009).

Teorem: Eğer i 'ninci hücreye ait kural KMT'lerinden özellik 1'i karşılamayan r_n ve r_m KMT'leri varsa; bu KMT'ler $r_n \in S_0$ (veya S_1) ve $r_m \in S_1$ (veya S_0) farklı kümelere ait ise $(i+1)$ 'inci hücreye ait kuralın ardışık KMT'lerinin seçimi çok uzunluk döngülü cezbedici oluşması için aşağıdaki gibi olmalıdır:

- a) $2r_n \bmod 8$, $(2r_n+1) \bmod 8$, $2r_m \bmod 8$, $(2r_m+1) \bmod 8$ KMT'leri aynı kümeden ise ya hepsi özellik 1'i karşılamalı; veya $2r_n \bmod 8$, $2r_m \bmod 8$ özellik 1'i karşılamalı; veya $(2r_n+1) \bmod 8$, $(2r_m+1) \bmod 8$ özellik 1'i karşılamalı
- b) Veya: $2r_n \bmod 8$, $(2r_n+1) \bmod 8$, $2r_m \bmod 8$, $(2r_m+1) \bmod 8$ KMT'leri farklı kümeden ise ya hiçbirisi özellik 1'i karşılamamalı; veya $2r_n \bmod 8$, $2r_m \bmod 8$ özellik 1'i karşılamamalı; veya $(2r_n+1) \bmod 8$, $(2r_m+1) \bmod 8$ özellik 1'i karşılamamalı

Teorem çok uzunluk döngülü cezbedici elde etmek için yol göstericidir. Tek uzunluk döngülü cezbedici üreten kural vektörleri oluşturulurken çok uzunluk döngülü cezbedici oluşumunun sebeplerinin bilmemiz gerekir. Bir sonraki bölümde tek uzunluk döngülü cezbedici üreten kural vektörünün elde edilmesine yer verilmiştir.

2.4.11. Tek uzunluk döngülü cezbedici hücrel otomatların sentezi

Bu bölümde kural vektörünün sadece döngü uzunluğu bir olan cezbedici üretmesini sağlayan algoritmaya yer verilmiştir.

Önceki bölümlerde yer alan algoritma 3 istenen sayıda kapsamlı bit üreten kural vektörünün elde edilmesi için gerekli adımları içeriyordu. Algoritma 3'ün çıktısı olarak verdiği kural vektörü cezbedici uzunluğu birden fazla cezbedici durumlar üretebilir. Bu bölümde kural vektörünün sadece cezbedici uzunluğu bir olan cezbediciler üretmesini odaklanmıştır. Kural vektörünün istenen sayıda kapsamlı bit üretmesi bir sonraki bölümde ele alınacaktır.

Sadece tek uzunluk döngülü cezbedici üreten ve çok uzunluk döngülü cezbedici oluşumunu kısıtlayan kural vektörü elde etmek için Mukherjee “özellik 3” adını verdiği teorem geliştirmiştir (Mukherjee, 2009).

Özellik 3: Eğer i 'ninci hücreye ait kural KMT'lerinden özellik 1'i karşılamayan r_n ve r_m KMT'leri varsa; bu KMT'ler $r_n \in S_0$ (veya S_1) ve $r_m \in S_1$ (veya S_0) farklı kümelere ait ise $(i+1)$ 'inci hücreye ait kuralın ardışık KMT'lerinin seçimi çok uzunluk döngülü cezbedici oluşmaması için aşağıdaki gibi olmalıdır:

- a) $2r_n \bmod 8$, $(2r_n+1) \bmod 8$ KMT'leri özellik 1'i karşılamalı; $2r_m \bmod 8$, $(2r_m+1) \bmod 8$ KMT'leri özellik 1'i karşılamamalı
- b) Veya; $2r_m \bmod 8$, $(2r_m + 1) \bmod 8$ KMT'leri özellik 1'i karşılamalı; $2r_n \bmod 8$, $(2r_n+1) \bmod 8$ KMT'leri özellik 1'i karşılamamalı
- c) Veya: $2r_n \bmod 8$, $(2r_n+1) \bmod 8$, $2r_m \bmod 8$, $(2r_m+1) \bmod 8$ KMT'leri aynı kümeden ise hiçbirisi özellik 1'i karşılamamalı
- d) Veya: $2r_n \bmod 8$, $(2r_n+1) \bmod 8$, $2r_m \bmod 8$, $(2r_m+1) \bmod 8$ KMT'leri farklı kümeden ise hepsi özellik 1'i karşılamalı

Dört hücreli bir hücreli otomata kurallarını özellik 3'ü dikkate alarak oluşturulur. Dört hücreli bir hücreli otomata dört kurala sahiptir. Her bir kural bir önceki kural dikkate alınarak oluşturulur.

Sıfır sınırlı hücreli otomata kullandığımız için birinci kuralın ilk dört KMT'si(KMT0, KMT1, KMT2, KMT3) kullanılacak ve son dört KMT'si(KMT4, KMT5, KMT6, KMT7) kullanılmayacaktır. Son dört KMT'nin seçimi rasgele yapılabilir ve bir sonraki kuralın oluşturulmasında dikkate alınmaz. İlk dört KMT'den KMT0 ve KMT3 özellik 1'i karşılamasın diğer KMT'ler KMT1 ve KMT2 özellik 1'i karşılasın. Bu şartlar dikkate alınarak ilk kuralımız şöyle olsun; 162: 10100101.

İkinci kuralı birinci kuralı dikkate alarak oluşturulacaktır. Birinci kuralda $KMT0 \in S_0$ ve $KMT3 \in S_1$ özellik 1'i karşılamadığı ve farklı KMT kümelerine ait oldukları için çok uzunluk döngülü cezbedici oluşabilir. Çok uzunluk döngülü cezbedici oluşumunu engellemek için ikinci kuralın KMT'lerini özellik 3 kullanarak seçilecektir. Birinci kuralda yer alan özellik 1'i karşılamayan KMT3 ve KMT0'ın ardışık KMT'leri sırasıyla KMT6, KMT7 ve KMT0, KMT1'dir. Bu KMT'ler $(KMT6, KMT7) \in S_1$ ve $(KMT0, KMT1) \in S_0$ kümelerine aittir. Ardışık KMT'ler iki farklı kümeye ait olduğu için ardışık KMT'lerin tümünün veya kümelerden birine ait KMT'lerin özellik 1'i karşılaması gerekir. Ardışık tüm KMT'ler özellik 1'i karşılamış olsun. Birinci kurala ait KMT1 ve KMT2 iki özellik 1'i karşılamaktadır. Bu KMT'lerin ardışık KMT'leri KMT2, KMT3 ve KMT4 ve KMT5'tir. Bu KMT gruplarından birinin özellik 1'i karşılaması diğerinin karşılamaması gerekir. KMT5 ve KMT4 özellik 1'i karşılasın; KMT2 ve KMT3 özellik 1'i karşılamasın bu durumda ikinci kural şöyle olacaktır; 192: 11000000

Üçüncü kuralı ikinci kuralı dikkate alarak oluşturulacaktır. İkinci kuralda iki farklı KMT kümesine ait ve özellik 1'i karşılamayan iki KMT olmadığı için, üçüncü kurala ait KMT seçiminde tüm KMT'lerin özellik 1'i karşılamaması dışında başka kısıt yoktur. Üçüncü kurala ait KMT'lerden S_1 kümesine ait KMT7, KMT6 ve KMT3 özellik 1'i karşılasın ve S_0 kümesine ait KMT'lerden KMT0, KMT1 ve KMT5 özellik 1'i karşılasın; KMT4 ve KMT2 özellik 1'i karşılamasın. Bu şartlarda üçüncü kural şöyle olacaktır; 216: 11011000.

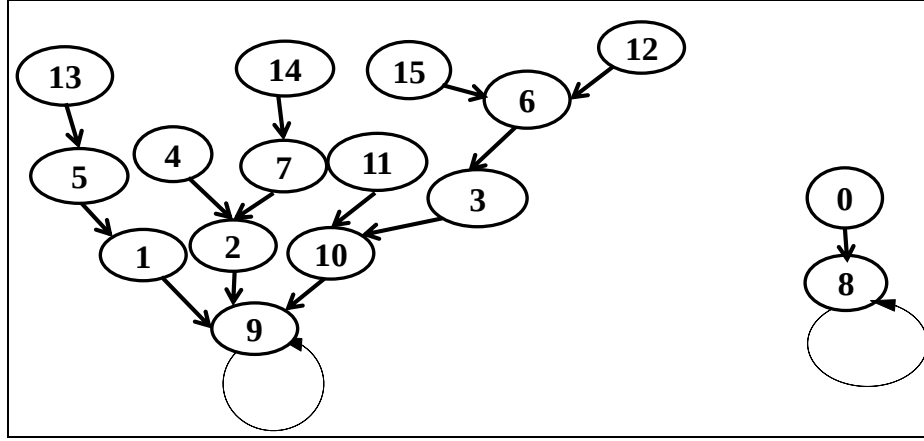
Dördüncü kuralı üçüncü kuralı dikkate alarak oluşturulacaktır. Üçüncü kuralda KMT4 $\in S_0$ ve KMT2 $\in S_1$ özellik 1'i karşılamadığı ve farklı KMT kümelerine ait oldukları için çok uzunluk döngülü cezbedici oluşabilir. Çok uzunluk döngülü cezbedici oluşumunu engellemek için dördüncü kuralın KMT'lerini özellik 3'ü kullanarak seçilecektir. Üçüncü kuralda yer alan özellik 1'i karşılamayan KMT4 ve KMT2'nin ardışık KMT'leri sırasıyla KMT0, KMT1 ve KMT4, KMT5'tir. Bu KMT'ler (KMT0, KMT1) $\in S_0$ ve (KMT4, KMT5) $\in S_0$ kümelerine aittir. Ardışık KMT'ler iki aynı kümeye (S_0) ait olduğu için ardışık KMT'lerin tümünün özellik 1'i karşılamaması veya kümelerden birine ait KMT'lerin özellik 1'i karşılaması gerekir. Ardışık tüm KMT'lerden bir kümeye ait olan KMT'ler özellik 1'i karşılamış olsun. KMT0 ve KMT1 özellik 1'i karşılasın; KMT4 ve KMT5 özellik 1'i karşılasın. Diğer KMT'lerden KMT2 ve KMT3 özellik 1'i karşılasın; KMT6 ve KMT7 özellik 1'i karşılasın. Bu şartlarda dördüncü kural şöyle olacaktır; 60:00111100

Aşağıdaki Tablo 2.11'de CA<165,192,216,60> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları yer almaktadır. Şekil 2.18'deki durum geçiş diyagramları ve Şekil 2.19'daki erişilebilirlik ağacı bu tabloya göre oluşturulmuştur.

Tablo 2.11. CA<165,192,216,60> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları

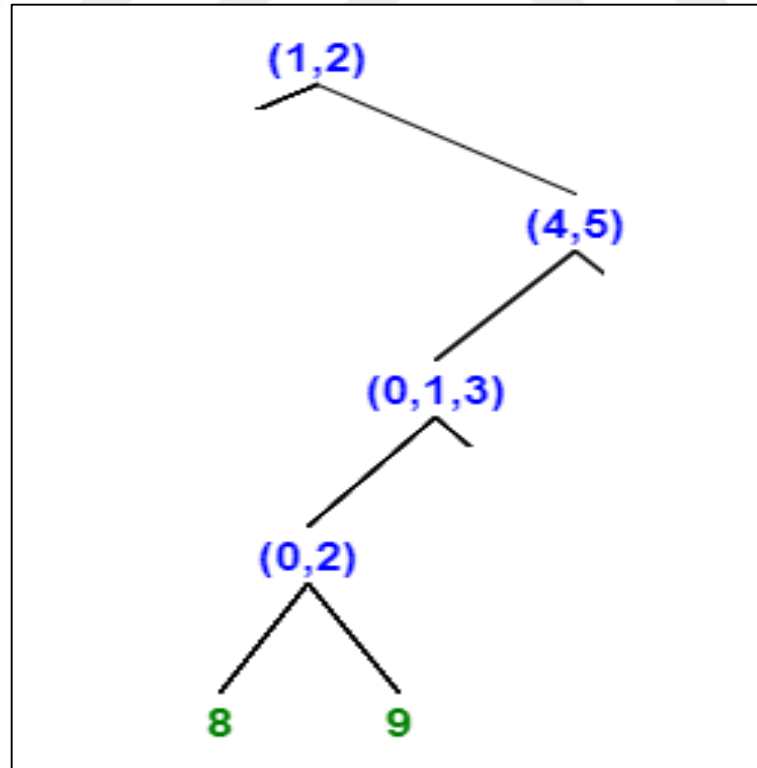
Kural	KMT							
	111	110	101	100	011	010	001	000
	KMT (7)	KMT (6)	KMT (5)	KMT (4)	KMT (3)	KMT (2)	KMT (1)	KMT (0)
165	1	0	1	0	0	1	0	1
192	1	1	0	0	0	0	0	0
232	1	1	0	1	1	0	0	0
60	0	0	1	1	1	1	0	0

CA<165,192,216,60> vektörüne ait durum geçiş diyagramı aşağıdaki Şekil 2.18.'de yer almaktadır. Kural vektörü iki cezbedici(8 ve 9) durum üretmektedir. Durumların çoğu 9 numaralı cezbedici etrafında toplanmıştır.



Şekil 2.18. CA<165, 192, 216, 60> kural vektörünün durum geçiş diyagramı

Aşağıdaki Şekil 2.19'da CA<165, 192, 216, 60> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı yer almaktadır. Cezbediciler için erişilebilirlik ağacı kural vektöründeki kurallara ait KMT'lerin özellik 1'i karşılamasına göre oluşturulur. Erişilebilirlik ağacı hakkında detaylı bilgiler yukarıdaki bölümlerde verilmiştir.



Şekil 2.19. CA<165, 192, 216, 60> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı

2.4.12. Algoritma 4: belirli kapsayıcı bitlere sahip tek uzunluk döngülü cezbedici hücrel otomatların sentezi

Bu bölümde kural vektörünün istenen sayıda kapsayıcı bite sahip ve sadece döngü uzunluğu bir olan cezbedici üretmesini sağlayan algoritmaya yer verilmiştir.

Önceki bölümlerde sadece tek uzunluk döngülü cezbediciler üreten kural vektörünün elde edilmesi ele alınmıştı. Yukarıdaki bölümün eksikliği cezbedici sayısının kontrol altına alınmamasıydı. Bu bölümde cezbedici sayısı kapsayıcı bitler ile kontrol altına alınması incelenecektir. Eğer n hücreli bir hücrel otomata k tane kapsayıcı bite sahip ise bu hücrel otomatın sahip olabileceği maksimum cezbedici sayısı; 2^k olacaktır. Bu bölümde kural vektörünün sadece cezbedici uzunluğu bir olan cezbediciler üretmesini ve kural vektörünün istenen sayıda kapsamlı bit üretmesi dikkate alınacaktır.

Kapsamlı bitlerin belirlenmesini ve istenen sayıda elde edilmesini yukarıdaki bölümlerde verilen Algoritma 2 ve Algoritma 3’de; kural vektörünün sadece tek uzunluk döngülü cezbedici üretmesini bir önceki bölümde açıklamıştık. Bu bölümde Algoritma 2 ve Algoritma 3’den farklı olarak kapsayıcı bitler elde edilirken çok uzunluk döngülü cezbedici oluşumuna izin vermeden kapsayıcı bitler elde edilecektir.

Das ve arkadaşları istenen sayıda kapsayıcı bite sahip tek uzunluk döngülü cezbedicileri üretecek kural vektörünün elde edilmesinde yardımcı olacak teoremler önermişlerdir. Teoremler kapsayıcı bite ait kuralın KMT’lerini çok uzunluk döngülü cezbedici oluşumuna izin vermeden oluşturulmasına yol göstermektedir.

Teorem 3.1: N hücreli bir hücrel otomata için, eğer i ’ninci bit kapsayıcı bit ise KMT 0, 1, 2, 3 veya KMT 4, 5, 6, 7 veya tüm sekiz KMT’lerinde $(i-1)$ ’inci kuralın özellik 1’i sağlayan KMT’lerine bağlı olarak sağlar (Das ve ark., 2009).

Açılama: n -hücreli bir hücrel otomata için; eğer birinci bit kapsayıcı bit ise erişilebilirlik ağacının 0’ıncı seviyesindeki düğümün 0(sol) ve 1(sağ) kenarlarında yer alan KMT’lerin özellik 1’i karşılaması gerekir. Bu nedenle eğer birinci ve ikinci bitler kapsayıcı bit ise K_1 kuralına ait 0, 1, 2 ve 3 KMT’lerinin özellik 1’i karşılaması gerekir. Eğer ikinci bit kapsayıcı bit değil ise K_1 kuralına ait 0, 1 KMT’lerinin en az birinin

özellik 1'i karşılaması gerekir ve K_1 kuralına ait 2, 3 KMT'lerinin en az birinin özellik 1'i karşılaması gerekir.

i 'ninci bir hücre için yukarıda verilen teoremi açıklayalım. Eğer i 'ninci bir hücre kapsayıcı bit ise, K_i kuralına ait KMT'lerin i 'ninci biti kapsayıcı bit yapacak şekilde seçilmesi gerekir. Seçim yapılırken K_{i-1} kuralına ait KMT'ler dikkate alınır. K_{i-1} kuralına ait KMT 0, 1 veya KMT 4, 5 veya KMT 0, 1, 4, 5 özellik 1'i karşılıyor ise i 'ninci kurala ait KMT 0, 1, 2 ve 3 özellik 1'i karşılar. Aynı mantıkla K_{i-1} kuralına ait KMT 2, 3 veya KMT 6, 7 veya KMT 2, 3, 6, 7 özellik 1'i karşılıyor ise i 'nin kurala ait KMT 4, 5, 6 ve 7 özellik 1'i karşılar. Çok uzunluk döngülü cezbedici oluşumunu engellemek için i 'ninci seviyede yer alan ve özellik 1'i $(i-1)$ 'inci seviyede karşılayan KMT'lerin ardışığı olan KMT'lerin özellik 1'i karşılaması gerekir. $(i-1)$ 'inci seviyede özellik 1'i karşılamayan KMT'lerin ardışığı olan ve aynı KMT kümesine ait olan i 'ninci seviyedeki KMT'ler rasgele seçilir.

Teorem 3.2: N hücreli bir hücresel otomata için: Eğer i 'ninci bit kapsayıcı bit değil ise K_i kuralının oluşturulması $(i+1)$ 'inci bitin kapsayıcı bit olup olmamasına göre oluşturulması aşağıdaki gibidir (Das ve ark., 2009):

- a) $(i+1)$ 'inci bit kapsayıcı bit olduğunda: K_{i-1} kuralının özellik 1'i karşılayan KMT'lerine bağlı olarak K_i kuralının 0, 1, 4, 5 KMT'leri özellik 1'i karşılar ve 2, 3, 6, 7 KMT'leri özellik 1'i karşılamaz. Ya da 2, 3, 6, 7 KMT'leri özellik 1'i karşılar ve 0, 1, 4, 5 KMT'leri özellik 1'i karşılamaz.
- b) $(i+1)$ 'inci bit kapsayıcı bit olmadığında: K_{i-1} kuralının özellik 1'i karşılayan KMT'lerine bağlı olarak K_i kuralının iki denk KMT'si (Örnek: $KMT0=KMT4$ veya $KMT1=KMT5$) özellik 1'i karşılar diğer altı KMT'si özellik 1'i karşılamaz.

Açıklama: N hücreli bir hücresel otomata düşünelim. Eğer birinci bit kapsayıcı bit değilse ve ikinci bit kapsayıcı bit ise birinci bitin 0, 1 KMT'leri veya 2, 3 KMT'leri özellik 1'i karşılamalıdır. İkinci bit kapsayıcı bit olduğundan 0, 1, 2, 3 KMT'leri veya 4, 5, 6, 7 KMT'leri özellik 1'i karşılamalı ki çok uzunluk döngülü cezbedici oluşumunu engellesin. Bu nedenle ikinci bit kapsayıcı bit olduğunda birinci kural özellik 1'i ya 0, 1 KMT'lerinde veya 2, 3 KMT'lerinde karşılamalıdır. Eğer ikinci bit kapsayıcı bit değil ise birinci kuralın 0, 1, 2, 3 KMT'lerinden yalnızca bir KMT özellik 1'i karşılamalıdır.

i 'ninci bir hücre için yukarıda verilen teoremi açıklayalım. Eğer i 'ninci bit kapsayıcı bit değilse ve $(i+1)$ 'inci bit kapsayıcı bit ise i 'ninci kuralın 0, 1, 4, 5 KMT'leri veya 2, 3, 6, 7 KMT'leri özellik 1'i karşılamalıdır. Çok uzunluk döngülü cezbedici oluşumunu engellemek için Kural K_i KMT'lerinin aynı kümede olanları ya hep beraber özellik 1'i karşılamalı yada hiç karşılamamalıdır. Eğer i 'ninci bit kapsayıcı bit değilse ve $(i+1)$ 'inci bit kapsayıcı bit değil ise i 'ninci bitin kuralı K_i ait KMT'lerden yalnızca iki denk KMT'si (Örnek: $KMT_0=KMT_4$ veya $KMT_1=KMT_5 \dots$) özellik 1'i karşılayacak şekilde değer seçimi yapılır. K_{i-1} kuralında özellik 1'i karşılayan KMT'ler olduğunda, çok uzunluk döngülü cezbedici oluşumunu engellemek için K_i kuralı şöyle oluşturulur: $(i-1)$ 'inci bit yer alan ve özellik 1'i karşılayan KMT'ler ile aynı KMT kümesine ait i 'ninci bitteki KMT özellik 1'i karşılamazlar. Diğer KMT kümesinde yer alan i 'ninci bitte yer alan KMT'lerden yalnızca iki denk KMT özellik 1'i karşılar.

Das ve arkadaşları yukarıdaki teoremlerim belirlediği çerçeve içerisinde istenen kural vektörü uzunluğuna ve istenen kapsayıcı bite sahip kural vektörü elde edilmesi sağlayan algoritma geliştirmişlerdir. Aşağıda Şekil 2.20'de algoritmada kural vektörü elde etmek için kullanılan sentez yönteminin algoritması yer almaktadır. Algoritma girdi olarak kural vektörü uzunluğunu ve kapsamlı bit sayısını almakta ve çıktı olarak istenen uzunluğu sahip kural vektörü vermektedir.

Girdi: n (Kural vektörü uzunluğu) ve K (Kapsamlı bit sayısı)
Çıktı: Kural Vektörü
Adım 1: Kapsamlı bit yeteneğine sahip K tane biti rasgele belirle
Adım 2: S_0 ve S_1 iki KMT kümesi olsun. S_0 KMT'leri 0 ve S_1 KMT'leri 1'dir. $S_0=\{0, 1, 4, 5\}$, $S_1=\{2, 3, 6, 7\}$
Adım 3:
a) Eğer birinci ve ikinci bit kapsayıcı bit ise:
 K_1 ve K_2 kurallarına ait tüm KMT'ler özellik 1'i karşılar
b) Eğer birinci bit kapsayıcı bit ve ikinci bit kapsayıcı bit değilse:
 K_1 kuralına ait KMT'lerden S_0 kümesine ve S_1 kümesine ait KMT'lerden en az birer tanesi özellik 1'i karşılar. K_1 kuralının özellik 1'i karşılayan KMT'lerinden türetilen ardışık KMT'ler K_2 kuralında özellik 1'i karşılarlar. Eğer K_1 kuralının diğer KMT'leri özellik 1'i karşılamıyorsa K_2 kuralına ait KMT'lerden K_1 kuralı KMT'leri aynı kümeye ait (S_0 ve S_1) ise özellik 1'i karşılamaz. Eğer K_1 kuralının diğer KMT'leri özellik 1'i karşılamıyorsa K_2 kuralına ait KMT'lerden K_1 kuralı KMT'leri farklı kümeye ait (S_0 ve S_1) ise özellik 1'i karşılar.
c) Eğer ikinci bit kapsayıcı bit ve birinci bit kapsayıcı bit değilse:
 K_1 kuralının S_0 veya S_1 kümelerinden sadece birine ait KMT'leri özellik 1'i karşılar diğer kümedeki KMT'ler özellik 1'i karşılamaz. K_2 kuralının K_1 kuralında özellik 1'i karşılayan KMT'lerinden türetilen ardışık KMT'ler özellik 1'i karşılar, diğer KMT'ler rasgele seçilir.
d) Eğer birinci ve ikinci bit kapsayıcı bit değillerse:
 K_1 kuralında, S_0 kümesinden sadece bir KMT özellik 1'i karşılar ve S_1 kümesindeki KMT'ler özellik 1'i karşılamazlar; veya S_1 kümesinden sadece bir KMT özellik 1'i karşılar ve S_0 kümesindeki KMT'ler özellik 1'i karşılamazlar. K_2 kuralında, K_1 kuralında özellik 1'i karşılayan KMT'den türetilen ardışık KMT'ler özellik 1'i karşılar. K_2 kuralında, K_1 kuralında özellik 1'i karşılamayan KMT'den türetilen ardışık KMT'lerden aynı kümede olanlar özellik 1'i karşılamaz, farklı kümede olanlar özellik 1'i karşılar.
Adım 4: For $i=3$ 'den n 'e
a) Eğer i 'ninci ve $(i+1)$ 'inci bitler kapsayıcı bit ise:
 K_i kuralının 0, 1, 2, 3 KMT'leri veya 4, 5, 6, 7 KMT'leri veya tüm sekiz KMT'si özellik 1'i karşılar.
b) Eğer i 'ninci bit kapsayıcı bit ve $(i+1)$ 'inci bit kapsayıcı bit değilse:
 K_i kuralının 0, 1, 2, 3 KMT'leri veya 4, 5, 6, 7 KMT'leri özellik 1'i karşılar. Diğer KMT'ler rasgele seçilir.
c) Eğer i 'ninci bit kapsayıcı bit değil ve $(i+1)$ 'inci bit kapsayıcı bit ise:
 K_i kuralının S_0 kümesine ait KMT'leri özellik 1'i karşılar ve S_1 kümesine ait KMT'leri özellik 1'i karşılamaz; veya S_1 kümesine ait KMT'leri özellik 1'i karşılar ve S_0 kümesine ait KMT'leri özellik 1'i karşılamaz
d) Eğer i 'ninci ve $(i+1)$ 'inci bitler kapsayıcı bit değiller ise:
 K_i kuralının S_0 kümesine ait KMT'leri özellik 1'i karşılamaz ve S_1 kümesine ait KMT'lerden yalnızca iki denk KMT özellik 1'i karşılar; veya S_1 kümesine ait KMT'leri özellik 1'i karşılamaz ve S_0 kümesine ait KMT'lerden yalnızca iki denk KMT özellik 1'i karşılar
Adım 5: K tane kapsamlı bite sahip kural vektörünü rapor et.

Şekil 2.20. Algoritma 4: Çok uzunluk döngülü cezbedici oluşumuna izin vermeden, istenen sayıda kapsamlı bit üreten kural vektörünü çıktı olarak veren çoklu cezbedici hücresel otomata sentez algoritması (Das ve ark., 2009)

Algoritmanın karmaşıklığı kural vektörünün uzunluğuna ve KMT sayısına bağlıdır. N tane kurala sahip kural vektörü ve 8 KMT değeri için zaman karmaşıklığı şöyledir; $T(n)=8n=O(n)$.

Yukarıdaki algoritmayı kullanarak belirli sayıda kapsayıcı bite sahip ve sadece tek uzunluk döngülü cezbediciler üreten kural vektörü elde edelim. Algoritma girdi olarak kural vektörünün uzunluğunu ve kapsayıcı bit sayısını almaktadır. Kural vektörünün uzunluğu $n=6$ ve kapsayıcı bit sayısı $k=3$ olsun. Kapsayıcı bitlerimiz 1, 4 ve 6'ncı bitler olsun.

Birinci bit kapsayıcı bit olduğu için birinci kural KMT'lerinin kapsayıcı bit şartlarını sağlaması dikkate alınarak seçilmesi gerekir. Kapsayıcı bit KMT'lerini seçerken dikkat edilmesi gereken bir diğer durum; bir sonraki bitin kapsayıcı bit olup olmamasıdır. İkinci bit kapsayıcı bit olmadığı için birinci kuralın ilk dört KMT'sinin özellik 1'i karşılamasına gerek yoktur. Kapsayıcı bit şartlarını sağlaması için birinci kurala ait S_0 ve S_1 KMT kümelerinden en az birer KMT'nin özellik 1'i sağlaması gerekir. Birinci kuralda üç KMT özellik 1'i karşılıyor olsun. Özellik 1'i sağlayan KMT'ler KMT0, KMT1 ve KMT3 olsun; KMT2 özellik 1'i sağlamasın. Sıfır sınırlı hücresele otomata kullandığımız için son dört KMT'nin hangi değeri aldığına bir önemi yoktur. Son dört KMT'nin değerleri 0 seçilsin. Bu şartlarda birinci kural şöyle olacaktır; 8:00001000.

İkinci bit kapsayıcı bit değildir. Bu nedenle ikinci kural KMT'lerinin kapsayıcı bit şartlarını sağlamaması gerekir. Ayrıca ikinci bitten sonra gelen üçüncü bitte kapsayıcı bit değildir. Bu nedenle ikinci kurala ait S_0 ve S_1 KMT kümelerinden birinin özellik 1'i tamamen karşılaması, diğerinde sadece iki denk KMT'nin özellik 1'i karşılaması gerekir. S_0 KMT kümesi özellik 1'i karşılamasın ve S_1 KMT kümesinden iki denk KMT olan KMT3 ve KMT7 özellik 1'i karşılasın, diğer iki denk KMT olan KMT2 ve KMT6 özellik 1'i sağlamasın. Bu şartlarda ikinci kural şöyle olacaktır; 187:10111011

Üçüncü bit kapsayıcı bit değildir. Bu nedenle üçüncü kural KMT'lerinin kapsayıcı bit şartlarını sağlamaması gerekir. Ancak dördüncü bit kapsayıcı bit olduğu için üçüncü kurala ait S_0 ve S_1 KMT kümelerinden birinin özellik 1'i tamamen karşılaması, diğerinin tamamen karşılamaması gerekir. Hangi KMT kümesinin özellik 1'i karşılayacağı bir önceki kural olan ikinci kuralın özellik 1'i sağlayan KMT'lerinin ait olduğu kümeye göre yapılır. İkinci kuralda özellik 1'i sağlayan KMT'ler S_1 kümesine ait olduğu için üçüncü kuralın S_1 kümesine ait KMT'lerinin özellik 1'i karşılaması, S_0

kümesine ait KMT'lerin özellik 1'i karşılamaması gerekir. Bu şartlarda üçüncü kural şöyle olacaktır; 255:11111111.

Dördüncü bit kapsayıcı bit olduğu için dördüncü kural KMT'lerinin kapsayıcı bit şartlarını sağlaması dikkate alınarak seçilmesi gerekir. Dördüncü kuralda özellik 1'i sağlayan KMT'ler üçüncü kuralda özellik 1'i sağlayan KMT'lerin ardışığı olacaktır. Üçüncü kuralda özellik 1'i sağlayan KMT şöyledir; KMT2, KMT3, KMT6 ve KMT7. Bu KMT'lerin ardışıkları şöyledir; $KMT2=\{KMT4, KMT5\}$, $KMT3=\{KMT6, KMT7\}$, $KMT6=\{KMT4, KMT5\}$, $KMT7=\{KMT6, KMT7\}$. KMT2 ve KMT6; KMT3 ve KMT7 denk KMT'ler oldukları için aynı ardışık KMT'lere sahiptir. Bu durumda dördüncü kuralda yer alan KMT4, KMT5, KMT6 ve KMT7'nin özellik 1'i karşılaması gerekir. Geriye kalan KMT'ler rasgele seçilebilir. KMT1, KMT2 ve KMT3 özellik 1'i karşılasın ve KMT0 özellik 1'i karşılamasın. Bu şartlarda dördüncü kural şöyle olacaktır; 205:11001110.

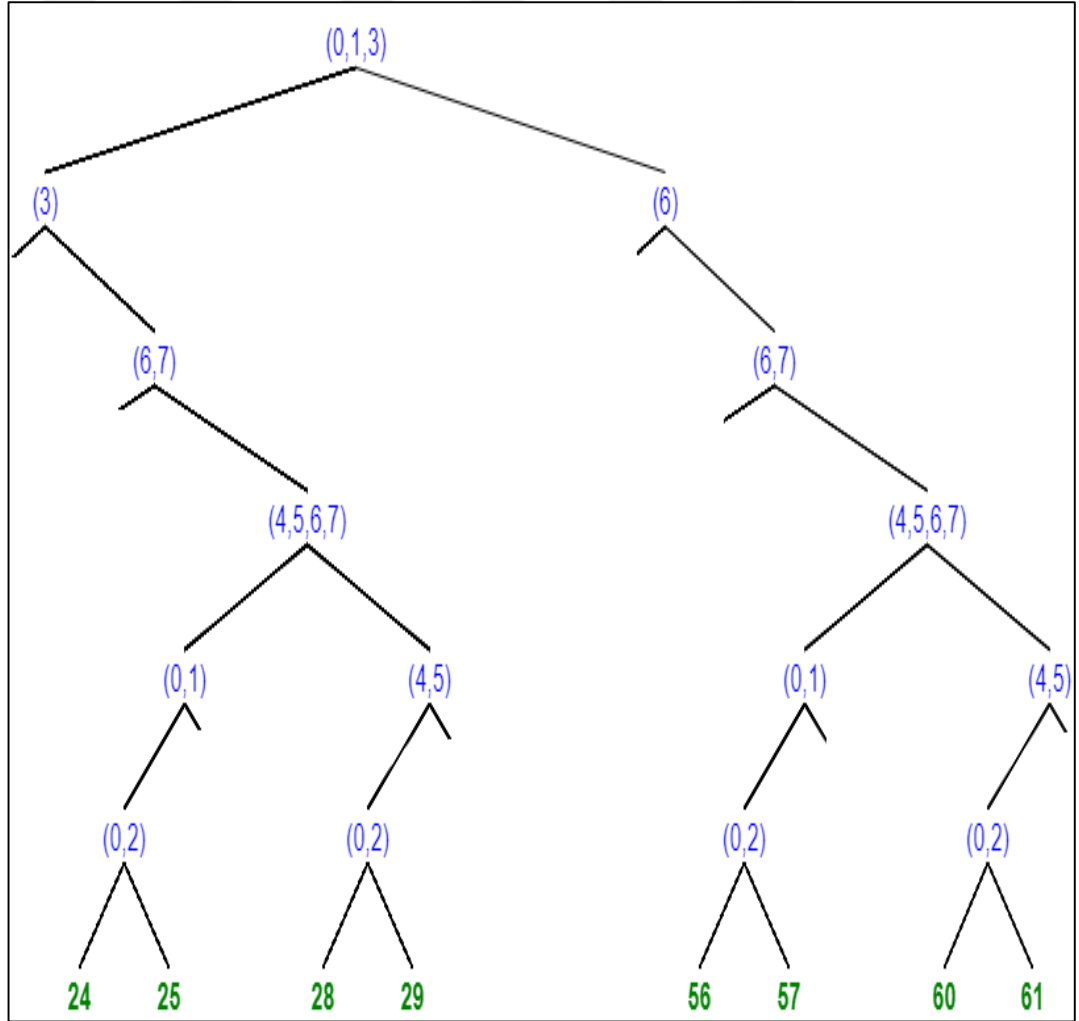
Beşinci bit kapsayıcı bit değildir. Bu nedenle beşinci kural KMT'lerinin kapsayıcı bit şartlarını sağlamaması gerekir. Ancak altıncı bit kapsayıcı bit olduğu için üçüncü kural ait S_0 ve S_1 KMT kümelerinden birinin özellik 1'i tamamen karşılaması, diğerinin tamamen karşılamaması gerekir. S_0 kümesi özellik 1'i karşılasın ve S_1 kümesi özellik 1'i karşılamasın. Bu şartlarda beşinci kural şöyle olacaktır; 0:00000000.

Altıncı bit kapsayıcı bit olduğu için altıncı kural KMT'lerinin kapsayıcı bit şartlarını sağlaması dikkate alınarak seçilmesi gerekir. Altıncı kurala ait KMT0, KMT2, KMT4 ve KMT6 özellik 1'i karşılasın; KMT1, KMT3, KMT5 ve KMT7 özellik 1'i karşılamasın. Bu şartlarda altıncı kural şöyle olacaktır; 102:01100110.

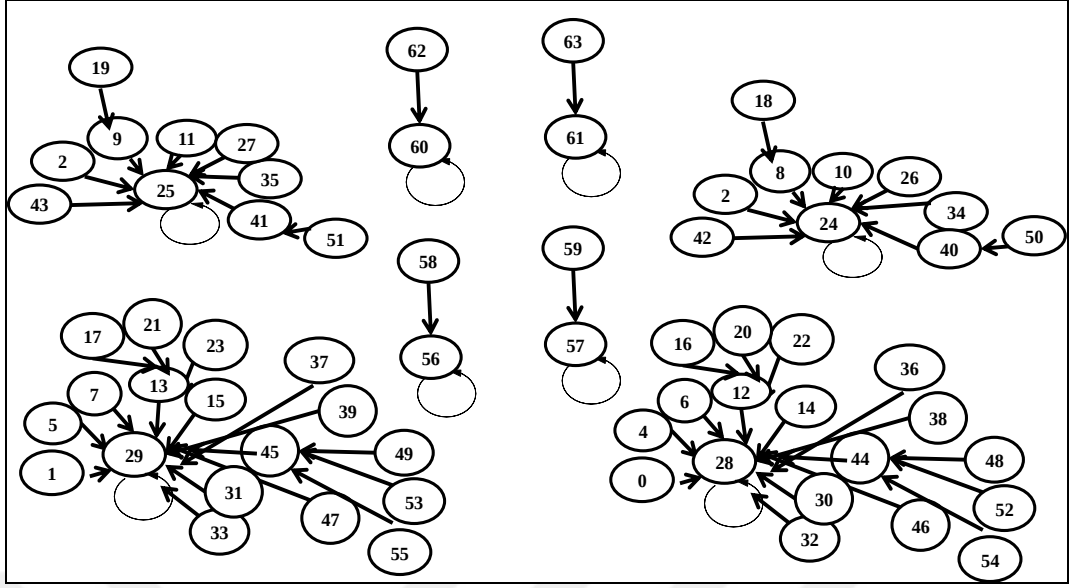
Aşağıdaki Tablo 2.12'de $CA<8, 187, 255, 205, 0, 102>$ kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları yer almaktadır. Şekil 2.21.'deki erişilebilirlik ağacı ve Şekil 2.22'deki durum geçiş diyagramları Tablo 2.12'ye göre oluşturulmuştur.

Tablo 2.12. CA<8, 187, 255, 205, 0, 102> kural vektörüne ait kuralların ikilik değerleri ve KMT karşılıkları

Kural	KMT							
	111	110	101	100	011	010	001	000
	KMT (7)	KMT (6)	KMT (5)	KMT (4)	KMT (3)	KMT (2)	KMT (1)	KMT (0)
8	0	0	0	0	1	0	0	0
187	1	0	1	1	1	0	1	1
255	1	1	1	1	1	1	1	1
205	1	1	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0
102	0	1	1	0	0	1	1	0



Şekil 2.21. CA<8, 187, 255, 205, 0, 102> kural vektörünün ürettiği cezbediciler için erişilebilirlik ağacı



Şekil 2.22. CA<8, 187, 255, 205, 0, 102> kural vektörünün durum geçiş diyagramı

2.4.13. Belirli kapsayıcı bitlere ve tek uzunluk döngülü cezbediciler için erişilebilirlik ağacı

Bir önceki bölümde belirli kapsayıcı bitlere sahip tek uzunluk döngülü cezbedici hücresel otomatları üreten kural vektörünün elde edilmesi açıklanmıştır. Bu bölümde belirli kapsayıcı bitleri ve tek uzunluk döngülü cezbedicileri temsil eden erişilebilirlik ağacının oluşturulması açıklanmıştır.

Erişilebilirlik ağacı, belirli kapsayıcı bitlere sahip tek uzunluk döngülü cezbedicileri temsil edecek şekilde oluşturulabilir (Mukherjee, 2009). N hücreli hücresel otomata için $n+1$ seviye sahip erişilebilirlik ağacı oluşturulması gerekir. Ağacın kök düğümü 0'ıncı seviyede yaprak düğümü $(n+1)$ 'inci seviyede yer alır. Erişilebilirlik ağacı hakkında detaylı bilgiyi “Erişilebilirlik Ağacının Oluşturulması” ve “Cezbediciler İçin Erişilebilirlik Ağacı” başlıklı bölümlerde bulunabilir.

Erişilebilirlik ağacındaki düğümler kapsayıcı bit olup olmamalarına göre KMT değerlerine sahip olurlar. Farklı durumlar için ağaç düğümlerinin KMT seçimini inceleyelim:

Eğer i 'ninci bir kapsayıcı bit ise; $(i-1)$ 'inci seviyedeki düğümün 0-kenarında(sol kenar) yer alan KMT0, KMT1, KMT4, KMT5 KMT'lerinden özellik 1'i taşıyacak KMT'ler ya KMT0 ve KMT1; veya KMT4 ve KMT5; veya hepsi olabilir. Aynı yaklaşımla: $(i-$

1)'inci seviyedeki düğümün 1-kenarında(sağ kenar) yer alan KMT2, KMT3, KMT6, KMT7 KMT'lerinden özellik 1'i taşıyacak KMT'ler ya KMT2 ve KMT3; veya KMT6 ve KMT7; veya hepsi olabilir. KMT'lerin bu yaklaşımla seçilmesi, i'ninci bitin kapsayıcı bit olmasını sağlarken, (i-2)'inci seviyede yer alan KMT'lerin özellik 1'i karşılamasına bağlı olarak (i-1)'inci seviyede farklı KMT kümelerindeki KMT'lerin özellik 1'i karşılamasını sağlayarak çok uzunluk döngülü cezbediciler oluşumunu engellemektedir.

Eğer i'ninci bit kapsayıcı bit değil ve (i+1)'inci bir kapsayıcı bir ise: (i-2)'inci seviyede özellik 1'i karşılayan KMT'ler 0-kenarında veya 1-kenarında yer almalıdır. Bu yaklaşım farklı iki kümeye ait KMT gruplarından bir grubun özellik 1'i karşılamasını, diğer grubun özellik 1'i karşılamamasını sağlayarak çok uzunluk döngülü cezbedici oluşumunu engellemektedir.

Eğer i'ninci ve (i+1)'inci bitler kapsayıcı bitler değil ise: Ağacın (i-1)'inci seviyedeki düğümünün 0-kenarından iki denk KMT veya 1-kenarından iki denk KMT özellik 1'i karşılamalı diğer KMT'ler karşılamalıdır. Özellik 1'i karşılayacak KMT'lerin seçimi (i-2)'inci seviyede özellik 1'i karşılayan KMT'lere bağlı olarak gerçekleştirilir. Eğer 0-kenarından iki denk KMT özellik 1'i karşılayacaksa bunlar KMT0 ve KMT4; veya KMT1 ve KMT5 olabilir. Eğer 1-kenarından iki denk KMT özellik 1'i karşılayacaksa bunlar KMT2 ve KMT6; veya KMT3 ve KMT7 olabilir. Bu yaklaşım, iki farklı KMT kümesinden birinin özellik 1'i tamamen karşılaması, diğerinde sadece iki denk KMT'nin özellik 1'i karşılamasını sağlayarak çok uzunluk döngülü cezbedici oluşumunu engellemektedir.

2.5. GMACA'nın Sınıflandırıcı Olarak Kullanımı

Çoklu cezbedici hücrel otomatların cezbedici durum davranışları rasgele sayı üreteçleri, elektronik devre testleri, örüntü tanıma, veri madenciliği, veri sıkıştırma vb. birçok uygulamada kullanılmıştır. Birbirinden farklı uygulamalarda kullanılabilmesi cezbedici durumların esnek ve ayrık yapısından kaynaklanmaktadır.

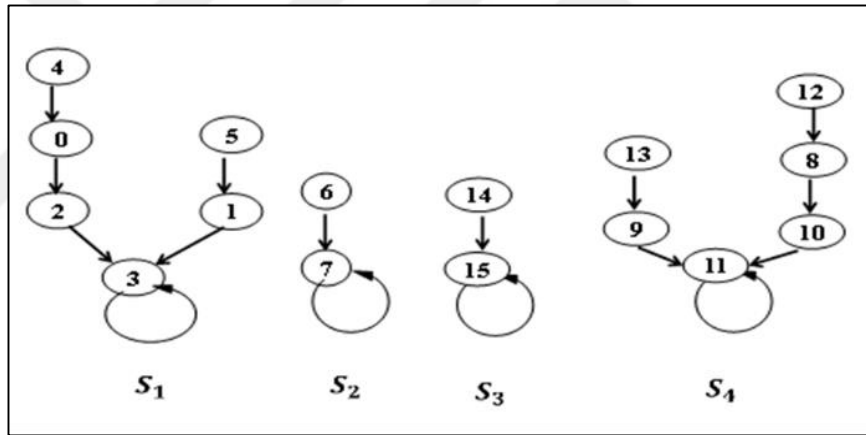
Mühendislik uygulamalarında önemli olan; probleme uygun çözüm modelinin geliştirilmesidir. Tek uzunluk döngülü cezbedicilerin örüntü sınıflandırma problemlerine çözüm sunma potansiyeline sahiptir. Tek uzunluk döngülü

cezbedicilerin, başka örüntüleri etrafında toplama yeteneğine sahip çekim merkezi formunda davranışlar sergilemesi örüntü sınıflandırma potansiyeli olarak değerlendirilebilir.

Bu bölümde tek uzunluk döngülü cezbediciler, gerçek veri setleri üzerinde örüntü sınıflandırıcı olarak kullanılacaktır. Bunun için cezbedicilerin sınıflandırıcı olarak değerlendirilmesi ve veri setlerinin hücresel otomata hesaplama yeteneğine uygun hale getirilmesi gerekir.

2.6. Cezbedicilerin Sınıflandırıcı Olarak Kullanımı

Kural vektörünün ürettiği n tane cezbedici, doğal bir sınıflandırıcı olarak düşünülebilir. Aşağıdaki şekilde kural vektörünün ürettiği cezbedicilerin farklı sınıfları temsil etmesi gösterilmektedir.



Şekil 2.23. Cezbedicilerin sınıflandırıcı olarak gösterimi (Das ve ark., 2009)

Şekil 2.23'te dört farklı cezbedici, dört farklı sınıf temsilcisi olarak düşünülmüştür. Burada kritik soru şudur; Cezbedicilerin hangi sınıfı temsil edeceği nasıl belirlenecektir? Cezbedicilerin hangi sınıfı temsil ettiği modelin eğitim aşamasında gerçekleştirilecektir.

Eğitim aşamasında, eğitim için ayrılan veri setindeki her bir örnek, kural vektörüne yüklenerek ait olduğu cezbedici bulunur. Eğitim veri setindeki tüm örneklerin ait oldukları cezbediciler bu şekilde bulunur. Tüm örneklerin cezbedicileri bulunduktan sonra, cezbedicilerin temsil edeceği sınıfa karar verilir. Cezbediciye en çok atanan örnek sınıf cezbedicinin sınıfı olarak kabul edilir. Eğer cezbediciye sınıflardan eşit

sayıda atama yapılmışsa cezbedicinin sınıfı rasgele atanır. Bu eğitim modeli iki sınıfa ait veri setleri için uygundur. Hücresel otomata kullanılarak yapılan sınıflandırmalar, iki sınıf içeren veri setleri kullanılarak gerçekleştirilmiştir.

Eğitim aşamasında cezbedicilerin ait oldukları sınıfların bulunması hata fonksiyonuna veya tekrar sayısına bağlı olarak tekrar edilir. Eğitim aşamasında elde edilen en verimli kural vektörü test veri seti üzerinde uygulanır.

2.7. Veri Setinin Hücresel Otomata Hesaplama Modeline Uygun Hale Getirilmesi

Hücresel otomatlar ikili örüntüler üzerinde işlem yapan hesaplama modeline sahiptir. Veri seti kullanılmadan önce veri setindeki örnekler ön işlemlerle hücresel otomata modeline uygun ikilik örüntülere dönüştürülür.

Makine öğrenmesi için kullanılan veri setlerinin büyük çoğunluğu ön işlemlere ihtiyaç duyarlar. Makine öğrenmesi modellerinin hepsi sayısal veriler üzerinde işlem yapabilirler. Eğer veri setinde metin değerler varsa bunların gerekli ön işlemler kullanılarak sayısal veriye dönüştürülmesi gerekir.

Ön işlemlerden sonra veri setinde kategorik ve sürekli sayısal değerler kalır. Kategorik verilerin ikili örüntülere dönüştürülmesi için OneHot kodlama yöntemi kullanılır. Sürekli sayısal veriler ikilik tabana dönüştürülmeden önce, normalleştirilmesi gerekir. Normalleştirmenin sınıflandırma performansı üzerinde önemli etkisi vardır. Normalleştirme işleminden sonra veriler istenen aralığa çekilir ve ikilik tabana dönüştürülür.

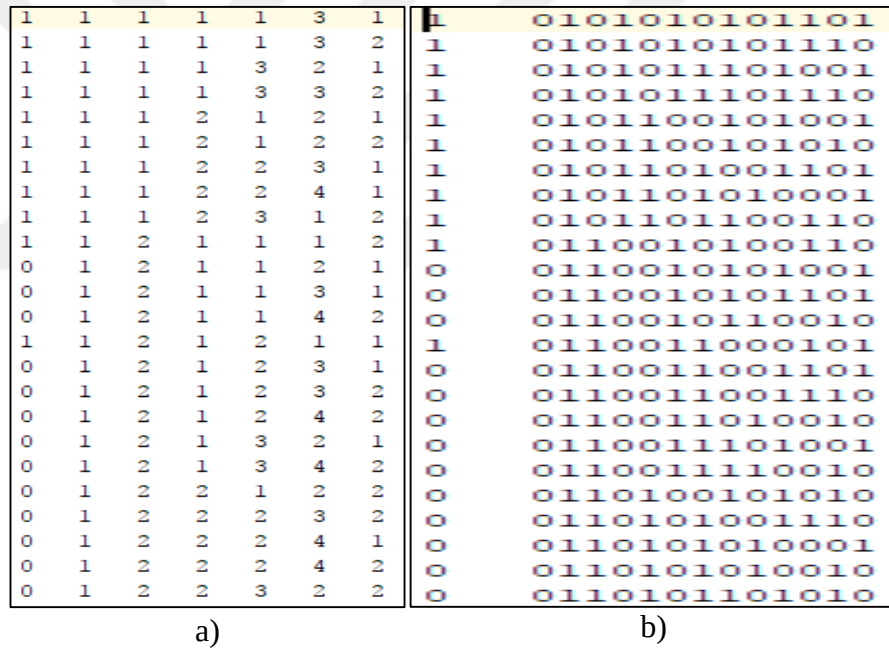
Bu çalışmada iki farklı veri seti üzerinde hücresel otomata tabanlı sınıflandırma yöntemi kullanılmıştır. Kullanılan veri setleri UCI makine öğrenmesi veri tabanında yer alan MONKs veri setleri ve diabet hastalarının bilgilerini içeren Pima Indian veri setidir. Veri setlerine ait bilgiler aşağıdaki Tablo 2.13'te yer almaktadır.

Monks veri seti üç ayrı veri setinden oluşmaktadır ve tüm veri setlerindeki tüm özellikler kategorik değerlere sahiptir. Veri setinin hücresel otomata modeline uygun hale getirilmeden önce, kategorik değerler OneHot kodlama yöntemi ile ikilik tabana dönüştürülmüştür.

Tablo 2.13. MONKS ve Pima Indian Diabet veri setlerine ait bilgiler

Veri seti	Özellik Sayı	Özellik Yapısı	Ön İşlem	Örnek Sayısı
Monk1	6	Kategorik	İkilik tabana dönüşüm	556
Monk2	6	Kategorik	İkilik tabana dönüşüm	601
Monk3	6	Kategorik	İkilik tabana dönüşüm	554
Pima Indian Diabets	8	Numerik	Normalleştirme , Kategorik ve ikilik tabana dönüşüm	768

Şekil 2.24'te Monks veri setine ait orijinal görüntü ve hücrel otomata için uygun ikilik tabandaki görüntü yer almaktadır.



Şekil 2.24. Monks veri setindeki kategorik değerler ikilik tabana dönüştürülmüştür a)Orijinal veri seti b)Dönüştürülmüş veri seti

Pima Indian veri seti özellikler kategorik ve numerik değerlerden oluşmaktadır. Ayrıca veri setinde eksik veriler vardır. Eksik veriler numerik değerlere ait olduğu için özelliğin ortalama değeri ile doldurulmuştur. Veri setinin hücrel otomata modeline uygun hale getirilmeden önce, kategorik değerler OneHot kodlama yöntemi ile ikilik tabana dönüştürülmüştür. Daha sonra numerik değerler normalleştirme işlemi yapıldıktan sonra 0 ile 100 aralığına çekilmiş ve ikilik tabana dönüştürülmüştür.

Aşağıdaki Şekil 2.25'te Pima Indian veri setine ait orijinal görüntü ve hücrel otomata için uygun ikilik tabandaki görüntü yer almaktadır.

6	148	72	35	0	33.6	0.627	50	1	1	00110100101001001000010001100100110110100001001111001100100
1	85	66	29	0	26.6	0.351000000000000003	31	0	0	00001010101011000010001110100100110110011010001000110011110
8	183	64	0	0	23.3	0.672	32	1	1	01000101101110000000011101001001101100101110100001101000000
1	89	66	23	94	28.1	0.166999999999999998	21	0	0	00001010110011000010001011100010111000111000001000000101010
0	137	40	35	168	43.1	2.288000000000000003	33	1	1	00000100010010101000010001100101010000101011110010001000010
5	116	74	0	0	25.6	0.201	30	0	0	00101011010010010100011101001001101100110010001010000111100
3	78	50	32	88	31.0	0.248	26	1	1	00011010011100110010010000000101100000111110001100000110100
10	115	0	0	0	35.3	0.134	29	0	0	010100111001110010000011101001001101101000110000110100111010
2	197	70	45	543	30.5	0.158	53	1	1	00010110001011000110010110100001111000111000001110110101010
8	125	96	0	0	0.0	0.231999999999999998	54	1	0	010000111101100000001110100100110110100000001011101101100
4	110	92	0	0	37.6	0.191	30	0	0	0010001101101011100001110100100110110100010100010011001100
10	168	74	0	0	38.0	0.537	34	1	1	01010101000010010100011101001001101101001100011010101000100
10	139	80	0	0	27.1	1.440999999999999998	57	0	0	01010100010111010000001110100100110110011011001000001110010
1	189	60	23	846	30.1	0.397999999999999996	59	1	1	0000101111010111000010111101000111000111000010011011011010
5	166	72	19	175	25.8	0.587000000000000001	51	1	1	00101101001101001000001001100101011100110010011101001100110
7	100	0	0	0	30.0	0.484	32	1	1	00111011001001001000001110100100110110011100011000001000000
0	118	84	47	230	45.8	0.551	31	1	1	00000011101010101000101110011100110011001010101100111110
7	107	74	0	0	29.6	0.254	31	1	0	001110110101110010100011101001001101100111010001100100111110
1	103	30	38	83	43.3	0.183	33	0	0	000010110011100111001001000010100110101010010010010000010
1	115	70	30	96	34.6	0.529	32	1	1	000010111001110001100011110000110000001000100011010001000000
3	126	88	41	235	39.3	0.704000000000000001	27	0	0	0001101111101011000010100100111010101001110100011000110110
8	99	84	0	0	35.4	0.387999999999999996	50	0	0	010000110001110101000011101001001101101000110010011001100100
7	196	90	0	0	39.8	0.451	41	1	1	0011111000100101101000111010010011011000110010110101010010
9	119	80	35	0	29.0	0.263	29	1	1	010010111011011010000010001100100110110011101000110100011010
11	143	94	33	146	36.6	0.254	51	1	1	0101110001111011100100001001001001001000001100101100110
10	125	70	26	115	31.1	0.205	41	1	1	0101001111011000110001101000011100110011110001010001010010
7	147	76	0	0	39.4	0.257	43	1	1	0011110010011100110000111010010011011001110001100101010110
1	97	66	15	140	23.2	0.487000000000000004	22	0	0	000010110000110000100001111001000110000101110011000000101100
13	145	82	19	110	22.2	0.245	57	0	0	010110010000110100100010011000110111000101100001100001110010
5	117	92	0	0	34.1	0.337	38	0	0	00101011101010111000011101001001101101000100010000101001100
5	109	75	26	0	36.0	0.546	60	0	0	00101011011011001011001101000100110110100100001101100111000

a)

b)

Şekil 2.25. Pima Indian veri setindeki kategorik ve normalleştirme işleminden sonra numerik değerler ikilik tabana dönüştürülmüştür a) Orijinal veri seti b) Dönüştürülmüş veri seti

2.8. İkili Sınıflandırma İçin Deneysel Çalışma

Bu bölümde hücrel otomata tabanlı sınıflandırıcılar kullanılarak elde edilen çapraz doğrulama başarı oranlarına ve sınıflandırma raporlarına yer verilmiştir.

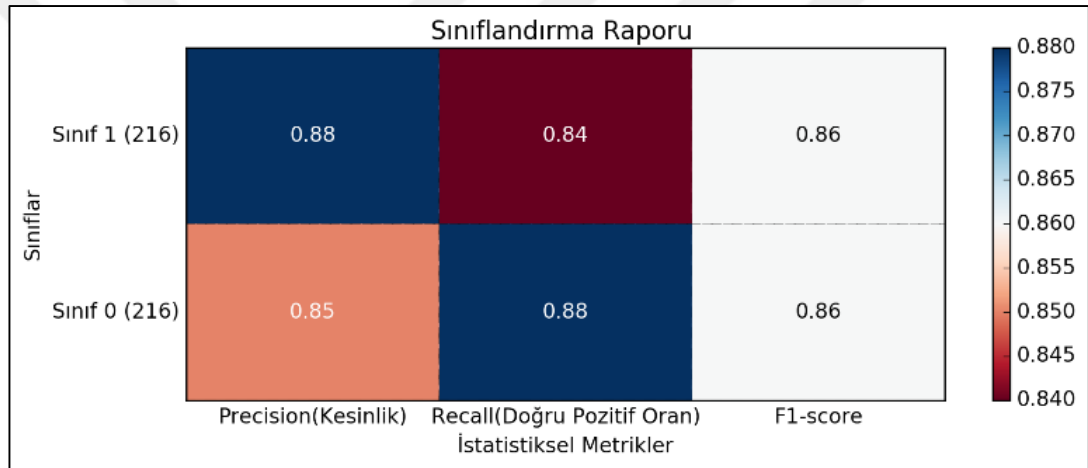
Çapraz doğrulama veri setinin birden fazla kez eğitim ve test veri setlerine ayrıştırılmasıyla gerçekleştirilir. Birçok çapraz doğrulama yaklaşımları olmasına karşın en bilinen çapraz doğrulama yaklaşımı K-fold(K-katlı) yöntemidir. Bu yöntemle göre veri seti K eşit parçaya bölünür ve bu parçalardan bir tanesi eğitim aşamasında kullanılır. Bu işlem K defa yapılarak her bir parçanın eğitim ve test aşamasında kullanılması sağlanır. K tane sınıflandırma sonucunun ortalaması modelin başarı oranı olarak kabul edilir.

Başarı oranı, sınıflandırma modelinin performansını değerlendirmek için yeterli değildir. Başarı oranı, sınıflandırma modelinin hatayı nasıl yaptığını söylemez. Bu nedenle modelin değerlendirmesi için gerekli detayları içermez.

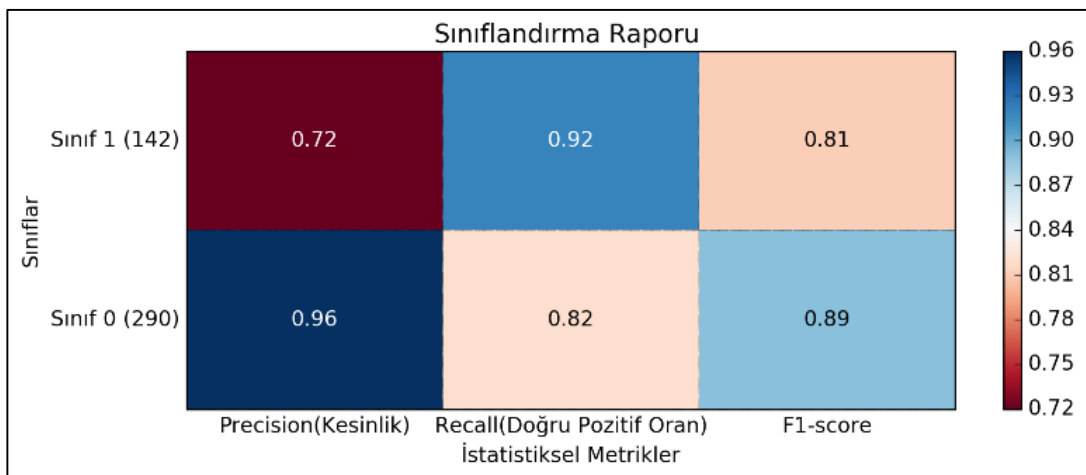
Sınıflandırma modelinin performansının daha iyi değerlendirilmesi için karışıklık matrisi kullanılır. Karışıklık matrisi, sınıflandırma modelinin performansını daha iyi değerlendirmeyi sağlayan bir tablolama tekniğidir. Karışıklık matrisi kullanılarak sınıflandırma raporunda yer alan kesinlik, yanlış tanıma ve f1 skorları hesaplanır.

Çapraz doğrulama ve karışıklık matrisi hakkında detaylı bilgiler 6.1. alt bölümünde yer almaktadır. Detaylı bilgi için bu alt bölüm incelenebilir.

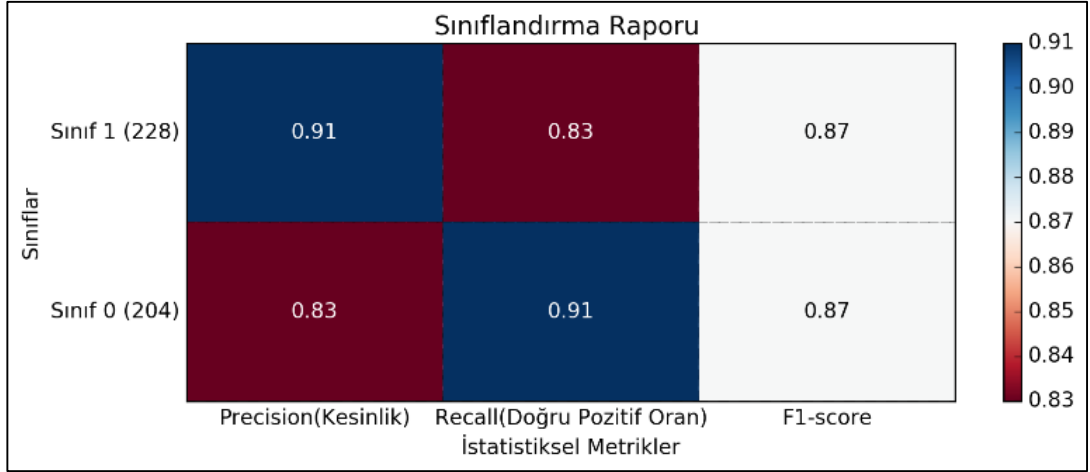
Aşağıdaki Şekil 2.26, Şekil 2.27 ve Şekil 2.28’de sırasıyla Monk1, Monk2 ve Monk3 veri setlerine ait sınıflandırma raporları yer almaktadır. Şekil 2.29’de Pima Indian veri setine ait sınıflandırma raporu yer almaktadır.



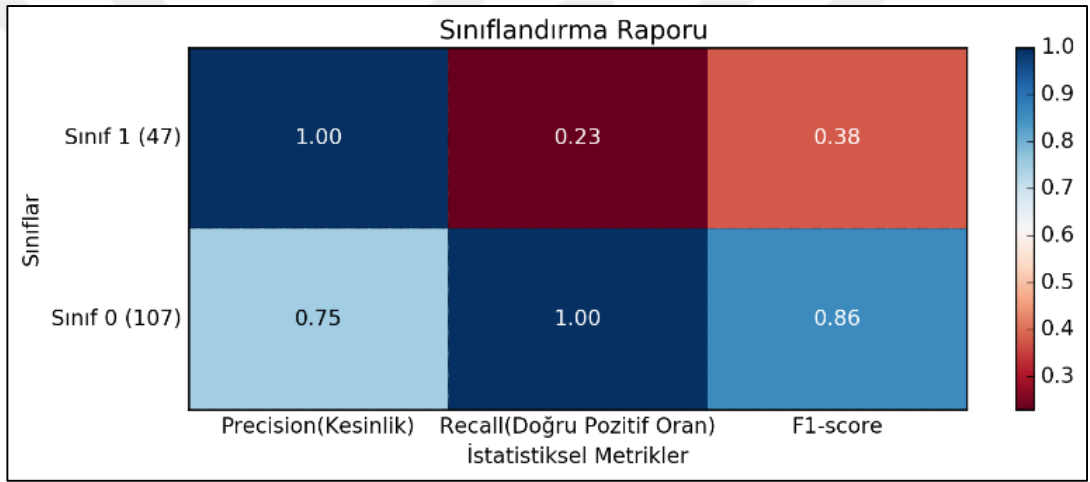
Şekil 2.26. Monk1 veri setine ait sınıflandırma raporu



Şekil 2.27. Monk2 veri setine ait sınıflandırma raporu



Şekil 2.28. Monk3 veri setine ait sınıflandırma raporu



Şekil 2.29. Pima Indian veri setine ait sınıflandırma raporu

Hücrel otomata kullanılarak elde edilen başarı oranlarının diğer çalışmalar ile karşılaştırılması iki farklı tabloda verilmiştir. Tablo 2.14'te bilinen makine öğrenmesi yöntemleriyle aynı veri setleri üzerinde bizim elde ettiğimiz sonuçlar yer almaktadır. Tablo 2.15'te aynı veri seti üzerinde elde edilen hücrel otomata tabanlı ve literatürdeki diğer makine öğrenmesi sonuçlarına yer verilmiştir.

Tablo 2.14. Bilinen makine öğrenmesi modelleri kullanarak elde edilen başarı oranları

Veri Seti	Yöntem	Performans
Monks1	Lojistik Regresyon	66,67
	Destek Vektör Makineleri: kernel=rbf	81,71
	Destek Vektör Makineleri kernel=lineer	66,67
	Lineer Ayırıştırma Analizi	66,20
	Karar Ağaçları	93,52
	En Yakın Komşu Algoritması	81,48
	Rasgele Orman	91,20
	Gauss Naive Bayes	67,59
	Perceptron	50,69
	CA İle Elde Edilen Sonuç	86,11
Monks 2	Lojistik Regresyon	65,51
	Destek Vektör Makineleri: kernel=rbf	67,59
	Destek Vektör Makineleri kernel=lineer	67,13
	Lineer Ayırıştırma Analizi	62,96
	Karar Ağaçlar	82,87
	En Yakın Komşu Algoritması	68,75
	Rasgele Orman	78,94
	Gauss Naive Bayes	64,35
	Perceptron	57,41
	CA İle Elde Edilen Sonuç	84,86
Monks 3	Logistik Regresyon	80,09
	Destek Vektör Makineleri: kernel=rbf	96,3
	Destek Vektör Makineleri kernel=lineer	81,25
	Lineer Ayırıştırma Analizi	81,48
	Karar Ağaçlar	90,05
	En Yakın Komşu Algoritması	89,35
	Rasgele Orman	93,52
	Gauss Naive Bayes	89,58
	Perceptron	75,23
	CA İle Elde Edilen Sonuç	87,03
Pima Indian	Logistik Regresyon	75,08
	Destek Vektör Makineleri: kernel=rbf	64,33
	Destek Vektör Makineleri kernel=lineer	76,21
	Lineer Ayırıştırma Analizi	76,05
	Karar Ağaçlar	69,72
	En Yakın Komşu Algoritması	69,68
	Rasgele Orman	74,41
	Gauss Naive Bayes	74,74
	Perceptron	63,06
	CA İle Elde Edilen Sonuç	76,62

Tablo 2.15. Monks ve Pima Indian elde edilen sonuçları hücresele otomata tabanlı ve diğer makine öğrenmesi yöntemleri ile karşılaştırılması

Veri Seti	Yöntem	Performans
Monks1	CA Tabanlı (Mukherjee, 2009)	61,11
	CA Tabanlı (Naskar, 2009)	72,87
	CA Tabanlı (Das ve ark., 2009)	91,93
	KNN (Duch ve Grudziński, 2001)	89,5
	Weighted KNN (Duch ve Grudziński, 2001)	99,7
	CA İle Elde Edilen Sonuç	86,11
Monks 2	CA Tabanlı (Mukherjee, 2009)	67,13
	CA Tabanlı (Naskar, 2009)	67,13
	CA Tabanlı (Das ve ark., 2009)	75,73
	CA İle Elde Edilen Sonuç	84,86
Monks 3	CA Tabanlı (Mukherjee, 2009)	80,65
	CA Tabanlı (Naskar, 2009)	84,33
	CA Tabanlı (Das ve ark., 2009)	95,05
	KNN (Duch ve Grudziński, 2001)	98,6
	Weighted KNN (Duch ve Grudziński, 2001)	98,6
	CA İle Elde Edilen Sonuç	87,03
Pima Indian	CA Tabanlı (Mukherjee, 2009)	73,79
	CA Tabanlı (Naskar, 2009)	67,59
	CA Tabanlı (Das ve ark., 2009)	81,54
	DTDN (Bozkurt ve ark., 2014)	76,00
	CA İle Elde Edilen Sonuç	76,62

Tablo 2.15’te daha önceki CA tabanlı makine öğrenmesi modelleriyle, bu tez çalışması kapsamında elde edilen sonuçlar arasında farklılık görülmektedir. Bu farklılığın olası iki nedeni olabilir. Birincisi: veri setlerinin hücresele otomata sınıflandırma modeline uygun hale getirmek için yapılan ön işlemlerin farklı olmasıdır. Mukherjee yaptığı çalışmada Monks ve Pima Indian veri setleri için elde edilen özellik vektörlerinin uzunluğunun sırasıyla 11 ve 16 olduğunu belirtmektedir (Mukherjee, 2009). Hücresele otomata kullanılarak elde edilen sonuçlarda, ön işlemler sonucu kullanılan özellik vektörlerinin uzunluğu Monks ve Pima Indian veri setleri için sırasıyla 13 ve 61’dir.

CA tabanlı yöntemlerin farklı çıkmasının ikinci olası sebebi: CA makine öğrenmesi modelinin farklı eğitilmelerinden kaynaklanıyor olabilir. Mukherjee (Mukherjee, 2009) ve Naskar (Naskar, 2009) kullanılan cezbedici sayısını kapsayıcı bitlerle sınırlandırmışlardır. Das ve diğ., yaptıkları çalışmada cezbedici sayısı özellik vektörünün %10’uyla sınırlandırılmıştır (Das ve ark., 2009). Bu tez çalışmasındaki hücresele otomata modelinin eğitimi önceki hücresele otomata çalışmalarından farklılıkları vardır. Öncelikle cezbedici sayısı kapsayıcı bitlerle sınırlandırılmamıştır.

İkincisi; Pima Indian veri setini hücresele otomata için uygun hale getirilmesinden sonra kullanılan özellik vektörü çok uzun olduğu için her bir özellik sütununun eğitimi ayrı yapılmıştır. Üçüncüsü eğitim aşamasında başarı oranını ve zaman verimliliğini artırma için Tablo 2.5'te verilen kural gruplarından 6'ncı, 7'inci ve 8'inci gruplara ağırlık verilmiştir.

2.9. Sonuç

Bu bölümde erişilebilirlik ağacı kullanılarak hücresele otomata kural vektörünün sadece tek uzunluk döngülü cezbedici üretecek şekilde nitelenmesi incelenmiştir. Tek uzunluk döngülü cezbediciler ikilik sınıflandırmada kullanılmıştır.

Elde edilen sonuçlar çoklu cezbedici otomatların örüntü sınıflandırma ve tanımda kullanılabileceğini göstermektedir. Hücresele otomatların ayrık yapısı bilgisayar uygulamalarında kullanılmasını avantajlı hale getirmektedir. Ayrıca hücresele otomatların hücresele düzlem üzerinde işleyen çalışma mantığı eş zamanlı hesaplamaya imkan sağlamaktadır.

Hücresele otomatların ikili örüntüler üzerinde kullanılabildiği için kullanılacak veri setinin ikilik örüntülere dönüştürülmesi gerekir. Çoğu makine öğrenmesi veri seti ikilik tabanda olmadığından hücresele otomata sınıflandırıcılarına uygun hale getirilmesi için ön işleme tabi tutulması gerekir. Sayısal değerlerin ikilik tabana dönüşümünde veri kayıpları yaşanabilmektedir. Sadece ikili örüntüler üzerinde çalışması hücresele otomata sınıflandırıcılarının eksik yanı olarak değerlendirilebilir.

2.10. GMACA'nın Çoklu Sınıflar İçin Kullanımı

Önceki alt bölümde GMACA sınıflandırıcıları ikili sınıflandırmada kullanılmıştır. Kullanılan veri setlerinde sadece iki sınıf yer almaktaydı ve sınıflandırma işlemi bu iki sınıf arasında yapılmaktaydı. Bu alt bölümde, GMACA sınıflandırıcılarının çoklu sınıflar üzerinde uygulanma potansiyeli araştırılmıştır.

Çoklu sınıflandırmayla, ikiden fazla sınıf içeren veri setlerinin sınıflandırılması yapılabilirken; ikili sınıflandırma ile sadece iki sınıf içeren veri setlerinin sınıflandırılması yapılabilir. GMACA sınıflandırıcıları gibi, sınıflandırma için kullanılan Destek Vektör Makineleri, Lojistik Regresyon, Perceptron vb. birçok

makine öğrenmesi modeli ikili sınıflandırıcılardır. Bu modeller iki sınıf içeren veri setlerinin sınıflandırılmasında kullanılabilirler.

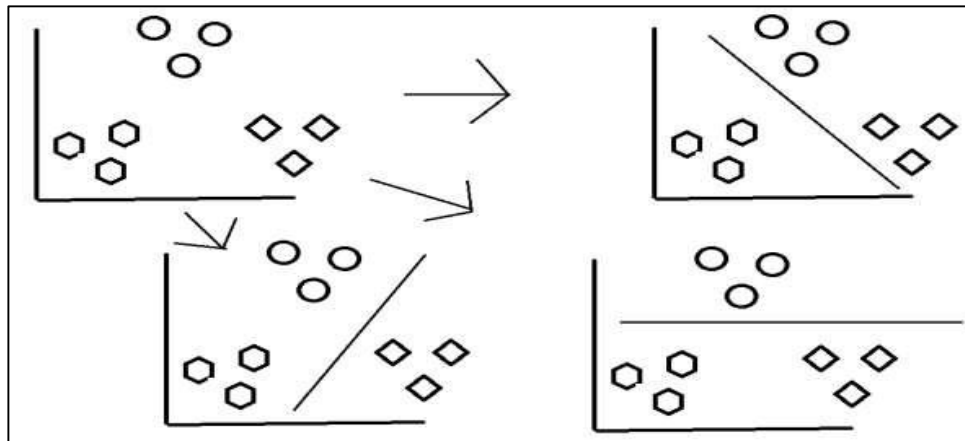
İkili sınıflandırıcıların, ikiden fazla sınıf içeren veri setleri üzerinde kullanılabilmesi için stratejiler geliştirilmiştir. En yaygın kullanılan çoklu sınıflandırma stratejileri; Bire Karşı Hepsi(One-vs-All) ve Bire Karşı Bir(One-vs-One) yaklaşımlarıdır. Bire Karşı Bir yaklaşımı Hepsine Karşı Hepsi(All-vs-All) olarak da adlandırılmaktadır.

İkili sınıflandırıcılar eğitim aşamasında, iki sınıfı birbirinden en iyi ayıran ağırlık vektörünü(GMACA için bu kural vektörüdür) elde ederler. Elde edilen ağırlık vektörü, test veri setindeki sınıfı bilinmeyen özellik vektörlerinin ait oldukları sınıfın tahmin edilmesinde kullanılırlar.

Çoklu sınıflandırma stratejilerinde birden fazla ağırlık vektörü elde edilir. Elde edilen ağırlık vektörleri test aşamasında birden fazla sınıf içeren test verilerinin sınıflandırılmasında kullanılır.

2.10.1. Bire karşı hepsi (one-vs-all)

Bire Karşı Hepsi(One-vs-All) yaklaşımı, N tane sınıf için N farklı ağırlık vektörü elde edilmesi fikrine dayanır. N tane ağırlık vektörü, her bir sınıf için elde edilir. Bunu için veri seti her bir sınıf için pozitif ve negatif olmak üzere ikiye ayrılır. Pozitif sınıf, ağırlık vektörü bulunacak sınıftır. Negatif sınıf, pozitif sınıf dışındaki diğer tüm sınıflardır. Aşağıdaki Şekil 2.30'da üç farklı sınıfın Bire Karşı Hepsi yaklaşımı ile ayrıştırılması gösterilmiştir.



Şekil 2.30. Bire Karşı Hepsi yaklaşımında her bir sınıfı diğer sınıflardan ayırtıran N tane ağırlık vektörü elde edilir.

Şekil 2.30'da yuvarlak, kare ve beşgen olmak üzere üç farklı sınıfın Bire Karşı Hepsini yöntemine göre ayrıştırılması alt resimlerle gösterilmiştir. Şekildeki en üstte soldaki resim, veri setindeki özelliklerin koordinat sistemine iz düşümünü göstermektedir. Sağ üstteki resim, beşgen sınıfını diğer sınıflardan ayıran ağırlık vektörünü göstermektedir. Sol alttaki resim kara sınıfı diğer sınıflardan ayıran ağırlık vektörünü göstermektedir. Sağ alttaki resim, yuvarlak sınıfını diğer sınıflardan ayıran ağırlık vektörünü göstermektedir. Şekilde, üç tane sınıf olduğu için üç farklı ağırlık vektörü elde edilmiştir.

Yukarıda bahsedildiği gibi sınıfları birinden ayıran ağırlık vektörleri eğitim aşamasında elde edilir. Test aşamasında, elde edilen ağırlık vektörleri kullanılarak sınıfı bilinmeyen bir özellik vektörünün ait olduğu sınıf tahmin edilmeye çalışılır.

Sınıfı bilinmeyen bir özellik vektörünün sınıf tahmini, tüm ağırlık vektörlerinin kullanılması sonucu en yüksek değeri alan sınıf olarak kabul edilir. Şekil 2.3'deki sınıflar üzerinden bir örnekle açıklamak faydalı olacaktır. Örneğin tahmin edilecek özellik vektörünün gerçek sınıfı yuvarlaklar olsun. Şekildeki alt resimlere göre sınıfları ayırtmak için elde edilen ağırlık vektörleri, özellik vektörünün ait olduğu sınıf bulmak için kullanılır. Sağ üstteki resimde gösterilen ağırlık vektörü kullanıldığında, sınıf tahmin edilecek özellik vektörü yuvarlaklar sınıfına ait olduğu için diğer sınıflar tarafında kalacaktır. Tahmin edilecek özellik vektörü, diğer sınıflar tarafında kaldığı için bir sonraki ağırlık vektörü kullanılacaktır.

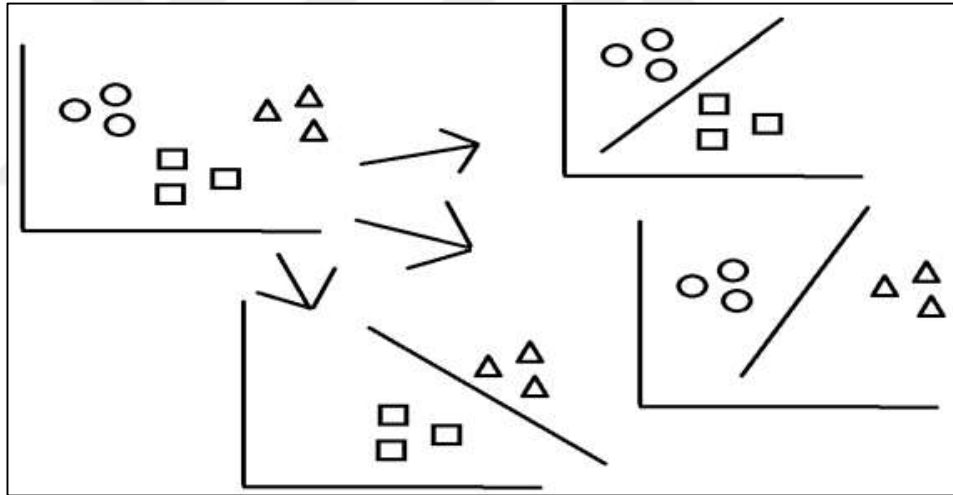
Bir sonraki ağırlık vektörü, sol alttaki resimde yer alan ağırlık vektörüdür. Sol alttaki resimde gösterilen ağırlık vektörü kullanıldığında, sınıfı tahmin edilecek özellik vektörü yuvarlaklar sınıfına ait olduğu için diğer sınıflar tarafında kalacaktır. Tahmin edilecek özellik vektörü, diğer sınıflar tarafında kaldığı için bir sonraki ağırlık vektörü kullanılacaktır. Bir sonraki ağırlık vektörü, sol alttaki resimde yer alan ağırlık vektörüdür.

Bir sonraki ağırlık vektörü, sağ alttaki resimde yer alan ağırlık vektörüdür. Sağ alttaki resimde gösterilen ağırlık vektörü kullanıldığında, sınıfı tahmin edilecek özellik vektörü yuvarlaklar sınıfına ait olduğu için yuvarlaklar tarafında kalacaktır. Diğer sınıflar tarafında kalmayan özellik vektörü kullanılan ağırlık vektörünün pozitif sınıf

tarafında olduğu için özellik vektörünün ait olduğu sınıf “yuvarlak” olarak tahmin edilir.

2.10.2. Bire karşı bir (one-vs-one)

Bire Karşı Bir(One-vs-One) yaklaşımı, N tane sınıf için $N(N-1)/2$ farklı ağırlık vektörü elde edilmesi fikrine dayanır. Ağırlık vektörleri, veri setindeki her bir sınıf kombinasyonu için elde edilir. Diğer bir ifade ile; her bir sınıf için $N-1$ tane ağırlık vektörü elde edilir. Peki, neden toplam elde edilecek ağırlık vektörü sayısı $N(N-1)$ değil de, $N(N-1)/2$ 'dir? Bunun nedeni bir ağırlık vektörünün iki sınıfı ayırtmak için kullanılmasından kaynaklanmaktadır. Örneğin; A ve B iki sınıf olsun. A ve B sınıflarını ayırtan ağırlık vektörü ile B ve A sınıflarını ayırtan ağırlık vektörü aynıdır. Aşağıdaki Şekil 2.31'de üç farklı sınıfın Bire Karşı Bir yaklaşımı ile ayrıştırılması gösterilmiştir.



Şekil 2.31. Bire Karşı Bir yaklaşımında her bir sınıfı diğer başka sınıftan ayırtan $N(N-1)/2$ tane ağırlık vektörleri elde edilir.

Şekil 2.31'de yuvarlak, kare ve üçgen olmak üzere üç farklı sınıfın Bire Karşı Bir yöntemine göre ayrıştırılması alt resimlerle gösterilmiştir. Şekildeki en üstte soldaki resim, veri setindeki özelliklerin koordinat sistemine iz düşümünü göstermektedir. Sağ üstteki resim, yuvarlak ve kare sınıflarını birbirinden ayıran ağırlık vektörünü göstermektedir. Sol alttaki resim yuvarlak ve üçgen sınıflarını birbirinden ayıran ağırlık vektörünü göstermektedir. Sağ alttaki resim, yuvarlak ve üçgen sınıflarını birbirinden ayıran ağırlık vektörünü göstermektedir. Şekilde, üç tane sınıf olduğu için $3(3-1)/2=3$ tane farklı ağırlık vektörü elde edilmiştir.

Sınıfı bilinmeyen bir özellik vektörünün sınıf tahmini, $N(N-1)/2$ olan tüm ağırlık vektörlerinin kullanılmasıyla elde edilir. En çok elde edilen sınıf, tahmin edilen sınıf olarak belirlenir. Şekil 2.31'deki sınıflar üzerinden bir örnekle açıklamak faydalı olacaktır.

Örneğin tahmin edilecek özellik vektörünün gerçek sınıfı üçgen olsun. Şekildeki alt resimlere göre sınıfları ayırtırmak için elde edilen ağırlık vektörleri, özellik vektörünün ait olduğu sınıf bulmak için kullanılır. Sağ üstteki resimde gösterilen ağırlık vektörü kullanıldığında, özellik vektörünün sınıfı yuvarlak veya kare olacaktır. Sol alttaki resimde gösterilen ağırlık vektörü kullanıldığında, özellik vektörünün sınıfı üçgen olacaktır. Sağ alttaki resimde gösterilen ağırlık vektörü kullanıldığında, özellik vektörünün sınıfı üçgen olacaktır. Tüm belirlemeler sonucunda; kare ve yuvarlaklar sınıf en fazla 1 ve üçgen sınıfı 2 belirleme sayısına sahip olacaktır. Bu durumda üçgen sınıfı, en fazla belirlenen sınıf olduğu için tahmin edilen sınıf olarak seçilir.

2.10.3. Cezbedicilerin doğal çoklu sınıf sınıflandırıcı olarak kullanımı

GMACA kural vektörünün ürettiği cezbediciler ikili sınıflandırmada başarılı bir şekilde kullanılması 2.5. alt bölümünden itibaren anlatılmıştır. Bu yaklaşıma göre: Kural vektörünün ürettiği cezbediciler eğitim aşamasında, etrafında toplanan diğer durumların ait oldukları en fazla sınıfa göre belirlenir. Test aşamasında, sınıfı bilinmeyen özellik vektörünün ait olduğu cezbedici bulur ve cezbedicinin ait olduğu sınıf özellik vektörünün sınıfı olarak tahmin edilir.

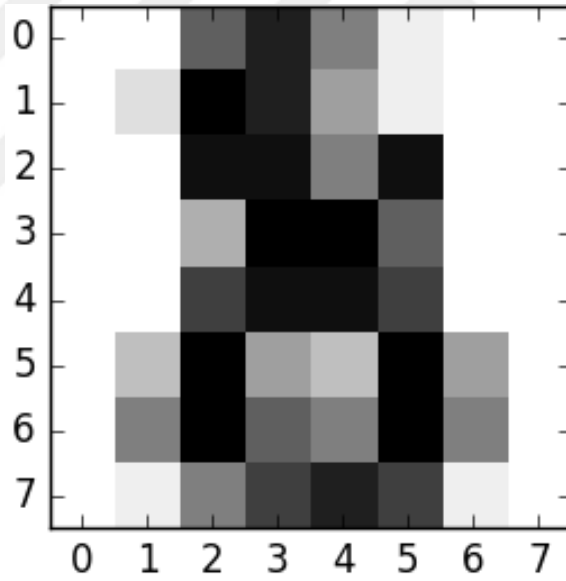
Cezbedicilerin ikili sınıfların ayrıştırılması için kullanılan yaklaşımı, çoklu sınıfların ayrıştırılmasında da kullanılabilecek yapıdadır. Bunun için; cezbedicilerin ait oldukları sınıflar, ikiden fazla sınıf içeren veriler üzerinde aynı mantıkla atanır. Yani: Eğitim aşamasında, cezbedicinin temsil edeceği sınıf etrafında toplanan diğer durumların ait oldukları en fazla sınıfa göre belirlenir. Test aşamasında, sınıfı tahmin edilmek istenen özellik vektörünün ait olduğu cezbedici bulunur. Cezbedicinin sınıfı, özellik vektörünün sınıfı olarak tahmin edilir.

2.10.4. Çoklu sınıflandırma için deneysel çalışma

Bu alt bölümde, farklı çoklu sınıflandırma stratejileri kullanılarak elyazması rakam veri seti üzerinden elde edilen deneysel sonuçlara yer verilmiştir. Kullanılan elyazması rakam görüntülerini içeren veri seti, Kaliforniya Üniversitesi Irvine Makine Öğrenmesi veri tabanından alınmıştır (Lichman, 2013). Veri setindeki görüntüler 8x8 boyutundadır ve 0 ile 16 arasında 17 farklı gri seviye değere sahiptir. Veri setindeki 1000 görüntü eğitim için ve 797 görüntü test için kullanılmıştır.

2.10.5. Görüntü verilerinin GMACA'ya uygun hale getirilmesi

Veri setinde yer alan rakam görüntülerinden biri Şekil 2.32'de verilmiştir. Görüntünün boyutu 8x8 ebatlarındadır. Tanıma işlemi yapılmadan önce 64 uzunluklu bir vektöre dönüştürülür.



Şekil 2.32. El yazması rakamlar veri setindeki orijinal görüntü örneği

GMACA ikili görüntüler üzerinde kullanılabilir. Bu nedenle görüntünün ikili değerlere dönüştürülmesi gerekmektedir. El yazması rakam görüntüleri, eşik değeri uygulanarak ikili değerlere dönüştürülmüştür. Daha sonra 8 uzunluklu her bir satır 8 bitlik ikili değerlere dönüştürülmüştür. Orijinal görüntü 8 satır ve 8 sütundan oluşmaktaydı. Dönüştürme işlemi yapıldıktan sonra 8 sekiz bit uzunluğunda 8 tane ikili değer elde edilmiştir. Veri setinin ikilik değerlere dönüştürüldükten sonraki hali aşağıdaki şekilde yer almaktadır.

0	1	2	3	4	5	6	7	target
00111000	00111110	01110110	01100110	01100110	01100110	01111100	00111000	0
00011100	00011100	00111100	01111100	00011100	00011100	00011100	00011100	1
00011100	00111100	00111100	00011100	00111000	01111000	01111110	00011110	2
00111000	01111100	01011000	00111000	00001100	00000110	00111110	00111100	3
00001000	00011000	00011110	00110110	01110110	01111100	00011100	00011100	4
00110000	00111100	00111100	00111100	00011110	00001110	00111110	00111100	5
00011000	00111000	00111000	00110000	00111100	00111110	00111110	00011110	6
00111110	00111110	00001100	01111110	01111100	00011000	00110000	00110000	7

Şekil 2.33. Elyazması veri setinin GMACA'ya uygun hale getirilmesi. Her bir satır bir rakama aittir.

Yukarıdaki şekilde her bir satır bir rakama aittir. En sağdaki sütunda rakamın değeri yer almaktadır. Eğitim aşamasında, veri setinde yer alan sekiz sütundan her biri eğitim için kullanılır. Yani; her bir sütun için ağırlık vektörü elde edilir.

2.10.6. Deneysel sonuçlar

Bu alt bölümde farklı çoklu sınıf sınıflandırma yaklaşımlarına göre elde edilen deneysel sonuçlara yer verilecektir. Sonuçlar; başarı oranı, karışıklık matrisi ve sınıflandırma raporu gibi istatistiksel değerlendirme ölçütleriyle verilmiştir.

Yukarıdaki alt bölümlerde bahsedilen; Bire Karşı Hepsi, Bire Karşı Bir ve Cezbedicilerin Doğal Çoklu Sınıf Sınıflandırıcı Olarak Kullanımı yaklaşımlarından ilkinde sonuç elde edilememiştir. Bire Karşı Hepsi yaklaşımından sonuç elde edilememesinin nedeni şöyledir: GMACA ile sınıfları ayırtıran kural vektörü Bire Karşı Hepsi yaklaşımı ile elde edilememiştir. Açıklaması aşağıdaki paragraflarda detaylıca anlatılmıştır:

Bire Karşı Hepsi yaklaşımında bir sınıf pozitif sınıf olarak seçililer ve diğer tüm sınıflar negatif sınıf kabul edilir. Eğitim aşamasında; pozitif ve negatif sınıfları birbirinden ayırtıran ağırlık vektörü bulunur. Ağırlık vektörü bulma işlemi her bir sınıf için gerçekleştirilir. GMACA kural vektörünün ürettiği cezbediciler eğitim aşamasında, etrafına toplanan diğer durumların en fazla ait olduğu durumu göre belirlenir.

Eğitim aşamasında cezbedicilerin sınıfını belirleme işlemi, elyazması veri setindeki her bir sütun için yapılır. Bu şekilde her bir örnek için sekiz farklı ağırlık vektörü elde

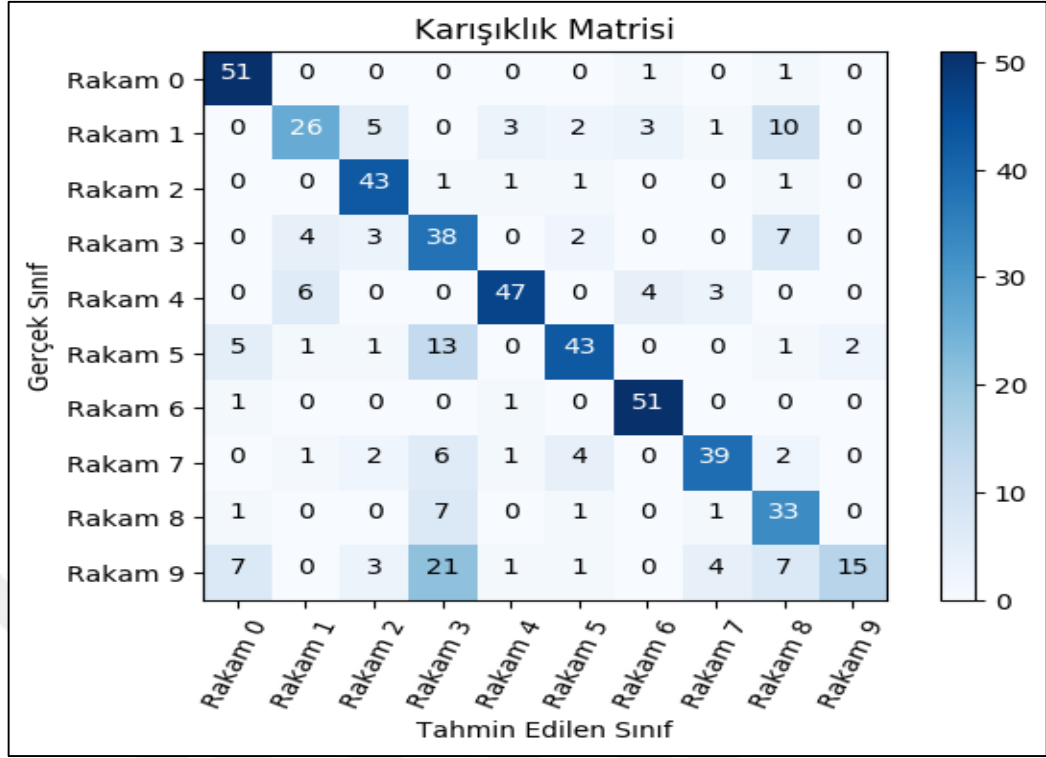
edilir. Test aşamasında, girdi olarak verilen ve sınıfı tahmin edilmek istenen özellik vektöründe yer alan sekiz özelliğin ait oldukları sınıflar eğitim aşamasında elde edilen ilgili ağırlık vektörlerine göre belirlenir. Bu şekilde özellik vektöründeki özellikler pozitif ve negatif sınıflara atanır. En fazla atanan sınıf, girdi olarak verilen özellik vektörünün sınıfı olarak tahmin edilir.

Bire Karşı Hepsini için yapılan deneysel çalışma yukarıdaki iki paragrafta göre ayarlanmıştır. Ancak yapılan deneysel çalışmada, test aşamasında girdi olarak verilen özellik vektörü negatif sınıf olarak tahmin edilmektedir. Tahminler hep negatif sınıf tarafında olduğu için pozitif sınıf hiçbir zaman tahmin edilememiştir. GMACA'ların Bire Karşı Hepsini yaklaşımına göre eğitilmesi çözülmesi gereken bir problem olarak ele alınabilir. Bu yönde yapılacak araştırmalar hücreli otomatların örüntü sınıflandırma yeteneklerine önemli katkı sağlayacaktır.

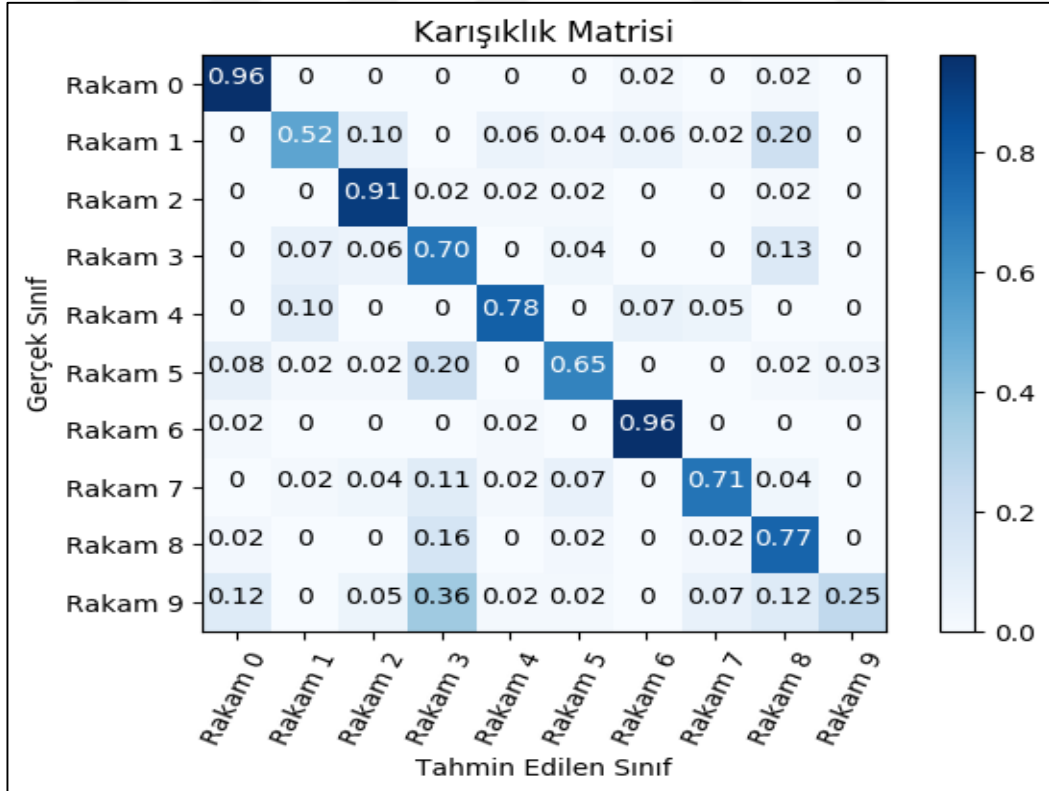
2.10.6.1. Bire karşı bir ile elde edilen sonuçlar

Bire Karşı Bir çoklu sınıf stratejisine göre, eğitim aşamasında veri setindeki yer alan tüm ikili sınıf kombinasyonlarını birbirinden ayıran ağırlık vektörü elde edilir. Test aşamasında, sınıfı bilinmeyen özellik vektörü tüm kombinasyonlara göre sınıfı bulunur ve en fazla bulunan sınıf tahmin edilen sınıf olarak kabul edilir.

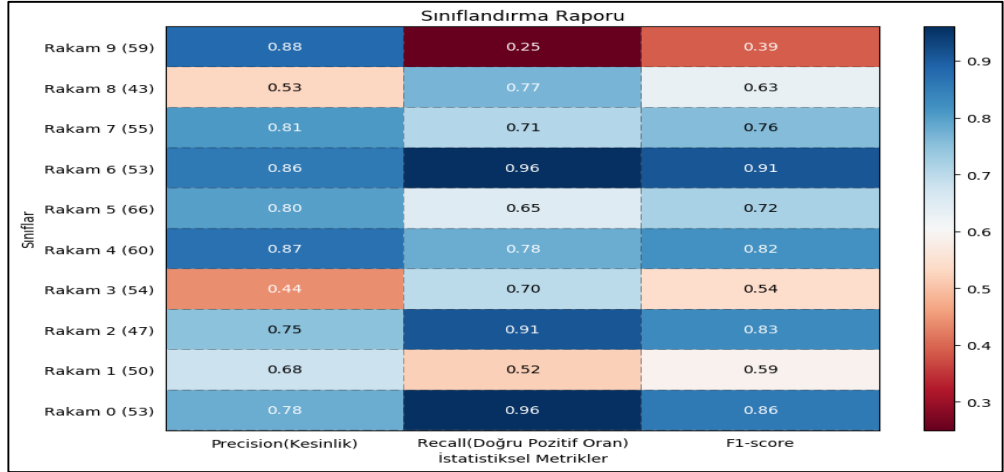
N tane sınıfı olan bir veri seti için eğitilmesi gereken $N(N-1)/2$ tane sınıf kombinasyonu olacaktır. Kullandığımız elyazması veri setinde 0-9 arası 10 sınıf olduğu için; eğitilmesi gereken $10(10-1)/2=45$ sınıf kombinasyonu olacaktır. Test aşamasında girdi olarak verilen özellik vektörünün sınıfı, 45 kombinasyona göre test edilecek ve en fazla tahmin edilen sınıf tahmin edilen sınıf olarak belirlenecektir. Aşağıda Şekil 2.34, 2.35 ve 2.36'te Bire Karşı Bir çoklu sınıf yaklaşımıyla GMACA sınıflandırıcıları kullanılarak elyazması veri seti üzerinde elde edilen sonuçların karışıklık matrisleri ve sınıflandırma raporlarına yer verilmiştir. Veri seti üzerinde elde edilen başarı oranı 0.72'dir.



Şekil 2.34. Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Bire Karşı Bir çoklu sınıflandırma yaklaşımı ile elde edilen karışıklık matrisi



Şekil 2.35. Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Bire Karşı Bir çoklu sınıflandırma yaklaşımı ile elde edilen normalleştirilmiş karışıklık matrisi



Şekil 2.36. Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Bire Karşı Bir çoklu sınıflandırma yaklaşımıyla elde edilen sınıflandırma raporu

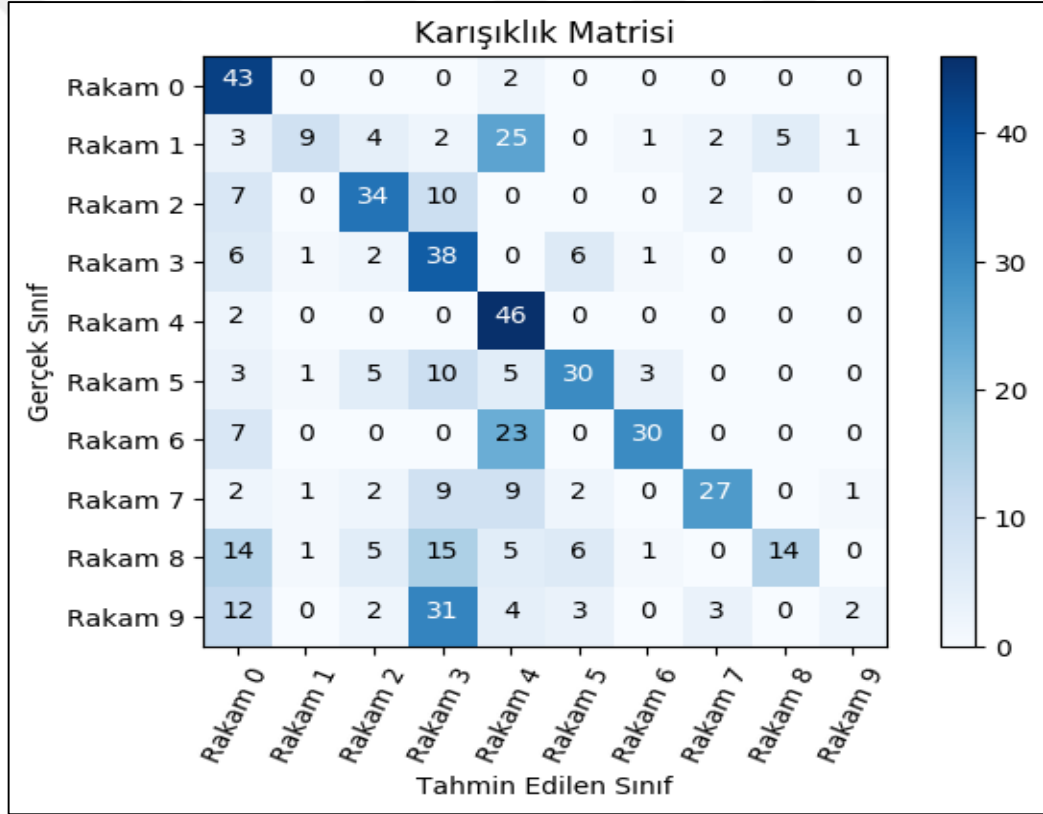
Bire Karşı Bir yaklaşımıyla tüm kombinasyonlar için elde edilen sonuçlar aşağıdaki Tablo 2.16’da verilmiştir.

Tablo 2.16. Bire Karşı Bir tüm kombinasyonlar için elde edilen başarı oranları

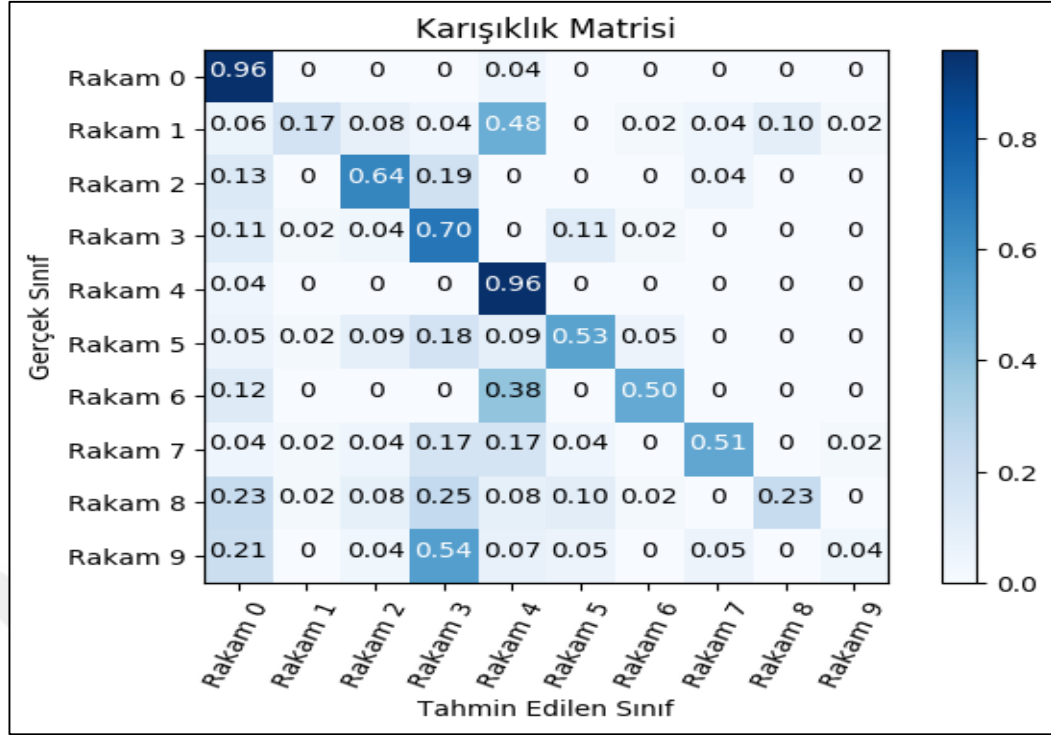
Kombinasyon	Başarı Oranı(%)	Kombinasyon	Başarı Oranı(%)
0-1	91,67	2-9	88,89
0-2	98,13	3-4	100
0-3	99,08	3-5	84,55
0-4	95,38	3-6	96,36
0-5	97,22	3-7	95,41
0-6	96,3	3-8	83,33
0-7	99,07	3-9	86,24
0-8	92,46	4-5	96,33
0-9	93,52	4-6	97,25
1-2	97,22	4-7	90,74
1-3	93,64	4-8	97,2
1-4	88,07	4-9	96,33
1-5	90	5-6	98,17
1-6	89,91	5-7	97,25
1-7	93,58	5-8	92,52
1-8	82,24	5-9	91,74
1-9	87,16	6-7	97,22
2-3	81,48	6-8	93,46
2-4	96,3	6-9	96,33
2-5	100	7-8	95,28
2-6	100	7-9	92,6
2-7	96,27	8-9	78,5
2-8	92,45	Ortalama	93,26

2.10.6.2. Cezbedicilerin doğal çoklu sınıf sınıflandırıcı olarak kullanımı ile elde edilen sonuçlar

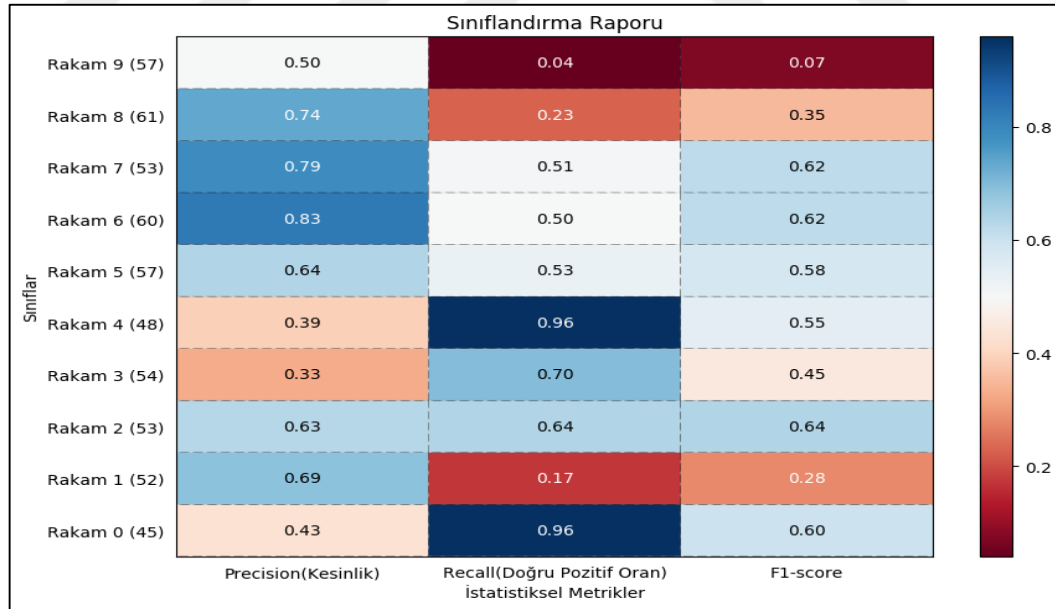
Cezbedicilerin ikili sınıfların ayrıştırılması için kullanılan yaklaşımı, çoklu sınıfların ayrıştırılmasında da kullanılabilecek yapıdadır. Bunun için; cezbedicilerin ait oldukları sınıflar, ikiden fazla sınıf içeren veriler üzerinde aynı mantıkla atanır. Yani: Eğitim aşamasında, cezbedicinin temsil edeceği sınıf etrafında toplanan diğer durumların ait oldukları en fazla sınıfa göre belirlenir. Test aşamasında, sınıfı tahmin edilmek istenen özellik vektörünün ait olduğu cezbedici bulunur. Cezbedicinin sınıfı, özellik vektörünün sınıfı olarak tahmin edilir. Bu yaklaşım için elde edilen sonuçların karışıklık matrisleri ve sınıflandırma raporları aşağıdaki şekillerde verilmiştir.



Şekil 2.37. Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Doğal çoklu sınıflandırma yaklaşımı ile elde edilen karışıklık matrisi



Şekil 2.38. Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Doğal çoklu sınıflandırma yaklaşımı ile elde edilen normalleştirilmiş karışıklık matrisi



Şekil 2.39. Elyazması rakamlar veri seti üzerinde GMACA sınıflandırıcılarının Doğal çoklu sınıflandırma yaklaşımıyla elde edilen sınıflandırma raporu

Tablo 2.17’de bilinen makine öğrenme yöntemleri ve GMACA sınıflandırıcısı ile elde edilen sonuçlara yer verilmiştir.

Tablo 2.17. Bilinen makine öğrenme modelleriyle karşılaştırma

Veri Seti	Yöntem	Performans(%)
MNIST orijinal	Logistik Regresyon	95,37
	Destek Vektör Makineleri: kernel=rbf	39,25
	Destek Vektör Makineleri kernel=lineer	97,96
	Lineer Ayrıştırma Analizi	94,81
	Karar Ağaçları	85
	En Yakın Komşu Algoritması	99,25
	Rasgele Orman	95,55
	Gauss Naive Bayes	85,18
	Gradient Boosting	96,85
MNIST, hücre sel otomata için uygun hale getirildikten sonra	Logistik Regresyon	45,37
	Destek Vektör Makineleri: kernel=rbf	10,55
	Destek Vektör Makineleri kernel=lineer	Yok
	Lineer Ayrıştırma Analizi	36,66
	Karar Ağaçlar	72,59
	En Yakın Komşu Algoritması	65,37
	Rasgele Orman	82,96
	Gauss Naive Bayes	37,59
	Gradient Boosting	89,07
	GMACA doğal çoklu sınıflandırma stratejisiyle	52
	Bire Karşı Hepsini stratejisiyle	72,22
	Bire Karşı Hepsini kombinasyonlar ortalaması	93,26

3. BİRLEŞTİR/BUL (UNION/FIND) VERİ YAPISI KULLANILARAK GMACA'LARIN KÜMELERE AYRIŞTIRILMASI

Bilim insanları mikroskoplardan galaksilere farklı ölçekteki konularla ilgilenirler. İlgilenilen konunun fiziki büyüklük veya küçüklüğüne göre farklı araç ve gereçler geliştirilmiştir. Aynı şekilde bilgisayar bilimiyle uğraşan kişilerde farklı ölçekteki konular üzerine yoğunlaşırlar. Ancak bilgisayar bilimiyle uğraşanların dikkate alması gereken ölçü fiziki büyüklük ve küçüklükten ziyade verinin büyüklük ve küçüklüğüdür (Goodrich ve Tamassia, 2015). Algoritmalar ve veri yapıları bilgisayarlar kaynaklarının büyük veriler üzerinde de verimli çalışmasına odaklanan iki uzmanlık alanıdır.

Algoritmalar bilgisayarın hesaplama kaynaklarının verimli kullanılmasına odaklanırken veri yapıları ana hafızada verinin organizasyonu, eklenmesi, çıkarılması ve güncellenmesi gibi işlemlerin verimliliğine odaklanır. Basitçe ifade edersek algoritma belirli bir zaman içerisinde bir görevin yerine getirilmesi için takip edilen adımlardır. Veri yapıları veriyi sistematik bir şekilde organize etme ve erişme yoludur. Algoritmalar ve veri yapıları bilgisayarlı hesaplamanın çekirdeğini oluştururlar ve doğru veri yapısı ve algoritmanın seçimi hafıza kullanımı ve işlemci performansı için hayati öneme sahiptir.

Küme ayırıştırma, kısımlara ayırma ve vb. isimlerle adlandırılan union/find(birleştir/bul) veri yapısı n elemanlı bir dizideki verileri farklı kümelere ayırştırmak için kullanılır. Birleştir/bul veri yapısı verinin ait olduğu kümenin bulunması, farklı iki kümeye ait verilerin birleştirilerek tek bir küme oluşturulması işlemlerini gerçekleştirmek için hızlı bul, hızlı birleştir, yol sıkıştırırmalı hızlı bul ve ağırlıklı hızlı bul yöntemleri içerir.

Genelleştirilmiş Çoklu Cezbedici Hücresel Otomatlar(Generalized Multiple Attractor Cellular Automata-MACA) hücrelerden oluşan ızgara düzlemi üzerinde n tane hücrenin kendisi, sol ve sağ komşularının etkileşimine göre hücrenin durumunu güncellemesi mantığına dayanır. GMACA'lar bir Hücresel Otomata(Cellular

Automata) türüdür. GMACA'lar n tane hücrenin her birine ait bir kuralın uygulanmasıyla farklı geçiş diyagramları formu oluşturur. Bu geçiş diyagramları grup ve grup olmayan hücresel otomatlar olarak iki farklı şekilde adlandırılır.

Grup ve grup olmayan durum geçiş diyagramlarına ait durumlar bir küme olarak düşünülebilir. Birleştir/bul veri yapısı GMACA'ların oluşturduğu grup ve grup olmayan durum geçiş diyagramlarına ait durumların ait oldukları kümelerin oluşturulmasında ve bulunmasında; hafıza ve uygulama olarak verimli bir yol sunmaktadır. Tez çalışmasının bu bölümünde birleştir/bul veri yapısı, GMACA durum geçiş diyagramlarının ayrıştırılmasında kullanılmıştır.

Birinci alt bölümde birleştir/bul veri yapısı ve bu veri yapısının gerçekleştirilmesinde önerilen farklı yöntemlere yer verilmiştir. İkinci alt bölümde birleştir/bul veri yapısının GMACA gruplarının ayrıştırılmasında kullanımı anlatılmıştır. Üçüncü alt bölümde küme ayrıştırma işlemi görüntü verileri üzerinde test edilmiştir. Sonuç bölümünde değerlendirme ve önerilere yer verilmiştir.

3.1. Birleştir/Bul (Union/Find) Veri Yapısı

Birleştir/bul veri yapısı denklik probleminin çözümüne etkili bir yol sunar. Bir kümedeki eleman çiftinin birbirleriyle bağıntısının olup olmaması doğru ve yanlış olarak ifade edilir. Doğru ise bağıntı vardır, yanlış ise bağıntı yoktur. Bir bağıntının denklik bağıntısı olabilmesi için üç özelliğinin olması gerekir; yansıyan, simetrik ve geçişken. Yansıyan: aRa tüm $a \in S$, Simetri: aRb ancak ve ancak bRa Geçişken: aRb ve bRc ise aRc (Weiss, 2012).

Denklik ilişkisi($a \sim b$) verilen iki noktanın bağlantılı olup olmadığını bulmak kolaydır. İki boyutlu bir dizi oluşturularak noktalar arası bağlantı varsa ilgili hücreye 1 yoksa 0 yazılarak bağlantılar tespit edilebilir. Ancak zorluk, bağlantıların direk olduğu durumlarda değil dolaylı olduğu durumlarda ortaya çıkıyor. Beş elemanlı bir küme $\{a_1, a_2, a_3, a_4, a_5\}$ için elemanların tümünün bağlantılı olup olmadığını belirleyecek 25 farklı bağlantı çifti vardır. Ancak a_1 'in a_2 ile, a_3 'ün a_4 ile, a_5 'in a_1 ile ve a_4 'ün a_2 ile bağlantılı olduğu bilirse bu kümedeki tüm elemanların bağlantılı olduğu söylenebilir.

Bir S kümesindeki a elemanı ait olduğu alt kümedeki tüm elemanlarla bağlantılıdır. Bir kümedeki elemanların ait olduğu alt kümelere ayrılması bu kümenin alt bölümlere bölünmesi anlamına gelir. Her bir elemanın ait olduğu sadece bir alt küme vardır. Eğer a ve b elemanlarının bağlantılı olup olmadığına karar vermek istiyorsak elemanların ait oldukları alt kümelerin bilinmesi yeterli olacaktır. Elemanlar aynı alt kümeye ait ise bağlantılı, değilse bağlantılı değildir.

Birleştir/bul veri yapısı n elemanlı bir kümeyi n tane alt küme şeklinde ele alır. Daha sonra bu kümeler arasında birleştirme ve bulma işlemi yapar. Birleştirme işlemi iki farklı alt kümeye ait elemanları bir alt kümede birleştirir. Birleştirilen iki alt kümedeki tüm elemanlar bir alt kümeye ait olur. Bul işleminde elemanın hangi kümeye ait olduğu belirlenir.

Birleştir/bul veri yapısının iki temel fonksiyonu vardır; birleştir ve bul fonksiyonları. Bul fonksiyonu girdi olarak verilen bir elemanın hangi kümeye ait olduğunu verir. Birleştir fonksiyonu verilen iki elemanın birleştirir. Birleştirme işlemi yapılırken ilk önce bu elemanların ait oldukları alt kümeler bulunur. Eğer elemanların ait oldukları kümeler aynı ise birleştirme işlemi yapılmaz. Eğer elemanlar farklı alt kümelere aitse bu iki alt küme birleştirilerek tek bir alt küme elde edilir. Birleştir/bul veri yapısını gerçekleştirmek için kullanılan algoritmalar dinamik yapıdadır. Çünkü algoritmanın çalışması esnasından birleştir işleminden dolayı kümeler değişebilmektedir.

Tablo 3.1. Farklı Birleştir/bul algoritmalarının zaman karmaşıklığı (Sedgewick ve Wayne, 2011)

Algoritma	Yapıcı Metot	Birleştir	Bul
Hızlı-bul	N	N	1
Hızlı-birleştir	N	Ağaç yüksekliği	Ağaç yüksekliği
Ağırlıklı-hızlı-birleştir	N	lgN	lgN
Yol-sıkıştırılmalı-ağırlıklı-hızlı-birleştir	N	Sabit zamana çok yakın ancak sabit zamanlı değil	
İmkansız	N	1	1

Birleştir/bul algoritmasını gerçekleştirmek için önerilen farklı yaklaşımlar vardır; hızlı-birleştir, hızlı-bul, ağırlıklı-hızlı-birleştir ve yol-sıkıştırılmalı-ağırlıklı-hızlı-birleştir. Yaklaşımların zaman karmaşıklığı Tablo 3.1.'de verilmiştir.

Algoritmanın gerçekleştirilmesi için n elemanlı bir dizi yeterli olacaktır. Algoritma ilk çalışmaya başladığında yapıcı metot dizi deki tüm elemanlara kendi indeks değerini atayacaktır.

3.1.1. Hızlı-bul

Bir alt kümedeki tüm elemanlar aynı değere sahiptirler. Bul fonksiyonu girdi olarak verilen bir elemanın ait olduğu alt kümeyi sabit zamanda verir. Birleştirme fonksiyonu aynı kümeye ait olmayan iki elemanın alt kümelerini birleştirmek için iki alt kümedeki elemanlar bir değerle temsil edilir. Bunun için alt kümelerden birinin değeri diğer alt kümenin değeri ile değiştirilir. Değiştirilecek alt kümenin tüm elemanlarının değerinin değişmesi gerektiği için dizi baştan sona taranır. Bu nedenle zaman karmaşıklığı N 'dir.

3.1.2. Hızlı-birleştir

Hızlı-birleştir yaklaşımı iki farklı alt kümeye ait elemanların birleşimini alt kümeye ait kök elemanların birleştirilmesi ile gerçekleştirir. Girdi olarak verilen iki elemanın kök elemanları bulunur eğer kök elemanlar farklı ise birleştirilmek istenen elemanlar farklı alt kümelere aittir. Kök elemanlardan birinin değeri diğerine atanarak birleştirme işlemi gerçekleştirilir. Hızlı-birleştir yaklaşımı ile alt kümelerin birleşimi ağaç yapısını meydana getirecektir. Birleştirmede kök dizinlerin bulunması ağaç yüksekliği kadar olacaktır. Ancak en kötü durumda algoritmanın çalışma performansı N 'dir.

3.1.3. Ağırlıklı-hızlı-birleştir

Hızlı birleştirmede iki farklı alt kümenin birleştirmesinde alt kümelerden biri diğerine rasgele dahil olur. Bu işlemde alt kümelerin büyüklüklerine bakılmaz. Ağırlıklı-hızlı-birleştir yaklaşımında alt kümelerin sahip oldukları eleman sayısı yani büyüklüğü dinamik olarak hesaplanır. Birleştirme işlemi yapılacağı zaman küçük ağaç büyük ağaca dahil olur. Böylece ağacın yüksekliğinin $\log N$ olması garanti edilmiş olur. Oluşan ağaç yüksekliğinin maksimum boyutu $\log N$ olduğundan dolayı bulma işlemi ve birleştirme işlemi $\log N$ zamanda gerçekleştirilir.

3.1.4. Yol-sıkıştırma-ağırlıklı-hızlı-birleştir

Bu yaklaşım ağacın yüksekliğini minimum seviyeye indirir. Her bulma işlemi yapılırken ağacın yüksekliği azalır ve minimum seviyede ağaç yüksekliği elde edilir. Sabit zamana yakın bir performans elde edilir.

3.2. Birleştir/Bul Veri Yapısının GMACA Sınıflarının Ayırıştırılmasında Kullanılması

GMACA'lar örüntü sınıflandırma ve tanımda kullanılmışlardır. GMACA'ların geçiş diyagramları tek bir diyagram olabileceği gibi bir den fazla alt diyagramlardan da oluşabilir. Örüntü tanıma ve sınıflandırmada kullanılan GMACA'lar genelde tek uzunluk döngülü cezbedicilerden oluşur. Örüntü sınıfları bu cezbediciler etrafında gruplanır. Cezbedici örüntü etrafından oluşan örüntü grupları tüm örüntüleri kapsayan kümenin alt kümeleri olarak düşünülebilir. Çok uzunluk döngülü cezbedicilerden oluşan GMACA diyagramları da örüntülerin farklı gruplar etrafından toplanmasıyla oluşur. Aynı şekilde örüntü grupları tüm örüntüleri kapsayan kümenin alt kümeleri olarak düşünülebilir.

Birleştir/bul algoritması GMACA örüntü gruplarının kümelere ayırıştırılmasında elverişli bir yol sunuyor. Algoritmanın uygulanabilmesi için küme elemanları arasındaki bağıntının denklik bağıntısı olması gerekir. Bir bağıntının denklik bağıntısı olabilmesi için üç özelliğinin olması gerekir; yansıyan, simetrik ve geçişken. GMACA'nın oluşturduğu geçiş diyagramları yönlü graf formundadır. Grafların yönlü olarak belirtilmesinin nedeni durumların geçiş yönüdür. Örüntü sınıflandırmada ve tanımda durumların geçiş yönü önemli değildir. Önemli olan durumların ait oldukları diyagramlardır. Diğer bir ifade ile önemli olan örüntüleri ait oldukları kümelerdir.

Birleştir/bul algoritmasının uygulanması için kullanımı basit veri yapısı olan diziler yeterli olmaktadır. Farklı birleştir/bul yaklaşımlarının zaman karmaşıklığını Tablo 3.1'de verilmiştir. Uygulamanın yapısına göre istenen yaklaşım tercih edilebilir.

Yukarıda verilen GMACA örneklerinde vektörü uzunluğu genelde dört bitten yani dört kuraldan oluşuyordu. Dört bit uzunluklu kural vektörünün oluşturacağı durum sayısı $2^4=16$ dır. Bit sayısı arttıkça durum sayısı da üstel olarak artmaktadır. 32

kuraldan oluşan bir kural vektörünün oluşturacağı kural sayısı 4294967296'dır. Büyük sayılar için örüntülerin kümelere hızlı bir şekilde ayrıştırılması ve örüntünün ait olduğu kümenin bulunması önemlidir. Birleştirme işlemi tamamlandıktan sonra bulma işleminin yapılacağı uygulamalarda hızlı-bul yöntemi tercih edilebilir. Uygulanan yöntemin sözde kodu aşağıda verilmiştir.

```
Girdi: n elemanlı kural vektörü
Çıktı: Örüntü grupları
For mevcutDurum=0 dan  $2^n-1$  e kadar
    mevcutDurum'u n basamaktan oluşan ikilik sayıya dönüştür
    ikilik sayıyı stringe dönüştür
    sıfır sınırlı CA olduğu için baş ve sona sıfır ekle
    mevcutDurum=0+mevcutDurum+0
    birSonrakiDurum=CA(mevcutDurum): Kural vektörüne mevcutDurum
    yüklenir ve bir sonraki durum ikilik sayı olarak bulunur.
    mevcutDurum ve birSonrakiDurum değişkenlerini tamsayıya dönüştür
    HızlıBulBirleştire(birSonrakiDurum, mevcutDurum): tercih edilen
    birleştir/bul yöntemine birSonrakiDurum ve mevcutDurum girdi olarak verilerek
    bu durumları birleştir.
For sonu
GruplarıYazdır
```

Şekil 3.1. Birleştir/Bul veri yapısı ile GMACA kümelerinin ayrıştırılmasına ait sözde kod.

Algoritma n tane kurala sahip bir kural vektörünü girdi olarak alır ve kural vektörüne göre oluşan örüntü gruplarını çıktı olarak verir. Her bir örüntü grubu, kural vektörünün oluşturduğu durum geçiş diyagramlarını temsil eder.

N tane kurala sahip bir kural vektörü, 0 ile 2^{n-1} arasındaki tamsayıların ikili değerlerinin bir sonraki durumunu belirleyebilir. Bu nedenle 2^n tane durum için bir sonraki durum kural vektörüne göre elde edilir ve tercih edilen birleştir/bul yaklaşımına göre birleştirilir.

GMACA ikili değerler üzerinde işlem yapabildiği için 0 ile 2^{n-1} arasındaki tamsayı değerler kural vektörüne yüklenmeden önce ikili değerlere dönüştürülür. Tamsayılar ikili değere dönüştürülürken kural vektörünün uzunluğu dikkate alınır. Her bir ikili değer uzunluğu kural vektörünün uzunluğu kadar olmalıdır. Kural vektörünün uzunluğundan daha az sayıda uzunluğu sahip ikili değerlerin soluna gerekli sayıda sıfır eklenir.

CA, durum komşuluklarına göre sıfır sınırlı ve periyodik sınırlı olmak üzere iki ayrılır. Bu çalışmada sıfır sınırlı CA kullandığı için ikili değerin en soluna ve en sağına sıfır eklenir.

Algoritma her bir ikili durum için çalıştırıldıktan sonra elde edilen gruplar çıktı olarak verilir.

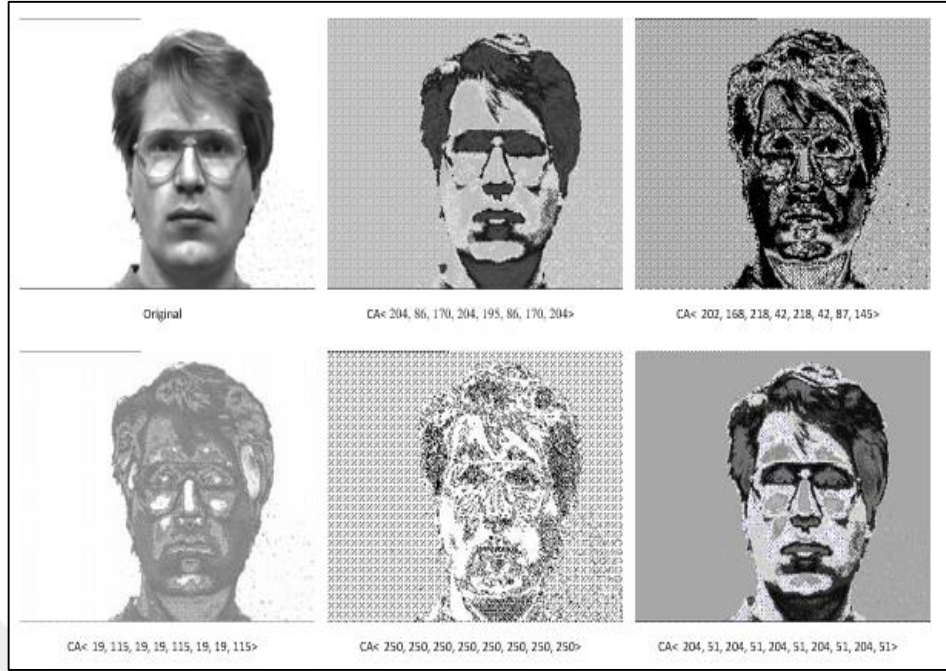
3.3. Gerçek Veriler Üzerinde Test Edilmesi

Önerilen yöntem, görüntü verileri üzerinde test edilmiştir. Görüntü piksel değerleri, kural vektörü tarafından üretilen geçiş şemasına göre gruplandırılır. Örneğin, geçiş diyagramı sayısı 16 ise, o zaman gruplandırılmış piksel sayısı 16 olacaktır. Her bir piksel grubu temsil eden pikselin değerini alır. Diğer bir deyişle, bir gruptaki tüm piksellerin aynı değeri vardır.

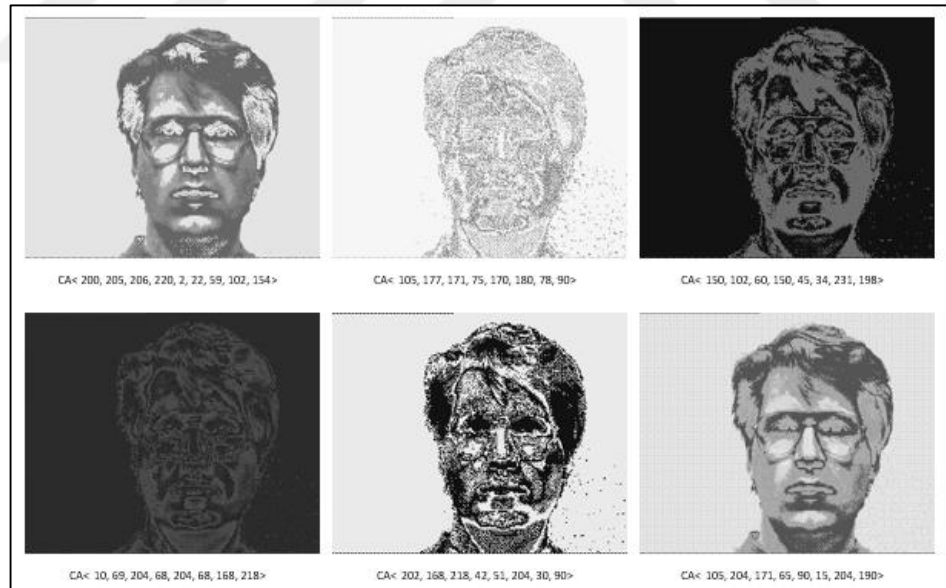
Kural vektörüne göre gerçekleştirilen piksellerin gruplanması. Farklı kural vektörleri kullanıldığında farklı gruplar elde edilecektir. Bu işlem, deseni veya özelliği görüntülerden çıkarmak anlamına gelir. Ortaya çıkan desenler veya özellikler yüz tanıma (Peldek, 2012) veya diğer bilgisayarlı görme uygulamaları için kullanılabilir.

Özellik çıkarma, boyut küçültmenin bir yoludur. Mevcut verilerin boyutunu azaltmak için iki yöntem vardır; özellik seçimi ve özellik çıkarma. Özellik seçimi, özellik vektöründeki doğru veya ayırt edici özellikleri kullanmayı amaçlamaktadır. Bunun için düzenleme yöntemleri ve özellik seçimi algoritmaları kullanılmaktadır. Özellik çıkarma yöntemleri ayrıca veri boyutunu azaltmaya yardımcı olur. Özellik seçim yöntemlerinden farkı: Mevcut özellik vektöründen belirgin şekilde farklı değerler elde eder ve daha az boyutlara sahip özellik vektörlerine sahip olur. Özellik çıkarmanın temel amacı, söz konusu özellik vektörünün daha küçük boyutlu bir özellik vektörü ile temsil edilmesidir. Önerilen yöntemin görüntüler üzerindeki kullanımı, aynı zamanda boyut küçültme görevi anlamına da gelmektedir.

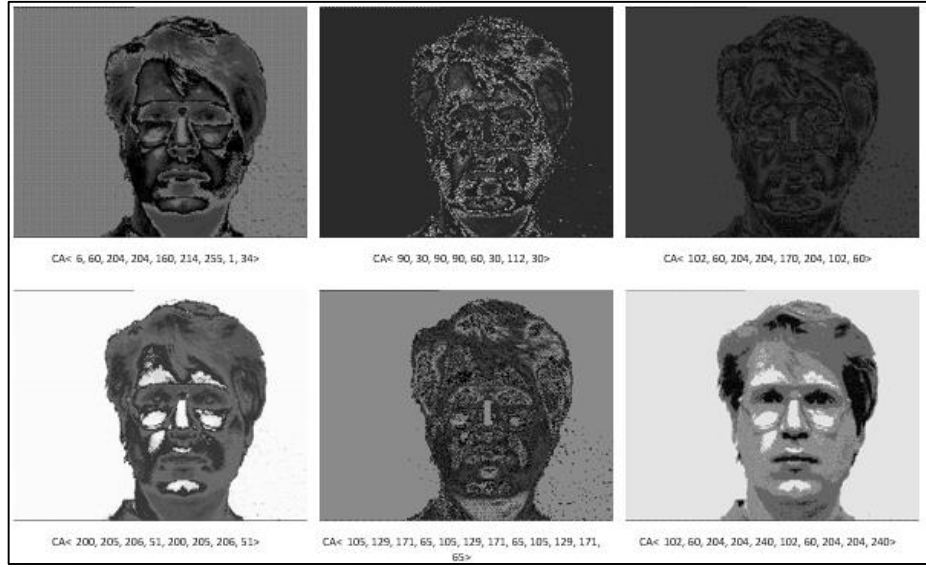
Farklı kural vektörlerini kullanarak görüntülerin piksellerinden desenler çıkarılmıştır. Aşağıdaki şekillerde elde edilen sonuçlar yer almaktadır.



Şekil 3.2. Yukarıdaki görüntülerin ilk resmi Yale Face Database [25] 'den elde edilen orijinal görüntüdür ve takip eden görüntüler farklı kural vektörlerine göre elde edilen sonuçlardır. Her bir resmin altında uygulanan kural vektörü yer almaktadır



Şekil 3.3. Farklı kural vektörlerine göre elde edilen sonuçlar. Her bir görüntünün altında uygulanan kural vektörü yer almaktadır.



Şekil 3.4. Farklı kural vektörlerine göre elde edilen sonuçlar. Her bir görüntünün altında uygulanan kural vektörü yer almaktadır

Deneyisel sonuçlar, her kural vektörünün neredeyse benzersiz bir desen çıkardığını göstermektedir. Bu bize görüntülerden farklı özellikler çıkarma yeteneği sağlar. Kullanılacak olan kural vektörü, (Das ve ark., 2009; Ganguly ve ark., 2002) çalışmalarında belirtildiği gibi eğitim aşamasında belirlenebilir.

Dört uzunluklu bir kural vektörü, $2^8 \times 2^8 \times 2^8 \times 2^8 = 4294967296$ farklı kural dizilimine sahip olabilir. Kural vektörü uzunluğu arttıkça, elde edilebilecek farklı kural vektörü sayısında üstel olarak artmaktadır. Bu bize çok geniş aralıkta kural vektörü seçme imkanı verir. Çok fazla seçeneğin olmasının dezavantajı; seçim yapmayı zorlaştırmasıdır. Bu dezavantaj hücrel otomata ve görüntü işleme ile ilgili araştırmacılar için bir fırsat olabilir.

Kural vektörünü oluşturan kurallar, elde edilecek durum geçiş diyagramlarının yapısına doğrudan etki ettiği gibi görüntülerden elde edilecek görüntülerin yapısına da doğrudan etki etmektedir. Kuralların durum geçiş diyagramlarının oluşumuna nasıl etki ettiği, önceki bölümde detaylıca ele alınmıştır. Hücrel otomatlar durum geçiş diyagramlarının yapısına göre iki ana gruba ayrılmıştır; ters döndürülebilir ve ters döndürülemez hücrel otomatlar. Önerilen yöntem iki farklı hücrel otomatın durum geçiş diyagramlarını sorunsuz bir şekilde ayırtmaktadır

Kural vektörünün görüntü verileri üzerinde uygulanırken dikkat edilmesi gereken bir durum; kural vektörünün uzunluğudur. Kural vektörünün uzunluğu kural vektöründe yer alan kural sayısı kadardır. Örneğin kural vektörü dört kuraldan oluşuyorsa bu kural vektörünün uzunluğu dördür. Kural vektörü ikili verilerin bir sonraki durumunu belirler. Bir kural vektörü sahip olduğu kural sayısına göre farklı sayılarda durumun bir sonraki durumunu belirleme kapasitesine sahiptir. Dört kuraldan oluşan bir kural vektörü $2^4 = 16$ farklı durumun bir sonraki durumunu belirleme kapasitesine sahiptir. Bu bölümdeki uygulamada kural vektörü 0 ile 255 arası değer alabilen gri görüntü verileri üzerinde kullanıldığı için kural vektörünün uzunluğunu en az sekiz olarak belirlenmiştir. Böylece kural vektörü 0 ile 255 arasında olabilecek tüm değerlerin bir sonraki durumunu belirleyebilecektir.

Uygulamama gri görüntüler üzerinde gerçekleştirilmiştir. Uygulama renkli görüntüler üzerinde de kolaylıkla gerçekleştirilebilir. Renkli görüntüler üzerinde işlem yapılırken her bir renk kanalı üzerinde yöntemin gerçekleştirilmesi gerekir.

Bazı görüntüler noktalı değerler içerebilir. Noktalı değere sahip görüntüleri üzerinde yöntemin uygulanabilmesi için, noktalı değerler istenen tamsayı aralığına ölçeklendirilmelidir. Tamsayı değerler elde edildikten sonra yöntem kolaylıkla uygulanabilir. Kural vektörü uzunluğunun, belirlenen renk aralığındaki değerlerin bir sonraki durumunu belirleyecek uzunlukta olması gerekmektedir.

3.4. Sonuç

Hücrel otomatların dinamik ve ayrık yapısı birleştir/bul veri yapısı ile kolay ve verimli bir şekilde uyum sağlamıştır. Kural vektöründeki kuralların özelliklerine göre oluşan ters döndürülebilir ve ters döndürülemez hücrel otomatların kümelere ayrıştırılması ve girdi olarak verilen bir elamanın hangi kümeye ait olduğu birleştir/bul veri yapısının gerçekleştirilmesi için önerilen hızlı-bul, hızlı-birleştir, ağırlıklı-hızlı-birleştir ve yol-sıkıştırılmalı-hızlı-birleştir yöntemleri ile gerçekleştirilmiştir. Birleştir/bul için önerilen bu farklı yaklaşımlar uygulamanın veya problemin gereksinimlerine göre tercih edilebilir. Eğer kümelerin ayrıştırılması ile elemanların hangi alt kümeye ait olduklarının bulunma işlemi eş zamanlı yapılacaksa yol-sıkıştırılmalı-hızlı-birleştir yöntemi zaman karmaşıklığı açısından en iyi sonucu

verecektir. Eđer birleřtirme iřlemi 6n hazırlık amacı ile yapılp uygulamanın alıřma zamanında sadece bul iřlemi yapılacaksa hızlı-bul y6ntemi en iyi sonucu vermektedir.

Birleřtir/bul veri yapısı tek bir diziye ihtiya duyan ve yerinde iřlem olarak adlandırılan; elemanların bulunduđu dizide iřlem yapan bir algoritmaya sahiptir. Bu alıřmada da farklı h6cresel otomatların ayrıřtırılma iřlemi tek bir dizi kullanılarak ve yerinde iřlem yapılarak gerekleřtirilmiřtir.



4. GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL OTOMATLARIN GÖRÜNTÜ BÖLÜMLEMEDE KULLANILMASI

Hücresele Otomatlar(Cellular Automata(CA)) basit yerel etkileşimlerle karmaşık global davranışlar sergileyebilen hücresele yapılardır. Hücrelerin yerel etkileşimleri basit kurallar ile gerçekleştirilir. Kural sayıları hücrelerin komşuluğu ve hücrelerin aldığı durum sayısına göre belirlenir. Bu çalışmada üç komşu ve iki duruma sahip hücresele otomatlar kullanılmıştır. Kullanılan hücresele otomata türüyle ilgili daha kapsamlı açıklamalar ikinci bölümde yer almaktadır.

Tez çalışmasının bu bölümünde, Genelleştirilmiş Çoklu Cezbedici Hücresele Otomatlar (Generalized Multiple Attractor Cellular Automata(GMACA)) görüntü bölümlemede kullanılmıştır. GMACA, her bitişik hücre ile etkileşime giren kafes düzlemindeki n sayıda hücrenin güncelleme mantığına dayanan bir hesaplama modelidir. Görüntü bölümleme görüntülerin anlaşılması ve analizi için önemli bir görüntü işleme işlevidir. Bu kısmın ana katkısı, GMACA'nın ürettiği cezbedicilerin görüntüleri bölümlere ayırtırmada kullanılabileceğini göstermesidir.

Görüntü bölümleme, bir görüntünün farklı gruplara sınıflandırılması işlemidir. K cezbedici durumları olan bir n -hücreli CA, k tane sınıf sınıflandırıcı olarak görülebilir. Kural vektörü tarafından üretilen her cezbedici durum, görüntü parçalanması için kullanılabilir. Bunu başarmak için, her cezbedici durum bir bölgenin temsilcisi olarak değerlendirilebilir. Bu basit fikir, uygun kural vektörleri seçilerek görüntülerde kolayca uygulanabilir.

Bu bölümde geliştirilen yöntem, kümeleme görüntü bölümleme yöntemleri kapsamında değerlendirilebilir. Geliştirilen görüntü bölümlendirme yönteminin temel varsayım; bir bölgeye ait piksellerin yakın değeri sahip olacak olmasıdır.

Yoğunluk görüntüleri için kural vektörünün minimum uzunluğunun sekiz olması gerekir. Önerilen yöntem gri tonlamalı görüntüler üzerinde uygulanmıştır. Ancak

yöntem herhangi bir renk formatı için kolaylıkla uyarlanabilir. Ayrıca piksellerin renk değeri tamsayı olmalıdır.

4.1. Motivasyon

Görüntü bölümleme, ayrıntıların azaltılması ve önemli olmayan özelliklerin ortadan kaldırılması ile görüntüleri yorumlamaya ve analizine yardımcı olur (Waseem, 2013). Görüntü bölümleme, bazı varsayımlara göre bir görüntünün bölmelere ayrıştırılmasını ele alır (Waseem, 2013; Wang ve ark., 2016; Zaitoun ve Aqelb, 2015). Bu varsayım komşu pikseller arasındaki değer farkı, global veya lokal eşik değeri veya piksellerin değer yakınlıklarına dayanabilir.

GMACA (Ganguly ve ark, 2002), her bitişik hücre ile etkileşime giren kafes düzlemindeki n sayıda hücrenin güncelleme mantığına dayanan daha genel bir Çoklu Cezbedicili Hücresel Otomata(MACA) sınıfıdır (Chaudhuri, 1997). GMACA, hücrelere uygulanan kural vektörüne bağlı olarak diğer durumları etrafında toplayan cezbedici durumlar üretirler. Cezbedicilerin diğer durumları etrafında toplaması görüntü bölümlendirme kapsamında ele alındığında bölge merkezi olarak düşünülebilir.

Hücresel otomatları görüntü işlemede kullanmanın birçok avantajı vardır. Rosin hücresel otomatların avantajlarını dört ana başlıkta toplamıştır (Rosin, 2010):

1. Her bir hücre etrafındaki komşu hücrelerle basit kurallar içeren etkileşime girer. Buna rağmen global olarak kompleks davranışlar sergileyebilir.
2. Uygulamasının kolay olması ve kompleks davranışlar sergileyebilmesi, CA'yı kompleks sistemlerin davranışlarını modellemede avantajlı hale getirmektedir.
3. CA paralel programlama için doğal ve basit bir yol sunar. Mantık devrelerindeki temel mantık kapıları VE/VEYA kullanılarak hesaplama performansı açısından verimli uygulamalar gerçekleştirilebilir.
4. CA genişletilebilir hesaplama yeteneğine sahiptir. Problemin yapısına göre kurallar eklenebilir, çıkartılabilir veya revize edilebilir.

Ayrıca CA lokal etkileşimli bir yapıya sahip olmasına nedeniyle geniş paralel hesaplama imkanı sunan Single Instruction Multiple Data (SIMD) bilgisayarlarda çalışacak şekilde tasarlanabilir (Kauffmann ve Piché, 2010).

CA; kenar algılama (Patel ve More, 2013), gürültü filtreleme (Popovici ve Popovici, 2002), inceltme (Rosin, 2010), görüntü morfolojisi (Ikenaga ve Ogura, 2000), bölümlendirme (Vilarino ve ark., 2003), görüntü iyileştirme (Rosin, 2010) ve görüntü sınıflandırılması (Bi ve ark., 2006) gibi farklı amaçlar için görüntü işleme alanında yaygın bir şekilde kullanılmaktadır. GMACA, ilişkisel bellek (Ganguly ve ark., 2002), görüntü sınıflandırma (Ganguly ve ark., 2002), VLSI devrelerde tasarım (Das, 2006) ve hata teşhisi (Sikdar ve ark., 2001) ve biyoenformatik (Sree ve ark., 2013) gibi çeşitli uygulamalar için kullanılmaktadır. GMACA'nın desen çekim gücü, görüntü işleme işlemleri için kullanılmamıştır.

GMACA'nın görüntü işleme görevleri için önemli bir potansiyele sahip olduğu bu tez çalışması kapsamında değerlendirilmektedir. Bu amaçla GMACA'nın görüntü bölümlendirmedeki potansiyeli araştırılmıştır ve tek uzunluk döngülü cezbediciler ile görüntü bölümlendirme gerçekleştirilmiştir.

4.2. Benzer Çalışmalar

CA kullanılarak yapılan görüntü işleme çalışmalarının tarihi 1970'lere (Deutsch, 1970; Triendl, 1970; Duff ve ark., 1973; Smith, 1971; Preston ve ark., 1979) kadar gitmektedir. CA kullanılarak yapılan erken görüntü işleme çalışmaları CA'nın paralel hesaplama gücünden faydalanmak için yapılmıştır.

Deutschs dörtgen, sekizgen ve üçgen komşuluk yapısına sahip iki boyutlu diziler kullanarak görüntüdeki nesneleri inceltme yapmıştır. Yapılan çalışmada en iyi sonucun sekiz komşuluğa sahip iki boyutlu diziler ile elde edilmiştir. Üç komşuluğa sahip iki boyutlu diziler gürültüye karşı hassas sonuçlar vermiştir. Ayrıca hesaplama maliyeti açısından üç komşuluğa sahip dizeler daha maliyetli olmuştur. (Deutsch, 1970).

Duff ve ark. 16x12 boyutlara sahip, 192 hücreli ve sekiz komşuya sahip hücrelerden oluşan iki boyutlu dizeler üzerinde CA tabanlı görüntü işlemci tasarlamıştır. İşlemci ikili görüntüler üzerinde çalışmaktadır (Duff ve ark., 1973). Triendl el ile yazılan sembollerin iskeletinin çıkarılmasında kullanılmak üzere paralel görüntü işlemciler üzerinde işlenebilen görüntü işleme yöntemi önermiştir (Triendl, 1970). Yapılan işlem Deutch'ın yaptığı çalışma gibi görüntü inceltme olarak değerlendirilebilir.

CA paralel hesaplama gücü kullanılarak yapılan ilk çalışmaların tümü ikili görüntüler üzerinde çalışmaktaydı. Bu nedenle görüntülerin işlenebilmesi için ilk önce ikili görüntülere dönüştürülmesi gerekirdi. CA ile yapılan erken dönen çalışmalarında dikkat çeken diğer bir durum; yapılan uygulamaların hem metotları ve donanımsal gerçekleştirilmesi beraber veriliyordu. Çoğu çalışmada, uygulamada oluşturulan görüntü işlemcilerinin komut setine yer verilmekteydi.

Daha önce yapılan CA tabanlı görüntü işleme çalışmalarının çoğu basit görüntü işleme uygulamaları için kullanılmıştır (Nayak ve ark.,2014). Bunlardan öne çıkanlar şöyledir: Kenar bulma (Rosin, 2006), gürültü giderme (Chandran ve Navina, 2012), inceltme (Deutsch, 1970), morfolojik işlemler (Ikenaga ve Ogura, 2000), görüntü bölümlendirme (Ghosh ve ark.,2011) ve görüntü iyileştirme (Rosin, 2010) uygulamalarıdır.

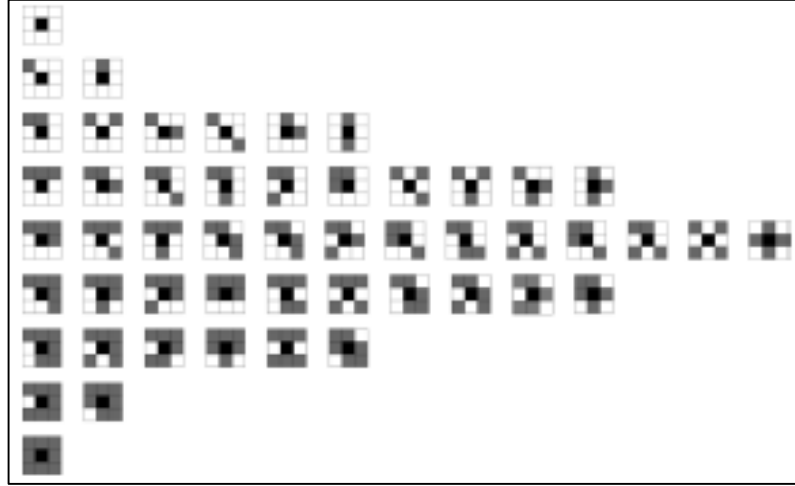
Ikenaga ve Ogura hücrel otomata mimarisi üzerine kurulmuş paralel hesaplama yapabilen morfolojik yöntemler önermişlerdir (Ikenaga ve Ogura, 2000). Geliştirdikleri paralel hesaplama yapan CA mimarisi hata silme, kenar bulma, ikili hale getirme, gürültü giderme ve boşluk doldurma görevlerini çeşitli morfolojik işlemler yaparak gerçekleştirmişlerdir.

Deterministik olmayan kurallara sahip CA tabanlı görüntü işleme çalışması (Popovici ve Popovici, 2002) tarafından önerilmiştir. Kenar bulma ve gürültü giderme için iki ayrı kural belirlemişlerdir. Belirledikleri kurallar CA'nın hesaplama mekanizmasına ait kural çıkarımından ziyade sezgisel kural özelliğine sahiptir. Hücre komşuluğu için Neumann komşuluğunu kullanmışlardır. Görüntülerdeki gürültüyü gidermek için belirledikleri sezgisel: Bir hücrenin bir sonraki durumu hücre komşuluklarının durumlarına göre belirlenir. Neumann komşuluğundan oluşan beş hücrede en fazla hangi değer varsa o değer orta hücrenin bir sonraki değeri olur. Popovic ve Popovic önerdikleri yöntemin gri seviye görüntüler için de çalışacağını belirtmişlerdir. Ancak gürültü giderme için belirledikleri sezgiselin renk sayısını fazla olduğu görüntülerde iyi sonuç vermeyeceği düşünülmektedir. Görüntülerdeki kenar algılama için belirledikleri sezgisel: Merkez hücrenin Neumann komşuluğunda yer alan tüm hücrelerden farkı alınır. Eğer fark belirlenen eşiğin üstünde ise hücre kenar hücresi olarak kabul edilir.

Wongthanavasas ve Tangvoraphonkchai medikal görüntüler üzerinde test ettikleri CA tabanlı görüntü işleme yöntemi geliştirmişlerdir (Wongthanavasas ve Tangvoraphonkchai, 2007). Yaptıkları çalışma (Popovici ve Popovici, 2002) benzemektedir. Yazarlar kenar algılama, gürültü giderme ve meme kanseri görüntülerinde kanserli bölge tespiti için kurallar belirlemişlerdir. Geliştirdikleri yöntemler ikili ve gri görüntüler üzerinde uygulamışlardır. Gri görüntüler içerisinde kenar algılama için görüntüleri histogram değerlerinden faydalananarak görüntü renklerini dört sınıfa ayırmışlardır. Görüntüdeki piksel değerleri sınıflara atandıktan sonra görüntü dört farklı renkle temsil edilir. Kenar algılama bu dört renk üzerinden yapılır. Merkez hücrenin Neumann komşuluğundaki hücreleri kontrol edilir, eğer farklı sınıfa ait değerler varsa bu hücre kenar hücresi olarak kabul edilir. Gürültü giderme işleme yapan diğer çalışmalarda (Popovici ve Popovici, 2002) olduğu gibi merkez hücre etrafındaki hücrelerde en fazla olan sınıfın değerini alır.

Rosin sadece ikili görüntüler üzerinde çalışan CA tabanlı görüntü işleme çalışması yapmıştır. Uygulanacak kural otomatik seçimi için deterministik bir yöntem geliştirmiştir (Rosin, 2006). İkili görüntüler üzerinde kullanılacak kuralları elde etmek için bir eğitim yöntemi araştırmıştır. Hücresel otomatların ikili görüntüler üzerinde uygulanacak olması bile büyük sayıda kural kümesinden amaca uygun kuralların seçimini zorunlu kılar. Böyle bir kural kümesini seçmek CA tabanlı görüntü işleme çalışmalarının en zor kısmıdır.

Bir pikselin bir sonraki durumu etrafındaki sekiz piksele göre belirlenir. Siyah beyaz görüntüleri için pikselin etrafındaki piksel sekizlik bir dize olarak ele alınır. Rosin başlangıç pikselini sol üst köşedeki piksel olarak belirlemiştir. Dize başlangıç pikselinden saat yönü doğrultusunda tüm piksellerin yan yana dizilmesiyle elde edilir. Başlangıç pikseli en az değere (least significant bit) sahip bit olarak değerlendirilir. Böyle bir diziliş sekiz bitlik bir değer oluşturur. Sekiz bitten oluşan ikilik tabandaki sayı 0 ile 255 arasındaki değerlere sahip olabilir. Bunun anlamı; ortadaki pikselin değerini belirlemek için 256 farklı ihtimal vardır. Rosin 256 farklı kural ihtimallerini simetri ve yansımaları aynı olan kuralları kaldırarak 51'e indirmiştir. Rosin'in basit bir sezgiselle elde ettiği 51 kural Şekil 4.1'de görülmektedir.



Şekil 4.1. Rosin'in simetri ve yansıyanı aynı olan kuralları çıkardıktan sonra elde ettiği 51 kural (Rosin, 2006).

Rosin gürültü giderme, inceltme ve dış bükey bulma görüntü işleme görevlerinde kullanılacak kuralları 51 kural içerisinden sıralı kayan ileri arama (sequential floating forward search) yöntemi ile elde etmiştir. İkili görüntüler üzerine uygulanan kurallar eşik bozunumu(threshold decomposition) ile gri görüntülere uygulanmıştır.

Choudhury ve diğ., tekdüze ve melez iki boyutlu hücresel otomatları görüntü işlemede kullanmışlardır (Choudhury ve ark., 2008). Yaptıkları görüntü işleme çalışmasında lineer kuralları kullanmışlardır. Bu kuralları, dikkate alınan hücreyi etkileyen komşu hücrelerin sayısına bağlı olarak dokuz gruba ayırmışlardır. Tekdüze CA'nın uygulanmasında görüntü içerisindeki nesnelerin kopyasının oluşturulduğu ortaya koymuşlardır. Melez kurallar ile görüntüdeki nesneleri büyültme, küçültme ve inceltme görevlerini gerçekleştirmişlerdir. CA ile yapılan görüntü işleme uygulamaları ikili görüntüler üzerinde gerçekleştirilmiştir.

Rosin yaptığı çalışmada (Rosin, 2010) hücresel otomatlar (CA) gürültü giderme ve özellik tespiti gibi çeşitli görüntü işleme görevlerine uygulanması anlatılmaktadır. Rosin önceki çalışmasında (Rosin, 2006) ikili görüntüler üzerine çalışmıştır. Bu çalışmasında yoğunluk görüntüleri üzerine araştırma yapmıştır. Hücre durumlarının artan sayısı (yani piksel yoğunlukları) muhtemel kuralların sayısının büyük ölçüde artmasına neden olur. Bu nedenle, daha pratik olan üç durumlu bir CA'yı değerleri temsil etmek için kullanmıştır. Buna ek olarak, CA eğitim aşamasındaki uygun kural setlerinin seçimini hızlandırmak için ardışık yüzen ileri arama(sequential floating

forward search) mekanizmasını optimize etmişlerdir. Önce yaptıkları çalışmadan daha iyi sonuç elde etmişlerdir. CA'nın pek çok görevi yerine getirecek şekilde eğitime tabi tutulabildiğini ve bu sonuçların kalitesinin, bilinen algoritmalarla karşılaştırılabilir veya daha iyi olduğunu göstermiştir.

Kauffman ve Piché, medikal görüntü işlemede Ford–Bellman algoritmasına dayalı çoklu en kısa yol ağacını hesaplamak için bir Hücresel Otomat (CA) kullanmışlardır (Kauffmann ve Piché, 2010). Yöntem verilen k tane merkez noktaya göre resmi k bölüme ayırır. K merkezlerinin en yakın komşularının bulunması hücresel otomatlar ile yapılmıştır. Çalışmanın gösterdiği önemli bir sonuç hücresel otomata yaklaşımlarının, paralel hesaplamaya uygun hale getirmek için bir araç olduğunu göstermesidir (Dioşan ve ark., 2017).

CA temelli birçok etkileşimli bölümlendirme yöntemi önerilmiştir. Önerilen yöntemlerde öne çıkan olgu; doğadaki etkileşimlerden esinleniyor olmasıdır. CA'nın çıkışındaki temel motivasyon; biyolojik canlıların kendini yenileme yeteneklerinin modellenmektir. Bu açıdan bakıldığında hücresel otomata mantığının görüntü işlemedeki birçok çalışmaya ilham verdiği söylenebilir.

GrowCut (Vezhnevets ve Konouchine, 2005), hücresel otomat fikrine dayalı interaktif bir görüntü bölümlendirme algoritmasıdır. Algoritma, rekabetçi bölge büyümesini andıran yineleyici bir etiketleme prosedürü kullanarak, orta ve zor nesnelerin 2D ve 3D'de güvenilir ve makul derecede hızlı bölünmesini sağlar. Kullanıcının etkileşimleri, piksellerin etiket alanlarına atanmasını sağlar. Komşu pikselin ağırlığı veya "gücü" ile ağırlığı belirlenen iki pikselin benzerlik ölçüsü, mevcut ağırlığını aştığı zaman, piksele komşusunun etiketi atanır. Etiket ataması da piksel ağırlığının güncellenmesine neden olur. Etiketleme prosedürü, piksel etiketlerinin modifikasyonu artık mümkün olmadığında kararlı bir yapılandırmaya ulaşılan kadar tekrar tekrar devam eder. Algoritma, görünür ön planda ve arka planda boyanmış vuruşların dışında, kullanıcıdan ek girdilere ihtiyaç duymaz (Egger ve ark., 2013). Ayrıca, kullanıcı, görüntüye ilave etiketler ekleyerek bölümlmeyi değiştirebilir, böylece bölümleme sonucunu etkiler.

Vezhnevets ve Konouchine yaptıkları çalışmayı açıklamak için biyolojik metafor kullanmışlardır. Piksel etiketleme sürecini, K tipi bakterilerin baskınlığı için büyüme

ve mücadelesi olarak düşünülebilir. Bakteriler tohum piksellerinden yayılmaya (büyümeye) başlar ve tüm görüntüyü kaplamaya çalışır. Bu nedenle de 'GrowCut' yöntemi adı verilmiştir. Bakteri üremesi ve rekabet kuralları açıktır - her ayrı zaman adımında, her hücre komşularına 'saldırmaya' çalışır. Eğer saldırı gücü savunucunun gücünden daha büyük ise - savunan hücrenin 'fethedilmesi' gerçekleştirir ve onun etiketi ve kuvveti değiştirilir. Bu yerel rekabetin sonucunda, en güçlü bakterilerin komşu alanları işgal etmesi ve kademeli olarak görüntü üzerine yayılması sağlanmış olur. GrowCut yönteminin ilgi çekici bir diğer yanı CA'nın ilham kaynağı olan biyolojik canlıların davranışlarından esinleniliyor olmasıdır. Neumann'nın CA'yı geliştirmesinin amacı biyolojik canlıların kendini yenileme yeteneklerinin mantıksal kurallarla modellenmesini sağlamak olduğunu, hatırlayalım.

“GrowCut” çalışmasına benzer bir çalışma Ghosh ve diğ., tarafından gerçekleştirilmiştir. Ghosh ve diğ., denetimsiz büyüme kesme algoritması (unsupervised grow-cut algorithm) ile etkileşim halinde bir görüntü bölümlendirme yöntemi önermişlerdir (Ghosh ve ark., 2011). Yapılan çalışmanı saf bir CA çalışması olduğunu söylemek zordur. Çünkü herhangi bir kural seçimi veya kural kümesi oluşturma kullanılmamıştır. Bunun yerine piksel değerleri durum geçiş kuralı olarak kullanılmıştır.

Gao ve Yang hücresel otomata temelli etkileşimli bir görüntü bölümlendirme yöntemi geliştirmişlerdir (Gao ve Yang, 2014). Kullanıcı; nesnenin veya arka planın olduğu kısımları işaretlemek suretiyle bölümlendirilecek bölgeyi belirler. Görüntü bölümlendirme işlemi hücresel otomata dönüşümlerinin hücrelere(piksel), bölgelere ve bloklara uygulanmasıyla gerçekleşir. Yöntem hücrelerin işaretli veya işaretli olma durumunu bakmamaktadır. CA tabanlı geliştirdikleri strateji, paralel uygulama esnekliği ve çok sınıflı problemlerle başa çıkma becerisi olduğunu belirtmişlerdir. Gao ve Yang önerdikleri yöntemin özelliklerini şöyle özetlemişlerdir (Gao ve Yang, 2014):

1. Orta derecede zor bölümlendirme görevlerini çözme becerisine sahiptir
2. Birden fazla boyuta sahip görüntüler üzerinde çalışabilir
3. Birden fazla etiketlenmiş görüntülerin bölümlendirilmesinde kullanılabilir. Hesaplama zamanı etiketlenen piksel sayısı ile doğru orantılı değildir.
4. Genişleyebilir yapıya sahiptir. Kendine has özelliklere sahip yeni algoritmalar oluşturma imkânı sağlar.

5. Etkileşim olanağı sunar. Bölümleme, her yineleme sonrası rafine edildiğinde kullanıcı dönüşümü izleyebilir ve 'anında' bölünmeyi hassaslaştırabilir.
6. Anlaması ve uygulaması kolaydır.
7. Hücresel otomat kullanıldığında hızlı paralel uygulama sağlanır

Hücresel otomatların farklı türleri görüntü işlemede kullanılmıştır. Safia ve diğ., sürekli hücresel otomatları(continuous cellular automata) kenar bulmada kullanmışlardır (Safia ve ark., 2011). Yaptıkları çalışmayı kenar tabanlı görüntü bölümlendirme olarak adlandırmışlardır. CA tabanlı uygulamalar için en büyük problem uygun kural kümesinin seçilmesidir. Kural sayısı kullanılacak piksel sayısı ile üstel bir ilişkiye sahiptir. Sürekli CA hücre sayısı ile üstel artan kural sayısı seçimine ihtiyaç bırakmıyor. Bunun yerine komşu piksellerin ortalamasının alınması veya piksel değerlerin birinden çıkarımı gibi işlemler hücrenin bir sonraki durumunu belirliyor.

Farklı bir hücresel otomata türü olan toplamli hücresel otomatlar(totalistic cellular automata) görüntü işlemede kullanılmıştır (Zhao ve ark., 2012). CA'nın bilim insanlarının ilgisini çekmesinde önemli bir yeri olan Hayat Oyunu (Gardner, 1970) toplamli hücresel otomata mantığına göre çalışmaktadır. Zhao ve ark. hücresel otomata kullanılarak gürültü giderme performansını artırmak için, basit toplamli CA (BTCA) modeli ve yeni ağırlıklı toplamli CA (WTCA) modeli önermişlerdir. Her iki yöntem de az bellek depolamaya ihtiyaç duyuyor ve pratikte çok büyük komşuluğa sahip uygulamalar için bile uygulanabilir olduğu belirtilmiştir. Önerilen yöntemlerin, düşük gürültü seviyeleri için medyan filtre ve geleneksel CA yönteminden daha iyi bir performans sergilediğini ve yüksek gürültülü seviyelerde filtreleme yapmak için WTCA modelinin diğer yöntemlerle karşılaştırıldığında mükemmel bir performans gösterdiğini göstermek için çeşitli sonuçlar sunulmuştur.

Diğer bir toplamli hücresel otomata tabanlı görüntü işleme çalışması Amrogowicz ve diğ., tarafından yapılmıştır (Amrogowicz ve ark., 2016). Yapılan çalışmada diğer geliştirilmiş CA tabanlı kenar bulma yöntemlerinin performansını ve önerilen yöntemin gürültü türleri ve seviyeleri açısından kapsamlı bir şekilde karşılaştırma yapıldır.

Dioşan ve diğ., çok durumlu hücresel otomatlar için genel bir kural önermişlerdir (Dioşan ve ark., 2017). Önerdikleri kural herhangi bir sayıda durum ve durumlar arasında anlamsal ilişki dikkate alınarak oluşturulmuş. Önerdikleri kuralı görüntü bölümlendirme için kullanmışlardır. Önerdikleri yöntemin kurallarını bazı gözlemlere dayandırmışlardır. Gözlemlerini şöyle özetlenebilir:

- Gri görüntülerde değerleri yakın olan piksellerin aynı bölgeye ait olması ve değerleri uzak olan piksellerin farklı bölgelere ait olması gerekir.
- Bir hücrenin durumu komşu hücrenin durumundan etkilenir.
- Bir hücre komşu hücrenin durumu alacaksa bu belirli şarlat altında mümkün olur.

Dioşan ve diğ., hücrelerin bir sonraki durumunu komşu hücrelerin komşularına göre belirlemişlerdir (Dioşan ve ark., 2017). Merkez hücresi M olarak adlandırılırsa, M hücresinin dört farklı komşuluk ilişkisi aşağıdaki gibi tanımlanmıştır:

- M 'nin komşu hücreleri arasında değeri en yakın olan hücrenin komşularından değeri M 'den az olan hücre.
- M 'nin komşu hücreleri arasında değeri en yakın olan hücrenin komşularından değeri M 'den fazla olan hücre.
- M 'nin komşu hücreleri arasında değeri en uzak olan hücrenin komşularından değeri M 'den az olan hücre.
- M 'nin komşu hücreleri arasında değeri en uzak olan hücrenin komşularından değeri M 'den fazla olan hücre.

Dioşan ve ark. yukarıdaki komşuluklara göre üç farklı sezgisel kural belirlemişlerdir; Yakın Çekim Kuralları(Close Attraction Rule (CAR)), Uzak Çekim Kuralı(Far Attraction Rule (FAR)) ve İnce Ayarlanmış Çekim Kuralları (FTAR). Önerilen birinci kuralın başlıca sezgiseli; belirli bir hücrenin komşularındaki görelî gri tonlama değerlerine göre kendi durumunu değiştirmesidir. FAR kuralı, hücrenin komşulardan değer almasını sağlar. Her bir hücre, en yakın komşularından değerler almak yerine, değeri daha da uzak olan değerleri alarak daha hızlı ilerlemeyi hedeflemektedir. Son kural olan FTAR kurallı eşik değerlerine bağlı ayarlamalarla bölümlendirme başarısını artırmayı hedeflemiştir.

Tezin bu bölümdeki çalışmada, Genelleştirilmiş Çok Cezbedici Hücresel Otomatları(Generalized Multiple Attractor Cellular Automata(GMACA)) görüntü bölümlendirmede kullanılmıştır. GMACA, kural vektörü kullanılarak gir görüntü verilerine uygulanmıştır. Kural vektörü melez bir yapıya sahiptir ve lineer olmayan kurallar içerebilir. Kural vektörünün seçimini Das ve diğ., örüntü sınıflandırma için yaptıkları çalışmalarında (Das ve ark., 2009) önerdiği yöntemlere göre elde edilmiştir.

Geliştirilen yöntemi daha önceki çalışmalardan ayıran birkaç önemli özellik vardır. Öncelikle bu çalışmada GMACA'yı görüntü bölümlendirmede ilk defa kullanılmıştır. GMACA çekim merkezi üretme yeteneği genellikle örüntü tanıma, örüntü sınıflandırma, VLSI devrelerde tasarım vb. ileri seviye problemlerin çözümünde kullanılmıştır. GMACA'nın görüntü bölümlendirmede kullanılması, görüntü işleme açısından önemli bir kazanım olacaktır. İkincisi, daha önceki çalışmaların tamamı CA'nın yerel etkileşimini kullanarak görüntü işleme yapmışlardır. Bu çalışmada CA'nın yerel etkileşimle oluşturduğu global davranışlara göre görüntü bölümlendirme yapılmıştır. Bu açıdan da bir ilktir. Üçüncüsü, kullandığımız kural seçim yöntemi daha önce görüntü işlemede kullanılan kural seçim yöntemlerinden farklıdır. Kullanılan kural seçim yönteminin zaman karmaşıklığı $O(n)$ ve hafıza karmaşıklığı $O(\lg n)$ dir. Kural seçim yönteminin daha önceki CA görüntü işleme çalışmalarında sıklıkla yapılan kenar bulma, gürültü giderme, inceltme, iyileştirme ve morfolojik işlemler için kullanılabileceği değerlendirilmektedir.

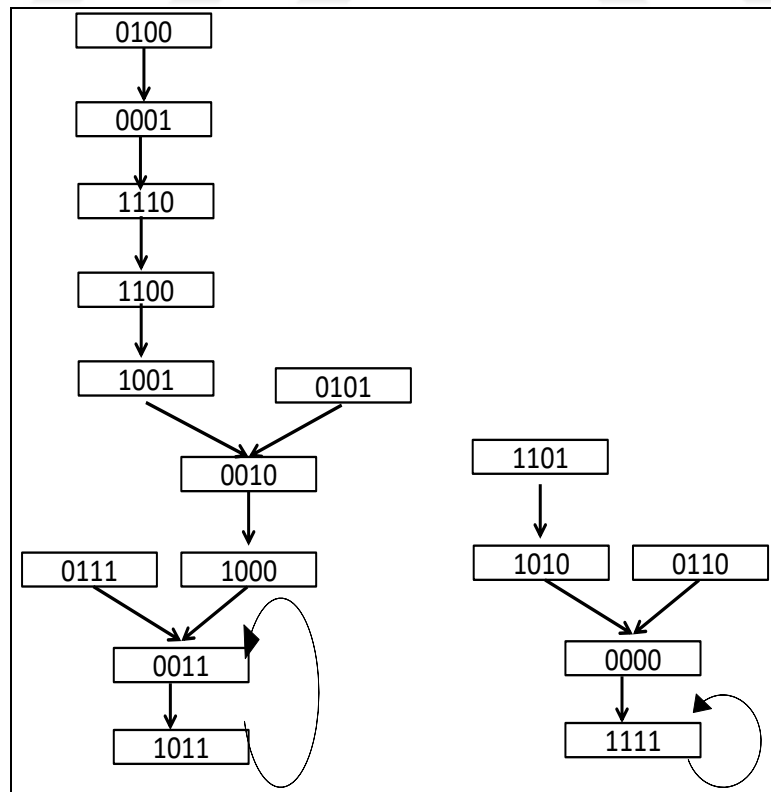
4.3. Cezbedici Kap Üreten Çoklu Cezbedici Hücresel Otomatların Tasarımının Yapılması

Daha önce de belirtildiği gibi, CA kuralları kural vektörü kullanılarak birden fazla hücreye uygulanmaktadır. Birçok araştırmacı, birden fazla hücre üzerinde kural vektörünün uygulanmasını araştırmıştır (Das ve ark., 2009; Das S. , 2006; Chaudhuri, 1997). Bazı durumlar diğer durumları cezbetmektedir. Bir n-hücreli hücresel düzlemi, ikili değerler içeren n-bitlik bir model olarak ele alınabilir. Bazı durumların başka bir durum tarafından çekilmesi, bazı örüntülerin başka bir örüntü tarafından çekilmesi anlamına gelir (Ganguly ve ark., 2002).

Çoklu cezbedici hücresel otomatlar kural vektörünün yapısına göre iki türe ayrılır; Çoklu Cezbedici Hücresel Otomatlar (MACA) ve Genelleştirilmiş Çoklu Cezbedici

Hücresel Otomatlar (GMACA). MACA'lar yalnızca XOR ve XNOR mantıksal operatörlerine göre işlem yapan kurallardan ibaretken, GMACA XOR ve XNOR mantıksal operatörlerine göre işlem yapmayan kurallardan oluşabilir. XOR ve XNOR mantıksal operatörlerine göre işlem yapan kurallar toplamsal kural olarak adlandırılır. Toplamsal kurallar, MACA'nın kural nitelemesinde, cebirsel işlemler yapılmasına olanak sağlar (Chaudhuri, 1997). Cebirsel işlemler, toplamsal olmayan kurallar içeren GMACA hücrelerinde yapılamaz.

GMACA'lar kural vektörünün yapısına göre farklı geçiş diyagramlarına sahip olurlar. MACA geçiş diyagramlarında bazı durumlar diğer durumları çekerler ve bu tip diğer durumları çeken durumlar cezbedici(attractor) durum olarak adlandırılır. Ters döndürülemez hücresel otomatlar olarak adlandırılan GMACA'lar diğer durumları çeken cezbedici durumun bir sonraki durumda kendine veya başka bir duruma dönmesine göre iki ayrılırlar; Tek Uzunluklu Döngülü Cezbedici(Single Length Cycle Attractor(SLCA)) ve Çok Uzunluklu Döngülü Cezbedici(Multi Length Cycle Attractor(MLCA)). Şekil 4.2'de iki fark cezbedici üreten CA <105, 129, 171, 65> kural vektörünün durum geçiş diyagramı görünmektedir.



Şekil 4.2. CA <105, 129, 171, 65> kural vektörünün ürettiği durum geçiş diyagramları

Yukardaki şekilde durumların değerleri ikili tabanda verilmiştir. Durumların ikilik tabandaki sayısal karşılığı dört basamaktan oluşmaktadır. Bu nedenle bir sonraki durumun elde edilmesi için dört uzunluklu bir kural vektörüne ihtiyaç duyulur. Uygulanan kural vektörü 105, 129, 171 ve 65 olmak üzere dört kuraldan oluşmaktadır. Kural vektörüne göre bir sonraki durumların nasıl elde edildiğini birkaç örnek vererek inceleyelim. Kuralların ikili değerleri ve KMT karşılıkları Tablo 2.'de verilmiştir. Mevcut durum 0100(4) olsun, sıfır sınırlı CA olduğu için en soldaki hücreden an sağdaki hücreye kadar bir sonraki durumu belirlemek için bakmamız gereken komşuluklar sırasıyla 001, 010, 100 ve 000'dir. Bu dört komşuluk için uygulanacak kurallar kural vektöründeki sıraya göre olacak. 001 komşuluğu için 105 kuralının KMT1 değerine bakmamız gerekir. 105 kuralının KMT1 değeri 0'dır. 010, 100 ve 000 komşulukları için 129, 171 ve 65 kurallarına karşılık gelen KMT2, KMT4 ve KMT0 değerleri sırasıyla 0, 0 ve 1'dir. Yani 0100(4) durumunun bir sonraki durumu 0001(1) durumudur. 0011 durumu için bir sonraki durumu bulalım. 0011 durumu için kural vektöründeki kuralların soldan sağa olacak şekilde KMT0, KMT1, KMT3 ve KMT6 değerleri 1, 0, 1 ve 1'dir. 0011(3) durumunun bir sonraki durumu $CA < 105, 129, 171, 65 >$ kural vektörüne göre 1011(11) dir. Son olarak 1011 durumunu ele alalım. 1011 durumu için kural vektöründeki kuralların soldan sağa olacak şekilde KMT2, KMT5, KMT3 ve KMT6 değerleri 0, 0, 1 ve 1'dir. 1011(11) durumunun bir sonraki durumu $CA < 105, 129, 171, 65 >$ kural vektörüne göre 0011(3) tür.

Tablo 4.1. $CA < 105, 129, 171, 65 >$ kural vektörünün KMT'ye göre bir sonraki durum değerleri

KMT	111	110	101	100	011	010	001	000
	7	6	5	4	3	2	1	0
105	0	1	1	0	1	0	0	1
129	1	0	0	0	0	0	0	1
171	1	0	1	0	1	0	1	1
65	0	1	0	0	0	0	0	1

GMACA'lar örüntü tanıma, veri sıkıştırma ve sınıflandırma gibi alanlarda kullanılmaktadır. GMACA'nın istenilen uygulamaya göre tasarımı kullanılan kural vektörünün seçimine göre olur. Kural vektörleri cezbedici oluşumuna yaptıkları katkıya göre dokuz gruba ayrılmışlardır (Naskar, 2009).

Özellik 1: Bir kuralın KMT0, KMT1, KMT4 ve KMT5 değerlerinin kuraldaki karşılığı 0 ve KMT2, KMT3, KMT6 ve KMT7 değerlerinin kuraldaki karşılığı 1 olmasına göre tek uzunluklu cezbedici oluşuma katkı sağlarlar (Das ve ark., 2009).

Tablo 4.1.'de yer verilen 204 kuralı özellik biri tüm KMT'lerinde karşılarken 51 kuralı hiçbir KMT'sinde karşılamamaktadır. 204 kuralı özellik 1'i tüm KMT'lerinde karşıladığı için grup8'de ve 51 kuralı özellik 1'i hiçbir KMT'sinde karşılamadığı için grup0'da yer almaktadır. Tablo 2.'de yer verilen 105 ve 171 kurallarını inceleyelim. 105 kuralı özellik 1'i KMT1, KMT3, KMT4 ve KMT6'da karşıladığı için özellik 1'i 4 KMT'de karşılıyor demektir, bu nedenle grup4'de yer alır. 171 kuralı özellik 1'i KMT3, KMT4 ve KMT7'de karşıladığı için özellik 1'i 3 KMT'de karşılıyor demektir, bu nedenle grup3'de yer alır.

Das ve diğ., tek uzunluk döngülü cezbedici elde etmek için bir algoritma tasarlamışlardır. Tasarladıkları algoritmaya göre kural sayısı ve cezbedici sayısı algoritmaya girdi olarak verilir. Algoritma çıktı olarak isten uzunluktaki kural sayısına sahip kural vektörünü üretir. Üretilen kural vektörü istenen sayıda cezbedici üretir. Aşağıdaki şekilde Das ve diğ., tarafından geliştirilen algoritma yer almaktadır.

Girdi: n (Kural vektörü uzunluğu) ve K (Kapsamlı bit sayısı)
Çıktı: Kural Vektörü
Adım 1: Kapsamlı bit yeteneğine sahip K tane biti rasgele belirle
Adım 2: S_0 ve S_1 iki KMT kümesi olsun. S_0 KMT'leri 0 ve S_1 KMT'leri 1'dir. $S_0 = \{0, 1, 4, 5\}$, $S_1 = \{2, 3, 6, 7\}$
Adım 3:
e) Eğer birinci ve ikinci bit kapsayıcı bit ise:
 K_1 ve K_2 kurallarına ait tüm KMT'ler özellik 1'i karşılar
f) Eğer birinci bit kapsayıcı bit ve ikinci bit kapsayıcı bit değilse:
 K_1 kuralına ait KMT'lerden S_0 kümesine ve S_1 kümesine ait KMT'lerden en az birer tanesi özellik 1'i karşılar. K_1 kuralının özellik 1'i karşılayan KMT'lerinden türetilen ardışık KMT'ler K_2 kuralında özellik 1'i karşılar. Eğer K_1 kuralının diğer KMT'leri özellik 1'i karşılamıyorsa K_2 kuralına ait KMT'lerden K_1 kuralı KMT'leri aynı kümeye ait (S_0 ve S_1) ise özellik 1'i karşılamaz. Eğer K_1 kuralının diğer KMT'leri özellik 1'i karşılamıyorsa K_2 kuralına ait KMT'lerden K_1 kuralı KMT'leri farklı kümeye ait (S_0 ve S_1) ise özellik 1'i karşılar.
g) Eğer ikinci bit kapsayıcı bit ve birinci bit kapsayıcı bit değilse:
 K_1 kuralının S_0 veya S_1 kümelerinden sadece birine ait KMT'leri özellik 1'i karşılar diğer kümedeki KMT'ler özellik 1'i karşılamaz. K_2 kuralının K_1 kuralında özellik 1'i karşılayan KMT'lerinden türetilen ardışık KMT'ler özellik 1'i karşılar, diğer KMT'ler rasgele seçilir.
h) Eğer birinci ve ikinci bit kapsayıcı bit değillerse:
 K_1 kuralında, S_0 kümesinden sadece bir KMT özellik 1'i karşılar ve S_1 kümesindeki KMT'ler özellik 1'i karşılamazlar; veya S_1 kümesinden sadece bir KMT özellik 1'i karşılar ve S_0 kümesindeki KMT'ler özellik 1'i karşılamazlar. K_2 kuralında, K_1 kuralında özellik 1'i karşılayan KMT'den türetilen ardışık KMT'ler özellik 1'i karşılar. K_2 kuralında, K_1 kuralında özellik 1'i karşılamayan KMT'den türetilen ardışık KMT'lerden aynı kümede olanlar özellik 1'i karşılamaz, farklı kümede olanlar özellik 1'i karşılar.
Adım 4: For $i=3$ 'den n 'e
e) Eğer i 'ninci ve $(i+1)$ 'inci bitler kapsayıcı bit ise:
 K_i kuralının 0, 1, 2, 3 KMT'leri veya 4, 5, 6, 7 KMT'leri veya tüm sekiz KMT'si özellik 1'i karşılar.
f) Eğer i 'ninci bit kapsayıcı bit ve $(i+1)$ 'inci bit kapsayıcı bit değilse:
 K_i kuralının 0, 1, 2, 3 KMT'leri veya 4, 5, 6, 7 KMT'leri özellik 1'i karşılar. Diğer KMT'ler rasgele seçilir.
g) Eğer i 'ninci bit kapsayıcı bit değil ve $(i+1)$ 'inci bit kapsayıcı bit ise:
 K_i kuralının S_0 kümesine ait KMT'leri özellik 1'i karşılar ve S_1 kümesine ait KMT'leri özellik 1'i karşılamaz; veya S_1 kümesine ait KMT'leri özellik 1'i karşılar ve S_0 kümesine ait KMT'leri özellik 1'i karşılamaz
h) Eğer i 'ninci ve $(i+1)$ 'inci bitler kapsayıcı bit değiller ise:
 K_i kuralının S_0 kümesine ait KMT'leri özellik 1'i karşılamaz ve S_1 kümesine ait KMT'lerden yalnızca iki denk KMT özellik 1'i karşılar; veya S_1 kümesine ait KMT'leri özellik 1'i karşılamaz ve S_0 kümesine ait KMT'lerden yalnızca iki denk KMT özellik 1'i karşılar
Adım 5: K tane kapsamlı bite sahip kural vektörünü rapor et.

Şekil 4.3. Algoritma 4: Çok uzunluk döngülü cezbedici oluşumuna izin vermeden, istenen sayıda kapsamlı bit üreten kural vektörünü çıktı olarak veren çoklu cezbedici hücresel otomata sentez algoritması (Das ve ark., 2009)

Bu alt bölümde GMACA hakkında kısaca bilgiler verilmiştir. GMACA kural vektörünün nitelenmesi hakkında detaylı bilgiler üçüncü bölümde yer almaktadır. İlgili bölümde erişilebilirlik ağacı tabanlı kural niteleme yöntemine ait tüm teorik alt yapı yer almaktadır.

4.4. Görüntü Bölümleme

Görüntü işleminde temel hedeflerden biri, görüntüden istenen özelliklerin elde edilmesidir. İstenen özellikler elde edilirken diğer özelliklerin etkisinden mümkün olduğu kadar uzak durmaya çalışılır. Bunu için en bilinen yöntemler gürültü giderme, iyileştirme ve görüntü bölümlendirme yöntemleridir. Görüntüyü alt görüntüler şeklinde ele almak için görüntü bölümleme temel işlemlerden biridir. Her bir ayrıştırılan bölüt renk, doku ve yoğunluk gibi farklı bilgiler içerebilir (Waseem, 2013). Görüntü bölümlenmede hedef tüm görüntü olabildiği gibi görüntünün bir kısmı da olabilir. Görüntünün tümü için yapılan bir bölümlenmede görüntü istenen sayıda farklı karakteristiğe sahip bölgelere ayrılır. Kısmi bölümlenmede ise belirli bir bölge içerisindeki alakasız bölümler çıkarılır (Sanko ve ark., 2008).

Görüntü bölümleme için birçok yöntem ve algoritma önerilmiştir. Ancak hangi yöntem veya algoritmanın iyi olduğuna karar vermek zor bir görevdir. Farklı görüntü bölümlenme yöntemlerinin ortaya koyduğu görüntü bölümlenme sonuçlarının bir uzman kişi tarafından yapılmasına sübjektif değerlendirme denir ve en çok bilinen görüntü bölümlenme değerlendirme yaklaşımıdır. Yaklaşım önceden tanımlanmış belirli görüntüler üzerinde sınırlı bir değerlendirme sağlar. Gözetimli değerlendirme görüntü bölümlenme performansı için kullanılan diğer bir ölçüm yaklaşımıdır. Bu yaklaşımda önerilen görüntü bölümlenme sonuçları ile önceden manuel olarak bölümlenen sonuçlar karşılaştırılır (Zhang ve ark., 2008).

Görüntü bölümleme yaklaşımları üç ana kategoriye ayrılır; gözetimli bölümleme, gözetimsiz bölümleme ve yarı gözetimli bölümleme. Gözetimli bölümlemede eğitim görüntülerinde eğitilen bölümleme modeli test görüntüleri üzerinde başarıları test edilir. Gözetimsiz bölümlemede görüntüler daha önceki bir deneyime veya eğitim sürecine dayanmadan bölümlenir. Genelde görüntünün histogramına göre bir strateji belirlenir. Yarı gözetimli bölümlemede görüntüler bir kişi tarafından bölümlenecek bölgenin işaretlenmesine esas olarak bölümleme gerçekleştirirler (Artan, 2013).

Bu çalışmada görüntü bölümleme gözetimsiz bölümleme kategorisi altında değerlendirilebilir. Görüntü bölümlemede bölgeler arasındaki ayrışımı belirginleştirmek için her bir bölgeye farklı renkler atanır. Bu şekilde bir yandan görüntünün insanlar tarafından analizi kolaylaştırılırken diğer yandan bilgisayar

tarafından yapılacak görüntü işleme yöntemlerine ve görüntü analizine yardımcı olunmaktadır.

Her görüntünün kendi karakteristiği olduğu için tüm görüntü türleri için en iyi sonucu veren bölümlleme yöntemi yoktur. Hangi bölümlleme yönteminin özel bir görüntü için en iyi sonucu vereceğini bulmakta zor bir görevdir. Bir görüntüde iyi sonuç veren bir yöntem başka bir görüntüde istenen sonuçları vermeyebilir. Kenar tabanlı bölümlleme (Brejl ve Sonka, 2000), bulanık mantık tabanlı bölümlleme (Saha ve ark., 2000), kısmi diferansiyel denklem tabanlı bölümlleme (Chung ve Sapiro, 2000), yapay sinir ağları tabanlı bölümlleme (Dong ve Xie, 2005), eşik değeri tabanlı bölümlleme ve bölge tabanlı bölümlleme en bilinen bölümlleme yöntemleridir (Waseem, 2013).

4.5. Cezbedici Kapların Görüntü Bölümllemede Kullanımı

Görüntü bölümlleme, bir görüntünün farklı gruplara sınıflandırılması işlemidir (Dhanachandra ve ark., 2015). K tane cezbedici durumları olan bir n-hücreli CA, k tane sınıf sınıflandırıcı olarak görülebilir (Das ve ark., 2009). Kural vektörü tarafından üretilen her cezbedici durum, görüntü parçalanması için kullanılabilir. Bunu başarmak için, her cezbedici durum bir bölgenin temsilcisi olarak değerlendirilebilir. Bu basit fikir uygun kural vektörleri seçilerek görüntülerde kolayca uygulanabilir. İkinci bölümde belirtildiği gibi arzu edilen cezbedici sayısını elde etmek için kural vektöründeki kuralların seçimi kilit öneme sahiptir.

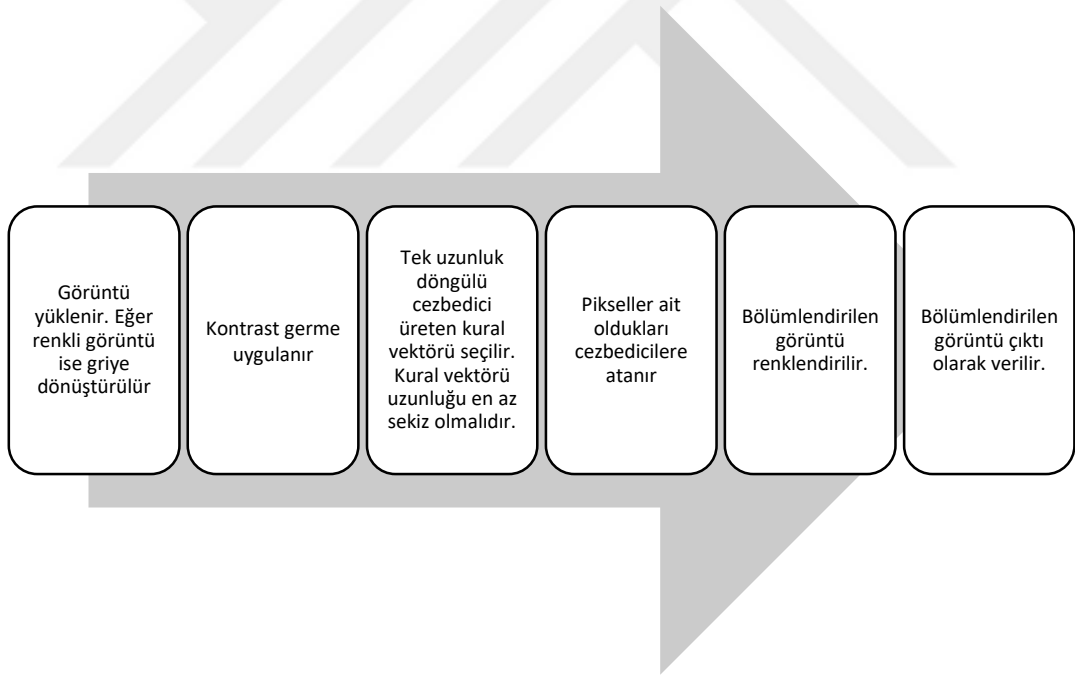
Önerilen görüntü bölümlleme yönteminde tek uzunluk döngülü cezbedici türü kullanılmıştır. Tek uzunluk döngülü cezbedici elde etmek için, kural vektöründeki tüm kurallar 4'üncü, 5'inci, 6'ıncı, 7'inci ve 8'inci gruplardan seçilmiştir.

Cezbediciler kullanılarak yapılacak görüntü bölümllemede dikkat edilecek bir konu; durum sayısını belirleyen kural vektörünün uzunluğudur. Bu uygulama bağlamında, durum değeri piksel değerini ifade eder. Test için gri yoğunluk görüntüleri kullanılmıştır. Test görüntüsünün piksel değeri 0'dan 255'e kadar değişebilir. Değerler 0'dan 255'e kadar olabileceği için, kural vektörü en az 8 kuraldan oluşmalıdır. Kural vektöründeki kuralların sayısı sekiz ise, durum değeri için sekiz bit olabileceği anlamına gelir. Sekiz bitlik ikili sayılar 0'dan 255'e değişebilir.

Önerilen görüntü bölümle yöntemi kontrast germe, kural vektörü tarafından oluşturulan cezbedici durum değerlerini piksel değerlerine atama işlemlerinden oluşur. Kontrast germe görüntü bölümleme kalitesini artırır (Dhanachandra ve ark., 2015). Gri görüntünün her piksel değeri, seçilen kural vektörü tarafından üretilen cezbedici durum değeri olarak değiştirilir. GMACA tabanlı görüntü bölümleme yönteminin algoritma adımları aşağıda listelenmiştir:

1. Görüntü yüklenir ve renkli görüntü ise griye dönüştürülür
2. Kontrast gerdirme uygulanır
3. Tek uzunluklu döngülü cezbedici üreten bir kural vektörü seçilir. Kural boyutu en az sekiz olmalıdır.
4. Görüntü pikseline karşılık gelen cezbedici durumun değeri atanır.
5. Bölümlendirilmiş görüntü renklendirilir

Şekil 4.4, GMACA tabanlı görüntü bölümlemesinin iş akışını göstermektedir



Şekil 4.4. GMACA tabanlı görüntü bölümlemenin iş akış diyagramı

Aşağıdaki şekilde geliştirilen yönteme ait sözde kod yer almaktadır. Sözde kod uygulama adımlarını, programlama açısından daha açıklayıcıdır. Her bir adımda yapılan işlemler sözde kod içersinde yer almaktadır.

```

1. Input: Gray scale image
2. Output: Color Segmented Image
3. Identify the rule vector. Minimum size of the rule vector should be eight and
   attractor
4. basin generated by rule vector may vary 0 to 255
5. For i=0 to height of image
6.     For j=0 to width of image
7. Image(i , j)=GMACA(Image(i , j)) Find the attractor basin of each pixel
   according
8. to the rule vector
9.     For End
10. For End
11. If it is required merge closest attractor basin pairs
12. Color the each attractor basin to different color
13. Output the segmented image
14. Function GMACA(currentState)
15.     Convert currentState to binary number of n digit
16.     Convert currentState to string
17.     do
18.         Add zero to both left and right of currentState because of null boundary
19.         CA
20.         currentState=0+ currentState +0
21.         nextState= Load currentState to the rule vector in order to find next
           state.
22.         while currentState==nextState
23.             Convert nextState to int
24.             Return nextState
25. Function End

```

Şekil 4.5. GMACA tabanlı görüntü bölümlendirmeye ait sözde kod

Yukarıdaki sözde kodu açıklamak faydalı olacaktır. Sözde kod girdi olarak gri görüntü alır. Çıktı olarak bölümlendirilmiş görüntü verir. Öncelikle, görüntülere uygulanacak kural vektörünü 3 numaralı satırda belirtildiği gibi seçilir. Kural vektörünün seçimi elde edilecek cezbediciler için çok önemlidir. Elde edilecek cezbedicilere göre kural seçimi yapılır.

Cezbedici kapların nasıl elde edildiği, ikinci bölümde detaylıca açıklanmıştır. İlgili bölümü, tekrardan gözden geçirmek faydalı olabilir. Tek uzunluk döngülü cezbedicilerin elde edilmesini Das ve diğ., yaptıkları çalışmada önerdikleri algoritmaya göre elde edilmiştir. Algoritma Şekil 4.3'te yer almaktadır. Algoritma istenen sayıda cezbedici üreten n uzunlukta kural vektörünü çıktı olarak vermektedir. Daha detaylı bilgiler orijinal çalışmada (Das ve ark., 2009) yer almaktadır.

5'inci ve 10'uncu satırlar arasında görüntüdeki pikseller üzerinde gezinmek için iç içe döngüler oluşturulmuştur. Görüntüdeki her bir değer kural vektörüne yüklenmekte ve ait olduğu cezbedici duruma atanmaktadır. Piksel değerlerinin kural vektörüne yüklenmesi 14'üncü satırda yer alan GMACA başlıklı fonksiyon ile gerçekleştirilir.

GMACA fonksiyonu parametre olarak piksel değerini alır. Parametre ismi olarak "currentState" kullanılmıştır. GMACA'lar iki değerler üzerinde işlem yapabilen hesaplama mekanizmalarıdır. Bunun için onluk tabanda değer tutan "currentState" değişkenini ikili tabana dönüştürülür. İkilik tabana dönüştürürken dikkat edilmesi gereken bir durum; ikilik sayının basamak sayısıdır. İkilik dönüşümü yapılan sayının en az kural vektörünün uzunluğuna sahip olması gerekir. Örneğin, sekiz uzunluğa sahip bir kural vektörüne sahip ise 24 sayısının ikilik tabandaki gösterimi şöyle yapılır; $24 = 00011000$. Normalde 24 sayısını ikilik tabanda ifade etmek için beş bit yeterli iken kural vektörü uzunluğu sekiz olduğu için ikilik sayının soluna üç sıfır eklenir.

Elde edilen ikilik tabandaki sayı üzerinde gezinmek için sayıyı dizge haline getirilir. 17'inci ve 21'inci satırlar arasındaki döngüde eldeki mevcut durumun ait olduğu cezbedici durum bulunur. Bunun için mevcut durum kural vektörüne yüklenir bir sonraki durum bulunur. Bir sonraki durum mevcut duruma atanır. Daha sonra mevcut durum tekrardan kural vektörüne yüklenir ve bir sonraki durum elde edilir. Bir sonraki durumu bulma işlemi mevcut durumla bir sonraki durum birbirine eşit oluncaya kadar devam eder. Eğer mevcut durumla bir sonraki durum birbirine eşitse cezbedici duruma ulaşılmış demektir. Döngünün dışındaki nextState değişkeni cezbedici durumu tutmaktadır. Son olarak nextState 22'inci satırda tam sayı değerine dönüştürülür ve fonksiyon bu değere döner. 20'inci satırda yer alan bir sonraki durumu elde etmek için kural vektörünün kullanılması üçüncü bölümde anlatılmıştır. Üçüncü bölümde, ilgili örnekler incelenerek, bir sonraki durumun nasıl elde edildiği anlaşılabılır.

Görüntüdeki tüm piksel değerleri cezbedici durumlara atandıktan sonra 11'inci satıra geçilir. On birinci satırda, istendiği takdirde bölüm sayısının azaltılabileceği belirtilmektedir. Bölüm sayısını azaltılması en yakın iki cezbedicinin birleştirilmesiyle gerçekleştirilir. Son olarak bölümlendirilen görüntü sahip olduğu cezbedici sayısına göre renklendirilir.

4.5.1. Kullanılan veri seti

Görüntü bölümlendirme için Berkeley Bölümlendirme Veri seti(Berkeley Segmentation Dataset) kullanılmıştır. Veri seti, Martin ve arkadaşları tarafından görüntü bölümlendirme ve kenar algılama araştırmalarına deneysel çalışma imkânı sağlamak için oluşturulmuştur (Martin ve ark., 2001). Veri setinde 300 tane görüntü renkli ve gri tonlarda yer almaktadır. Görüntülerin 200 tanesi eğitim amaçlı ve 100 tanesi test amaçlı olarak ayrılmıştır. Veri setinde yer alan görüntüler aşağıdaki Şekil 4.6, Şekil 4.7 ve Şekil 4.8’de küçük ebatlarda verilmiştir.

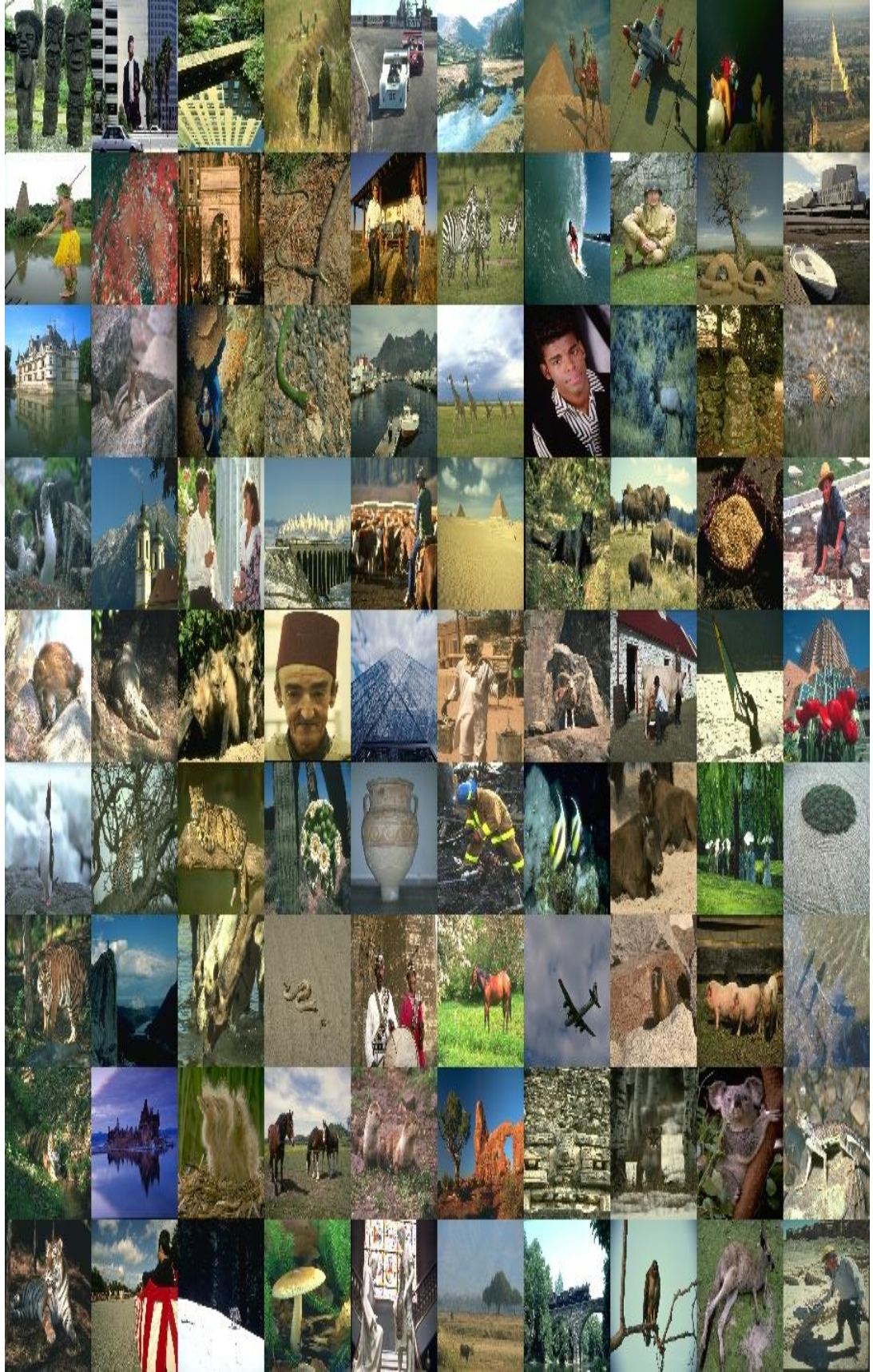




Şekil 4.6. Berkeley Bölümlendirme Veri Setinde yer alan 0-100 arası eğitim görüntüleri.



Şekil 4.7. Berkeley Bölümlendirme Veri Setinde yer alan 100-200 arası eğitim görüntüleri.



Şekil 4.8. Berkeley Bölümlendirme Veri setinden yer alan 200-300 arası test görüntüleri

Veri setinin oluşturulmasını Berkeley üniversitesinde okuyan öğrencilerin görüntüleri bölümlendirmesi suretiyle gerçekleştirilmiştir. 19-23 yaşları arasındaki 30 öğrenci 8 ay boyunca görüntülerin bölümlendirilmesini gerçekleştirmiştir.

Veri setindeki görüntülere dikkat edildiğinde görüntülerin çoğunun dış mekân ve doğal ortamlardan alındığı fark ediliyor. Görüntülerdeki renkler şekiller birbirinden çok farklıdır. Ayrıca görüntülerin içerisindeki çizgiler ve nesneler belirgin değildir. Veri setinde yer alan görüntüler bölümlendirme açısından zor bir karakteristiğe sahiptir. Görüntülerdeki bazı nesneler insan tarafından bile zor algılanmaktadır.

Tezin bu bölümündeki çalışmada, eğitimsiz ve insan etkileşimi olmayan bir görüntü bölümlendirme işlemi yapılmıştır. Elde edilen sonuçlar, denetimsiz bölümlendirme yaklaşımları açısından bakıldığında iyi olduğu değerlendirilmektedir.

4.5.2. Deneysel sonuçlar

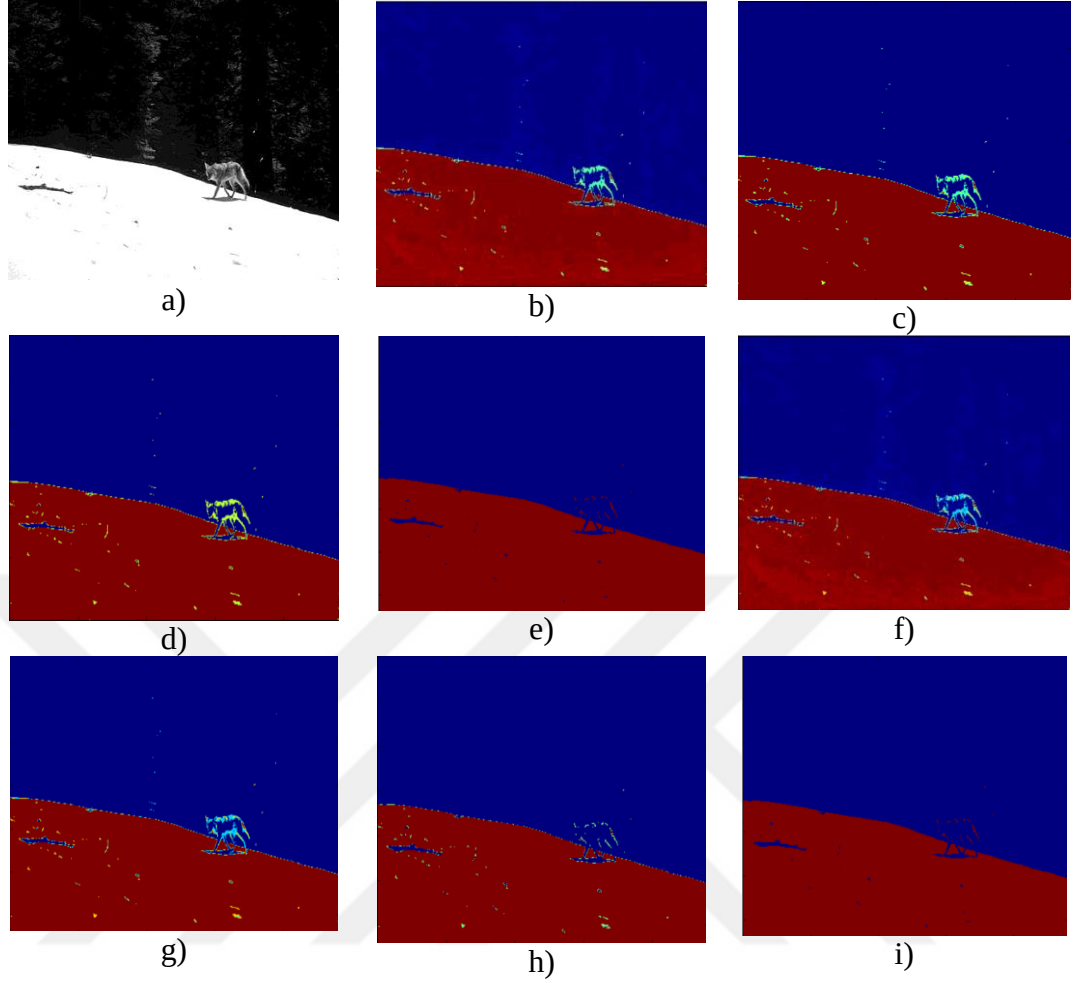
Görüntü bölümlendirme işlemi kural vektörü kullanılarak gerçekleştirilmiştir. Oluşturulan kural vektörü tek uzunluk döngülü cezbediciler üretmektedir. Görüntüler kural vektörünün ürettiği cezbedicilere göre bölümlendirilmiştir. Kural vektörünün ürettiği cezbediciler kademeli olarak yarı oranda azaltılmıştır. Azaltma işlemi en yakın cezbedicilerin birleştirilmesi yolu ile gerçekleştirilmiştir. Aşağıdaki şekillerde orijinal görüntünün gri hali ve cezbedici sayısına göre bölümlendirilmesi verilmiştir.

Görüntüleri farklı kural vektörleri kullanarak bölümlendirilmiştir. Bölümlendirilen görüntüler Şekil 4.9 ve Şekil 4.10'da verilmiştir.

Şekil 4.9 ve Şekil 4.10'daki görüntüler The Berkeley Segmentation Dataset veri tabanından alınmıştır. Orijinal resimden sonraki resimler farklı cezbedici sayılarına göre sonuçları göstermektedir. Her bir görüntü için iki farklı kural vektörü uygulanmıştır. Her bir kural vektörü 16 farklı cezbedici durum üretmektedir. Cezbedici sayıları kademeli olarak azaltılmıştır. İlk önce 16 olan cezbedici sayısı sırasıyla 4, 3 ve 2 cezbedici sayılarına azaltılmıştır.



Şekil 4.9. Orijinal görüntü ve farklı kural vektörüne göre bölümlenmiş görüntüler a)Orijinal görünüt. b)-d) CA <102, 60, 204, 204, 240, 102, 60, 204, 204, 240> kural vektörüne göre sırasıyla 16, 8, 4 ve 2 cezbediciye göre bölümlendirilmiş görüntüler f)-i) CA CA <6, 60, 204, 204, 160, 6, 60, 204, 204, 160>kural vektörüne göre sırasıyla 16, 8, 4 ve 2 cezbediciye göre bölümlendirilmiş görüntüler



Şekil 4.10. Orijinal görüntü ve farklı kural vektörüne göre bölümlenmiş görüntüler
a)Orijinal görünüt. b)-d) CA <102, 60, 204, 204, 240, 102, 60, 204, 204, 240> kural vektörüne göre sırasıyla 16, 8, 4 ve 2 cezbediciye göre bölümlendirilmiş görüntüler
f)-i) CA CA <6, 60, 204, 204, 160, 6, 60, 204, 204, 160>kural vektörüne göre sırasıyla 16, 8, 4 ve 2 cezbediciye göre bölümlendirilmiş görüntüler

Orijinal resimden sonra gelen 4 resim CA<102, 60, 204, 204, 240, 102, 60, 204, 204, 240 > kural vektörü kullanılarak elde edilmiştir. Son 4 resim ise CA <6, 60, 204, 204, 160, 6, 60, 204, 204, 160> kural vektörü kullanılarak elde edilmiştir. 2, 3, 4, 5 numaralı görüntüler için cezbedici durum sayısı sırasıyla 16, 4, 3, ve 2'dir. 6, 7, 8, 9 numaralı görüntüler için cezbedici durum sayısı sırasıyla 16, 4, 3, ve 2'dir.

Aşağıdaki Şekil 4.11, Şekil 4.12, Şekil 4.13 ve Şekil 4.14'te sırasıyla görüntülerin 16, 8, 4, 2 cezbediciye göre elde edilmiş görüntüleri verilmektedir. En son Şekil 4.15'te 2 cezbediciye göre bölümlendirilmiş görüntülerin siyah beyaz görüntüsü verilmektedir. Siyah beyaz görüntülerde bölümlendirme daha iyi görüldüğü için görüntüler siyah beyaz renklere dönüştürülmüştür.



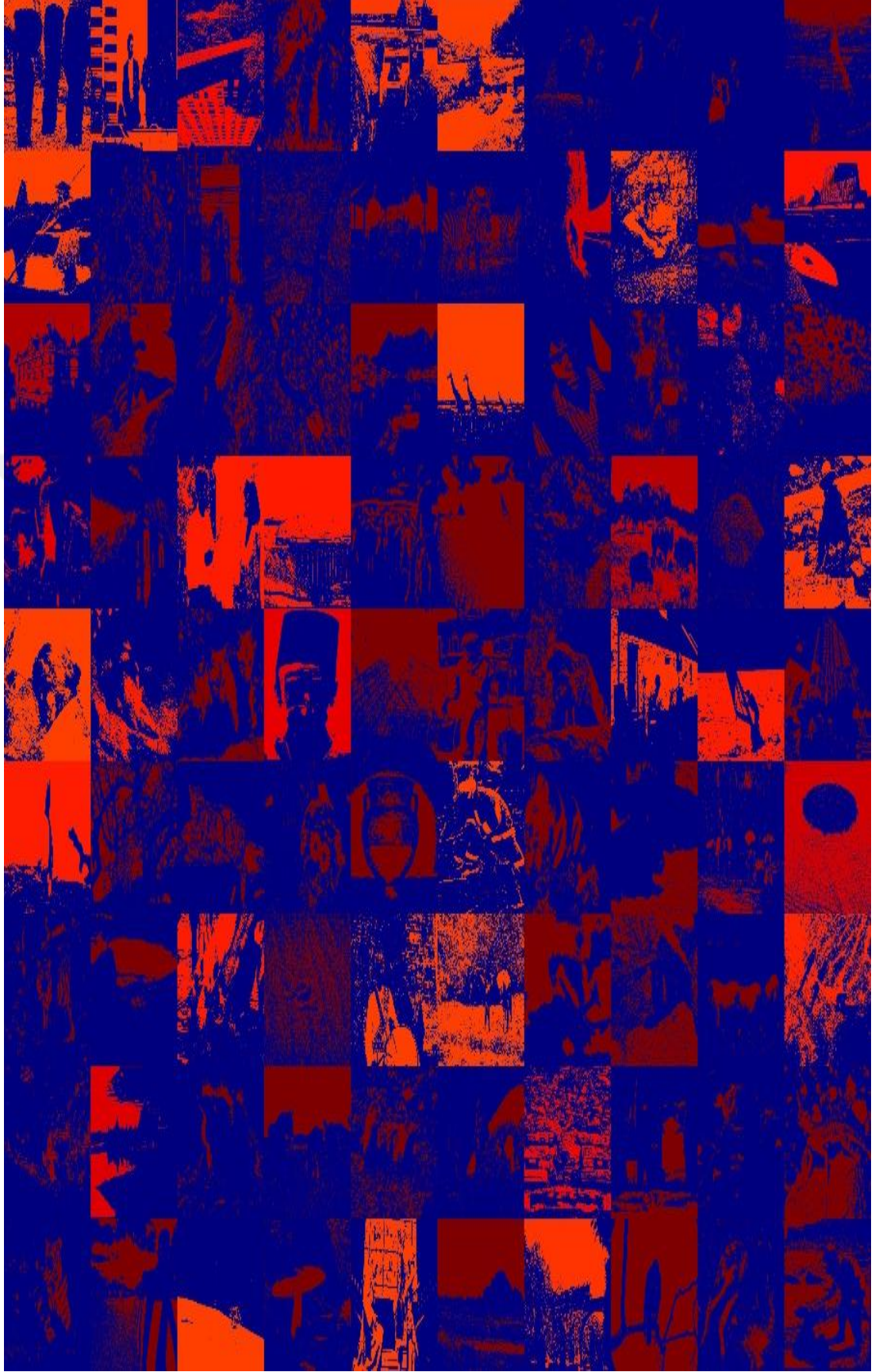
Şekil 4.11. 16 cezbediciye göre elde edilen bölümlendirilmiş görüntüler



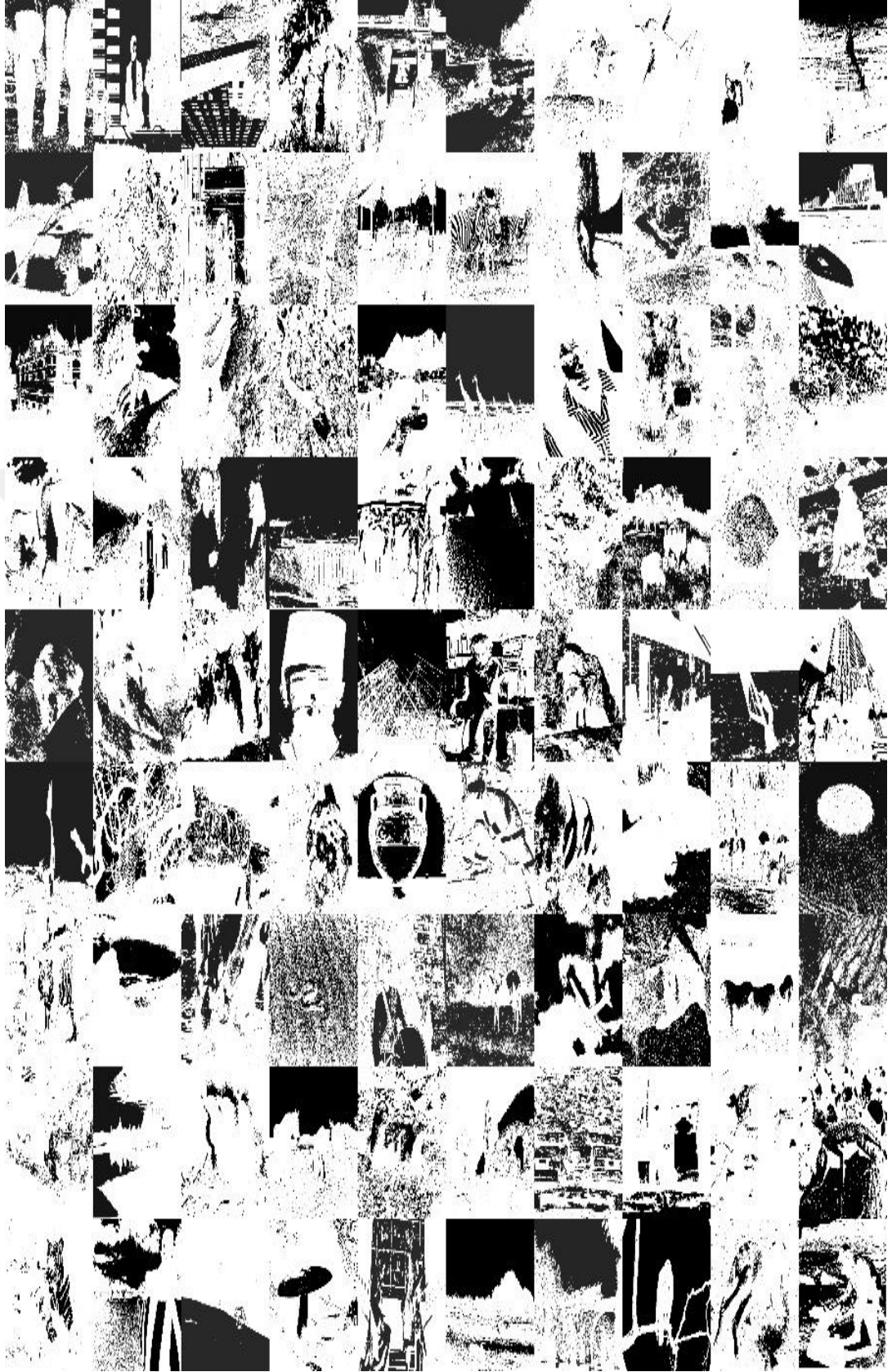
Şekil 4.12. 8 cezbediciye göre elde edilen bölümlendirilmiş görüntüler



Şekil 4.13. 4 cezbediciye göre elde edilen bölümlendirilmiş görüntüler



Şekil 4.14. 2 cezbediciye göre elde edilen bölümlendirilmiş görüntüler



Şekil 4.15. 2 cezbediciye göre elde edilen bölümlendirilmiş görüntülerin siyah beyaz renklerle görünümü

Yukarıdaki şekillerde görüntüler küçük olduğu için bölümlendirmeler belirgin bir şekilde görünmemektedir. Tüm görüntüleri bir görüntü karesi içerisinde verilmesinin amacı; tüm görüntüler üzerinde geliştirilen bölümlendirme yönteminin test edildiğini göstermektir.

Berkeley görüntü bölümlendirme veri setinin oluşturulmasındaki bir diğer amaç görüntü bölümlendirme yöntemlerinin performansının sayısal sonuçlarla ortaya konmasına olanak sağlamaktır. Bu amaçla veri setinden yer alan tüm görüntüler insanlar tarafından bölümlendirilmiştir. İnsanlar tarafından bölümlendirilen görüntüler “gerçek doğruluk görüntüleri” olarak adlandırılmaktadır. Gerçek doğruluk görüntüleri araştırmacılar tarafından geliştirilen görüntü bölümlendirme yöntemlerinin karşılaştırması için kullanılır. Karşılaştırma işlemi karışıklık matrisi oluşturularak elde edilir. Karışıklık matrisi kullanılarak kesinlik, hassaslık ve bunların birbirine oranı olan f1 skorları elde edilir.

Martin ve arkadaşları önerilen görüntü bölümlendirme yöntemlerinin performansının belirlenmesini hassaslık ve f1 skorları üzerinden değerlendirilmesini önermişlerdir (Martin ve ark., 2001). F1 değeri kesinlik ve hassaslığın harmonik ortalaması ile elde edilir.

Bölümlendirme yönteminin hassaslık ve f1 değerlerinin ölçümü bölgelerin karşılaştırılmasıyla elde edilmez. Bunun yerine görüntü bölümlendirme yönteminin ortaya koyduğu bölge sınırlarının gerçek doğruluk görüntülerinin karşılaştırılmasıyla elde edilir. F1 skoru, önerilen yöntemin ortaya koyduğu sınır çizgilerinin gerçek sınır pikseli olma olasılığıdır. Hassaslık değeri, gerçek bir sınır pikselinin algılanma olasılığıdır. F1 skoru, yöntemin çıktısında ne kadar gürültü olduğunu gösteren bir ölçüsüdür. Hassaslık değeri, gerçek sınırların ne kadar doğru ölçüldüğünün göstergesidir.

Karışıklık matrisi, sınıflandırma algoritmasının performansını özetlemek için kullanılan bir tablolama tekniğidir. Karışıklık matrisi, geliştirilen sınıflandırma modelinin tahmin sonuçlarını özetleyen bir tablodur. Doğru ve yanlış tahminler her bir sınıf için gösterilir. Sınıflandırma modelinin yaptığı yanlış tahminlerin detayını verir (Kuyumcu, 2016). Sadece, sınıflandırma modelinin hata yaptığını söylemek yerine

hatayı nasıl yaptığını söyler. Böylece “başarı oranı” sonucunda gizli kalan detaylar karışıklık matrisi kullanılarak ortaya çıkar.

Karışıklık matrisinde yer alan hücrelerin genel bir adlandırması vardır. Bu adlandırma her bir sınıf için ayrı bir anlam ifade eder ve her problem için aynı terimleri kullanma imkanı verir. Tablo 4.2’de karışıklık matrisi hücrelerinin adları yer almaktadır:

- TP: Doğru tahmin edilen Sınıf 0
- TN: Doğru tahmin edilen Sınıf 0 olmayan
- FP: Yanlış tahmin edilen Sınıf 0
- FN: Yanlış tahmin edilen Sınıf 0 olmayan

Tablo 4.2. Karışıklık matrisi hücrelerinin adları

		Tahmin Edilen	
		Sınıf 0	Sınıf 1
Gerçek	Sınıf 0	True Positive(TP)	False Positive(FP)
	Sınıf 1	False Negative(FN)	True Negative(TN)

Tablo 4.2.’ye göre bazı istatistiksel metrikleri Denklem (4.1), (4.2), (4.3) ve (4.4)’teki gibi hesaplanır;

$$\text{Başarı Oranı(Accuracy Rate)} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.1)$$

$$\text{Hassaslık (Doğru Pozitif Oranı(Recall))} = \frac{TP}{TP+FN} \quad (4.2)$$

$$\text{Kesinlik(Precision)} = \frac{TP}{TP+FP} \quad (4.3)$$

$$\text{F1 Skoru} = \frac{2TP}{2TP+FP+FN} \quad (4.4)$$

4.6. Yöntemin Performansı

Sonuçların elde edilmesinde herhangi bir hazır kod kullanılmamıştır. Hassaslık ve f1 değerlerini elde etmek için ilk önce karışıklık matrisi oluşturulur. Karışıklık matrisi oluşturulduktan sonra, bir önceki bölümde yer verilen sınıflandırma için kullanılan hassaslık ve f1 sırasıyla Denklem (4.3) ve Denklem (4.4)’e göre elde edilir.

Bu bölümde Berkeley veri setinden yer alan test görüntülerinin, önerilen yöntemle göre bölümlendirilmesiyle elde edilen sonuçlara yer verilecektir. Berkeley veri setindeki her bir görüntü numara ile adlandırılmıştır.

Tablo 4.3.'teki isimlerin sıralanması alfabetik sıraya göre yapılmıştır. Aynı sıralama yukarıda cezbedici sayısına göre verilen Şekil 11-15'teki toplu görüntülerde de yapılmıştır. Görüntüler adlarının alfabetik sıralanmasına göre verilmiştir. Görüntülerin sıralanışı sütun merkezli ilerlemektedir. Yani ilk sütunda ilk on görüntü, ikinci sütunda sonraki on görüntü olacak şekilde yer almaktadır.

Tablo 4.3. Berkeley test veri setinde yer alan her bir görüntü için elde edilen f1 ve hassasiyet değerleri

Görüntü Adı	F1 skoru	Hassasiyet (Recall)	Görüntü Adı	F1 skoru (Recall)	Hassasiyet
101085.jpg	0,67	0,67	241048.jpg	0,61	0,61
101087.jpg	0,67	0,68	253027.jpg	0,58	0,59
102061.jpg	0,72	0,72	253055.jpg	0,66	0,68
103070.jpg	0,63	0,65	260058.jpg	0,67	0,67
105025.jpg	0,64	0,64	271035.jpg	0,63	0,64
106024.jpg	0,69	0,70	285079.jpg	0,63	0,65
108005.jpg	0,64	0,65	291000.jpg	0,55	0,54
108070.jpg	0,62	0,63	295087.jpg	0,63	0,65
108082.jpg	0,67	0,68	296007.jpg	0,66	0,66
109053.jpg	0,59	0,61	296059.jpg	0,65	0,67
119082.jpg	0,61	0,63	299086.jpg	0,66	0,67
12084.jpg	0,61	0,62	300091.jpg	0,68	0,68
123074.jpg	0,64	0,63	302008.jpg	0,61	0,64
126007.jpg	0,64	0,65	304034.jpg	0,64	0,63
130026.jpg	0,65	0,65	304074.jpg	0,64	0,64
134035.jpg	0,54	0,57	306005.jpg	0,60	0,62
14037.jpg	0,65	0,67	3096.jpg	0,70	0,70
143090.jpg	0,65	0,67	33039.jpg	0,58	0,57
145086.jpg	0,59	0,60	351093.jpg	0,62	0,62
147091.jpg	0,61	0,64	361010.jpg	0,62	0,66
148026.jpg	0,62	0,62	37073.jpg	0,58	0,64
148089.jpg	0,52	0,56	376043.jpg	0,60	0,59
156065.jpg	0,63	0,64	38082.jpg	0,62	0,63
157055.jpg	0,57	0,61	38092.jpg	0,59	0,60
159008.jpg	0,60	0,62	385039.jpg	0,59	0,62
160068.jpg	0,64	0,65	41033.jpg	0,66	0,68
16077.jpg	0,63	0,64	41069.jpg	0,62	0,60
163085.jpg	0,61	0,64	42012.jpg	0,66	0,68

Tablo 4.3(Devamı) Berkeley test veri setinde yer alan her bir görüntü için elde edilen f1 ve hassasiyet değerleri

Görüntü Adı	F1 skoru	Hassasiyet (Recall)	Görüntü Adı	F1 skoru	Hassasiyet (Recall)
167062.jpg	0,70	0,71	42049.jpg	0,65	0,67
167083.jpg	0,64	0,63	43074.jpg	0,66	0,69
170057.jpg	0,54	0,57	45096.jpg	0,64	0,68
175032.jpg	0,57	0,55	54082.jpg	0,62	0,64
175043.jpg	0,64	0,63	55073.jpg	0,62	0,63
182053.jpg	0,63	0,64	58060.jpg	0,61	0,62
189080.jpg	0,67	0,69	62096.jpg	0,67	0,68
19021.jpg	0,59	0,61	65033.jpg	0,59	0,60
196073.jpg	0,59	0,57	66053.jpg	0,61	0,65
197017.jpg	0,66	0,67	69015.jpg	0,53	0,60
208001.jpg	0,52	0,58	69020.jpg	0,63	0,64
210088.jpg	0,55	0,61	69040.jpg	0,60	0,62
21077.jpg	0,64	0,64	76053.jpg	0,61	0,62
216081.jpg	0,57	0,61	78004.jpg	0,63	0,65
219090.jpg	0,65	0,66	8023.jpg	0,68	0,67
220075.jpg	0,63	0,64	85048.jpg	0,55	0,57
223061.jpg	0,66	0,65	86000.jpg	0,67	0,67
227092.jpg	0,64	0,66	86016.jpg	0,57	0,57
229036.jpg	0,62	0,61	86068.jpg	0,61	0,60
236037.jpg	0,63	0,62	87046.jpg	0,62	0,62
24077.jpg	0,57	0,60	89072.jpg	0,63	0,64
241004.jpg	0,62	0,65	97033.jpg	0,63	0,64
Ortalama:				0,62	0,64

Tablo 4.3.'te her bir görüntü için elde edilen hassaslık ve f1 değeri yer almaktadır. Tabloda yer alan bilgilerin hangi görüntüye ait olduğunu bir önceki paragrafı dikkate alarak anlaşılabilir.

Önerilen görüntü bölümlendirme yönteminin ortalama performansı 0,63'tür. Daha önce aynı görüntü veri seti üzerinde yapılan görüntü bölümlendirme yöntemlerinin performansı Tablo 4.4'te verilmiştir.

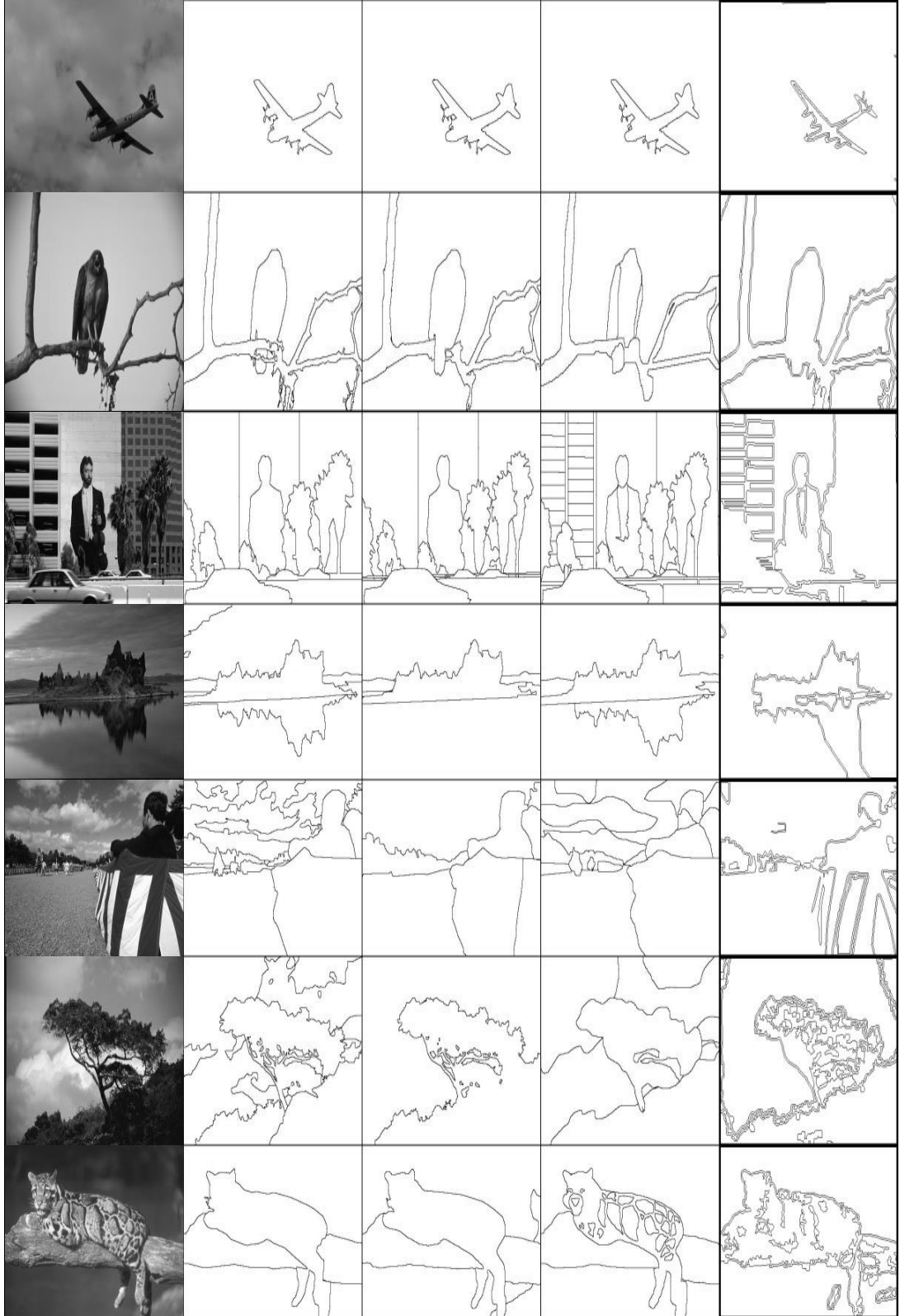
Tablo 4.4'te Berkeley görüntü bölümlendirme veri seti üzerine yapılan bölümlendirme yöntemlerinin performans sıralaması verilmektedir. Yukarıdaki sıralama gri görüntüler kullanılarak yapılan yöntemlere aittir. Bu çalışmada gri görüntüler kullanarak bölümlendirme yapılmıştır. Sıralama hassasiyet ve f1 değerlerine göre

yapılmıştır. Tabloda veri seti üzerinde yapılmış 13 çalışma yer almaktadır. GMACA kullanarak geliştirilen bölümlendirme yöntemi altıncı sırada yer almaktadır. Birinci sırada insanlar tarafından yapılan bölümlendirme performansı yer almaktadır. İnsanlar tarafından yapılan bölümlendirme yöntemi 0,79 başarı oranına sahiptir. Önerdiğimiz yöntemin başarı performansı 0,63'tür. Yaptığımız çalışma daha önceki yapılan çalışmalarda farklı bir metodoloji sahiptir. GMACA'nın eş zamanlı hesaplama yeteneği ve ayırık yapısı önerdiğimiz yöntemi önemli hale getirmektedir.

Tablo 4.4. Berkeley veri seti üzerinde yapılan çalışmaların başarı oranları

Sıra	Başarı Oranı	Algoritma
0	0,79	Humans (URL-5, 2017)
1	0,68	gPb-ucm (Arbelaez ve ark., 2011)
2	0,68	Ren et al. NIPS2012 (Ren ve Bo, 2012)
3	0,68	Global Probability of Boundary (Maire ve ark., 2008)
4	0,66	xren (Ren, 2008)
5	0,64	Boosted Edge Learning (Dollar ve ark., 2006)
6	0,63	GeliştirilenYöntem
7	0,63	Brightness / Texture Gradients (Martin ve ark, 2004)
8	0,60	Brightness Gradient (Martin ve ark., 2003)
9	0,58	Texture Gradient (Martin ve ark., 2003)
10	0,58	Multiscale Gradient Magnitude (URL-5, 2017)
11	0,57	Second Moment Matrix (URL-5, 2017)
12	0,56	Gradient Magnitude (URL-5, 2017)
13	0,48	Segmentation Induced by Scale Invariance (Yu, 2005)

Şekil 4.16 ve Şekil 4.17'de test veri seti üzerinde insanların yaptığı bölümlendirmeler ve önerilen yöntemin sonuçlarına yer verilmiştir. Berkeley veri setinde yer alan görüntüler iki farklı ebatla olabiliyorlar. Bu nedenle aynı ebattaki görüntüler beraber verilmiştir. Bazı görüntülerin genişliği uzun iken diğerlerinin yüksekliği uzun olabiliyor. Bu nedenle yükseklik ve genişliği aynı olan görüntüler aynı şekilde yer almaktadır.



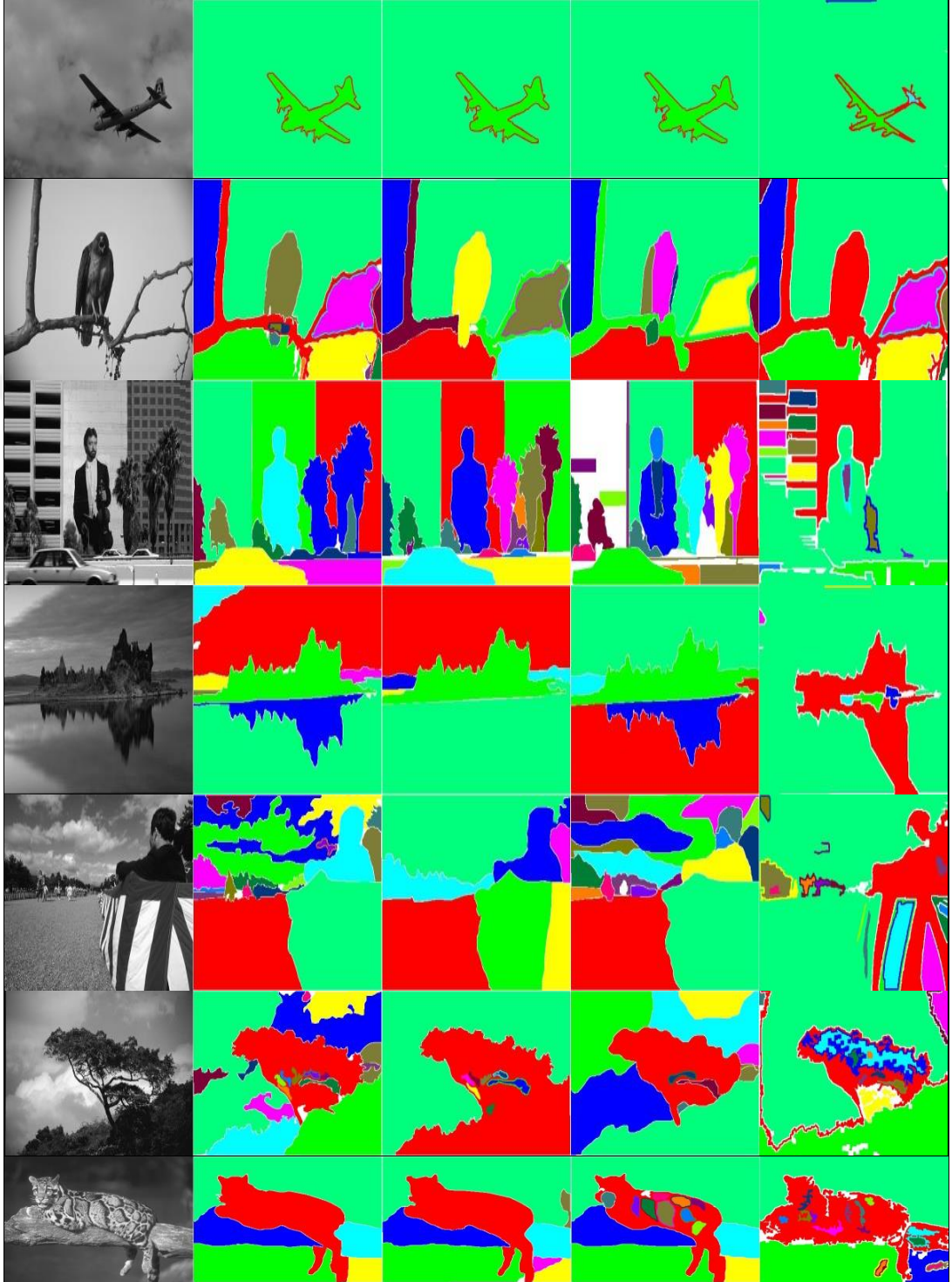
Şekil 4.16. En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş görüntüler, en sağda geliştirilen yöntemle elde edilen sonuç yer almaktadır.



Şekil 4.17. En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş görüntüler, en sağda geliştirilen yöntemle elde edilen sonuç yer almaktadır

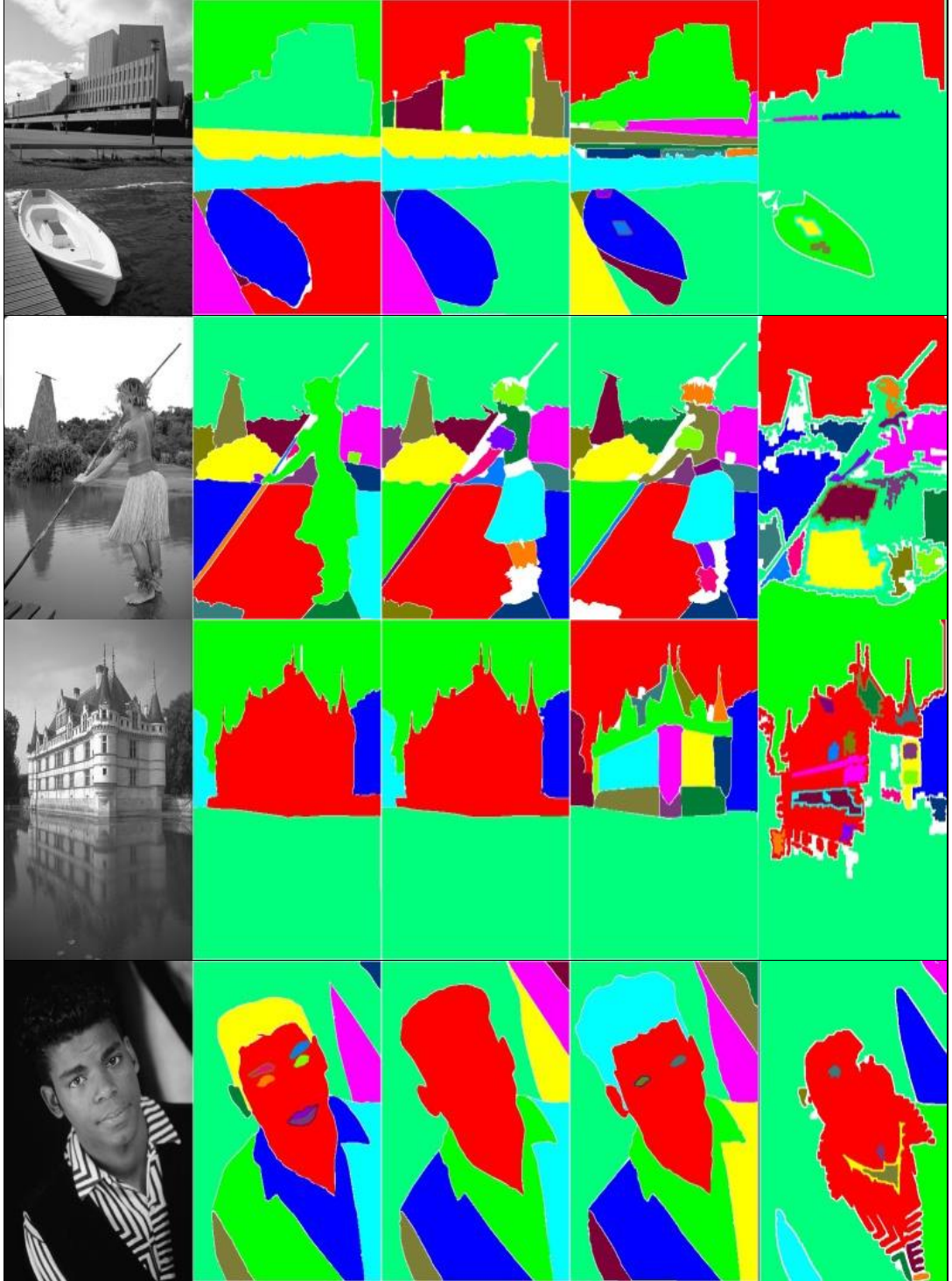
Şekil 4.16. ve Şekil 4.17.'de her bir satırda beş tane görüntü yer almaktadır. Satırdaki ilk görüntü Berkeley test veri setinde yer alan orijinal görüntüdür. Daha sonra gelen üç görüntü, orijinal görüntünün üç farklı insan tarafından bölümlendirilmiş halidir. En sağda, önerilen yöntemle elde edilmiş görüntü yer almaktadır. Geliştirilen yöntemin

ortaya koyduđu çıktılar insanlar tarafından yapılan bölümlendirmelere yakın karakteristiđe sahiptir.



Şekil 4.18. En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş renkli görüntüler, en sağda geliştirilen yöntemle elde edilen renklendirilmiş sonuç yer almaktadır

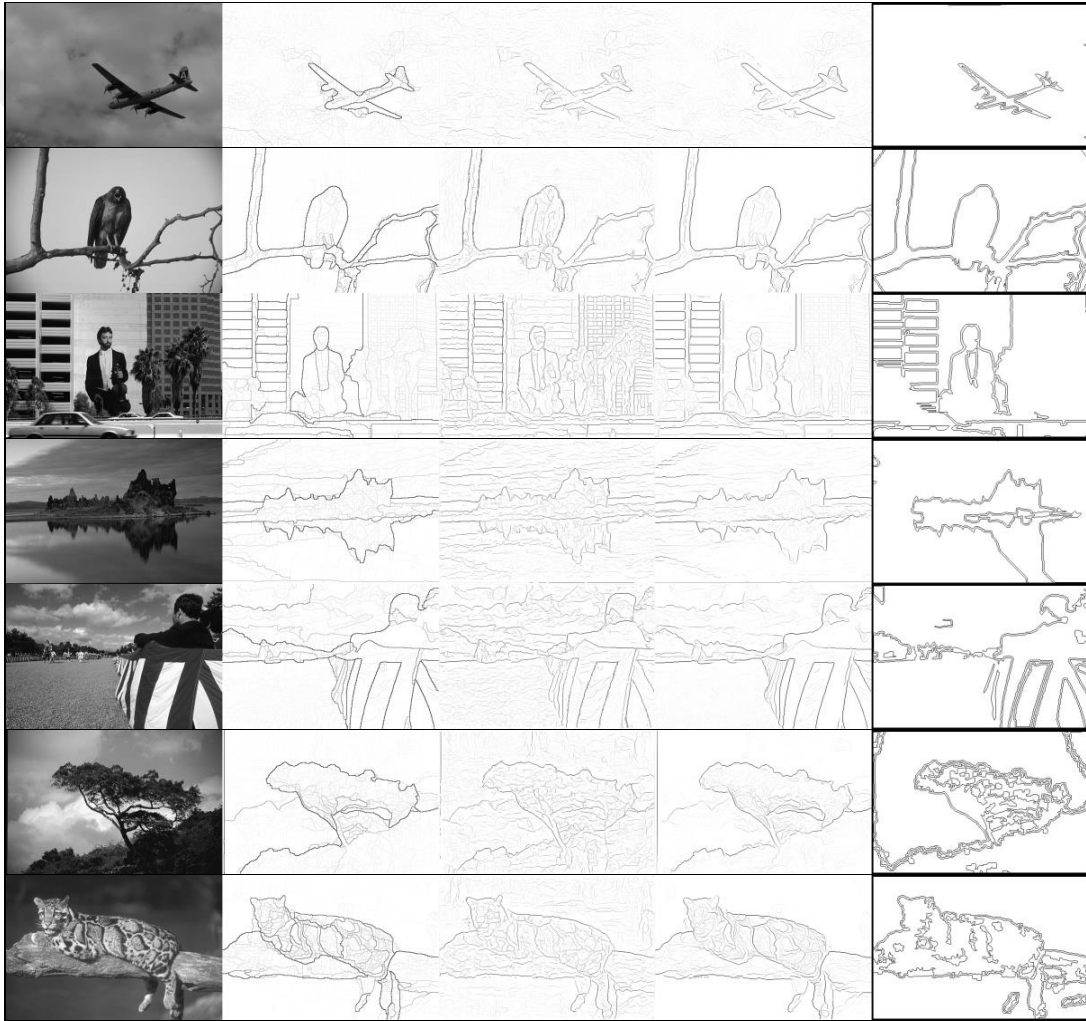
Şekil 4.18. ve Şekil 4.19.’da test veri seti üzerinde insanların yaptığı bölümlendirmeler ve önerdiğimiz yöntemin renklendirilmiş görüntülerine yer verilmiştir.



Şekil 4.19. En solda orijinal görüntü, ortada insanlar tarafından bölümlendirilmiş renkli görüntüler, en sağda geliştirilen yöntemle elde edilen renklendirilmiş sonuç yer almaktadır.

Yukarıdaki şekilde dört satır ve beş sütundan oluşmaktadır. Her bir satırda beş tane görüntü yer almaktadır. Satırdaki ilk görüntü Berkeley test veri setinde ver alan orijinal görüntüdür. Daha sonra gelen üç görüntü, orijinal görüntünün üç farklı insan tarafından bölümlendirilmiş halidir. En sağda, geliştirilen yöntemle elde edilmiş görüntü yer almaktadır. Geliştirilen yöntemin ortaya koyduğu çıktılar insanlar tarafından yapılan bölümlendirmelere yakın karakteristiğe sahiptir.

Şekil 4.20. ve Şekil 4.21.'de test veri seti üzerinde daha önceki literatür çalışmalarında elde edilen en iyi üç sonuca ve geliştirilen yöntemin görüntülerine yer verilmiştir.



Şekil 4.20. En solda orijinal görüntü, devamında aynı görüntüler üzerinde elde edilen en iyi üç yöntemle (Arbelaez ve ark., 2011; Ren ve Bo, 2012; Ren, 2008) ait sonuçlar ve en solda geliştirilen yöntemle elde edilen sonuç yer almaktadır.



Şekil 4.21. En solda orijinal görüntü, devamında aynı görüntüler üzerinde elde edilen en iyi üç yönteme (Arbelaez ve ark., 2011; Ren ve Bo, 2012; Ren, 2008)ait sonuçlar ve geliştirilen yöntemle elde edilen sonuç yer almaktadır.

Yukarıdaki şekillerde dört satır ve beş sütundan oluşmaktadır. Her bir satırda beş tane görüntü yer almaktadır. Satırdaki ilk görüntü Berkeley test veri setinde yer alan orijinal görüntüdür. Daha sonra gelen üç görüntü, orijinal görüntünün literatürde en iyi üç farklı yöntem tarafından bölümlendirilmiş görüntülerdir. En sağda, geliştirilen yöntemle elde edilmiş görüntü yer almaktadır. Geliştirilen yöntemin ortaya koyduğu

çıktılar literatürdeki en iyi üç yöntem tarafından yapılan bölümlendirmelere yakın sonuçlar vermiştir.

4.7. Önerilen Yöntemin Karmaşıklığı

Das ve arkadaşları yaptıkları çalışmada (Das ve ark., 2009) tek uzunluk döngülü cezbedicilerin niteliği hakkında detaylı bilgiler içermektedir. Kural vektörü tarafından üretilen cezbedicileri bulmanın karmaşıklığı $O(n)$ 'dir. Cezbedici durumlar PE (pseudo-exhaustive) bitleri ile temsil edilebilirler (Das ve ark., 2009). PE bitlerini tanımlamanın karmaşıklığı, n (kural vektör boyutu) ve RMT sayılarının sayısına bağlıdır. Toplam RMT sayısı iki durum ve üç komşu CA için her zaman 8 olduğu için karmaşıklık $O(n)$ (Das ve ark., 2009)'dır. m tane PE bit, bir n -hücreli CA'nın 2^m cezbedici durumunu benzersiz şekilde tanımlayabilir, $m \leq n$ (Das ve ark., 2009). PE bitleri, bir uygulama için GMACA modeli geliştirirken hesaplama karmaşıklığını ve bellek karmaşıklığını azaltır. Kısaca, önerilen yöntemin istene kural vektörünün elde edilmesi için zaman karmaşıklığı ve bellek karmaşıklığı sırasıyla $O(n)$ ve $O(\log_2 n)$ olup, burada n kural vektör uzunluğu ve m PE bit sayısıdır.

Kural vektörünün bir kere elde edilmesi, görüntülere uygulanması için yeterli olacaktır. Bir görüntüdeki piksel sayısına k diyecek olursak, görüntüler üzerinde kural vektörünün uygulanması için gerekli zaman en kötü durum için $O(k+n)$ ve gerekli hafıza en kötü durum için $O(k+\log_2(n))$ olacaktır.

4.8. Sonuç

Çalışmanın bu kısmının temel amacı, görüntü bölümleme bağlamında görüntü işleme için GMACA'nın kullanılabileceğini göstermektir. Yapılan literatür araştırmasında GMACA tabanlı görüntü işleme çalışmasına rastlanmamıştır. Çalışma, görüntü işleme için GMACA modelleme kabiliyetini kullanmaya yönelik ilk girişimdir. Yöntem, kümeleme görüntü bölümleme yöntemleri altında değerlendirilebilir. Temel varsayım, bir bölgeye ait piksellerin yakın değeri olacak olmasıdır. Yöntemde piksellerin konum bilgisinden yararlanılmıyor. Daha iyi bölümle sonucu elde etmek için piksel konum bilgisinden faydalanarak önerilen yöntem geliştirilebilir.

GMACA tarafından üretilen tek uzunluk döngülü cezbediciler, görüntü bölümlemesi için kullanılmıştır. Her cezbedici durum görüntüdeki bir bölgeyi temsil etmektedir. Yoğunluk imgeleri için kural uzunluğunun minimum boyutunun sekiz olması gerekmektedir. Gri tonlamalı görüntüler için görüntü bölümlendirme yapılmıştır, ancak geliştirilen yöntem herhangi bir renk formatı için kolaylıkla uyarlanabilir. Ayrıca pikselin renk değeri tamsayı olmalıdır. Piksel değerleri noktalı sayı ise, değerleri tamsayıya çevirip 0'dan 255'e yeniden ölçeklendirdikten sonra yöntem uygulanabilir. Kural boyutu sekizden büyükse, kural vektörü tarafından üretilen cezbedici durum değeri 0 ile 255 arasında olmalıdır.

Önerilen yöntemin n uzunluklu ve m tane kapsayıcı bit üreten kural vektörü için zaman karmaşıklığı ve bellek karmaşıklığı sırasıyla $O(n)$ ve $O(\log_2 n)$ olup, n kural vektörünün uzunluğu ve m , cezbedici durum sayısıdır.

Önerilen bölümleme yönteminin zayıf tarafı, kural vektörü için kullanılan kuralların seçilmesinde görüntü özelliklerinin kullanılmamasıdır. Daha sonra yapılacak çalışmalarda görüntünün histogramından faydalanılarak kural seçimine yardımcı olunabilir. Seçilecek kuralın oluşturacağı cezbediciler histogramın lokal maksimum veya minimum değerlerine göre elde edilebilir. GMACA ikili görüntüler üzerinde çalışabilen hesaplama yeteneğine sahiptir. GMACA'nın sadece ikili değerler üzerinde uygulanabilmesi yöntemin bir diğer eksikliğidir. Ancak şunu hemen belirtmek gerekir: Bu çalışmada gri görüntüler üzerinde görüntü bölümlendirme yapılmıştır. Bunun için onluk tabandaki gri değerleri ikilik tabana dönüştürdükten sonra GMACA uygulanmıştır.

Daha önceki temel CA tabanlı görüntü işleme çalışmalarının çoğu ikili görüntüler üzerinde çalışmaktadır. Bir kez daha vurgulamak içi şunu belirtelim: Önerilen görüntü bölümlendirme işlemi, gri görüntüler üzerinde gerçekleştirilmiştir. Gri görüntüler üzerinde çalışabilen CA tabanlı bir yöntem araştırmacıların dikkatini çekecektir.

5. GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL OTOMATLARIN BOYUT AZALTMADA KULLANILMASI

Çalışmanın bu bölümünde GMACA'nın örüntü çekim yeteneklerinin boyut azaltmada kullanılması araştırılmıştır. En Yakın Komşu (K nearest Neighbour KNN) algoritması, GMACA tarafından boyutu azaltılan verilerin sınıflandırılmasında kullanılmıştır (Peldek ve Becerikli, 2017).

Cezbediciler, çekim merkezi formunda davranışlar sergileyen ve diğer durumlarla dinamik ilişkiye sahip durumlardır. Kural vektörünün ürettiği birden fazla cezbedici olabilir. Her cezbedici farklı sayıda örüntüyü etrafından toplar. Cezbedici etrafında toplanan örüntüler cezbedici tarafından temsil edilir. Cezbedicilerin bu özelliği örüntü sınıflandırma ve tanıma uygulamalarında kullanılmaktadır.

Cezbedicilerin örüntü çekim merkezi formundaki davranışları, makine öğrenmesi modellerinin kullandığı özellik vektörlerinin boyutunun azaltılmasında kullanılabilir. Cezbediciler sayesinde birden fazla benzer özellik tek bir özellik tarafından temsil edilebilir.

Boyut azaltma veri setindeki özellikler üzerinde yapılan bir işlem olduğu için özelliklerin makine öğrenmesi modeline uygun yapıya dönüştürülmesinde yapılan ön işlemleri bilmek faydalı olacaktır. Bir sonraki alt bölümlerde, makine öğrenmesi modeli kullanılmadan önce yapılan ön işlemlere yer verilmiştir.

5.1. Makine Öğrenme Modeli İçin Verilerin Hazırlanması

Bu alt bölümde, makine öğrenmesi modeli kullanılmadan önce veri seti üzerinde yapılan ön işlem başlıklar halinde ele alınmıştır.

Gerçek uygulamaların çoğunda veri seti, makine öğrenmesi modelleri için uygun değildir. Öncelikle veri setindeki eksik bilgiler doldurulur. Daha sonra, tüm makine öğrenmesi modelleri sayısal verileri kullandığı için kategorik ve metin değerlerinin, numerik değerlerle temsil edilmesi gerekir. Veri setindeki tüm özellikler numerik

değerlerle temsil edildikten sonra normalleştirme yapılır. Karar ağaçları tabanlı modeller hariç tüm makine öğrenmesi modelleri için normalleştirme çok önemlidir. Özellik vektöründeki bazı özellikler makine öğrenmesi modelinin başarısına negatif yönde etki edebilir. Bunun için özellik vektöründeki önemli özelliklerin elde edilmesi için düzenleme ve alt küme seçimi işlemleri yapılır. Son olarak, veri seti eğitim ve test amaçlı olmak üzere ikiye ayrılır. Model, eğitim veri seti kullanılarak eğitildikten sonra, test veri seti üzerinde başarısı test edilir.

5.1.1. Eksik verilerle başa çıkmak

Verilerin kalitesi ve içerdiği faydalı bilgilerin miktarı, bir makine öğrenme algoritmasının ne kadar iyi öğrenebileceğini belirleyen temel faktörlerdir. Bu nedenle, bir veri kümesini bir öğrenme algoritmasına göndermeden önce incelemek ve ön işleme tabi tutmak kritik öneme sahiptir (Raschka, 2015). Verilerin eğitim aşamasına geçmeden önce girdiler ve hedefler analizine göre bir ön işleme tabi tutulursa makine öğrenme algoritmaları daha iyi sonuç verirler (Marsland, 2015).

Veriler ilk elde edildiğinde yontulmamış elmasa benzerler. Verinin raporlama ve analiz aşamasında kullanılmak üzere hazır hale getirilmesi gerekir. Veriyi ön işleme tabi tutmak, öğrenme modellerimizin daha iyi performans göstermesini sağlayacaktır ve beklenmeyen sonuçları düzeltmeye çalışırken harcayacağımız zamandan daha az zamana ihtiyaç duyduğu için muazzam derecede önemlidir (Cielen ve ark., 2016).

Veri temizliği, verilerimizdeki hataları ortadan kaldırmaya odaklanan veri bilimi sürecinin bir alt işlemi olup verilerimizi aslına uygun doğru ve tutarlı bir temsili haline dönüştürür (Cielen ve ark., 2016). Gerçek dünyadaki uygulamalarda, veri örneklerimizde çeşitli nedenlerle bir veya daha fazla değer eksik olması sık görülen bir durumdur. Örneğin; veri toplama sürecinde bir hata olabilir, bazı ölçümler uyumlu olmayabilir veya belirli alanlar özellikle boş bırakılmış olabilir (Raschka, 2015).

Ne yazık ki, hesaplama araçlarının çoğu bu tür eksik değerleri işleyemiyor veya bu eksiklikleri görmezden geldiğimizde öngörülemeyen sonuçlar verebiliyor. Bu nedenle, veriler üzerinde analiz yapmadan önce eksik değerlerle ilgilenmek önemlidir. Kayıp verilerle başa çıkmanın en kolay yollarından biri, karşılık gelen özellikleri (sütunları) veya örnekleri (satırları) tamamen veri kümesinden kaldırmaktır (Raschka, 2015). Bu

eksik verilerle başa çıkmanın kolay bir yöntemi olsa da var olan diğer özellikleri veya gözlemleri veri setinden çıkarmak bilgi kaybına neden olmaktadır. Eksik verilen yerine null değeri atamak diğer bir kolay yöntemdir. Bu şekilde var olan özelliği veya gözlemi tamamen kaldırmak yerine olmayan değerın hesaba katılmaması gerektiğini gösteren null değeri atanır. Ancak bu yöntemin dezavantajı; tüm hesaplama tekniklerinin veya uygulamalarının null değerini işlemeye uygun olmamasıdır (Cielen ve ark., 2016).

Eksik değerlerle başa çıkmanın diğer bir yolu ara değer hesaplama (enterpolasyon) teknikleridir. Veri setimizdeki diğer eğitim örneklerinden eksik değerleri tahmin etmek için farklı istatistik teknikler kullanılabilir. En yaygın istatistik tekniklerinden biri, eksik değeri tüm özellik sütununun ortalama, sıklık veya sıfır değeri ile yer değiştirmektir (Raschka, 2015; Cielen ve ark., 2016).

Eksik kategorik özellik değerlerini, özellik sütunundaki en sık değerle, numerik özellik değerlerinin özellik sütunun ortalamasıyla değiştirmek daha faydalı olmaktadır. Bu yöntemlerin avantajları; uygulamasını kolay olması ve gözlemlenen diğer özelliklerdeki bilgilerin kaybolmamasıdır. Dezavantajı ise hesaplama modelinden yanlış tahminler yapılmasına neden olmasıdır (Cielen ve ark., 2016).

5.1.2. Kategori verilerini işleme

Gerçek dünya veri kümelerinin bir veya daha fazla kategorik özellik sütunu içermesi sık rastlanan bir durumdur. Kategorik veriler; nominal ve ordinal olmak üzere ikiye ayrılır. Ordinal özellikler sıralanabilir veya sıralanabilir kategorik değerler olarak anlaşılabilir. Örneğin, $XL > L > M$ bir sipariş tanımlayabildiğimiz için, tişört boyutu sıra düzenli ordinal bir özellik olacaktır.

Ordinal özelliklerin aksine, nominal özellikler herhangi bir düzeni ima etmez. Önceki örnek üzerinden gidersek; T-shirt rengi normal bir özelliktir, çünkü örneğin kırmızı maviden daha büyük olduğunu söylemek mantıklı değildir (Raschka, 2015).

Öğrenme algoritmasının ordinal özellikleri doğru bir şekilde yorumladığından emin olmak için, kategorik dize değerleri tam sayılara dönüştürülmelidir (Raschka, 2015). Nominal değerler uğraşmak biraz daha zordur. Çünkü nominal değerleri doğrudan

sayısal bir değere çevirdiğimizde farklı değerlere sahip olmanın sebep olacağı hatalar olacaktır. Yukarıdaki T-shirt örneğindeki renklere sayısal değerler verirsek mavi, yeşil ve kırmızı değerleri sırasıyla 0, 1 ve 2 değerlerini alacaktır. Bu da hesaplama modelinin kırmızıyı diğer değerlerden daha fazla ağırlığa sahip bir özellik olarak yorumlamasına sebep olacaktır. Bu sorun için önerilen çözüm, bir sıcak kodlama(one-hot encoding) adı verilen bir teknik kullanmaktır. Bu yaklaşımın arkasındaki fikir, nominal özellik sütunundaki her benzersiz değer için yeni bir kukla özellik oluşturmaktır (Raschka, 2015). Değişkenleri kuklalara dönüştürmek, birden çok sınıflı olan bir değişkeni birden çok değişkene bölen ve her birini yalnızca iki olası (0 veya 1) değere sahip olan bir veriye dönüştürür (Cielen ve ark., 2016). Kullandığımız örnekte renk özelliğini üç yeni özelliğe dönüştüreceğiz: mavi, yeşil ve kırmızı. Böylece, bir numunenin belirli rengini belirtmek için ikili değerler kullanılabilir; Örneğin, mavi bir renk için, mavi = 1, yeşil = 0, kırmızı = 0, yeşil bir renk için, mavi = 0, yeşil = 0, kırmızı = 1 ve kırmızı bir renk için, mavi = 0, yeşil = 0, kırmızı = 1, olarak kodlanabilir (Raschka, 2015).

5.1.3. Özellik vektöründeki değerlerin aynı aralığa çekilmesi

Gerçek hayat uygulamalarında kullanılan veriler çok boyutludur. Bu çok boyutlu veriler özellik vektörü kullanılarak temsil edilirler. Geliştirilen makine öğrenmesi modelinin özellik vektörü üzerinde çalıştıracağımızda dikkat edilmesi gereken önemli bir konu; özellik vektöründeki tüm özellik değerlerinin aynı aralıkta olmasıdır. Özellik vektöründe birbirinden farklı aralıklarla temsil edilen özelliklerin olması sık görülen bir durumdur. Mesela sağlık analizi için kullanılan bir özellik vektöründe kiloyu belirten özellik olabileceği gibi kandaki bir kimyasalın değerini taşıyan bir özellikte olabilir. Bu iki özelliğin değer aralığı çok farklı olabilir. Kilo değer aralığı genellikle 60 ile 120 arasında bir değer olabilirken, bir kimyasalın değeri miligram cinsinden 600 ile 2000 arasında bir değer olabilir. Bu durumda kimyasalın değeri kilo değerinin etkisini önemli oranda azaltacaktır.

Özellik değerlerinin aynı aralığa çekilmesi, makine öğrenme modelinin oluşturulmasında en önemli adımlardan biridir. Karar Ağaçları ve Rastgele Ormanlar gibi, özellik ölçeklemeye ihtiyaç duymayan az sayıdaki makine öğrenme algoritması vardır. Bununla birlikte, makine öğrenimi ve optimizasyon algoritmalarının

çoğunluğu, özellikler aynı ölçeğe sahipse daha iyi sonuç verirler (Raschka, 2015). Özellik vektöründeki değerlerin belirli bir aralığa çekilmesi işlemi normalleştirme ve standartlaştırma olarak adlandırılır. Normalleştirmede değerler [0,1] arasında iken standartlaştırmada değerler [-1,1] aralığındadır. Normalleştirme ve standartlaştırma dönüşümleri aşağıda sırasıyla Denklem (5.1) ve (5.2)'de verilmiştir;

$$x_{\text{norm}}^{(i)} = \frac{x^{(i)} - x_{\min}}{x_{\max} - x_{\min}} \quad (5.1)$$

$$x_{\text{std}}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x} \quad (5.2)$$

Normalleştirme kullanarak özellik değerleri istenen aralığa çekebilir. Değer aralığını, istenilen aralığa çekmek için normalleştirme daha kullanışlıdır. Ancak çoğu makine öğrenme algoritması standartlaştırmada daha iyi sonuç vermektedir. Bunun nedeni Lojistik Regresyon ve Destek Vektör Makineleri gibi makine öğrenme algoritmaları ağırlıkların başlangıç değerlerini 0 veya 0'a yakın rasgele değerler atamasıdır. Standartlaştırma kullanıldığında, özellik sütünün ortalamasını 0 ve standart sapması 1 olarak belirlenir. Bu şekilde özellik sütünü normal dağılıma sahip olur ve ağırlıkların öğrenmesini kolaylaştırır (Raschka, 2015).

Bazen, ordinal veya kategorik bir özelliği ölçeklenebilir hale getirmek gerekebilir. Bu işleme özellik kalibrasyonu denir. Birçok önemli uygulamaya sahip olan denetimli bir özellik dönüşümü işlemidir. Doğrusal sınıflandırıcılar gibi makine öğrenmesi modellerinin ihtiyaç duyduğu ölçeklendirilmiş veriler sağlar. Ayrıca, modellere özellikleri sıralı, kategorik veya kantitatif olarak işleme esnekliği kazandırır (Julian, 2016).

5.1.4. Modelin kullanabileceği kaynak: “özellikler”

İstatistiksel makine öğrenme, örüntü tanıma ve veri madenciliği alanlarında, veriler bir özellik matrisi veya veri matrisi kullanılarak temsil edilir. Genellikle, veri matrisinin sütunları özelliklere karşılık gelir. Bir özellik, bir örüntünün niteliğini veya karakteristiğini gösterir (Murty ve Devi, 2015). Veri matrisinin satırları, veri kümesindeki örüntülere karşılık gelir. Örüntü, makine öğrenmesinde ve diğer veri kullanarak model oluşturan disiplinlerde nokta, vektör ve örnek olarak da adlandırılır.

İnsanların verdiği kararların isabeti, sahip oldukları bilgiyle doğru orantılıdır. Aynı şekilde bir makine öğrenme modeli, modele sağlanan özellikler kadar iyi sonuçlar verir. Matematiksel olarak özellikler, örnek uzayından belirli bir alandaki değerler grubuna eşleyen bir fonksiyon olarak düşünülebilir. Makine öğrenmesinde yaptığımız hesaplamaların ve ölçümlerin çoğu sayısalıdır ve bu nedenle bir örüntünün özellikleri en yaygın olarak gerçek sayılar kullanılarak ifade edilir (Julian, 2016). Özellikler yeni özellik üretmek için kullanılabilir. Örneğin iki özelliği harmanlayıp yeni bir özellik oluşturmak çoğu zaman çok kullanışlıdır. Yeni bir özellik türetme işlemine özellik mühendisliği adı verilir. Yeni özellik üretimi algoritma oluşturmaktan daha az öneme sahipmiş gibi görünebilir. Ancak makine öğrenmesi modelinin başarımı için çok önemli olabilir (Richert ve Coelho, 2013).

Kullanışlı bir “özellik”, önemli olan şeylerle eşzamanlı olarak değişir ve olmayan şeylerle değişmez yapıda olmalıdır. İyi bir “özellik” elde etmek önemlidir. Birçok araştırma alanı için, üzerine inşa edebileceğimiz olası özellikler ve özellik türleri hakkında çok geniş bir literatür oluşmuştur (Richert ve Coelho, 2013). Makine öğrenme algoritmasının performansını arttırmak için doğru veya ayırt edici özelliklerin kullanılması gereklidir. Alakasız özelliklerin kullanılması gereksiz hesaplamalara ve sonuçlara neden olabilir (Murty ve Devi, 2015). Bunun yanında modelin doğruluğunu azaltacak bilgileri atmamak önemlidir. Hangi özelliğin etkisiz veya alakasız olduğunu belirlemek önemli bir makine öğrenmesi görevidir. Bunu için iki yaklaşım vardır (Julian, 2016); özellik seçimi ve özellik çıkarımı. Özellik seçimi, özgün özellik değişkenlerinden yeni bir alt kümesini bulmaya çalışır. Öte yandan özellik çıkarma, birbiriyle ilişkili değişkenleri birleştirerek veya özellikleri dönüştürerek yeni özellik değişkenleri oluşturur (Julian, 2016).

Özellik seçimi için önerilen birçok yöntem var, ancak pratikte çok basit fikirler en iyi sonucu verebiliyor. Özellik sayısının küçük olduğu durumlarda, özellik seçimi yapmak mantıklı değil, ancak binlerce özelliğe sahip bir özellik vektörünüz varsa, bu özelliklerden çoğunu atmak makine öğrenme sürecinin geri kalanını daha hızlı gerçekleştirilmesini sağlayabilir (Richert ve Coelho, 2013). Bazı durumlarda, modelin eğitimi kabul edilebilir bir süre içerisinde tamamlanmayabilir. Özellik seçimi ilk gerçekleştirilirse, eğitim süresi kabul edilebilir sınırlar içinde olabilir (Murty ve Devi, 2015). Öğrenme modeli için yararlı olan özelliklerin tanımlanmasında dikkat edilmesi

gereken bir husus; özellikleri elde etmenin aşırı maliyetli ve zaman alıcı olmamasıdır (Marsland, 2015).

Üç özellik seçim yöntemi vardır. Bunlar (Murty ve Devi, 2015): Filtre yöntemler, Alt küme seçim yöntemleri, Gömülü yöntemler. Filtre yöntemleri, her özellik makine öğrenmesi algoritmasını koşturarak bir puan hesaplar ve daha sonra puanına göre özellikleri seçer. Alt küme seçim yöntemleri, makine öğrenme algoritmasını özellik vektörünün alt kümelerinde koşturur ve performansına göre alt kümeleri derecelendirir. Gömülü yöntemler, özellikleri eğitim sürecinde seçer.

Bir modelin eğitim veri setindeki performansı test veri setindeki performansında çok iyiye, öğrenme modelinin kullanılan veriler üzerinde aşırı uygunluk (overfitting) sorunsalına sahip olabileceğini gösterir. Aşırı uygunluğun bir nedeni, modelin verilen eğitim verileri için çok karmaşık olmasıdır. Aşırı uygunluğu gidermenin bir çözümü özellik sayısının azaltılmasıdır (Raschka, 2015).

5.1.5. Modelin basitleştirilmesi: düzenleme(regularization)

Düzenleme makine öğrenmesinde popüler bir kavramdır. Düzenleme modelimizin basitleştirilmesine yardımcı olur. En yaygın kullanılan düzenleme yöntemleri L1 ve L2 düzenleme yöntemleridir. L2 ve L1 düzenlemeleri bazen sırasıyla Ridge ve Lasso regresyonu isimleriyle de adlandırılmaktadır.

1990'lı yıllardan başlayarak önce biyomedikal alanda ve daha sonra Web'de P, N'den büyük sorunu olarak adlandırılan sorunlar ortaya çıkmaya başladı. Bu durumda, özelliklerin sayısı örneklerin sayısından daha büyük olduğu için, eğitim verilerinde mükemmel sonuçlar elde etmek mümkündür. Bununla birlikte, bu önemli bir sorundur, sıfır eğitim hatası çözümünüzün genelleşeceği anlamına gelmez. Aslında, çok zayıf genelleme yapısına sahip olduğunu gösterir (Richert ve Coelho, 2013).

L2 ve L1 düzenlemeleri, büyük ağırlıkları cezalandırarak bir modelin karmaşıklığını azaltan yaklaşımlardır (Raschka, 2015). L2 ve L1 normlarındaki ağırlık vektörleri w ; aşağıdaki Denklem (5.3) ve (5.4)'teki gibi tanımlanır;

$$L2: \|w\|_2^2 = \sum_{j=1}^m w_j^2 \quad (5.3)$$

$$L1: \|w\|_1 = \sum_{j=1}^m |w_j| \quad (5.4)$$

L2 normda ağırlıkların kareleri alınırken L1 normunda ağırlıkların mutlak değerleri alınır. Düzenleştirme ağırlık vektörünün seyrekleştirilmesini sağlar. Ağırlık vektörünün seyrekleştirilmesi, birbirinden alakasız özelliklerin yer aldığı çok boyutlu özellik vektörüne sahip olduğumuzda oldukça kullanışlı olabilir (Raschka, 2015).

Düzenleştirme uygulayarak, modeli oluşturmak için kullanılan her ekstra değişken için bir ceza puanı verilir. L1 ile mümkün olduğunca az özellik kullanan bir model oluşturulur. Bu, modelin dayanıklılığı için önemlidir: basit çözümler, daha çok durumda doğru sonuç vermeye yatkındır. L2, özelliklerin katsayıları arasındaki varyansı mümkün olduğunca düşük tutmayı amaçlamaktadır. Basitçe ifade edecek olursak: Düzenleştirme, bir modelin çok fazla özellik kullanmasını ve dolayısıyla aşırı uyumluluğu önlemek için kullanılır (Cielen ve ark., 2016).

L2 ve L1 düzenleştirme yöntemleri ağırlıkların sıfır veya sıfıra yaklaşmasını sağlamaktadır. Bu, bazı özelliklerin büyük ölçüde makine öğrenmesi modeli tarafından göz ardı edileceği anlamına gelir. Bu yaklaşım, otomatik özellik seçimi biçimi olarak da görülebilir. Bazı katsayıların tam olarak sıfıra sahip olması, bir modelin daha kolay yorumlanmasını sağlar ve modelimizin sahip olduğu en önemli “özellikleri” ortaya çıkarabilir (Mueller ve Guido, 2016).

5.1.6. Alt küme seçimi

Modelin karmaşıklığını azaltmanın ve aşırı uygunluğun önüne geçmenin alternatif bir yolu, özellikle düzensiz modeller için kullanışlı olan özellik seçimi yoluyla özellik vektörünün boyutunun azaltılmasıdır. Özellik seçimi algoritmalarının arkasındaki motivasyon, hesaplama verimliliğini artırmak veya, alakasız özellik veya gürültüyü kaldırarak modelin genelleme hatasını azaltmak için problemle en alakalı özelliklerin bir alt kümesini otomatik olarak seçmektir. Bu yöntem düzenleştirmenin kullanılamayacağı algoritmalar için yararlı olabilir (Raschka, 2015).

Alt kümeyle seçim yöntemleri, makine öğrenme modelinin sonucuyla daha fazla ilişkili olunan özellikleri seçmeyi hedefler. Bu yöntemler En İyi Alt küme Seçimi Yöntemi ve Aşamalı Alt küme Seçim Yöntemleri olarak ayrılır (James ve ark., 2013). En iyi alt küme seçimi hesaplama maliyeti yüksek bir yöntemdir. Özellik sayısının fazla olduğu durumlarda uygulanması mümkün olmaya bilir. Bu yöntem, özellik sayısına n dersek, en iyi alt kümeyi seçmek için 1'den n 'e kadar olan tüm kombinasyonlara bakar ve en iyi sonucu veren alt kümeyi seçer.

En iyi alt küme seçim yöntemi 2^n ihtimali değerlendirir. Bu, n büyük olduğunda istatistiksel problemlere de neden olabilir. Özellik sayısı arttıkça, genelleme gücü düşük olmasına rağmen, eğitim verileri üzerinde iyi görünen modeller bulma şansınız artar. Bu da, katsayı tahminlerinin aşırı uygunluğuna ve, eğitim ve test sonuçlarının farklı olmasına neden olur (James ve ark., 2013).

En iyi alt küme seçim yönteminin eksiklerinden dolayı Aşamalı alt küme seçim yöntemleri iyi bir alternatif olabilir. Bu yöntemleri ileri aşamalı ve geri aşamalı olarak ayrılır.

En iyi alt küme seçimi, n sayıda özellik için, 2^n alt kümelerini içeren tüm modellerini göz önüne alırken, ileriye doğru aşamalı özellik seçimi yöntemi, çok daha küçük model kümesini göz önüne alır. Bu yöntem, hiçbir özellik içermeyen bir modelle başlar ve daha sonra, tüm öngörücüler modelde oluncaya kadar, modele birer birer “özellik” ekler. Modele en fazla katkı yapan özellikler seçilir (James ve ark., 2013). Seçilen “özellik”, özellik kümesinden çıkarıldıktan sonra geri kalan özellikler üzerinde bir sonraki en iyi özelliği seçme işlemi yapılır. Seçme işlemi istenen sayıda özellikler elde edilinceye kadar devam eder. Yöntemin zaman karmaşıklığı: $1+n(n+1)/2$.

Geriye doğru aşamalı özellik seçim yöntemi tüm özellikleri kullanan bir modelle başlar ve daha sonra kademeli olarak özellikleri azaltır. İstenen “özellik” sayısına ulaşınca kadar özellik azaltma işlemi devam eder. “Özellik” kaldırma kriteri; modelin performansına en az katkı yapan özelliğin kaldırılmasıdır (Raschka, 2015). İleriye doğru aşamalı özellik seçimi yöntemi ile aynı zaman karmaşıklığına sahiptir.

İleri aşamalı ve geri aşamalı özellik seçim yöntemleri en iyi alt kümeyi seçmeyi garanti etmez ancak hesaplama maliyetinin az olması sebebiyle tercih edilir.

Geriye doğru seçim, örnek sayısının, özellik sayısından daha büyük olmasını gerektirir. İleri doğru seçim örnek sayısının özellik sayısından az olduğu durumlarda da kullanılabilir. Özellik sayısının çok büyük olduğu durumlarda tek geçerli yöntem ileri doğru özellik seçim yöntemidir (James ve ark., 2013).

Aşamalı özellik seçimi algoritmaları, ilk d -boyutlu özellik alanını, $k < d$ olan bir k -boyutlu özellik alt uzayına indirgemek için kullanılan açgözlü arama algoritmalarıdır. Açgözlü algoritmalar, kombinatoryal bir arama sorununun her aşamasında lokal olarak en iyi seçimleri yapar ve olası tüm kombinasyonları değerlendiren ve en iyi çözümü bulmayı garanti eden kapsamlı arama algoritmalarının aksine soruna optimal olmayan bir çözüm getirir. Bununla birlikte, hesaplama maliyeti kombinatoryal algoritmalara göre düşük olduğu için tercih edilir (Raschka, 2015).

5.1.7. Uzmanların bilgilerinden faydalanmak

Faydalı özelliklerin elde edilmesinde, uzman bir kişinin bilgisini belirli bir uygulama için kullanmak genellikle önemlidir. Makine öğrenmesinin amacı, insanlar tarafından tasarlanmış bir kurallar dizisi oluşturmak zorunda kalmamaktır. Ancak bu, uygulanacak alan konusunda uzman bir kişinin görüşünü yok saymayı gerektirmez. Genellikle bir alanda uzman kişiler, faydalı bilgilerin tanımlanmasında yardımcı olabilirler (Mueller ve Guido, 2016). Anlamlı özellikler bulmak için genellikle bir uzmana danışmanız veya ilgili alanda daha önce yapılmış bilimsel çalışmaları okumanız gerekir (Cielen ve ark., 2016).

5.1.8. Veri setinin eğitim ve test veri setlerine ayrıştırılması

Veri setinin bir kısmı, kullanılan makine öğrenme algoritmasının eğitilmesinde geri kalan kısmını da eğitilen algoritmanın test edilmesinde kullanılır. Bir veri kümesini eğitim ve test veri kümelerine ayırdığında, öğrenme algoritmasının yararlanabileceği değerli bilgileri muhafaza edildiğinden emin olunmalıdır. Bu nedenle, test kümesine çok fazla veri tahsis edilmez. Bununla birlikte, test setinin küçük olması kullanılan makine öğrenme algoritmasının genelleme güvenirliğini azaltır. Bir veri setini eğitim ve test setlerine bölmek, bu dengeyi korumakla ilgilidir. Uygulamada, ilk veri kümesinin boyutuna bağlı olarak en sık kullanılan ayırma oranları eğitim: test olacak şekilde 60:40, 70:30 veya 80:20'dir. Bununla birlikte, büyük veri setleri için,

eğitim ve test alt gruplarına 90: 10 veya 99: 1 oranlarında ayrıştırılması yaygın ve uygundur (Raschka, 2015).

Daha iyi bir makine öğrenmesi modele oluşturmak için veri setinin bir kısmı doğrulama amaçlı kullanılabilir. Ayrılan bu kısma doğrulama seti adı verilir. Makine öğrenme modeli, eğitim seti kullanılarak oluşturulur ve doğrulama kümesi, modeli ayarlamak veya modelle ilişkili parametreleri elde etmek için test seti olarak kullanılır. Bu işleme çapraz doğrulama(cross validation) adı verilir.

5.1.9. Çapraz doğrulama (cross validation)

Veri biliminde birçok modelleme tekniği vardır. Asıl soru hangisinin daha iyi olduğudur. İyi bir modelin iki özelliği vardır: iyi tahmin gücüne sahiptir ve görmediği veriler üzerinde de genelleme yeteneğine sahiptir. Bunu başarmak için bir hata ölçütü (modelin ne kadar yanlış olduğunu) ve bir doğrulama stratejisi tanımlamamız gerekir (Cielen ve ark., 2016).

Çapraz doğrulama, genelleme performansını eldeki veriyi eğitim ve test setine ayırmadan daha istikrarlı ve kapsamlı bir şekilde artırmak için geliştirilmiş istatistiksel bir yöntemdir. Çapraz doğrulamada, veriyi eğitim ve test setine bölmek yerine, veriler tekrar tekrar bölünür ve model birden çok defa test edilir. (Mueller ve ark., 2016). Çapraz doğrulama için çeşitli yöntemler önerilmesine rağmen, K katlı(K-fold) çapraz doğrulama yaygın şekilde kullanılmaktadır. Bu yöntemde, eğitim seti K eşit parçaya bölünür ve bunlardan biri doğrulama seti olarak kullanılır. Kalan K-1 kısımları eğitim setini oluşturmaktadır. Her seferinde doğrulama seti olarak farklı bir bölüm kullanılır. Bu işlemi K kez tekrar edilir ve doğrulama seti üzerindeki başarısı her defasında hesaplanır. Böylece, K tane başarı değeri elde edilir. Elde edilen k tane başarı değerinin ortalaması modelin başarısı kabul edilir (Murty ve Devi, 2015).

Doğrulama son derece önemlidir çünkü modelinizin gerçek yaşam koşullarında çalışıp çalışmadığını belirler. Açıkça söylemek gerekirse, önerilen modelin bir değeri olup olmadığını gösterir (Cielen ve ark., 2016). Eldeki veri setini bir eğitim ve test setine bölme yerine çapraz doğrulamayı kullanmanın birçok avantajı vardır. Veri seti; eğitim ve test setlerine ayrıldığında, modellemesi zor verilerin eğitimde modellenmesi kolay

verilerin test setinde ya da tam tersi gelme ihtimali vardır. Bu önerilen modelin gerçek performansının çok altında veya üstünde sonuçlar üretmesine yol açacaktır.

Çapraz doğrulama, veri setindeki her örneğin eğitim setinde ve test setinde kullanılmasını garanti edecektir. Dolayısıyla, modelin tüm çapraz doğrulama skorlarının (ve ortalamalarının) yüksek olması için veri kümesindeki tüm örnekler üzerinde iyi sonuç vermesi gerekir. Ayrıca çapraz doğrulama önerilen modelin vereceği başarı sonucunun aralığını belirler (Mueller ve Guido, 2016). Örneğin geliştirilen öğrenme modeli 5 farklı eğitim ve test setlerini kullanarak en kötü %85 en iyi %95 başarı oranı ortaya koydu. Bu model için kullanılan veriye göre %10'luk bir başarı değişikliği olduğu varsayımında bulunulabilir.

Çapraz doğrulama yaklaşımları, hesaplama açısından maliyetli olabilir, çünkü eğitim verilerinin farklı alt kümelerini kullanarak aynı makine öğrenmesi yöntemi birden çok kez test edilir. Bununla birlikte, hesaplama gücünün geldiği nokta, yeniden örnekleme yöntemlerinin hesaplama gereksinimlerini genelde karşılayabilecek seviyededir (James ve ark., 2013).

5.2. Veri Boyutunun Azaltılması

Eldeki verinin boyutunu azaltmanın iki yöntemi vardır; özellik seçimi ve özellik çıkarımı. Özellik seçimi özellik vektöründeki doğru veya ayırt edici özelliklerin kullanılmasını amaçlamaktadır. Bunun için düzenlileştirme yöntemleri ve özellik seçim algoritmaları kullanılmaktadır. Özellik çıkarım yöntemleri de veri boyutunu azaltmaya yardımcı olur. Özellik seçim yöntemlerinden farkı; var olan özellik vektöründen tamamen farklı değerlere sahip ve daha az boyutta özellik vektörüne sahip olmasıdır. Özellik çıkarımındaki temel amaç eldeki özellik vektörünün daha az boyutta bir özellik vektörü ile temsil edilmesidir.

Özellik çıkarımını boyut azaltma bağlamında ele alınacak olursa; ilgili bilgilerin çoğunun korunduğu bir tür veri sıkıştırma yöntemi olarak düşünülebilir. Özellik çıkarımı, genel olarak hesaplama verimliliğini artırmak için kullanılır ve çok boyutlu verilerin sebep olduğu sorunları azaltmaya yardımcı olabilir; özellikle de düzenlileştirme ve özellik seçimi yapılmayan modellerle çalışılıyorsa (Raschka, 2015).

Özellik vektörünün boyutu birçok algoritmanın hesaplama maliyeti için belirgin bir faktördür. Hesaplama maliyetini azaltmak ve daha düşük hacimli eğitim verisi ile işlem yapılmasına olanak sağlamak boyut azaltılmanın temel faydalarıdır. Bununla birlikte, veri kümesindeki gürültüyü kaldırabilir, öğrenme algoritmasının sonuçlarını önemli ölçüde iyileştirebilir, veri kümesini daha kolay kullanabilir ve sonuçların daha kolay anlaşılabilir hale getirilebilir (Marsland, 2015).

Herhangi bir sınıflandırıcı veya regresyon makine öğrenmesi modelinin karmaşıklığı, girdi sayısına bağlıdır. Bu, zaman ve hafıza karmaşıklığını ve böyle bir sınıflandırıcıyı veya regressor'u eğitmek için gerekli eğitim örneklerinin miktarını belirler. Alpaydın, boyut azaltmanın neden önemli olduğunu maddeler halinde sıralamıştır (Alpaydın, 2014):

- Çoğu öğrenme algoritmasında zaman karmaşıklığı özellik vektörünün boyutuna ve örnek sayısına bağlıdır. Özellik vektörünün boyutunun azaltılması zaman karmaşıklığının azaltılmasına ve ihtiyaç olan örnek sayısının azaltılmasına yardımcı olur
- Özellik vektöründeki bir özelliğin gereksiz olduğuna karar verdiğimizde, onu kullanma maliyetinden tasarruf ederiz.
- Basit modeller küçük veri setlerinde daha kararlı sonuçlar verir. Ayrıca, basit modeller daha az varyanta sahiptir; diğer bir deyişle, bir numunenin ayrıntılarına, gürültüye, aşırılıklara ve benzeri durumlara göre daha az değişir.
- Örnekler daha az özelliklerle temsil edildiğinde, örnekler hakkında fikir veren özellikleri daha iyi anlayacağız ve bu bizim örnekler üzerinde yorum yapmamızı kolaylaştıracaktır. Daha az özellik, örnekler üzerindeki gizli ve baskın özellikler olarak yorumlanabilir.
- Özellik vektörü, bir kaç özelliğe sahip bir vektörle temsil edilebilirse grafiksel olarak gösterimi daha kolay olur.

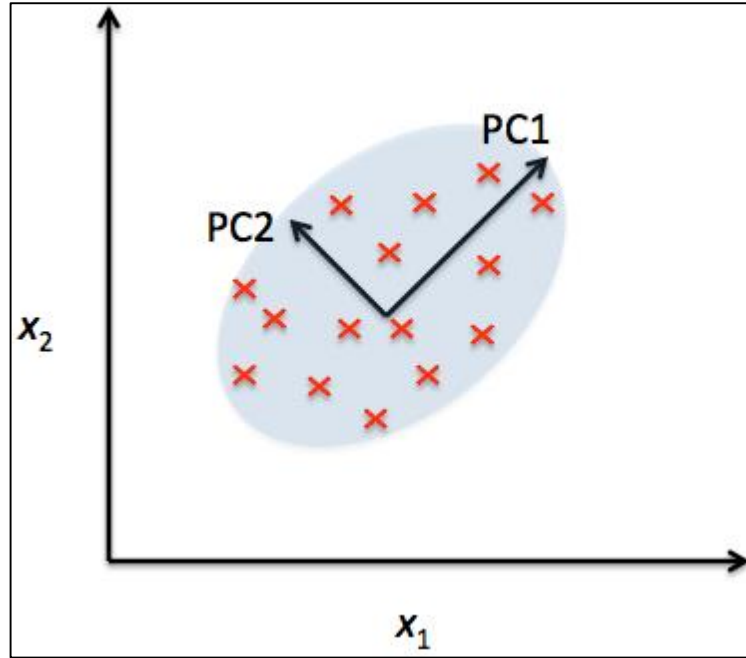
5.2.1. Temel bileşen analizi (principal components analysis-pca)

PCA, birbiriyle ilişkili değişkenlerden oluşan büyük bir kümeyi, temsil edebilen daha az sayıda değişken elde etmeye yarayan gözetimsiz bir makine öğrenme modelidir. PCA genellikle boyut azaltmada kullanılan bir doğrusal dönüştürme tekniğidir. Keşifci veri analizi, borsa işlemlerinde sinyallerin gürültülerinin giderilmesi ve biyoinformatik

alanındaki genom verileri ve gen ekspresyon seviyelerinin analizi PCA'nın popüler uygulamaları arasında bulunmaktadır (Raschka, 2015).

PCA, özellikler arasındaki korelasyona dayalı olarak veri setindeki önemli örüntüleri belirlememize yardımcı olur. Özetle; PCA, yüksek boyutlu verideki maksimum varyant yönlerini bulmayı ve özgün olan veri boyutuna eşit veya daha az boyuta sahip yeni bir alt uzay üzerine projeksiyon etmeyi amaçlamaktadır (Raschka, 2015). Algoritma ilk önce verilerin ortalamasını verilerden çıkararak verileri merkezlendirir ve daha sonra en büyük varyanta sahip olan yönü seçer ve bu yönde bir eksen yerleştirir. Ardından, kalan değişime bakar ve birinci bulunan eksene dik olan ve mümkün olduğunca kalan varyasyonların çoğunu kaplayan başka bir eksen bulur. Daha sonra olası eksenler bitene kadar bunu tekrarlar (Marsland, 2015).

Matematiksel olarak kovaryant matrisinin öz vektörlerinin hesaplanması anlamına gelir. Kovaryans matrisi simetriktir ve girdiler gerçek sayılardır; Bu nedenle, öz vektörler ortogonaldır. En büyük öz değere karşılık gelen öz vektör, birinci ana bileşen; bir sonraki en büyük öz değere karşılık gelen değer, ikinci temel bileşendir, vb. (Murty ve Devi, 2015).



Şekil 5.1. PCA temel bileşenleri.

PCA'yı boyut azaltma için kullanırsak, d uzunluğundaki bir x özellik vektörünü k uzunluklu bir özellik vektörüne dönüştürmek için $d \times k$ boyutunda W dönüşüm matrisi kullanırız. k uzunluğu genelde d uzunluğundan az olur. PCA ait temel bileşenler ölçeklendirmeye karşı çok hassastır. Eğer özellik vektörümüzdeki özellikler farklı değer aralıklarına sahip ise PCA uygulanmadan önce özellik vektörünün ölçeklendirmesinin yapılması gerekir. Aşağıdaki denklemlerde projeksiyon matrisi kullanılarak eldeki d uzunluğa sahip bir vektörün k uzunluğuna azaltılması için gerekli temel adımları verilmiştir. Denklemden (5.5)'te d uzunluğa sahip özellik vektörü yer almaktadır. Denklemden (5.6)'da x özellik vektörü W projeksiyon matrisi ile çarpılarak z özellik vektörü elde edilmektedir. Denklemden (5.7)'de k uzunluktaki özellik vektörü yer almaktadır;

$$x = [x_1, x_2, x_3, \dots, x_d], x \in \mathbb{R}^d \quad (5.5)$$

$$z = xW, W \in \mathbb{R}^{d \times k} \quad (5.6)$$

$$z = [z_1, z_2, z_3, \dots, z_k], z \in \mathbb{R}^k \quad (5.7)$$

Boyut azaltma için kullanılan PCA yöntemini birkaç adımda sıralayalım (Raschka, 2015):

1. d boyutlu veri kümesi standartlaştırılır.
2. Kovaryans matrisi oluşturulur
3. Kovaryans matrisinin öz değerleri ve öz vektörleri bulunur
4. Öz değerler büyükten küçüğe sıralanır ve k tane en büyük öz değere karşılık gelen öz vektörler alınır.
5. W projeksiyon matrisi k sayısındaki öz vektör kullanılarak oluşturulur.
6. d uzunluğundaki tüm özellik vektörler W projeksiyon matrisi kullanılarak k uzunluklu özellik vektörlerine dönüştürülür

5.3. En Yakın Komşu (K Nearest Neighbour Knn) Algoritması

KNN, gözetimli öğrenmede kullanılan en basit sınıflandırma algoritmalarından biridir. Test verilerinin özellik uzayındaki en yakın vektörlerin eşleştirilmesi fikrine dayanmaktadır. KNN sınıflandırma amaçlı kullanılabildiği gibi regresyon amaçlıda kullanılabilmektedir. Eğitim kümesinde yer alan etiketlenmiş örüntüler kullanılarak

yeni bir X örüntüsünün ait olduğu sınıf veya sahip olduğu değer eğitim örüntüleri içerisindeki en yakın komşuluğa sahip örüntülere göre belirlenir. Daha açık bir ifade ile KNN yöntemi; yeni bir örüntünün, X, ait olduğu sınıf veya sahip olduğu değeri en yakın k komşuluğundaki çoğunluğa sahip sınıfa veya değerlere göre belirler (Nilsson, 1998).

KNN yönteminde k değerini seçimi önemlidir (Nilsson, 1998; James ve ark., 2013). Mükemmel verilere için k değeri 1 alınabilir (Segaran, 2007). Bu durumda KNN'i sınıflandırma için kullanıyorsak tahmin edilen sınıf en yakın komşunun ait olduğu sınıftır. Eğer amacımız sayısal bir değer tahmini ise tahmin edeceğimiz değer en yakın sınıfın sahip olduğu değerdir. Göreceli olarak büyük bir k değeri kullanmak ait olduğu sınıfın bulunması istenen X'in yakınında yer alan gürültülerden az etkilenmesini sağlar. Bunun yanında k değerinin büyük olması sonucun kesinliğini azaltır (Nilsson, 1998).

KNN'de iki örüntü arasındaki uzaklık Öklid bağıntısıyla elde edilir. İki örüntüye ait özellik vektörleri $(x_{11}, x_{12}, \dots, x_{1n})$ ve $(x_{21}, x_{22}, \dots, x_{2n})$ ise bu iki örüntü arasındaki uzaklık Öklid bağıntısına göre $\sqrt{\sum_{j=1}^n (x_{1j}-x_{2j})^2}$ elde edilir (Nilsson, 1998).

Özellik vektöründeki değerler heterojen bir yapıya sahip ise bu değerlerin belirli bir aralığa çekilmesi gerekir (Segaran, 2007). Bunun için ölçeklendirme diğer adıyla normalleştirme işlemi yapılmalıdır. En basit ölçeklendirme her bir özelliğin bir sabit ile çarpılmasıdır (Segaran, 2007). Bu durumda iki özellik vektörü arasındaki uzaklığı elde etme için $\sqrt{\sum_{j=1}^n a_j^2 (x_{1j}-x_{2j})^2}$ kullanılır. a_j özellik vektöründeki j'ninci özelliğinin ölçeklendirme çarpanıdır. Bir diğer çözüm Z-skora göre özellik vektöründeki her bir özelliğe normalleştirme yapmaktır. Z-skor değeri standart sapmaya göre ortalamadan ne kadar uzakta olduğunu belirler (Richert ve Coelho, 2013). Orijinal değerden bağımsız olarak Z-skoru alınan değer eğer sıfırdan büyük ise ortalamanın üstünde, sıfırdan küçük ise ortalamanın altından bir değer olduğunu gösterir. Z-skor Denklem (5.8)'deki gibi elde edilir;

$$Z = \frac{x - \mu}{\sigma} \quad (5.8)$$

İki nokta arasındaki uzaklığı ölçen farklı yaklaşımlar vardır;

Manhattan Uzaklığı: İki vektör arasındaki mutlak farkı hesaplar. City-block ve Taxicab uzaklığı gibi farklı isimlendirmeler Manhattan uzaklığı yerine kullanılabilir (Marsland, 2015). İki nokta arası mesafeyi ölçmenin birden fazla yolu vardır ve radikal olarak farklı cevaplar sağlayabilirler. Öklid uzaklığı iki nokta arasındaki kuş bakışı uzaklığı verir. Ancak şartlar bizi farklı uzaklık ölçümü yapmaya zorlayabilir. Örneğin cadde ve sokaklardan oluşan bir şehirde bir noktadan diğer bir noktaya gidilmesi için uzaklık hesabı yapılması gerektiğinde Öklid uzaklığı geçerli olamayabilir (Marsland, 2015). Bu durumda Manhattan uzaklık ölçümü bize anlamlı sonuçlar verecektir. Aşağıda iki nokta arasındaki Manhattan uzaklığını Denklem (5.9)'da yer almaktadır;

$$d=|x_1-x_2|+|y_1-y_2| \quad (5.9)$$

Minkowski Uzaklığı: Öklid ve Manhattan uzaklıklarının genelleştirilmiş halidir. Makine öğrenmesi, örüntü tanıma, bilgisayarlı görü vb. alanlarda analizi yapılacak veriler genellikle çok boyutlu verilerdir. Öklid ve Manhattan uzaklık ölçümleri çok boyutlu veriler üzerinde çalışabilir. Aslında bu iki ölçüt, her iki boyutta da çalışan bir ölçüt sınıfının bir tür örneğidir. Minkowski ölçümü genel bir ölçüm sunar. Denklem (5.10)'da birden fazla boyutta çalışan Minkowski ölçümüne ait eşitlik yer almaktadır;

$$L_k(x,y)=\left(\sum_{i=1}^d |x_i-y_i|^k\right)^{\frac{1}{k}} \quad (5.10)$$

Eğer k değeri 1 olursa Manhattan ölçümünü verir ve k=2 olursa Öklid uzaklığını verir.

Hamming Uzaklığı: Eşit uzunluktaki vektörler arasındaki simge farkını ölçer. Yukarıda bahsi geçen üç uzaklık ölçümünden farklı ölçüm sağlar. Bilgi teorisinde, eşit uzunlukta iki dizgi arasındaki Hamming mesafesi, karşılık gelen simgelerin farklı olduğu konumların sayısıdır. Bir başka deyişle, bir dizgeyi diğerine dönüştürmek için gereken minimum sayıda ikame sayısını veya bir dizgeyi diğerine dönüştüren asgari hata sayısını ölçer (URL-4, 2016).

Örnek:

- "serkan" ve "tarkan" dizgelerinin Hamming mesafesi 2'dir.

- 1011101 ve 1110100, dizgelerinin Hamming mesafesi 3'tür.
- 45677233 ve 45777322, dizgelerinin Hamming mesafesi 4'tür.

5.4. GMACA İle Boyut Azaltma

Her biri 0 ile 9 arasındaki bir rakam içeren görüntü verilerinin boyutu GMACA kullanılarak azaltılmıştır. Boyutu azaltılan veriler KNN sınıflandırma algoritması kullanılarak tanınmıştır. Orijinal görüntüler 8x8 boyutlara sahiptir. Her bir görüntü 64 uzunluğuna sahip bir vektör ile temsil edilmiştir. GMACA ile orijinal boyutu 64 olan vektörler azaltılarak 16 ve 8 uzunluklu vektörlere dönüştürülmüştür. Her bir vektör boyutu için KNN ile tanıma işlemi yapılmıştır. Küçük uzunluktaki vektörlerde tanıma oranı daha düşük çıkmıştır. Elde edilen sonuçlar ileriki bölümlerde detaylıca verilecektir.

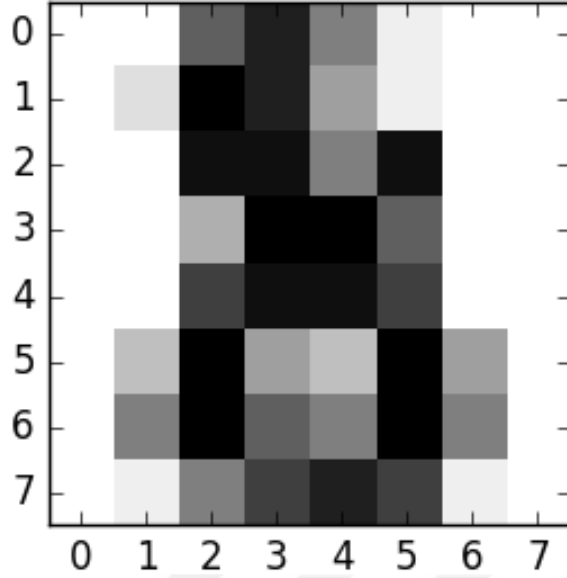
Yapılan çalışma, boyut azaltma olarak adlandırılabilceği gibi özellik çıkarma olarak da adlandırılabilir.

5.5. Kullanılan Veri tabanı

Kullanılan elyazması rakam görüntülerini içeren veri seti, Kaliforniya Üniversitesi Irvine Makine Öğrenmesi veri tabanından alınmıştır (Lichman, 2013). Veri setindeki görüntüler 8x8 boyutundadır ve 0 ile 9 arasında 10 farklı gri seviye değere sahiptir. Veri setindeki 1000 görüntü eğitim için ve 797 görüntü test için kullanılmıştır.

5.5.1. Görüntü verilerinin GMACA'ya uygun hale getirilmesi

Veri setinde yer alan rakam görüntülerinden biri Şekil 5.2'de verilmiştir. Görüntünün boyutu 8x8 ebatlarındadır. Tanıma işlemi yapılmadan önce 64 uzunluklu bir vektöre dönüştürülür.



Şekil 5.2. El yazması rakamlar veri setindeki görüntü örneği

GMACA ikili örüntüler üzerinde kullanılabilir. Bu nedenle görüntünün ikili değerlere dönüştürülmesi gerekmektedir. El yazması rakam görüntüleri, eşik değeri 2 alınarak ikili değerlere dönüştürülmüştür. Daha sonra 8 uzunluklu her bir satır 8 bitlik ikili değerlere dönüştürülmüştür. Orijinal görüntü 8 satır ve 8 sütundan oluşmaktaydı. Dönüştürme işlemi yapıldıktan sonra 8 bit uzunluğunda 8 tane ikili değer elde edilmiştir.

GMACA, elde edilen bu ikili değerler üzerinde kullanılmıştır. GMACA ilgili detaylı bilgilere birinci ve ikinci bölümlerde yer verilmiştir. GMACA'nın ikili değerler üzerinde uygulanması kural vektörü kullanılarak gerçekleştirilir. Kural vektörü istenen sayıda kurallardan oluşur. Ancak kuralların alabileceği değer kullanılan hücresel otomatının yapısına göre değişebilir. Bu çalışmada üç komşu ve iki durumlu hücresel otomatlar kullanıldığı için kural vektöründeki kuralların değeri 0 ile 255 arasında bir değer olabilir. Neden 0 ile 255 arası değer aldığını açıklayalım: Üç hücre sekiz farklı durum alabildiği için kural değerinin ikili tabanda sekiz bitten oluşabilir. Bu, $2^8=256$ farklı kural uygulanabileceği anlamına gelir. Uygulanabilecek kural sayısı ile komşu sayısı arasında üstel bir bağıntı vardır.

Komşu sayısına k diyecek olursak kural sayısı 2^{2^k} olacaktır. Örneğin; kendisi, sol, sağ, üst ve alt hücre komşuluklarından oluşan bir hücresel otomata için uygulanabilecek kural $2^{2^5}=4294967296$ sayısı kadardır.

5.6. Deneysel Sonuçlar

El yazması rakam görüntülerindeki her bir satır 8 bit uzunluklu ikilik tabandaki değerlere dönüştürüldükten sonra GMACA boyut azaltmada kullanılmıştır. 64 uzunluğundaki özellik vektörü 16 ve 8 uzunluklu özellik vektörlerine dönüştürülmüştür. Boyutu azaltılan özellik vektörleri KNN algoritması kullanılarak tanıma işlemi yapılmıştır. KNN ile yapılan tanıma işleminde farklı uzaklık metrikleri kullanılmıştır. Kullanılan uzaklık metrikleri Öklid, Manhattan ve Hamming uzaklık metrikleridir. En iyi tanıma oranları Hamming ve Manhattan uzaklık metrikleri ile elde edilmiştir. Özellik vektörünün boyutu azaldıkça tanıma oranında azalmalar olmuştur.

GMACA ile elde edilen yeni değerler $CA < 222, 221, 192, 68, 222, 221, 192, 68 >$ kural vektörü kullanılarak elde edilmiştir. Özellik vektörünün boyutu 8 uzunluklu vektöre dönüştürüldüğünde % 81 tanıma oranı Hamming uzaklığı metriği kullanan KNN algoritmasıyla elde edilmiştir. Özellik vektörünün boyutu 16 uzunluklu vektöre dönüştürüldüğünde % 86 tanıma oranı Manhattan uzaklığı metriği kullanan KNN algoritmasıyla elde edilmiştir.

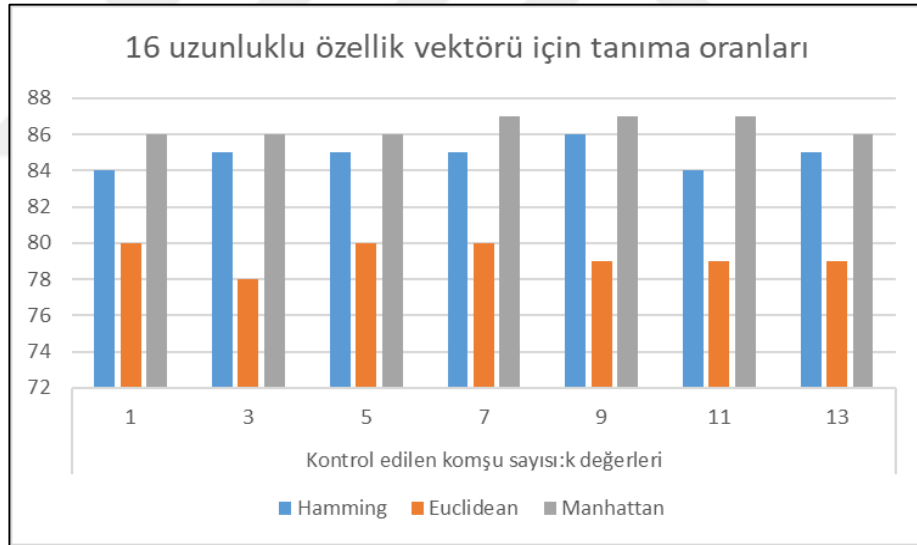
Tablo 5.1. 64 olan özellik vektörünün 16'ya düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklere göre elde edilen tanıma oranları

Farklı metriklere göre başarı oranları	Kontrol edilen komşu sayısı: k değerleri						
	1	3	5	7	9	11	13
Hamming metriğine göre başarı oranları(%)	84	85	85	85	86	84	85
Euclidean metriğine göre başarı oranları(%)	80	78	80	80	79	79	79
Manhattan metriğine göre başarı oranları(%)	86	86	86	87	87	87	86

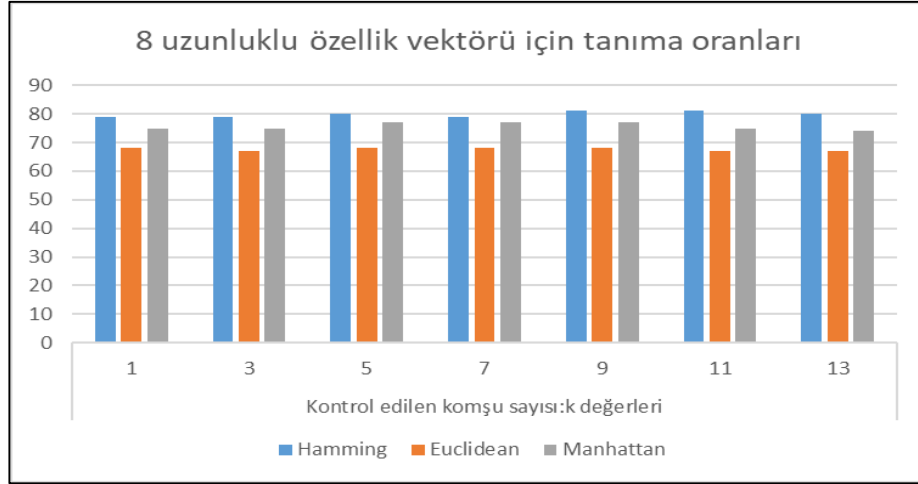
Tablo 5.1 ve Tablo 5.2’de ve şekillerde Şekil 5.3 ve Şekil 5.4’te elde edilen başarı oranları farklı metriklerle ve komşu sayılarına göre verilmiştir.

Tablo 5.2. 64 olan özellik vektörünün 8’e düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklerle göre elde edilen tanıma oranları

Farklı metriklerle göre başarı oranları	Kontrol edilen komşu sayısı: k değerleri						
	1	3	5	7	9	11	13
Hamming metriğine göre başarı oranları(%)	84	85	85	85	86	84	85
Euclidean metriğine göre başarı oranları(%)	80	78	80	80	79	79	79
Manhattan metriğine göre başarı oranları(%)	86	86	86	87	87	87	86



Şekil 5.3. 64 olan özellik vektörünün 16’ya düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklerle göre elde edilen tanıma oranları



Şekil 5.4. 64 olan özellik vektörünün 8'e düşürülerek, KNN algoritmasıyla farklı komşu sayısına ve metriklere göre elde edilen tanıma oranları

6. GENELLEŞTİRİLMİŞ ÇOKLU CEZBEDİCİ HÜCRESEL OTOMATLAR İLE VIDEO GÖRÜNTÜLERDE HAREKET TESPİTİ

Hareket tespit yaklaşımları, video görüntülerdeki noktaların veya bölgelerin ardışık iki zaman dilimi arasındaki renk değişimine odaklanır. Değişim tespit edilen yerlerin hareket içerdiği varsayılır. Videolar ardışık görüntü karelerinden(frame) oluştuğu için bu değişim belirli aralıklardaki görüntü karelerinin birbiriyle karşılaştırılmasından elde edilir. Bu karşılaştırma işlemi, ardışık görüntülerin karşılaştırması olabildiği gibi, sabit bir arka plan görüntüsünün karşılaştırması da olabilir. Görüntünün karşılaştırılması iki farklı zaman dilimine ait görüntü karelerinin aynı konumdaki piksellerin renk değerlerinin karşılaştırılması işlemidir. Karşılaştırma işlemi genelde aynı konumdaki piksellerin renk değerlerinin mutlak farkına göre yapılır. Eğer elde edilen mutlak fark belirlenen eşik değerinin üzerindeyse ilgili piksel bölgesinde hareket olduğunu, eğer mutlak fark eşik değerinin altındaysa ilgili piksel bölgesinde hareket olmadığı sonucunu verir.

Hareket tespiti video görüntü tabanlı uygulamalarda temel işlemlerden biridir. Bunlardan öne çıkanlar; güvenlik kameraları, trafik izleme, videolardaki insanları sayma, cihazların insan hareketleriyle uzaktan kontrolü, video konferanslarda baş ve vücut takibi gibi uygulamalardır. Hareket tespitinin doğruluğu, video görüntü tabanlı tanıma uygulamalarının performansını doğrudan etki etmektedir.

6.1. Arka Plan Çıkarma

Videolardaki hareket tespiti sabit arka plan resmi kullanılarak yapılıyorsa bu işleme “arka plan çıkarımı” adı verilir. Videolardaki görüntü kareleri ile arka plan görüntüsünün aynı konumdaki piksellerin mutlak farkı hesaplanır ve fark belirlenen eşik değerinin üzerindeyse, ilgili bölgede hareket olduğu sonucuna varılır. Arka plan çıkarımına dayalı farklı yöntemler önerilmiştir. Arka plan çıkarımını göre hareket tespiti Denklem (6.1)’deki gibi gösterilir;

$$D(x,y,t)=\begin{cases} 0, & |I(x,y,t)-B(x,y,t)|<eşik \\ 255, & |I(x,y,t)-B(x,y,t)|\geq eşik \end{cases} \quad (6.1)$$

Eşitlikte yer alan büyük harfler D, I ve B sırasıyla; hareketli alanları gösteren görüntü, video görüntüsü ve sabit arka plan görüntüsünü temsil etmektedir. Görüntüdeki piksel konumları x ve y ile ve görüntünün ait olduğu zaman t ile gösterilmektedir. Eğer t zamanındaki I görüntüsünün B arka plan görüntüsüne karşılık gelen pikselin mutlak farkı eşik değerinin altındaysa D hareket tespit görüntüsüne karşılık gelen pikselin değerine 0 atanır, fark eşik değerinin üstündeyse D hareket tespit görüntüsüne karşılık gelen pikselin değerine 255 atanır. Böylece hareket tespit edilen bölgeler beyaz renkle, hareket olmayan bölgeler siyah renkle temsil edilecektir.

6.2. Ardışık Görüntülerin Karşılaştırılması

Videolardaki hareket tespiti ardışık görüntülerin mutlak farkı hesaplanarak da yapılabilir. Bu yöntemle göre hareket tespiti, t anındaki görüntünün t-i zamandaki görüntünün karşılık gelen piksellerin renk değerlerinin mutlak farkı bulunarak elde edilir. Eğer mutlak fark belirlenen eşik değerinin üstündeyse hareket olduğu, değilse hareket olmadığı sonucuna varılır. Ardışık görüntülerin farkına göre hareket tespiti Denklem (6.2)'deki gibi gösterilebilir;

$$D(x,y,t)=\begin{cases} 0, & |I(x,y,t)-I(x,y,t-i)|<eşik \\ 255, & |I(x,y,t)-I(x,y,t-i)|\geq eşik \end{cases} \quad (6.2)$$

D, hareketli alanları gösteren resim, I(x,y,t) t zamanındaki video görüntüsü ve I(x,y,t-i) t-i zamanındaki video görüntüsünü temsil etmektedir. Görüntüdeki piksel konumları x ve y ile ve görüntünün ait olduğu zaman t ile gösterilmektedir. Eğer t zamanındaki I görüntüsünün t-i zamanındaki video görüntüsüne karşılık gelen pikselin mutlak farkı eşik değerinin altındaysa D hareket tespit görüntüsüne karşılık gelen pikselin değerine 0 atanır, fark eşik değerinin üstündeyse D hareket tespit görüntüsüne karşılık gelen pikselin değerine 255 atanır. Böylece hareket olan bölgeler beyaz renkle, hareket olmayan bölgeler siyah renkle temsil edilecektir.

6.3. Arka Plan Elde Edilmesi

Çoğu durumda sabit bir arka plan resmine sahip olunamaz. Bunun için arka plan elde etme yöntemleri geliştirilmiştir. Bu yöntemlerin yaklaşımı oldukça basittir. Arka plan belirlenen n tane önceki görüntünün ortalamasına göre elde edilir. Ortalama elde etme işlemi Ortalama Filtre veya Medyan Filtre yöntemlerine göre elde edilebilir. Ortalama Filtre yöntemine göre arka plan elde etme Denklem (6.3)'teki gibi gösterilir;

$$B(x,y,t)=\frac{1}{n}\sum_{i=0}^{n-1} I(x,y,t-i) \quad (6.3)$$

Medya Filtre yöntemine göre arka plan elde etme Denklem (6.4)'teki gibi gösterilebilir;

$$B(x,y,t)=\text{medyan}\{I(x,y,t-i)\}, \text{ için } i \in \{0, \dots, n-1\} \quad (6.3)$$

Video görüntülerdeki hareketleri tespit etmek için geliştirilen bir diğer yaklaşım optik akış(optic flow) yöntemidir. Bu yöntem yukarıda anlatılan arka plan çıkarma ve ardışık görüntülerin farkını alma yöntemlerinden farklıdır. Optik akış, nesnenin veya kameranin hareket etmesinin neden olduğu iki ardışık görüntü arasındaki görünür hız olarak adlandırılır. Piksel değerlerinin zamana göre değişmediği varsayımına dayanır. Yani t zamanındaki bir piksel değerinin t+1 zamanında konumu bulunarak hareketin yönü ve hızı elde edilir. Optik akış ışık değişimine karşı duyarlıdır. Eğer ardışık iki görüntü arasında ışık farklılığı varsa yöntem başarısız olur. Optik akışın bir diğer zayıf yanı belirli hızdaki hareketleri tespit edebilmesidir. Yani hareket belirli bir sınırın üzerinde ise optik akış yöntemi bu hareketi tespit edemez.

6.4. Önerilen Yöntem

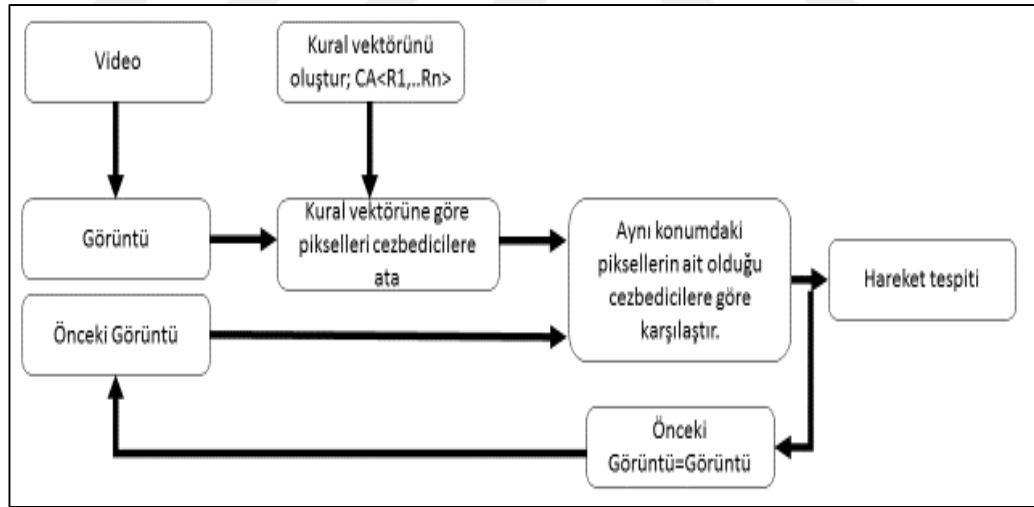
Geliştirilen yöntem ardışık görüntülerin karşılaştırmasına dayanmaktadır. Bu açıdan ardışık görüntülerin farkına dayalı hareket tespit etme yöntemleri altında değerlendirilebilir. Önerdiğimiz yöntemin daha önceki yapılan çalışmalardan en önemli farkı hareket tespitinin GMACA ile kullanılarak yapılmasıdır. Önerilen hareket tespiti yönteminde ardışık görüntüler arasındaki farka dayalı hareket tespiti yapılmıyor; bunun yerine ardışık iki görüntü arasında karşılık gelen piksel konumlarının aynı cezbediciye ait olup olmamasına göre hareket tespiti yapılıyor.

Daha açık bir ifadeyle; ardışık iki görüntünün karşılık gelen pikselleri aynı cezbediciye aitse hareket olmadığı, farklıysa hareket olduğu sonucuna varılıyor.

Önerilen yöntemin adımlarını şöyle sıralanabilir:

- İstenen sayıda cezbedici üreten kural vektörü belirlenir. Kural vektörünün nasıl belirlendiği ikinci bölümde detaylıca anlatılmıştır.
- O anki görüntüyü seçilen kural vektörüne göre cezbedicilerle temsil edilir. Görüntüdeki her bir piksel bir cezbediciye atanır
- Bir önceki görüntü ile o anki görüntüyü karşılaştırılır. Eğer görüntülerin aynı konuma karşılık gelen pikselleri aynı cezbediciye sahipse hareket olmadığı, aynı cezbediciye sahip değilse hareket olduğu sonucuna varılır. Karşılaştırma işlemi CA mantığına göre yapılır. Aynı konumdaki piksellerin durumlarına bakılır. Eğer pikseller durumunda bir değişim varsa bu hareket olduğunu gösterir. Eğer piksellerin durumunda bir değişim yoksa bu hareket olmadığını gösterir.

Önerilen yöntemin blok diyagramı şöyledir:



Şekil 6.1 Önerilen yöntemin blok diyagramı

6.4.1. Ardışık görüntülerin karşılaştırılması

Video görüntülerindeki hareket tespitini ardışık görüntü karelerinin karşılaştırmasını yaparak elde edilir. Videolardaki her bir görüntü karesi ilk önce belirlenen kural vektörüne göre bölümlendirilir. Art arda gelen üç görüntü karesinin karşılaştırılmasıyla hareket tespiti yapılır. Görüntülerin karşılaştırılması görüntülerdeki piksellerin ait olduğu cezbediciye göre yapılır. Aynı konuma denk

gelen ardışık görüntülerin pikselleri aynı cezbediciye sahip ise hareket olmadığı, aynı cezbediciye sahip değil ise hareket olduğu sonucunu verir.

Ardışık üç görüntü karesi $I(x,y,t)$, $I(x,y,t-1)$ ve $I(x,y,t-2)$ olsun. Zaman sıralaması o anki $I(x,y,t)$, bir önceki $I(x,y,t-1)$ ve iki önceki $I(x,y,t-2)$ olacaktır. İlk önce bu üç görüntü karesini kullanarak iki ayrı hareket tespit görüntüsü elde edilir. Birinci hareket tespit görüntüsünü $I(x,y,t)$ ve $I(x,y,t-1)$ görüntü karelerinin karşılaştırmasıyla elde edilir. İkinci hareket tespit görüntüsünü $I(x,y,t-1)$ ve $I(x,y,t-2)$ görüntü karelerinin karşılaştırmasıyla elde edilir.

Hareket tespit görüntüleri $D(x,y,t)$ ve $D(x,y,t-1)$ olsun. $D(x,y,t)$ o anki hareket tespit görüntüsünü, $D(x,y,t-1)$ bir önceki zamandaki hareket tespit görüntüsünü temsil etmektedir. Hareket tespit görüntülerin elde edilmesini Denklem (6.5) ve (6.6)'daki gibi gösterebilir;

$$D(x,y,t) = \begin{cases} 0, & \text{GMACA}(I(x,y,t)) == \text{GMACA}(I(x,y,t-1)) \\ 255, & \text{GMACA}(I(x,y,t)) \neq \text{GMACA}(I(x,y,t-1)) \end{cases} \quad (6.4)$$

$$D(x,y,t-1) = \begin{cases} 0, & \text{GMACA}(I(x,y,t-1)) == \text{GMACA}(I(x,y,t-2)) \\ 255, & \text{GMACA}(I(x,y,t-1)) \neq \text{GMACA}(I(x,y,t-2)) \end{cases} \quad (6.5)$$

Denklemlerde $\text{GMACA}()$ fonksiyonunu görüntüdeki piksellerin durumunu belirlemek için kullanılıyor. $\text{GMACA}()$ fonksiyonu belirlenen kural vektörüne göre belli sayıda cezbedici durum oluşturmaktadır. $\text{GMACA}()$ fonksiyonuna girdi olarak verilen piksel değerleri kural vektörüne göre bir cezbediciye atanır. $\text{GMACA}()$ ile bütün piksellerinin durumları belirlenmiş iki ardışık görüntü karşılaştırılır. Karşılaştırma işlemi aynı konumda yer alan piksellerin durumuna göre yapılır. Ardışık görüntülerdeki aynı konumda pikseller farklı durumlara sahip ise hareket tespiti yapılır.

Yukarıdaki işlemleri yaparak iki farklı hareket tespit görüntüsü elde edilmiştir. Bu görüntülerde hareketler tekrarlı bir şekilde yer alacaktır. Örneğin bir kişinin yürüdüğü bir videoda iki kişi yürüyormuş gibi bir çıktı verecektir. Bu iki görüntüyü kullanarak o anki gerçek hareket tespitini elde edilecektir. Elde edilen $D(x,y,t)$ ve $D(x,y,t-1)$ hareket tespit görüntülerinin kesişimi bize $t-1$ zamanındaki gerçek hareket tespitini verecektir. Gerçek hareket tespit görüntüsü $D_g(x,y,t-1)$ olsun. $t-1$ zamanına ait gerçek hareket tespit görüntüsünün elde edilmesi Denklem (6.7)'deki gibi gösterilebilir;

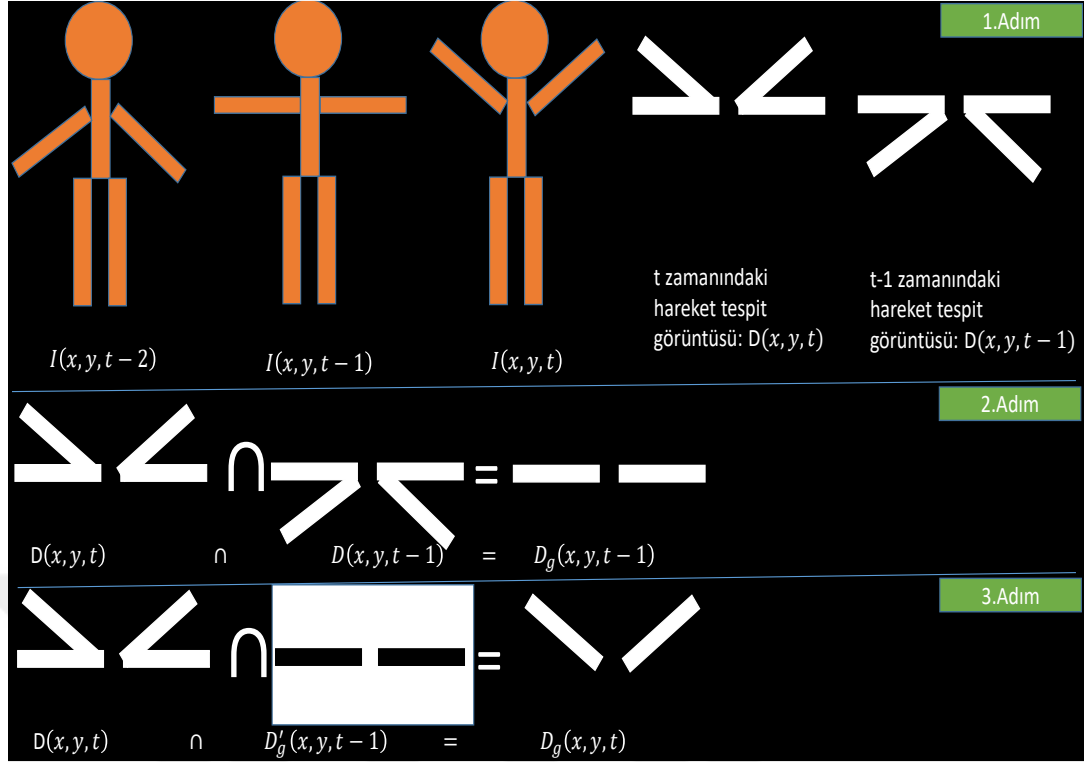
$$D_g(x,y,t-1)=D(x,y,t)\cap D(x,y,t-1) \quad (6.6)$$

t-1 zamanında ki gerçek hareket görüntüsünü kullanarak t zamanındaki gerçek hareket görüntüsünü elde edilecektir. Bunun için $D_g(x,y,t-1)$ görüntüsünün tamlayanını $D'_g(x,y,t-1)$ kullanacaktır. $D'_g(x,y,t-1)$ görüntüsünün $D(x,y,t)$ görüntüsüyle kesişimi t zamanındaki gerçek hareket görüntüsünü verecektir. Gerçek hareket tespit görüntüsünün elde edilmesi Denklem (6.8)'deki gibi gösterilebilir;

$$D_g(x,y,t)=D(x,y,t)\cap D'_g(x,y,t-1) \quad (6.7)$$

Neden iki hareket tespit görüntüsü elde ettikten sonra gerçek hareket tespiti yapılıyor? Bunun nedeni ardışık görüntü karelerinde hareket tespiti yaparken iki farklı zamana ait hareketlerin tek bir hareket tespit görüntüsünde yer almasından kaynaklanıyor. Hareketin yapıldığı zamana t denilirse, hareket tespit görüntüsünde t ve t-1 zamanına ait görüntüler yer alacaktır. Aynı şekilde t-1 ve t-2 zamanlarındaki görüntü kareleri arasındaki hareket tespitini tutan görüntü t-1 ve t-2 zamanlarındaki hareketleri gösterecektir. Elde edilen bu iki hareket tespit görüntüsünün kesişimi, t-1 zamanındaki gerçek hareket tespitini verecektir. t-1 zamanındaki gerçek hareket tespit görüntüsünü kullanarak t zamanındaki gerçek hareket tespit görüntüsünü elde edilir. Bunu t zamanındaki hareket tespit görüntüsü ile t-1 zamanındaki gerçek hareket tespit görüntüsünün tamlayanının kesişimi verecektir. Aşağıdaki şekli inceleyerek bu paragrafta anlatılanlar daha kolay kavranabilir.

Şekil 6.2'de ardışık görüntülerden hareket tespitinin adımları gösterilmektedir. Birinci adımda, ardışık iki zamana ait görüntülerden iki farklı hareket tespit görüntüsü elde edilmiştir. İkinci adımda birinci adımda elde edilen hareket tespit görüntülerinin kesişimiyle, t-1 zamanındaki gerçek hareket görüntüsü elde edilmiştir. Üçüncü adımda t-1 zamanına ait gerçek hareket görüntüsünün mantıksal tersi ile t zamanına ait hareket tespit görüntüsünün kesişimiyle t zamanına ait gerçek hareket görüntüsü elde edilmiştir.



Şekil 6.2. Ardışık görüntü karelerinden t zamanındaki hareketin tespiti.

Hareket tespitinin anlaşılmasını kolaylaştırmak için geometrik şekillerden oluşan ana hatlarıyla bir insan figürü oluşturulmuştur. Oluşturulan figür altı parçadan oluşmaktadır; baş, gövde, iki kol ve iki bacak. Tespit edilecek hareket, kolların aşağı yukarı sallama hareketidir. Hareket tespitinin elde edilmesi üç adımda gösterilmiştir. Şekil 6.2.'de işlem adımları yatay mavi çizgi ile birinden ayrılmıştır ve sağ üst köşeye adım numaraları yazılmıştır. Birinci adımda ardışık zaman dilimleri t, t-1 ve t-2 görüntüleri yer almaktadır. Yapılan hareketleri eskiden yeni doğru tanımlansın: T-2 zamanında kollar aşağı indirilmiş konumda, t-1 zamanında kollar iki yana açık konumda ve t zamanında kollar yukarı kaldırılmış konumdadır. T zamanında yapılan hareket t-1 zamanındaki konumdan farklılığa göre tespit edilecektir. Nihayetinde tespit edilecek hareket, üçüncü adımda görünen $D_g(x, y, t)$ hareket tespit görüntüsüdür. Bu görüntüyü elde etmek için yapılan işlemleri Şekil 6.2.'deki adımlar takip edilerek anlaşılabilir.

Birinci adımda t ve t-1 zamanına ait görüntüler $D(x, y, t)$ ve $D(x, y, t-1)$ elde edilmiştir. $D(x, y, t)$ hareket tespit görüntüsü; $I(x, y, t)$ görüntüsü ile $I(x, y, t-1)$ görüntüsünün karşılaştırılmasıyla elde edilmektedir. $D(x, y, t-1)$ hareket tespit görüntüsü; $I(x, y, t-1)$ görüntüsü ile $I(x, y, t-2)$ görüntüsünün karşılaştırılmasıyla elde edilmektedir.

Karşılaştırma işleminin nasıl yapıldığını yukarıda detaylıca bahsedildiği için burada detaylıca bahsedilmemiştir. Kısaca özetlenecek olursak; ardışık iki görüntünün karşılık gelen pikselleri aynı cezbediciye ait ise hareket olmadığı, farklı ise hareket olduğu sonucuna varıyoruz. $D(x,y,t)$ hareket tespit görüntüsünde kolların yana açık olması ve kolların yukarı doğru kaldırılması hareketleri tespit edilmiştir. Yani iki hareket birden tespit edilmiştir. Tek bir hareket yapılmasına karşın iki hareket tespit edilmesi sorundur. Bu sorunun çözümünü ikinci ve üçüncü adımlarda gerçekleştirilir. $D(x,y,t-1)$ hareket tespit görüntüsünde kolların yana açık olması ve kolların aşağı doğru indirilmesi hareketleri tespit edilmiştir. $D(x,y,t)$ 'de olduğu gibi $D(x,y,t-1)$ görüntüsünde de iki hareket birden tespit edilmiştir. Bu iki hareket tespit görüntüsünden tek bir hareket içeren gerçek hareket tespit görüntüsü ikinci adımda elde edilecektir.

İkinci adımda, birinci adımda elde edilen $D(x,y,t)$ ve $D(x,y,t-1)$ hareket tespit görüntülerinden $t-1$ zamanına ait gerçek hareket görüntüsü olarak adlandırılan $D_g(x,y,t-1)$ görüntüsünü elde edilir. $D(x,y,t)$ ve $D(x,y,t-1)$ görüntülerinin kesişimi bize $D_g(x,y,t-1)$ görüntüsünü verecektir. Bu Denklem (6.9)'daki gibi ifade edebilir;

$$D_g(x,y,t-1) = D(x,y,t) \cap D(x,y,t-1) \quad (6.8)$$

İki görüntünün kesişimi; bu iki görüntüyü “ve” mantıksal operatörüne tabi tutmak anlamına gelir. İki görüntünün “ve” operatörüne tabi tutulması iki görüntüdeki aynı konumdaki piksellerin “ve” operatörüne tabi tutulmasıyla gerçekleştirilir. Normalde mantıksal operatörler ikili değerler üzerinde uygulanır. Ancak elimizdeki görüntüler 0 ve 255 değerlerini içermektedir. Basit bir varsayım; 0 ve 255 değerleri 0 ve 1 kabul edilir. 0 değerine sahip pikseller siyah ile 255 değerine sahip pikseller beyaz ile temsil edilmektedir. İki görüntüde aynı konumdaki pikseller beyaz ise beyaz olacak, diğer durumlarda siyah olacaktır. Elde edilen hareket tespit görüntüsü $D_g(x,y,t-1)$ Şekil 6.2.'de ikinci adım bölümünde yer almaktadır. Üçüncü adımda elde edilen $t-1$ zamanına ait gerçek görüntü kullanılarak t zamanına ait gerçek görüntü elde edilecektir.

Üçüncü adımda, nihayetinde elde edilmesi istenen t zamanına ait gerçek hareket tespit görüntüsü $D_g(x,y,t)$ elde edilir. Bunun için ilk önce $D_g(x,y,t-1)$ tamlayanını alınır.

Yani $D_g(x,y,t-1)$ görüntüsü mantıksal “değil” operatörüne tabi tutulur. Mantıksal “değil” operatörü uygulandığı ikili değerleri tersine dönüştürür. Yani 0 değerini 1’e, 1 değerini 0’a dönüştürür. Görüntüler için söylenecek olursa; 0 değerini 255’e, 255 değerini 0’a dönüştürür. Diğer bir değişle beyaz pikseller siyaha, siyah pikseller ise beyaza dönüşür. Elde edilen $D'_g(x,y,t-1)$ görüntüsü Şekil 6.2.’de üçüncü adım bölümünde yer almaktadır. $D'_g(x,y,t-1)$ görüntüsü ile $D(x,y,t)$ görüntüsünün kesişimi bize t zamanındaki gerçek hareket tespit görüntüsünü $D_g(x,y,t)$ verecektir. $D_g(x,y,t)$ elde edilmesini Denklem (6.10)’daki gösterilebilir;

$$D_g(x,y,t) = D(x,y,t) \cap D'_g(x,y,t-1) \quad (6.9)$$

6.4.2. Hareket tespitinden sonra yapılan görüntü işlemleri

Ardışık görüntü karelerinden hareket tespit yaklaşımının iki zayıf yanı vardır (Yu ve Chen, 2009). Birincisi, üst üste gelen hareketlerde, tespit edilen hareket görüntüsünde boşluklara neden olmasıdır. İkincisi, ardışık iki çerçeve arasındaki tüm değişikliklere duyarlı olduğundan gürültülere sebebiyet vermesidir. Bu sorunların etkisini azaltmak için hareket tespitinden sonra morfolojik işlemler gerekmektedir. Önerilen GMACA tabanlı hareket tespit yöntemiyle elde edilen hareket tespit görüntülerinde çok fazla gürültü yoktur. Yöntem gürültüye karşı gürbüz sonuçlar vermektedir.

6.4.2.1. Morfolojik işlemler

Erozyon:

Erozyonun temel işlevi, ön plan nesnesinin sınırlarını aşındırmasıdır. Bu işlemi yapmak için filtreleme işlemlerinde olduğu gibi yapı elemanını görüntü boyunca gezdirilir. Orijinal görüntüdeki bir piksel (1 ya da 0) sadece yapı elemanı uygulandıktan sonra tüm piksellerin 1 olması durumunda 1 kabul edilir, aksi halde aşınır, yani sıfır yapılır. Böylece, sınırın yakınındaki bütün pikseller yapı elemanının boyutuna bağlı olarak yok olacaktır. Sonuç olarak, ön plan nesnesinin kalınlığı veya boyutu düşer veya görüntüde beyaz bölge azalır. Küçük beyaz gürültüleri kaldırmak, birbirine bağlı iki nesneyi ayırmak için yararlıdır. 5x5 bir yapı elemanı için erozyon etkisi Şekil 6.3’teki gibi olacaktır:



a)



b)

Şekil 6.3. Orijinal görüntü ve, 5x5 yapı elamanın erozyon uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Erozyon sonucu

Yayma:

Yayma işlemi, erozyonun tam tersidir. Orijinal görüntüdeki bir piksel (1 ya da 0) sadece yapı elanı uygulandıktan sonra tüm piksellerin 0 olması durumunda 0 kabul edilir, aksi halde yayılır, yani 1 yapılır. Böylece görüntüdeki beyaz bölgeyi artırır veya ön plan nesnesinin boyutu artar. Normalde, gürültüyü giderme gibi durumlarda erozyonu yayma izler. Çünkü, erozyon beyaz gürültüleri kaldırır, ancak aynı zamanda nesne küçülür. Yayma küçülen nesneyi eski hali döndürme için kullanılır. Ayrıca, bir nesnenin aşınmış bölümlerinin iyileştirilmesinde de yararlıdır. 5x5 bir yapı elamanı için yayma etkisi Şekil 6.4'teki gibi olacaktır:



a)

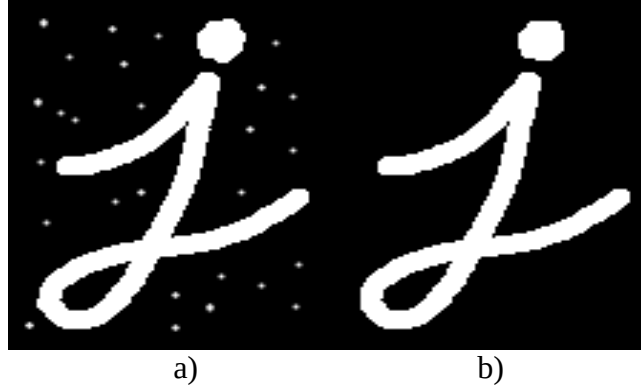


b)

Şekil 6.4 Orijinal görüntü ve 5x5 yapı elamanın yayma uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Yayma sonucu

Açma:

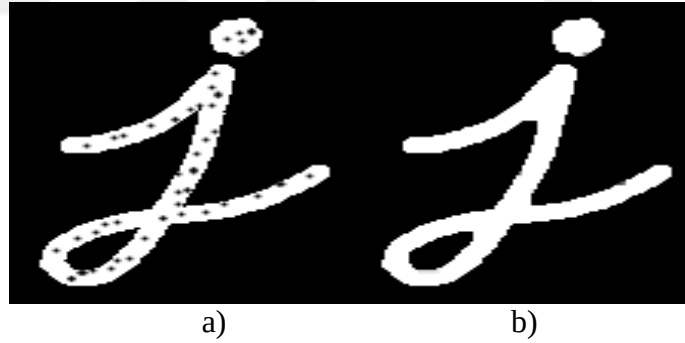
Açma işlemi erozyon işleminden sonra yama işleminin yapılmasıyla elde edilir. 5x5 bir yapı elamanı için açma etkisi Şekil 6.5'teki gibi olacaktır:



Şekil 6.5. Orijinal görüntü ve 5x5 yapı elamanın açma uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Açma sonucu

Kapama:

Kapama işlemi yayma işleminden sonra erozyon işleminin yapılmasıyla elde edilir. 5x5 bir yapı elamanı için kapama etkisi Şekil 6.6'daki gibi olacaktır:



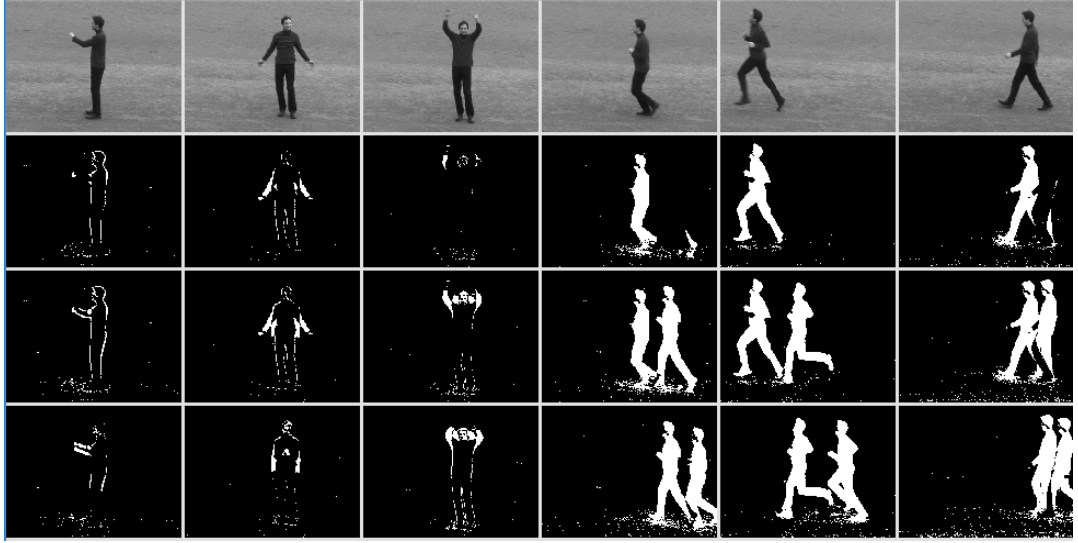
Şekil 6.6. Orijinal görüntü ve 5x5 yapı elamanın kapama uygulanmasıyla elde edilen görüntü a) Orijinal resim b) Kapama sonucu

Bu çalışmada hareket tespiti yapılan görüntülere sırasıyla açma ve kapama işlemleri uygulanmıştır. Açma işlemiyle hareket tespit görüntüsündeki ilgisiz beyaz noktalar giderilmiştir. Kapama işlemi, tespit edilen hareket bölgelerindeki boşlukların azaltılması için kullanılmıştır.

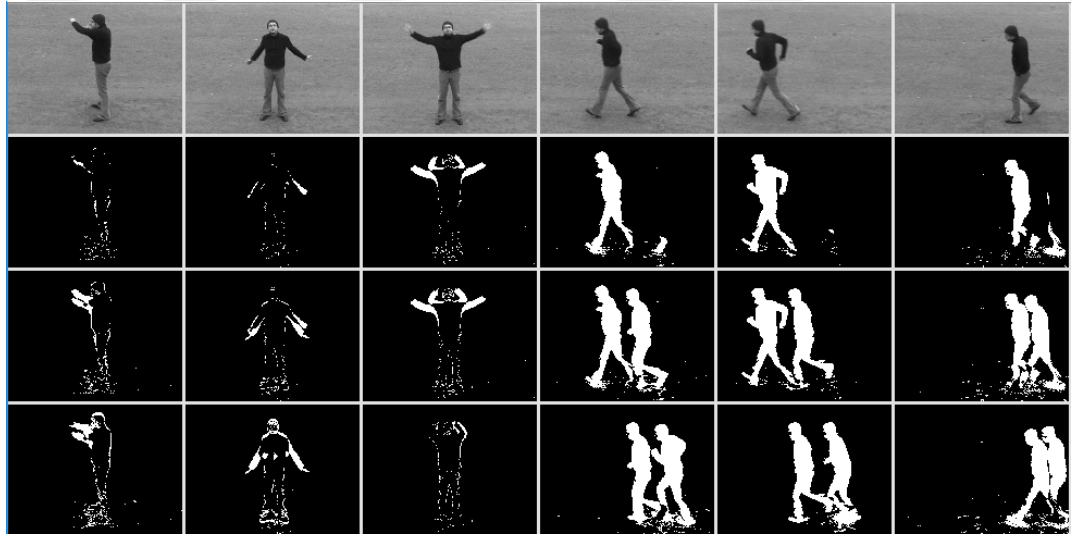
6.5. Deneysel Sonuçlar

Bu bölümde GMACA tabanlı hareket tespit yöntemine ait deneysel sonuçlara yer verilmiştir. Geliştirilen yöntem, KTH veri tabanı üzerinde test edilmiştir. KTH veri tabanında altı farklı hareket biçimine ait videolar vardır. Geliştirilen yöntem birbirinden farklı bu hareketler üzerinde iyi sonuçlar vermiştir. Şekil 6.7 ve Şekil 6.8’de KTH veri tabanında yer alan iki farklı kişiye ait sonuçlar yer almaktadır. KTH veri tabanında yer alan hareketler şunlardır; boks, alkış, el sallama, tempolu koşu, normal koşu ve yürüme. Bu hareketlere ait orijinal görüntülere ve hareket tespit sonuçlarına Şekil 6.7 ve Şekil 6.8’de yer verilmiştir. Şekillerde yer alan resimler yatay ve dikey bölümlere ayrılmıştır. Dikey bölümlerin her birisi ayrı bir hareketi göstermektedir. Yatay bölümler orijinal hareket görüntülerini ve farklı hareket tespit görüntülerini göstermektedir. Yatay ve dikey bölümlendirmeyi sırasıyla satır ve sütun olarak da düşünülebilir. Sütunlarda sırasıyla boks, alkış, el sallama, tempolu koşu, normal koşu ve yürüme hareketlerine ait görüntüler yer almaktadır. Satırlarda sırasıyla orijinal görüntü $I(x,y,t)$, t zamanındaki gerçek hareket tespit görüntü $D_g(x,y,t)$, t zamanındaki hareket tespit görüntüsü $D(x,y,t)$ ve $t-1$ zamanındaki hareket tespit görüntüsü $D(x,y,t-1)$ yer almaktadır. Gerçek hareket tespit görüntüsünün nasıl elde edildiğine yukarıda detaylıca anlatılmıştır.

İkinci satırda, en sonunda elde ettiğimiz hareket tespit görüntüsü yer almaktadır. Bu görüntüye gerçek hareket tespit görüntüsü adı verilmiştir ve sembolü $D_g(x,y,t)$ ile yapılmıştır. İkinci ve üçüncü satırlarda sırasıyla t ve $t-1$ zamanına ait hareket tespit görüntüleri yer almaktadır. Bu görüntülerin semboller, $D(x,y,t)$ ve $D(x,y,t-1)$ ile gösterilmiştir. $D(x,y,t)$ hareket tespit görüntüsü $I(x,y,t)$ ve $I(x,y,t-1)$ görüntülerinin karşılaştırılmasıyla elde edilmiştir. $D(x,y,t-1)$ hareket tespit görüntüsü $I(x,y,t-1)$ ve $I(x,y,t-2)$ görüntülerinin karşılaştırılmasıyla elde edilmiştir. Bu görüntülere dikkat edilirse hareket tespitlerinin tekrarlandığı görülecektir. Bu tekrarlar tempolu koşu, normal koşu ve yürüme hareketlerinin yer aldığı görüntülerde daha belirgin görünmektedir. Hareket tespitinin tekrarlı yer aldığı bu görüntüler kullanılarak birinci satırda yer alan t zamanındaki tekrarsız görüntü elde edilmiştir.



Şekil 6.7. Önerilen yöntemin KTH veri tabanı üzerindeki sonuçları: Kişi1



Şekil 6.8. Önerilen yöntemin KTH veri tabanı üzerindeki sonuçları: Kişi2

Şekil 6.7 ve Şekil 6.8’de KTH veri tabanında yer alan iki farklı kişiye ait hareket tespit görüntüleri yer almaktadır.

6.6. Sonuç

Bu bölümde, bir CA türü olan GMACA tabanlı hareket tespit yöntemi geliştirilmiştir. GMACA'nın hesaplama modeli ve uygulaması detaylıca anlatılmış ve gösterilmiştir. Yapılan literatür taraması, CA'nın görüntü işlemede kullanıldığını ancak GMACA'nın kullanılmadığını göstermektedir. Önerilen yöntem, GMACA tabanlı ilk hareket tespit yöntemidir. CA'nın paralel hesaplama imkanı sağlayan hesaplama modeli, önerilen

yöntemi ilgi çekici hale getirmektedir. Önerilen yöntemi, GMACA'nın ileri görüntü işlemede kullanılabilirliğini göstermesi açısından önemlidir.

Ardışık görüntülerin karşılaştırmasına göre yapılan GMACA tabanlı hareket tespit çalışması gayet iyi sonuçlar vermiştir. Çalışma, GMACA kullanılarak yapılan ilk hareket tespit çalışmasıdır. Elde edilen sonuçlar GMACA'nın hareket tespitinde kullanılabilirliğini göstermektedir. Daha önceki bölümde GMACA görüntü bölümlendirmede kullanmıştık. Yapılan görüntü bölümlendirme, GMACA kullanılarak yapılan ilk görüntü bölümlendirme çalışmasıydı. Bu bölümdeki hareket tespit çalışması, GMACA'nın gerçek zamanlı uygulamalarda da kullanılabilirliğini göstermektedir. Önerilen hareket tespit yönteminin önceki hareket tespit çalışmalarından ayıran en önemli özelliği ardışık görüntülerin farkından yararlanmamasıdır. Bunun yerine ardışık görüntülerin ait olduğu cezbediciye göre hareket tespiti yapmasıdır.

GMACA kural vektörleri iki farklı cezbedici oluşturmaktadır; tek uzunluk döngülü cezbediciler ve çok uzunluk döngülü cezbediciler. Bu çalışmada tek uzunluk döngülü cezbediciler kullanılmıştır. Cezbedici oluşturmak için kullanılacak kural vektörünün uzunluğu minimum sekiz olmalıdır. Eğer kural vektörünün uzunluğu sekizden büyük olduğu durumlarda cezbediciler, görüntü renk aralığında olması gerekir. Bu çalışmada üç komşu ve iki durumlu CA kullanıldığı için kural değerleri 0 ile 255 arası bir değer olabilir.

7. İNSAN HAREKETLERİNİN VIDEO GÖRÜNTÜLERDEN TANINMASI ÜZERİNE YAPILAN LİTERATÜR ARAŞTIRMASI

Video görüntülerdeki insan hareketlerinin tanınması, birçok görevin yerine getirilmesi ile gerçekleştirilir. İnsan hareketlerinin tanınması için ilk önce video görüntüdeki insan nesnesi tespit ve takip edilir. Daha sonra insan nesnesine ait özelliklerin çıkarımı ve bu özellikler kullanılarak harekete ait özelliklerin çıkarımı yapılır. Son olarak basit hareketlerden oluşan etkinliklerin tanınması gerçekleştirilir. Birbirine bağlı bu görevlerin hareket tanımaya uygun bir şekilde yapılması gerekir.

Hareket tanımının ilk adımı, ardışık görüntü çerçevelerindeki farklılıklardan veya insan ait ten rengi ve şekil gibi özelliklerden yararlanarak hareketli insan nesnesinin tespitidir. Nesne tespitini zorlaştıran; ışık yetersizliği, nesnenin gölgesi ve nesnelerin çakışması gibi zorluklarla uğraşılması gerekir. Tespit edilen nesnenin takip edilmesi de zor bir bilgisayarlı görü problemidir. Tespit ve takip edilen nesneden faydalı ve kararlı özelliklerin çıkarımı; nesne hareket özelliklerinin çıkarımı ve temsili için kritik öneme sahiptir. Bilgisayarlı görüde görüntü özelliklerinin ne olduğuna dair önemli bir tanım yoksa da kenarlar, köşeler, ilgi noktaları ve bölgeleri ve renk değerleri görüntülerdeki ayırt edici temel özellikler olarak kabul edilir ve bir sonraki aşama olan hareket temsillerinde kullanılır.

İnsan hareketlerini tanıma üzerine yapılan çalışmalarda dikkat çeken bir durum; hareket tanımaya ait aşamalarda kullanılan yöntemlerin ve yaklaşımların sınıflandırılmasında veya adlandırılmasında genel kabul gören bir kategorize veya adlandırma olmayışıdır. Hareket temsili üzerine yapılan çalışmalarında da bu durum dikkat çekmektedir. Weinland ve diğ., hareket temsil yöntemleri için yaptıkları değerlendirme çalışması, kapsamlı bir tasnif sunmaktadır (Weinland ve ark., 2011). Çalışmada hareket temsil yöntemleri iki ana kategoriye ayrılmıştır; konumsal hareket temsilleri ve zamansal hareket temsilleri. Konumsal hareket temsilleri vücut modelleri, görüntü modelleri ve yerel istatistikler olarak üçe ayrılmıştır. Zamansal hareket

temsilleri hareket üzerine yapılan gözlemlerin zamansal bileşenleri nasıl belirttiklerine göre üçe ayrılmıştır; hareket gramerleri, hareket şablonları ve zamansal istatistikler.

Hareketlerin zaman akışı içerisinde nerede başlayıp nerede bittiğini belirlemek, hareket tanıma kadar önemlidir. Çünkü hareket için yapılacak yanlış sınırlar, hareketlerin tanınamamasına veya yanlış tanınmasına doğrudan etki edecektir. Bu konuda yapılan çalışmalar üçe ayrılır; sınır tespiti, kayan pencereler ve gramer birleştirme.

Seçilen özellik temsillerine göre doğru algoritmanın uygulanması insan hareketleri tanıma başarısı açısından çok önemlidir. Dynamic Time Warping (DTW), Gizli Markov Model (GMM), Dinamik Bayesian Ağı (DBA), Destek Vektör Makinaları (DVM), Relevance Vector Machines (RVMs), Yapay, Sinir Ağları (YSA), Kalman filtresi, ikili ağaç, en yakın komşu algoritması (K-NN), context free gramer ve stocastic context free gramer yöntemleri hareket tanımada kullanılan makine öğrenmesi yaklaşımlarıdır.

7.1. Önemi

Eğer bir gün makineler insanların günlük yaşamdaki hareketlerini otomatik olarak yorumlayabilirlerse, çoğu işin halledilmesi kökten değişecektir. Bu amacı gerçekleştirmek için insan hareketlerini otomatik olarak ayrıştırıp, elde edip ve tanımak olan hareket analizi bilgisayarlı görme alanında üzerine yoğunlukla çalışılmaktadır (Chen ve ark., 2013). İnsan hareketlerini tanımanın amacı, bilinmeyen videolardan devam eden hareket, eylem ve etkinliklerin otomatik analiz edilmesidir. Daha basit bir durum için düşünülecek olursa; sadece bir insan hareketini içeren bir video parçasını, içerdiği eyleme göre doğru eylem kategorisine göre sınıflandırmaktır. Daha genel bir durum için, video görüntülerde süre giden insan etkinliklerin tanınması oluşan tüm eylemlerin başlangıç ve sonunun tespit edilmesiyle gerçekleştirilir (Aggarwal ve Ryoo, 2011).

Hareket analizi ve tanıma çalışmaları başlangıç tarihi itibarıyla eski bir çalışma alanıdır. Ancak gerçek hayat uygulamaları ve performans açısından olgunlaşmamıştır (Ahad, 2013; Ahad, 2011).

İnsan hareketlerinin tanınması ortam gözetleme, eğlence sektörü ve sağlık sistemleri gibi alanlarda uygulama talebinin artmasından dolayı dikkat çekmeye başlamıştır. Ortam gözetlemede anormal eylemlerin otomatik tespit edilmesi potansiyel olarak yasaklı ve tehlikeli eylemlerin doğrudan ilgili kişiler iletilmesi için kullanılabilir. Benzer şekilde eğlence sektöründe insan hareketlerinin tanınması, bilgisayarda tenis oyunu oynayan oyuncuların hareketlerinin otomatik tanınarak avatar oluşturup bilgisayar ara yüzü olarak kullanılabilir. Sağlık sistemlerinde hareket tanıma, hasta hareketlerinin otomatik tanınip iyileştirme süreçlerini kolaylaştırarak hasta iyileştirmeye yardımcı olabilir (Ke ve ark., 2013) .

7.2. Literatür Değerlendirmesi

Video görüntülerdeki insan hareketlerinin tanınması, birçok görevin yerine getirilmesiyle gerçekleştirildiği için literatür değerlendirmesi bu görevler dikkate alınarak oluşturuldu. İlk önce hareketin farklı formlarının, araştırmacılar tarafından nasıl adlandırıldıkları ele alındı. Hareket tanımada temel işlemler olan; hareket tespiti ve takibi üzerine yapılan çalışmalar, ayrı başlıklar altında incelendi. Makine öğrenmesi modeli için kullanılan özellik çıkarma yöntemleri değerlendirildi. Son olarak hareketlerin sınıflandırılması için kullanılan makine öğrenmesi yöntemleri listelendi.

7.2.1. Terim anlamları

Literatürde, hareketlerin sıralı görüntülerdeki kavramsal anlamlarının farklı seviyelere göre adlandırmaları vardır. Nagel bu seviyeleri değişim(change), olay(event), fiil(verb), hadise(episode) ve geçmiş(history) olarak adlandırmıştır. Nagel yaptığı adlandırmaları şöyle açıklamıştır: Değişim gözlenebilir hareketlerdir. Olay kompleks bir davranışın ilk değişimidir. Fiil adlandırılabilir eylemleri tanımlamak için kullanılır. Hadise birleşik eylemler ve geçmişle ilişkili eylemlerin sıralı uzantısı olarak tanımlanmıştır (Nagel, 1988). Gonzalez ve diğ., Nagel'in çalışmasını esas alarak yeni bir adlandırma yapmışlardır; hareket(motion), eylem(action), etkinlik(activity) ve durum(situation). Gonzalez ve diğ., yaptıkları adlandırmaları şöyle açıklamıştır: Hareket sıralı görüntü çerçeveleri içerisindeki insan pozunun veya yerinin değişimidir. Eylem insanın gerçekleştirdiği anlamı olan bir iştir. Etkinlik insanın yaptığı bir veya daha fazla eylemler dizisidir. Durum gerçekleştiği şartlara göre farklı anlamları olan etkinlikler şeklinde tanımlanmıştır (Gonzalez ve ark., 2002).

Fanti eylem-etkinliklerin zamansal seviye ayrımını yapmıştır. Bu seviyeler hareket(movement), eylem(action) ve etkinliktir(activity) yayılır (Fanti, 2008).

Tablo 7.1. Hareketlerin seviyeleri (Fanti, 2008)

Süre	Adlandırma	Gerçek dünya karşılığı
1 Gün 1 Saat	Etkinlik	Spor, yemek pişirme, antrenman, yüzme
1 Dakika	Eylem	Yürüme, koşma, içme, açma/kapama, işaret etme
1 kaç Saniye ve Frame	Hareket	Çekme, itme, sol/sağa adım atma, fırlatma

Fanti, yaptığı adlandırmaları şöyle açıklamıştır: Hareket eğitilmiş veri kullanılmadan elde edilen atomik hareketlerdir. Birkaç görüntülük süre alır ve bazı hareketler üst üste gelebilir. Eylem hareketlerden meydana gelmiş kısa olaylardır ve bunlar bir araya gelince bir etkinliği netice verme ihtimali vardır. Tipik olarak bir eylem bir kaç saniye içerisinde sonlanır. Yürümek, koşmak gibi bazı tekrarlı olaylar yine eylem olarak kabul edilir. Etkinlik en üst seviyede yer alır ve geniş bir zaman dilimine.

Diğer bir isimlendirme Krüger ve diğ., tarafından yapılmıştır; ilkel eylem(primitive action), eylem(action) ve etkinlik(activity) (Krüger ve ark., 2007). Krüger ve diğ., yaptıkları adlandırmaları şöyle açıklamıştır: İlkel eylem eylemleri oluşturan atomik hareketlerdir. Eylem bir görevi tamamlamak için gerekli sıralı ilkel eylemler dizisidir. Etkinlik çevrenin, insan etkileşimine veya nesneye bağlı olan kapsamlı olaylardır. Örneğin tenis sporu için; ilkel eylemler sağ vuruş, sol vuruş, sola koşmak, sağa koşmak vb. hareketlerdir. Eylem top tutmak için yapılan bir dizi ilkel eylemlerden meydana gelir. İlkel eylemlere bağlı olarak topu tutmanın farklı yolları olabilir. Nihayetinde tenis topu ile yapılan bu eylemler; tenis oynama etkinliğidir.

Eylemler sade ve insanlar tarafından gerçekleştirilen bir şeyin basit bileşenlerini oluşturan insan vücut hareketleri şeklide tanımlanabilir. Bir eylem bir hareketle yapılabildiği gibi birden fazla ilkel eylemden meydana da gelebilir. İkisi arasında belirgin bir çizgiden bahsetmek zordur (Gaidon ve ark., 2011).

Aggarwal ve Ryoo karmaşıklıklarına bağlı olarak insan hareketlerini kavramsal olarak dört farklı kategoriye ayırmışlardır; kıpırtı(gesture), eylem, etkileşim ve grup etkinlikleri. Kıpırtı insan vücut uzuvlarının basit hareketleridir ve bir kişinin anlamlı

hareketlerini tarif eden atomik bileşenlerdir. “Kolun gerdirilmesi” ve “dizin kaldırılması” kısırtı için uygun örneklerdir. Tek bir kişinin eylemleri, yürüme, el salama, vurma gibi, birçok kısırtıdan geçici olarak organize edilmiş bileşenler olabilir. Etkileşim iki veya daha fazla insan veya nesne içerir. Mesela iki kişi arasındaki kavga iki insan arasındaki etkileşimdir ve bir kişinin başka bir kişinin çantasını çalması iki insan ve bir nesne içeren insan ve nesne arasındaki etkileşimdir. Grup etkinlikleri birçok kişi ve/veya nesneden oluşan belirli bir grup tarafından gerçekleştirilen etkinliklerdir. Bir grup insanın yürümesi, bir grup insanın toplantı yapması ve iki grubun dövüşmesi grup etkinliği için tipik örneklerdir (Aggarwal ve Ryoo, 2011).

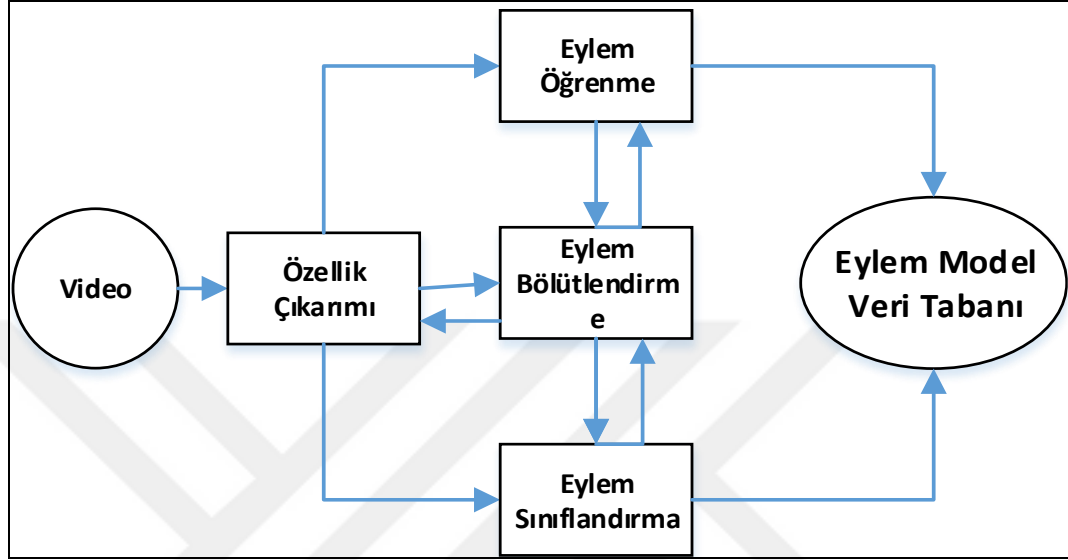
Etkinlik birçok basit eylemin bir araya gelmesinden oluşur. Bir veya daha fazla özne tarafından gerçekleştirilebilir. Basit eylemler olduğu gibi kompleks eylemler vardır. Kompleks eylemler etkinlik olarak da kabul edilebilir. İnsanın veya bir grup insanın davranışları bir etkinlik olabilir veya etkinliklerin bir araya gelmesinden meydana gelebilir.

Eylem ve etkinliklerin farklı oluşumları vardır. Bazı eylemler çok yavaş gerçekleştirilirken bazı eylemler çok hızlı gerçekleştirilir. Araba sürme gibi bazı eylemler durağandır. Koşmak, zıplamak gibi eylemler tüm vücudun hareket etmesini gerektirir. Alkışlamak, boks yapmak, ateş etmek eylemlerinde vücudun üst kısmı, top vururken alt kısmı kullanılır. Kahve içmek veya at sürmek vücudun kullanılan bölgeleri ve hareket bilgileri açısından birbirinden çok farklıdır. Bu nedenle eylem veya etkinliğin birçok yönü vardır ve tek bir yöntem altında ele alınması çok zordur.

7.2.2. Hareket tanıma

Bilgisayarlı görüde tanıma(Recognition), eldeki sayısal görüntü verisi ile özel bir nesneyi, özelliği veya hareketi tanımadır. Hareket tanıma video görüntüler içerisinde hareketi, eylemi veya etkinliği tanıma işlemidir. İnsan hareketlerinin tanınması birçok zorluğu içerisinde barındıran bir problemdir. Hareketler dinamik yapıdadır ve zaman içerisindeki bir noktaya yönelmekle tanınmazlar. Hareketlerin insanlar tarafından farklı yapılması veya aynı kişinin aynı hareketi farklı yapması hareket tanımayı zorlaştırmaktadır. Başarılı bir hareket tanıma için arka plan çıkarımının başarılı bir şekilde yapılması gerekir.

Hareket tanıma eylemlere isim verme işlemidir. Genelde bu hareket, bir eylem fiilinin basit formudur. Teknik olarak eylem insanın bir görevi yaparken ortaya koyduğu hareketler dizisidir. Şekil 7.1, tipik bir eylem tanıma sisteminin veri akış diyagramını temsil etmektedir.



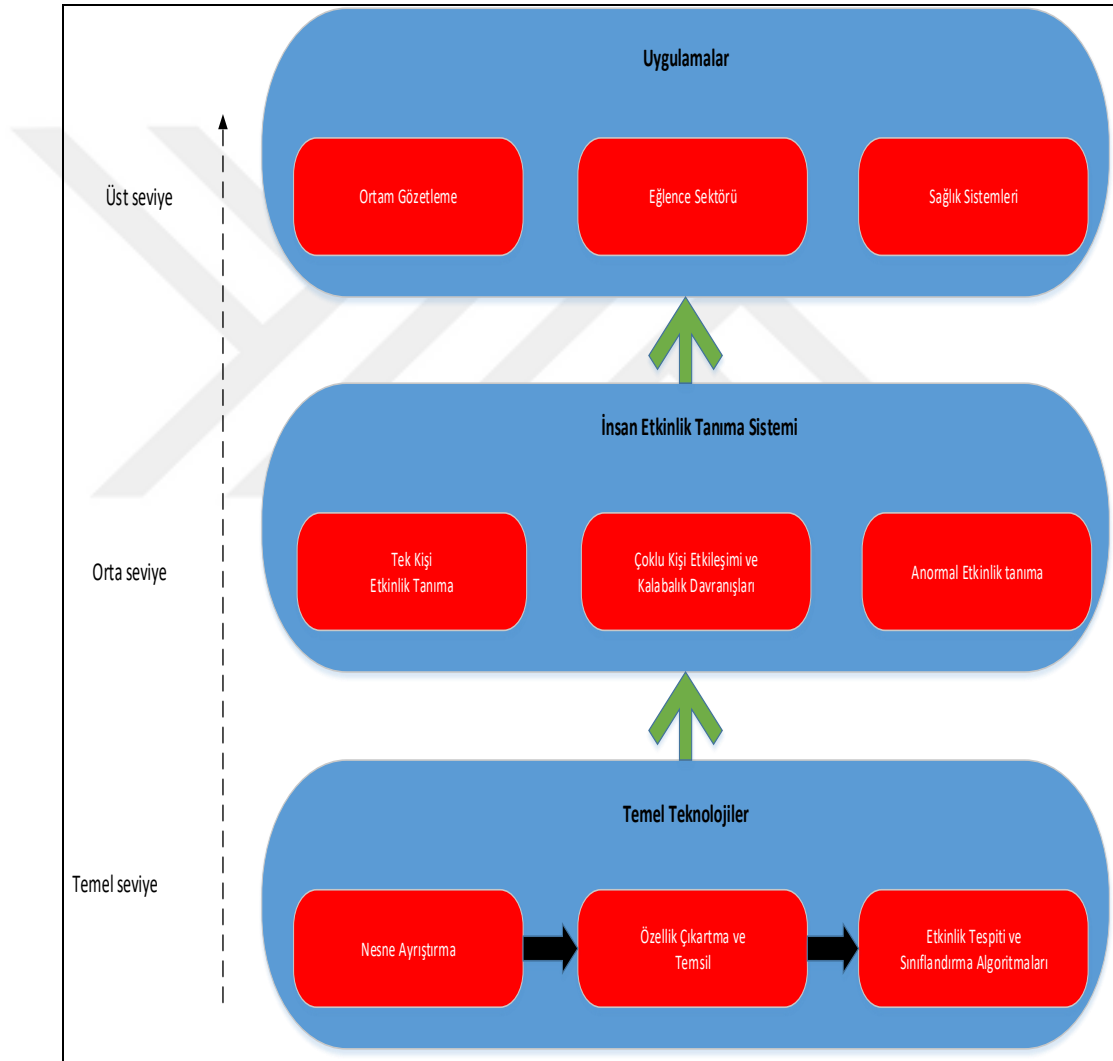
Şekil 7.1. Eylem tanıma sisteminin genel veri akış diyagramı gösterilmektedir (Weinland ve ark., 2011)

İnsan eylemlerine göre ayırt edici olan vücut duruşunun ve hareket işaretlerinin video görüntülerden çıkartılması özellik çıkarma olarak adlandırılmaktadır ve hareket tanımda temel işlemdir. Kompleks vücut modellerinden basit karartı görüntülerine geniş bir yelpazede farklı temsiller kullanılabilir. Her durumda kişinin konumu, bölgesel çakışmalara karşı dayanıklılık, arka plan karışıklığı, gölgeler ve farklı ışıktandırılmalar gibi etkenler göz önünde bulundurulmalıdır. Gelişmiş temsiller farklı elbise ve vücut yapılarından az etkilenirler.

Eylem öğrenme ve sınıflandırma çıkartılan özelliklerden istatistiksel öğrenme adımlarıdır ve gözlenen yeni özellikleri sınıflandırmak için bu özellikler kullanılır. Temel sorun gerçekleştirilen bir eylemin farklı cinsiyette ve ölçülerde kişiler tarafından farklı hız ve stillerde yapılmasıdır. İtme, vurma veya el sallama gibi eylemlerin sınıflandırılması insanlara kolay gelebilir ancak, pratikte bunların uygulanması çok farklı olabilir (Weinland ve ark., 2011). Bu nedenle eylem tanımaya özel bir zorluk; hareketi temsil eden model tanımlarken aynı zamanda bu modelin hareketin değişik formlarına da uyarlanır olması gerekir. Eylem bölümlendirme de tek

bir eylem örneğinin hareket akışının belli bir yerde kesilmesi veri tabanında modellenen eylemle tutarlı olması açısından önemlidir.

Ke ve diğ., yaptıkları literatür araştırmalarına göre insan hareketlerinin tanınmasını üzerine yapılan çalışmaları üç ayrı seviye ayırmışlardır. Birinci seviyede temel teknolojiler, orta seviyede insan etkinliklerinin tanınması ve üçüncü seviyede hareket tanıma ile yapılan pratik uygulamalar yer almaktadır. Şekil 7.2’de bu seviyelere ait görevler yer almaktadır.



Şekil 7.2. İnsan hareketleri tanıma sistemine genel bakış (Ke ve ark., 2013)

Şekildeki temel seviyede üç ana işlem aşaması vardır; nesne ayrıştırma, özellik çıkartma ve temsil, eylem tespiti ve sınıflandırma. İlk önce insan nesnesi video içerisinde ayrıştırılır. Şekil, silüet, renkler, duruş ve vücut hareketleri gibi insan nesnesinin karakteristik özellikleri çıkartılır ve temsil edilir. Ardından değişik insan

eylemelerinin tanınması için çıkartılan özelliklere tespit ve sınıflandırma algoritmaları uygulanır. İnsan hareketlerinin tanınma sisteminin orta seviyesinde üç farklı tanıma sistemi ele alınmıştır; tek kişi etkinliklerinin tanınması, birden fazla kişinin etkileşimi ve kalabalık davranışları ve son olarak anormal davranış tanıma. Son olarak üst seviyede insan hareketlerinin tanınması ile yapılan uygulamalar yer almaktadır; ortam gözetleme, eğlence endüstrisi ve sağlık sistemleri.

Bir hareket tanıma çalışması kavramsal bilgiler ve nesnenin tespiti hususlarını dikkate alması gerekir. Örneğin bardak kahve içme eylemi için, futbol topu şut çekme eylemi için kullanılan nesnelerdir (Ahad, 2013).

Hareket tanıma yöntemlerinde görüntülerden çıkartılan temel özelliklerin yanında eylemlerin semantik özelliklerini çıkartan yöntemlerde kullanılmıştır. Semantik özellikler “Bu hareketi yapmanın anlamı nedir ?” veya “Bir eylemi nasıl anlamalıyız ?” gibi soruların cevaplarını bulmayı amaçlar. Semantik terimi, hareket tanımada bağlamsal anlam çıkarma çalışmalarını ifade eder. Örneğin araba ve yolun aynı görüntüde olması anlamlıyken, zürafa ve mutfağın aynı görüntüde yer alması anlamlı değildir.

Semantik özellikler, aynı sınıftaki farklı eylem oluşumlarının tespit edilmesinde kullanışlıdır. Burada aynı sınıf, aynı eylemin farklı yapılarıdır ve aynı eylemin farklı oluşumlarının birbirlerine nasıl benzediğidir. Şekil 7.3’te görüldüğü gibi aynı eylemi farklı şekilde gerçekleştirebilirler veya aynı kişi aynı eylemi farklı durumlarda farklı yapabilir. İnsanlar giydikleri elbiseler, vücut şekilleri ve görünme açılarının çeşitliliğine göre çok farklı şekilde görünebilirler. Semantik özellikler ortak semantiğe sahip ancak görüntüde farklılık arz eden benzer eylemlerin ayırt edilmesine yardımcı olur.



Şekil 7.3. “Tekmeleme” eylemi. Aynı eylem kamera açısı, elbise, vücut şekilleri gibi farklılıklardan dolayı farklı görünmektedir (Ziaeeferd ve Bergevin, 2015)

İnsanlar bir hareketi tanıırken hareketi yapan kişinin sadece vücut duruşuna göre fikir edinmezler; içerik veya sahne, nesnelerin eylemlerle arasındaki ilişki veya eylemin görsel karakteristiği gibi ek bilgilere ihtiyaç duyarlar. Semantik olmayan yaklaşımlar eylemleri tanımak için sadece silüet, dönüşüm ve optik akış gibi alt seviye özellikleri kullanırlar. İnsanların eylem hakkındaki bilgilerinden istifade etmezler. Semantik olmayan yaklaşımlar görüntünün ve hareketin karakteristiğini elde ederken semantik yaklaşımlar eylem karakteristiğinin anlamsal ilişkisi ile de ilgilenir. Semantik olmayan yaklaşımlar basit eylemler için idealdir ancak karmaşık durumlarda anlamsal ilişkiden mahrum oldukları için başarısızdırlar (Ziaeeferd ve Bergevin, 2015).

Her ne kadar hareket tabanlı/video tabanlı insan hareket tanıma bilgisayarlı görme ve örüntü tanıma da yoğunlukla çalışılmakta ise de son zamanlarda yapılan çalışmalarda durağan görüntüler içerisinde hareket tanıma araştırılmaya başlamıştır. Şekil 7.4'te görüldüğü gibi birçok eylem tek bir görüntüden(hareket ve video sinyali içermeyen) tasvir edilebilir ve kişinin perspektifine göre kolayca anlaşılabilir. Bu gelişme durağan görüntülerde hareketlerin analizini ve tanınması için önerilen hesaplama algoritmalarının gelişimini desteklemektedir. İnternet üzerindeki büyük sayıdaki tek görüntü formatları düşünüldüğünde insan hareketlerinin bu görüntüler üzerinden analiz edilmesi değerli hale gelmektedir.

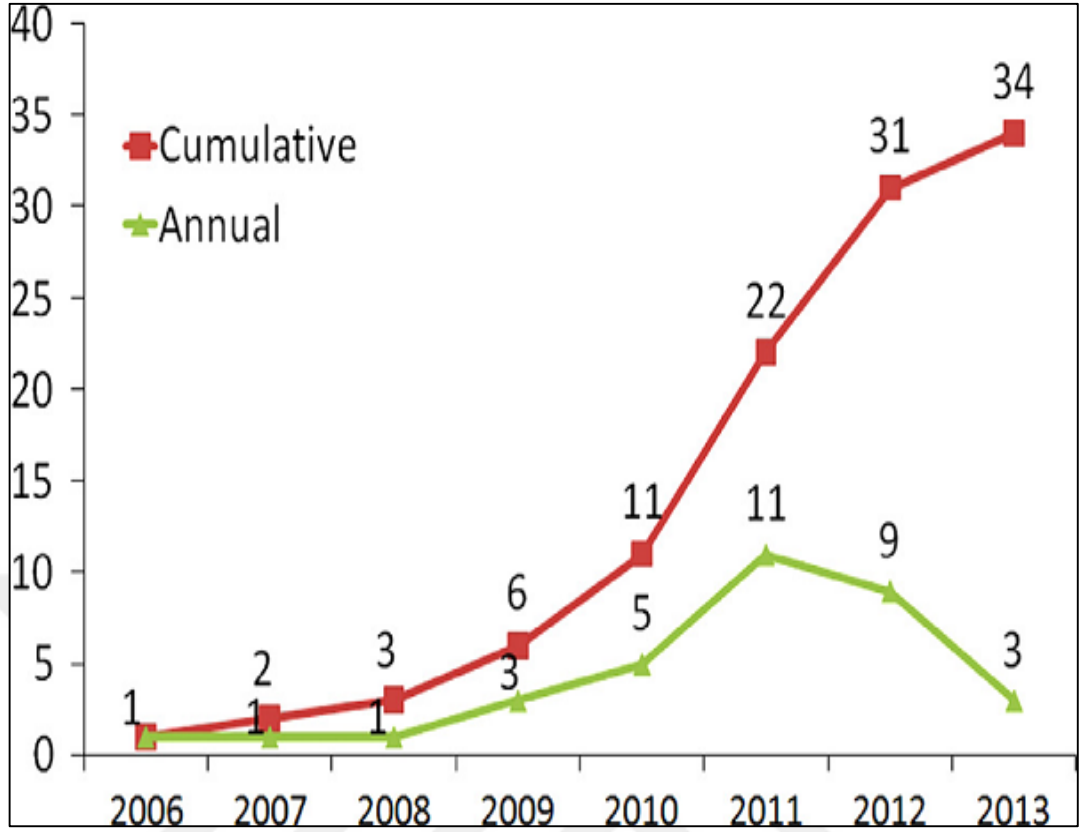


Şekil 7.4. Durağan görüntü tabanlı hareket tanımaya birkaç örnek. İlgili eylemi belirlemek için tek bir görüntü yeterlidir (Guo ve Lai, 2014)

Geleneksel video tabanlı hareket tanımayla karşılaştırıldığında durağan görüntü tabanlı hareket tanıma ayrı niteliğe sahiptir. Örneğin durağan görüntülerde hareket yoktur; bu nedenle geleneksel yöntemler için belirlenen uzay-zaman özellikleri ve geliştirilen özellik çıkarma yöntemleri durağan görüntülere uygulanamazlar. Hareket belirtileri olmadığından ve arka planın çok karmaşık olabileceğinden, durağan görüntülerde insan nesnesini arka plandan ayırtmak kolay değildir. Bahsi geçen durumlardan dolayı durağan görüntüler kullanılarak yapılan hareket tanımanın uğraşması gereken yeni zorluklar ortaya çıkmaktadır.

Durağan görüntü tabanlı hareket tanımanın kullanışlı olacağı uygulama alanları vardır. Guo ve Lai durağan görüntü tabanlı hareket tanımanın faydalarını dört ana kategoriye ayırmışlardır (Guo ve Lai, 2014). Birincisi: Hareket içeren görüntülere açıklayıcı not etiketlemektir. İnternette devasa miktarda durağan görüntü mevcuttur ve her geçen gün yeni görüntüler birikmektedir. Şekil 7.4'te görüldü gibi internet üzerindeki durağan görüntülerden hareket tanıma, görüntüdeki hareketlerin etiketlenmesine yardımcı olabilir. İkincisi: Görüntü veri tabanlarından eylem ve davranış tabanlı görüntü aramadır. Durağan görüntüler kullanılarak yapılan otomatik hareket tanıma, hareket sorgularına göre yapılan arama işlemlerine yardımcı olabilir. Üçüncüsü: Video tabanlı yapılan hareket tanıma görüntü karesi (frame) sayısının azaltmasına yardımcı olabilir. Durağan görüntü tabanlı hareket tanıma bazı eylemler için yüksek tanıma oranına sahip olduğunda, bu eylemleri içeren uzun video parçaları birkaç görüntü karesi ile temsil edilebilir ve böylece hareket tanıma performansına etki etmeden az sayıda bilgiyle tanıma gerçekleştirilebilir. Dördüncüsü: Geleneksel insan hareketlerini tanıma insan bilgisayar etkileşiminde uygulandığı gibi durağan görüntüler kullanılarak yapılan insan hareketlerinin tanınması uzun zaman periyoduna gereksinim duymayan dokunma, düşünme ve gülümseme eylemleri için kullanılabilir.

Pratikte kullanışlı uygulamaları olmasına rağmen durağan görüntülerden hareket tanıma araştırmaları 2006 yılından itibaren yaygınlaşmaya başlamıştır. İlgi çekici özelliği, araştırmacılar tarafından fark edilmediğinden dolayı 2006'dan sonra durağan görüntüler üzerine yapılan hareket tanıma çalışmaları fazla görülmemektedir. Çoğu çalışma son zamlarda yayınlanmıştır. Şekil 7.5'teki grafik durağan görüntüler üzerine yapılan yayınlarının eğilimini göstermektedir.



Şekil 7.5. Durağan görüntüler üzerine yapılan hareket tanıma yayınlarının yıllık ve birikimli eğilimi (Guo ve Lai, 2014)

2011 yılında zirveye ulaşan yayın sayısının düşüşe geçmesinin bir sebebi tanıma başarısında ulaşılan darboğaz olabilir. Durağan görüntülerdeki hareket tanıma sorunları üzerine yoğunlaşacak yeni fikirler ve yaklaşımlara ihtiyaç vardır (Guo ve Lai, 2014).

Görüş açısı insan hareketleri tanımda çok önemli dar boğazlardan biridir. Şekil 7.6'da görüldüğü gibi ortam izleme kameraları genelde rasgele açıyla görüntü alır. Bu nedenle yapılacak uygulamaların farklı görüş açılarına göre uygulanabilir olması gerekir. İnsan hareketlerini analiz eden yöntemlerin çoğu tek bir görüş açısına göre uyarlanır. Görüntüdeki insanlar çoğu ya yüzü kameraya dönüktür ya da kameraya paralel bir şekilde hareket eder. Gerçek dünyada böyle bir kısıtla görüntü almak imkânsızdır.



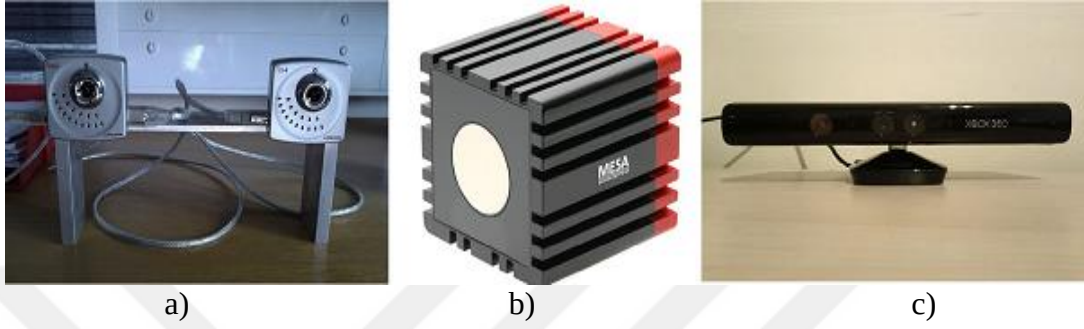
Şekil 7.6. UCM veri setinde ortam gözetleme sahnesi (Gross ve Shi, 2001)

Görüş açısı kısıtlamalarından dolayı çok sayıda insan hareketlerini tanıma yöntemi geniş kapsamlı olarak kullanılamamaktadır. Son yıllarda görüş açısında bağımsız poz kestirimi, eylem temsili ve tanıma gibi görüş açısı etkilerini azaltan çalışmalar yapılmıştır. Ji ve Liu yapılan bu çalışmaları iki ayırmışlardır; görüş açısından bağımsız poz temsili ve kestirimi; görüş açısından bağımsız eylem temsili ve tanıma. Bu ikisi arasındaki fark, birincinin görüntü dizilerinden 3D modeller oluşturmaya çalışırken ikincisi insan eylemlerini çıkarmak ve tanımaya yoğunlaşır (Ji ve Liu, 2010).

Şimdiye kadar ele alınan çalışmaların çoğu yoğunluk görüntüsü veya yoğunluk videosu verilerini girdi olarak kullanan çalışmalardır. Yoğunluk görüntüleri zengin bilgi içermesine karşın ışık değişimine duyarlıdır ve arka plandan nesne çıkarımı karmaşık arka planlarda zordur. Çıkartılacak özellikler, nesnenin genel şeklinden çok yüzeysel doku yapısına göre elde edilmesi durumlarında yoğun görüntüler sınırlı bilgiye sahiptir.

Derinlik görüntülerin yoğunluk görüntülere göre avantajı vardır. Öncelikli, ışık değişiminden etkilenmezler. İkinci olarak görüntünün üç boyutlu yapısını sağladıklarından dolayı arka plan çıkarımı, nesne ayrıştırma ve hareket kestirimi gibi işlemleri önemli ölçüde kolaylaştırırlar.

Derinlik görüntüleri derinlik algılayıcıları tarafından elde edilirler. Derinlik görüntüler yoğunluk görüntüler gibi ön işleme ihtiyaç duyabilirler. Bilgisayarlı görmede uygulanan üç ana derinlik algılayıcı teknoloji vardır; stereo kameralar, time-of-flight (ToF) kameralar ve kodlanmış ışık (structured light) kameralardır. Şekil 7.7’de üç farklı derinlik kamerasına ait görüntüler vardır.



Şekil 7.7. Derinlik kamera örnekleri. a) stereo kamera, b) time-of-flight, c) kodlanmış ışık (Microsoft Kinect) (Chen ve ark., 2013)

Stereo yapay görme, insan görme sisteminden esinlenerek yapılmıştır. İki veya daha fazla görüş açısından alınan görüntülerden 3D görüntüler oluştururlar. Stereo geometri hesaplamalarının karışıklığından dolayı stereo görüntülerden derin haritaların inşa edilmesi hale zorluğunu devam ettirmektedir. Derinlik haritalarının oluşturulması için kullanılan yoğunluk görüntüleri ışık değişimine duyarlı olduklarından dolayı üçgenleme eşleştirmesini zorlaştırmaktadır. Bu nedenle stereo görüntülerden derinlik haritalarının inşa edilmesi çoğu gerçek dünya uygulamaları için uygun değildir.

Time-of-flight (ToF) kameralar nesne yüzeyine gönderdiği darbeleri ışıkların geri dönüş zamanı ve ışık hızına göre nesnenin uzaklığını elde eder. ToF kameralar yüksek görüntü karesi oranlarına ulaştıklarından dolayı geçek zamanlı uygulamalarda elverişlidirler. ToF kameralarının en önemli eksiği yüksek fiyatlı ve düşük çözünürlükte olmasıdır.

2010 Kasım ayında Microsoft, kişisel tüketiciler alabileceği fiyata sahip Kinect adını verdiği görüntüleme cihazını piyasaya sürdü. Microsoft Kinect Guinness rekorlarına en hızlı satılan elektronik eşya olarak girdi. Kinect birer tane RGB kamerası ve derinlik sensöründen oluşmaktadır. Derinlik sensörü 640X480 çözünürlükte saniyede 30 görüntü karesi sağlayabilmektedir. Kameranın algılayabildiği uzaklık skalası 0.8 ile

3.5 metre aralığındadır. Tablo 7.2’de derinlik kameralarını özellik karşılaştırılması verilmiştir (Chen ve ark., 2013).

Tablo 7.2. Derinlik kameralarının karşılaştırılması (Chen ve ark., 2013)

Algılayıcı tipi	Stereo kamera	Time-of-flight (ToF)	Kodlanmış ışık
Çözünürlük	Yüksek: 640X480 veya daha fazla	Düşük: 64X48 ile 200X200 aralığı	Yüksek: 640X480
Hız	Yavaş	Hızlı	Hızlı
Kapsam alanı	Kamera özelliğine bağlı	5 ile 10 metre arası (iç ve dış mekan)	0.8 ile 3.5 metre arası (iç mekan)
Derinlik çözünürlüğü	Kamera özelliğine ve çözünürlüğüne bağlı	5 mm den az	1 cm den az
Görüş alanı	Kamera lensine bağlı	Yaklaşık yatay 43 dikey 69 derece açı	Yaklaşık yatay 43 dikey 57 derece açı
Derinlik haritasında boşluklar	Evet	Hayır	Evet
Fiyat	Ucuz	Pahalı	Ucuz
Işık duyarlılığı	Evet	Hayır	Hayır

İnsan hareketlerinin ve eylemlerinin görüntüler içerisinde tanınması üzerine yapılan çalışmaları, kullandıkları verilere göre üç ana başlığa ayrılabilir; yoğunluk görüntüsü tabanlı, derinlik kamera tabanlı ve durağan görüntü (still image) tabanlı yöntemler.

7.2.3. Görüntülerde hareketli nesne tespiti

Hareketli nesnelerin tespiti video görüntü analizinin ilk adımını oluşturmaktadır. Bir nesnenin takip edilebilmesi için ilk önce tespit edilmesi gerekir. Işık değişimi, nesnenin gölgesi, nesnelerin görüntülerinin çakışması ve karmaşık arka planlar nesne tespitini zorlaştırmaktadır.

İlk önce ön planın arka plandan ayrıştırılması gerekir. Bunun için arka planın elde hazır olması veya görüntüden elde edilmesi gerekir. Arka plan elde edildikten sonra görüntünün tamamından arka plan çıkarılarak hareketli nesne tespit edilir. Nesne bölgesi tespit edildikten sonra morfolojik ve etiketli bağlantılı bölge işlemleri uygulanarak nesnedeki gürültüler giderilmeye ve eksik yerler iyileştirilmeye çalışılır. Hareketli nesne takip yöntemleri genel olarak dörde ayrılır; arka plan çıkarımı

(Barnich ve Droogenbroeck, 2011; Maddalena ve Petrosino, 2008) istatistiksel yöntemler (Stauffer ve Grimson, 1999; Sheikh ve Shah, 2005), zamansal değişimler (Yu ve Chen, 2009) ve optik akış (Samsudin ve Ghazali, 2012).

Arka plan çıkarımı t zamanındaki görüntünün her bir pikselinin arka plan görüntüsündeki karşılık gelen pikselinden mutlak çıkarımına uygulanan eşik değerine göre elde edilir. Mutlak çıkarımda elde edilen sonuç eşik değerinden fazla ise bu bölgenin hareketli bir bölge olduğu varsayılır.

İstatistiksel yöntemler arka plan çıkarımını, istatistiksel işlemler kullanarak elde edilmesini esas alırlar. İstatistiksel yöntemler basit arka plan çıkarımı yönteminin arka planın değişken olduğu durumlardaki eksikliklerinin gidermek için arka planın dinamik olarak güncellenmesini istatistiksel yöntemler kullanarak aşamayı hedefler. Işık değişimi, gölge ve gürültüden kaynaklanan zorluklara karşı daha iyi sonuçlar verdiği için istatistiksel yöntemler daha popüler hale gelmektedir.

Zamansal farklılıklar yöntemi piksel bazında iki veya üç sıralı görüntü çerçevesindeki farklılıkları kullanarak hareketli bölgeyi tespit eder. Arka planın çok değişken olduğu durumlarda iyi sonuç vermesine karşın nesnenin tek düze yapıya sahip olduğu, yavaş hareket ettiği veya kameraya dikey hareket ettiği durumlarda nesnenin tamamının elde edilmesinde başarısızdır.

Optik akış aynı sahnenin farklı zamanlardaki görüntü sıralamalarındaki değişimine odaklanır. Dinamik arka planlarda daha iyi sonuç verir. Ancak hesaplama maliyeti diğer yöntemlere göre daha fazladır.

7.2.4. Hareketli nesnelerin takibi

Üst seviye bilgisayarlı görü problemi olan hareketli nesnelerin takibi insan hareket analizinde önemli bir yere sahiptir. Nesne takibinin amacı videolar içerisindeki ardışık görüntüler arasında hedefin takip edilmesidir. Literatürde nesne takibi yaparken karşılaşılan görüntüdeki gürültüler, başka nesnelerin takip edilen nesne ile çakışması, sahnedeki kalabalıklar, takip edilen nesnenin görüntüleme açısının değişmesi, takip edilen nesnede hareket ve ışık seviyesinin farklılıklarından kaynaklanan değişimlerin

ortaya çıkardığı zorlukların üstesinden gelmek için birçok nesne takip algoritması önerilmiştir.

Nesne takip yöntemleri özelliklerine göre üç kategoriye ayrılabilir; özellik tabanlı, model tabanlı ve optik akış tabanlı yaklaşımlar. Özellik tabanlı yaklaşımlar az sayıda belirgin özelliklerin ardışık görüntüler içerisinde takip edilmesi amacıyla geliştirilmiştir. Görüntülerdeki belirgin özelliklerin olduğu bölgelerin tespitini ve takip edilen nesnenin eşleniğinin olduğu bölgenin saptanmasını gerektirirler. Bu yaklaşım görüntü karelerindeki özgül uyuşmaları elde etmeye amaçladığından özelliklerin özgünlüğüne göre yanlış nesne takibinden korunmuş olurlar. Çoklu hipotez takibi(multiple hypothesis tracking), gizli Markov modeli, yapay sinir ağları, özel filtre kullanan yöntemler, Kalman filtreleme ve ortalama kayma(mean shift) yöntemleri özellik tabanlı nesne takip yöntemleridir.

Model tabanlı takip yöntemleri aslında özellik tabanlı yöntemlerin bir örneğidir. Gruplama, tasarlama ve oluşturma farklılığından dolayı ayrı olarak değerlendirilir. Çoklu nesne takip uygulamalarında nesnelerin ikili temsillerin öncelikli olarak elde edilmesi buna örnek olarak gösterilebilir.

Optik akış nesnenin zaman içerisinde nasıl değiştiğini tanımlayan vektör alanıdır. Kamerayla gözlemlenen üç boyutlu hareketli alanın iki boyutlu izdüşümünün kestirimini gerektirir. Çakışma ve diyafram etkisi zorluklarını aşmak için optik akış tabanlı farklı yaklaşımlar geliştirilmiştir. Ancak bu yöntemlerin hesaplama yükünün fazla olması iyileştirme taleplerini ortaya çıkarmıştır (Zhou ve ark., 2009).

7.2.5. Özellik çıkarımı

Özelliklerin ne olup ne olmadığı hakkında açık bir tanım yoktur. Ancak görüntü işlemede kenar, köşe, ilgi noktaları, blob (ilgi bölgeleri) vs. tipik görüntü özellikleri olarak kabul edilirler. Bu özelliklerden bir veya birkaçı görüntüden çıkarılır ve sonraki analiz işlemleri için kullanılır.

Kenar bulma çoğu uygulama için önemlidir ve hareket analizinde kullanılan kenar tabanlı hareket temsilleri vardır. Kenar bulma için birçok yöntem önerilmiştir;

Gradient, Canny, Sobel, Prewitt, Harris-Stephens, Robert ve Laplacian en çok kullanılan kenar bulma operatörleridir.

Köşe noktaları bazen ilgi noktası olarak da adlandırılabilir. Köşe noktaları kullanılarak hareket temsili oluşturan çok sayıda çalışma vardır. Köşe bulma için birçok yöntem önerilmiştir; Laplacian of Gaussian (LoG), Difference of Gaussians (DoG), Wang ve Brady köşe bulma yöntemleridir.

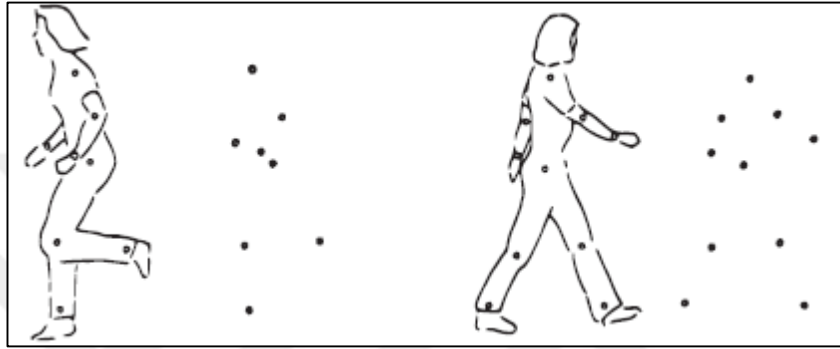
İlgi bölgesi bulma bazen köşe bulma yerinde de kullanılabilir. İlgi bölgeleri görüntülerin bulanık olduğu durumlarda köşe özelliklerinin yerine tercih edilmektedir. İlgi bölgeleri nokta tipi özellik bulma yöntemlerinden farklı olarak daha geniş daire veya elips tarzında bölgeleri bulur. Bazı ilgi bulma yöntemleri şöyledir; Laplacian of Gaussian (LoG), Difference of Gaussians (DoG), Determinant of Hessian, Maximally Stable Extremal Regions (MSER), Principal Curvature-based Region Detector (PCBR), Harris-affine, Hessian-affine, Edge-based regions (EBR), Intensity Extrema-based Region (IBR).

Öznitelik betimleyiciler görüntülerdeki yerel özellikleri tespit ve tanımlamak için önerilmiştir. Çok sayıda betimleyici olmasına rağmen en çok dikkat çeken ve kullanılan yöntemler vardır. Scale-Invariant Feature Transform(SIFT), Speeded Up Robust Features (SURF), Histogram of Oriented Gradients (HOG), Nonparametric Weighted Feature Extraction (NWFE), Lucas-Kanade-Tomasi (LKT), Local Energy-Based Shape Histogram (LESH), PCA-SIFT ve Gradient Location-Orientation Histogram (GLOH) en yaygın betimleyicilerdir.

7.2.6. Hareketlerin temsili

Weinland ve diğ., hareket temsil yöntemleri üzerine yaptıkları literatür çalışmasında hareket temsil yöntemlerini iki ana kategoriye ayırmışlardır; konumsal hareket temsilleri ve zamansal hareket temsilleri (Weinland ve ark., 2011). Konumsal hareket temsilleri vücut modelleri, görüntü modelleri ve yerel istatistikler olmak üzere üçe ayrılmıştır. Zamansal hareket temsilleri hareket üzerine yapılan gözlemlerin zamansal bileşenleri nasıl belirttiklerine göre üçe ayrılmıştır; hareket gramerleri, hareket şablonları ve zamansal istatistikler.

Vücut modelleme yöntemlerinde bir insanın vücut duruşu, görüntüdeki mevcut özellikler kullanılarak video akışındaki her bir görüntü için yeniden inşa edilir. Weinland ve diğ., bu yöntemlerin canlıların hareketlerinin insanlar tarafından nasıl yorumlandığı üzerine Johansson'ın yaptığı çalışmadan esinlendiğini belirtmiştir (Weinland ve ark., 2011). Bahsi geçen çalışmada insanların hareketleri anlamasında insan vücuduna konumlandırılan birkaç ışık noktasının yeterli olduğu gösterilmiştir. Şekil 7.8.'de yapılan deneyin gösterimi vardır.



Şekil 7.8. Johansson insan vücuduna bağlanan birkaç ışık noktasının hareketi gözlemleyen kişinin yapılan hareketi tanımasında yeterli olduğunu göstermiştir (Johansson, 1973).

Yapılan deney insan hareketlerinin doğrudan 2D görüntülerden mi yoksa hareketlerden 3D yapılar oluşturulduktan sonra mı insan tarafından tanındığı üzerine yapılan tartışmaların merkezini oluşturmaktadır. 3D modeller oluşturularak yapılan çalışmalar iki aşamadan oluşmaktadır. Öncelikler 3D modellerin vücut eklem yerlerine göre oluşturulması ve daha sonra oluşturan modeldeki eklemlerin izlerinin takip edilmesidir (İkizler ve Duygulu, 2009). 2D modelleme ile yapılan çalışmalarda vücudun belli yerleri çıkarılan özelliklere göre etiketlenerek takip edilir ve buna göre hareket tanıma işlemi gerçekleştirilir (Wilson ve Bobick, 1999).

Bütüncül temsil olarak da adlandırılan görüntü modelleri insan vücut kısımlarının tespit edilmesi yerine insanın merkezinde olduğu ilgi bölgelerinin tespitine ihtiyaç duyarlar (Blank ve ark., 2005). Görüntü tabanlı modeller vücut tabanlı modellerden daha basitçe gerçekleştirilebilirler. Hesaplama maliyeti olarak daha verimli ve kararlıdır. Bazı eylemler için vücut modelleri gibi ayırt edici oldukları gösterilmiştir.

Hareketi gerçekleştiren insanın silueti ve dış hatları bütüncül temsil yöntemlerinin en çok kullandığı özelliklerdir. Siluetlerin zaman akışındaki durumuna göre Motion History Image ve Motion Energy Image adı verilen model görüntüleri oluşturulmuştur (Ahad, 2013). Siluetlerin hareket hakkında güçlü ipuçları vermesini ve renk, doku ve ışık değişiminden etkilenmemesini avantajları olarak sıralaya biliriz. Vücut kısımlarının birbirinin önüne gelmesi durumlarında başarısızdırlar ve sağlam bir arka plan ayrıştırılmasına ihtiyaç duyarlar.

Görüntü tabanlı yöntemlerin kullandığı diğer bir önemli özellik, sıralı görüntülerden elde edilen optik akış hesaplamalarıdır (Kumar ve John, 2016). Optik akış kullanan görüntü modellerinin arka plan çıkarımına ihtiyaç duymamaları siluet kullanımından daha avantajlı hale getirmektedir. Optik akış tabanlı hareket temsil yöntemlerinin eksikliği; optik akışın hesaplanabilmesi hareketin olmasını varsaymasıdır. Görünüm modellerinin nesne hareketlerini basit temsil etmesinin sonucu olarak kamera görüş açısına ve vücut ölçütlerinin farklılığına aşırı derecede duyarlıdırlar.

Yerel veya konumsal istatistiksel yöntemler görüntüyü küçük parçalara ayırır ve tüm parçalardaki yerel özelliklerin istatistiksel çıkarımına göre tanımayı gerçekleştirir (Weinland ve ark., 2011). Bu yaklaşımların en belirgin avantajı vücut kısımlarının açıkça çıkarımına veya insanın bütünsel tespitine ihtiyaç duymamasıdır. Yerel özellikler yoğun veya seyrek biçimde hesaplanabilirler.

Konumsal istatistiksel yöntemlerde uzay-zaman ilgi noktaları nesne tanıma ve görüntü sınıflandırmada kullanılan yerel betimleyicileri ve ilgi noktalarını genelleştirmek için önerilmişlerdir. Bu yaklaşımlar tipik olarak tümevarım yolu izlerler. İlk önce köşe veya kabarcık tarzı yapılar ilgi noktası olarak tespit edilir; daha sonra, önceden seçilmiş kelime-özellik kümelerine atanırlar (Weinland ve ark., 2011). Laptev ve Lindeberg konumsal-zamansal olayları tespit etmek için, Harris ve Forstner ilgi noktalarını kullanmışlardır. Uzaysal zamandaki görüntü değerlerinin hem uzay hem de zaman içinde önemli yerel varyasyonlara sahip olduğu yerdeki yerel yapıları tespit edilmiştir. Daha sonra, tespit edilen olayların konumsal-zamansal uzantılarını tahmin edip, bunların ölçek-değişmeyen konumsal-zamansal tanımlayıcıları kullanılarak sınıflandırma yapılmıştır. (Laptev ve Lindeberg, 2003).

Tespit edilen ilgi noktalarının benzer gözlemler için tutarlık sergilemesi gerekir. Hareket yapının açıkça tespit edilmesi, uzay-zaman ilgi noktalarının hesaplanması için gerekli olmaması bu yaklaşımın avantajlarından. Yaklaşımın eksisi; tespit edilen özelliklerin sırasız ve değişik ölçülerde olmasının sonucu olarak geometrik ve geçici yapıların modellenmesi uzay-zaman özellikleriyle zordur.

Hareket gramerleri, hareketleri her birinin kendi görünümü ve dinamiği olan anlık dizgi şeklide temsil eder. Dinamik bir sistemin gözlemlenen özellikler üzerinden modellenmenin ortak yolu benzer yapıya sahip özelliklerin gruplandırılmasıdır. Hareket tanımada kullanılan birçok olasılık gramer yöntemi içerisinde en bilineni Gizli Markov Modelidir(HMM). Faisal ve diğ., önerdikleri hareket sınıflama çalışmasında, yörüngeler eğrilikteki değişim noktalarına bölünür ve alt bölümler, temel bileşen analizi (PCA) katsayıları ile temsil edilir. PCA katsayıları HMM modeline durum ve topoloji açısından veri güdümlü olarak verilerek tanıma işlemi yapılmıştır.

Hareket tanıma için kullanılan diğer dinamik hareket gramer modelleri; dinamik Bayesian ağı (Suk ve ark., 2010), öz bağımlı modeller(auto regressive models), zaman gecikmeli yapay sinir ağları, context-free gramerler ve özellik yapı gramerleridir. Hareket gramerlerinin önemli bir avantajı farklı hızda ve tarzda icra edilen hareketlerin genelleştirilmesinde yüksek derecede esneklik sağlamasıdır. Bu yaklaşımın eksikliği; olasılık gramer yapılarının elle seçilmesi gerekliliğidir. Bu nedenle eylem sınıfının çok olduğu durumlarda ciddi problemler yaşanmaktadır.

Hareket şablonları belli bir zaman aralığındaki özelliklerin toplam görünümünü kullanan yöntemlerdir. Tipik şablon tabanlı yöntemler hareket değişim dinamiğini örnek görüntü dizisinden ya özellikleri üst üste yığarak ya da konum-zaman hacmini oluşturarak tanıma yapmaya çalışır. Şablon kullanan çoğu yaklaşım görünüm modellerine dayanmaktadır. Görünüm modelindeki silüetler üst üste konarak şablon inşa edilir. Şablon kullanan yöntemlerin konum zaman özellikleri ve optik akış yaklaşımlarından farkı bu yöntemlerin 3 veya 4 görüntü çerçevesi kullanmasına karşın çok sayıda görüntü çerçevesi kullanılmasıdır. Bu alanda ufuk açıcı bir çalışma motion history image çalışmasıdır. Şablon tabanlı yöntemler hareket temsilde etkin ve ayır edicidirler çünkü güçlü sınıflandırıcılarla doğrudan entegre edilebilirler. Eksikliği

hareketin tespit edilmesine uyumlu değildir çünkü zaman bölümlenmesine için elverişli değildirler.

Zamansal istatistikler hareketlerin görünümünün istatistiksel modellerini oluşturmaya çalışırlar. Görüntülerde anahtar resmi(key frame) ve özelliklerin histogramını kullanan yöntemler bu yaklaşımlar için örnek verilebilir. Benzer özellikleri olan ancak farklı zaman sıralamasına sahip eylemler için uygulanamazlar.

7.2.7. Hareket bölümleme

Hareket tanımada süregiden farklı hareketlerin ayrıştırılması çok önemlidir. Bir hareket eylemi çok sayıda atomik hareketlere bölünerek süregiden ardışık eylemlerin sınıflandırılması yapılabilir. Weinland ve diğ., hareket bölümlendirmeyi üçe ayırmışlardır; sınır tespiti, kayan pencereler ve gramer birleştirme (Weinland ve ark., 2011).

Sınır tespiti, hareketlerin sınırlarını tespit ederek tespit edilen sınırlar arasında kalan hareketin tanınması için kullanılır. Bu hareket sınırları harekete ara verilmesi, harekete ait ivme, hız ve eğimde doruk nokta gibi özelliklerle belirlenir. Herhangi bir eylem sınıfına bağlı olmadan genel bir bölümleme sağladığı için sınır tespit yöntemleri teorik olarak çok ilgi çekicidir. Hareket bölgesinin tekrar elde edilmesinde hataya açık olması, farklı görüş açılarında başarısızlığı ve eş zamanlı hareketlerin olduğu durumlarda kolayca karıştırıldığından dolayı pratikte bazı önlemler alınarak kullanılmalıdır.

Kayan pencereler yönteminde belirli aralıklardaki video parçaları için tanıma işlemi yapılır ve tanıma sonucunun en yüksek olduğu durumlar mevcut harekete ait parça olarak kabul edilir. Sınır tespit yöntemlerinin zıddı olarak burada bölümlendirme tanıma aşamasına ciddi şekilde bağımlıdır. Tutarlı bölümlendirilmiş eğitim örneklerinin sağlanması bu yöntemlerin başarısında hayati öneme sahiptir. Sınır tespit yöntemleriyle kıyaslandığında kayan pencere yöntemleri yüksek hesaplama maliyetine sahiptir. Bilinmeyen eylemler için tahmin edilemeyen sonuçlar verebilir. Az sayıda varsayıma ihtiyaç duyması ve hareket sınıflandırma yöntemleriyle doğrudan uygulanabiliyor olması avantajlı yanlarıdır.

Gramer birleştirme yöntemlerinde ayırık eylem sınıflarının gramerlerle temsil edilmesi hareketin durumları arası geçişlerinde bir model veriyor. Bu model hareketler arası geçişlerde kullanılarak hareket bölümlenme için genel bir yaklaşım sağlamaktadır. Nihai olarak hareketlerin eş zamanlı bölümlenmesi ve tanınması için etkili bir yöntem sunmaktadır. Gramerlerin birleştirilmesi son durumlu makinalarda olduğu gibi tüm hareket modelleri ortak bir başlangıç ve son duruma sahip olur ve durumlar arası geçişler belirli girdilere göre kendine veya başka bir duruma gidebilir. Bu yaklaşımlar ne sınır tespit yöntemleri gibi bazı varsayımlara gerek duyarlar ne de kayan pencere yöntemleri gibi ağır işlem yüküne sebep olurlar. Çok sayıda eylemlerin gramer birleştirme ile öğrenilmesi için büyük çapta eğitim verisine ihtiyaç duyulduğu göz ardı edilmemelidir.

7.2.8. Hareket tanımada kullanılan yöntemler

Görüntü veya videolardan doğru özellikler seçildikten sonra eylem tespit ve sınıflandırma algoritmaları bir ileri aşama olarak insan hareket tanımada kullanılır. Seçilen özellik temsillerine göre doğru algoritmanın uygulanması çok önemlidir. Dynamic time warping (DTW), Gizli Markov model (GMM), dinamik Bayesian ağı (DBA), destek vektör makinaları (DVM), relevance vector machines (RVMs), yapay, sinir ağları (YSA), Kalman filtresi, ikili ağaç, en yakın komşu algoritması (K-NN), context free gramer ve stocastic context free gramer yöntemleri hareket tanımada uygulanan yaklaşımlardır.

DTW iki dizi arasındaki benzerliği(uzaklığı) ölçen dinamik bir programlama algoritmasıdır. DTW'nın avantajı hızlı ve kolay olmasıdır. Ancak tanınacak eylemin çeşitli olduğu durumlarda çok sayıda şablona ihtiyaç duyması hesaplama maliyetini artırmaktadır.

GMM bileşik olasılığı kullandığı için üretken yaklaşımlar içerisinde ele alınır. GMM gizli olasılık sürecini ve gözlemlenen olasılık sürecini içeren çift katmanlı bir olasılık süreci şeklide tanımlanır. GMM konuşma ve video gibi zamanlı ardışık yapıya sahip verileri modellemede oldukça popülerdir.

DBA durum model tabanlı yaklaşımlarda GMM gibi kullanılmıştır. İki yöntemde de etkinlik gizli durumlar kümesi elemanları olarak temsil edilmektedir. Aggarwal ve

Ryoo DBA'yı çoklu şartlı olarak bağımsız gizli durumlardan oluşan GMM'nin genişletilmiş hali olduğunu belirtmişlerdir (Aggarwal ve Ryoo, 2011). GMM ve DBA tek bir eylem için kullanıldığı gibi birçok eylemi ihtiva eden etkinlikler içinde kullanılmışlardır.

DVM sınır tabanlı popüler gözetimli bir sınıflandırma algoritmasıdır. Bir veri setini iki sınıfa ayırmak için kullanılır. İki sınıf arasındaki ayrımı maksimum seviyede sağlayan doğruyu bulmayı amaçlar. DVM'nin ana eksikliği öğrenme safhasında kullanılan kısıtlı eniyileme programlamanın yüklediği hesaplama maliyetidir.

Yönlü çizge ağlarındaki problemleri tarif etmek için kullanılan YSA hareket analizinde kullanılmıştır. Çizge ağındaki her bir düğüm yapay bir sinir ve düğümler arasındaki yönlü bağlantılarda yapar sinirler arasındaki bağlantıları ifade eder. YSA'lar doğrusal olmayan karmaşık girdi-çıkı ilişkisini öğrenmeye imkan veren evrensel bir yaklaşımdır. Sistematik sıralı bir eğitim prosedürüyle YSA girdi verisine uyarlanacak bir modeli eğitebilir.

Kalman filtresi ortalama kare hatalarını minimize eden matematiksel denklemler kümesidir ve dinamik süreç durumlarını kestirebilir. Kalman filtrede iki aşama özyineli(recursive) olarak icra edilir; süreç durumunun kestirimi ve ölçümlere göre süreç durumunun güncellenmesi. KF yaya takibinde kullanılmıştır ve en büyük dezavantajı iyi bir ön plan çıkarımına ihtiyaç duymasındır ve bu yüzde nesne kesişmelerine karşı zayıftır.

İkili ağaçlar her bir düğümün sadece iki alt düğüme sahip olduğu bir veri yapısı modelidir. İkili ağaç yapıları sınıflandırma problemlerine uygulandığında sınıflandırma ağacı olarak adlandırılır. İkili ağaç sınıflandırma basit ve hızlıdır ancak her bir düğüm için kural belirlemek zordur ve kompleks durumlar için uygulanmasını zorlaştırır.

KNN önceden tanımlı bir K sabitini esas alan bir sınıflandırma algoritmasıdır. KNN birkaç parametrelili basit bir modeldir. Test aşamasındaki hesaplama süresi sınıf sayısından bağımsızdır ve arama uzayındaki doğrusal olmayan veriler için bile kararlı sonuçlar verir. KNN için temel eksikliği; eğitim verilerine test aşamasında ihtiyaç duyması ve sınıflandırma performansının seçilecek K değerine duyarlı olmasıdır.

Context free Grammer (CFG) ve stocastic context free grammer (SCFG) atomik hareketlerden oluşan üst seviye etkinlik tanımda kullanılmışlardır. CFG'nin bir durumdan başka bir duruma geçiş kuralları doğal bir şekilde basit hareketlerden oluşan etkinliklerin modellenmesinde ve tanınmasında kullanılmasına yol açmıştır. SCFG üst seviye etkinlikler basit hareketlerden oluşan sözdizimleri olasılıksak sözdizimi analiz tekniği kullanılarak tanınır. CFG ve SCFG kullanan yöntemlerin birinci eksikliği eş zamanlı etkinliklerin tanımadaki zayıflığıdır.

Sözdizimi yaklaşımlar doğası gereği alt olaylardan oluşan etkinlikleri tanımaya elverişlidirler ancak eş zamanlı alt olaylardan oluşan etkinlikleri tanımda yetersizdirler. Sözdizimi yaklaşımlar basit hareketlerden oluşan etkinlikleri bir dizi şeklinde modellediğinden dolayı basit hareketlerin zorunlu bir şekilde ardışık olması gerekir. Sözdizimi yaklaşımlar tüm gözlemlerin söz üretme kuralları uygulanarak üretildiğini varsayar. Bu sistemlerde mümkün olan tüm olaylar için, örnek sayısı çok büyük olsa bile, kullanıcıların tüm olaylar için sözdizimi kurallarının belirlenmesi gerekir.

8. VIDEO GÖRÜNTÜLERİNDEN İNSAN HAREKETLERİNİN TANINMASI

Bu bölümün amacı uygulama alanı çok geniş ve güncel bir konu olan video görüntülerden hareketlerin tanınmasına ve analiz edilmesine yeni bir yaklaşımla katkıda bulunmaktır. Çalışmamızın birinci amacı hücresel otomatların örüntü tanıma ve sınıflandırmada kullanılan bir formu olan Genelleştirilmiş Çoklu Cezbedici Hücresel Otomatlar(Generalized Multiple Attractor Cellular Automata(GMACA)) kullanmaktır. Hücresel otomatların ızgara düzlemindeki çalışma mantığı görüntülerdeki piksellere benzediği için görüntü işleme ile ilgilenen araştırmacıların dikkatini çekmiştir. Cellular otomata görüntü işlemede kenar bulma, köşe bulma, gürültü giderme ve benzeri işlemlerde kullanılmıştır. GMACA'lar elektronik devrelerde hata teşhisi, veri madenciliği, görüntü sıkıştırma, örüntü tanıma ve VHLS devre tasarımı alanlarında kullanılmıştır. Bu çalışmada GMACA'ların örüntü çekim gücü hareket tespitinde kullanılmıştır.

Bu bölümde, GMACA ile yapılan hareket tespit görüntüleri kullanılarak video görüntülerindeki insan hareketlerinin tanınması yapılmıştır (Peldek ve Becerikli, 2019). Geliştirilen insan hareketlerini tanınma yönteminin performansı KTH ve WEIZMANN veri setleri üzerinden test edilmiştir. Elde edilen sonuçlar GMACA ile yapılan hareket tespit görüntülerinin ileri bir bilgisayarlı görü uygulaması olan insan hareketlerinin video görüntülerden tanınması görevi için kullanılabileceğini göstermektedir.

Bu bölüm, iki ana alt bölüme ayrılmıştır; “Geliştirilen Hareket Tanıma Yaklaşımı” ve “DeneySEL Sonuçlar”. Geliştirilen yöntemin anlatıldığı bölümde, insan hareketlerinin tanınması için yapılan adımlar anlatılmıştır. DeneySEL sonuçların yer aldığı bölümde, geliştirilen yöntem insan hareketlerini içeren videoların yer aldığı KTH ve WEIZMANN veri setleri üzerinde test edilmiştir. Video görüntülerdeki insan hareketlerinin tanınma başarısı istatistiksel metriklerle verilmiştir. Elde edilen sonuçlarla literatürde daha önce aynı veri setleri üzerinde elde edilmiş sonuçlar karşılaştırılmıştır.

8.1. Geliştirilen Hareket Tanıma Yaklaşımı

Bu çalışmada insan hareketlerinin tanınması KTH ve WEIZMANN veri tabanlarında yer alan insan hareketlerini içeren videolar kullanılarak gerçekleştirilmiştir. KTH veri tabanında altı çeşit insan eylemi içeren (yürümek, tempolu koşu, koşu, boks, elle sallama ve elle el çırpma) videolar yer almaktadır. Eylemler 25 farklı kişi tarafından dört farklı senaryo içerisinde gerçekleştirilmiştir. Eylemlerin gerçekleştirildiği senaryolar şöyledir; açık alan, açık alan ve ölçek değişikliği, farklı giysilerle açık alan ve kapalı alan. Veri tabanı 600 video içermektedir. Tüm videolar, 25fps kare hızı ile statik bir kamera kaydedilmiştir. Videolardaki arka planlar sabittir. Videolar, 160x120 piksel çözünürlüğe sahiptir (Schuldt ve ark., 2004).

WEIZMANN veri tabanında yer alan hareket çeşitleriye şöyledir; eğilme, tek el sallama, iki el sallama, jack(iki el ve iki bacağın yerinde zıplayarak beraber açılıp kapanması), yerinde zıplama, ileri doğru zıplama, yan yürüme, yürüme, koşma ve tek ayak üzerinde sekerek yürüme. Hareketler dokuz farklı kişi tarafından açık alanda gerçekleştirilmiştir. Veri tabanı, statik kamerayla çekilmiş 25 fps hıza ve 180x144 çözünürlüğe sahip 90 video içermektedir (Blank ve ark., 2005) .

Gerçekleştirilen hareket tanıma yönteminin adımları şöyledir:

1. Görüntüler gri renk uzayına dönüştürülür
2. Kullanılacak GMACA kural vektörü, Bölüm 2’de anlatıldığı gibi elde edilir
3. GMACA kullanılarak video görüntülerdeki insan hareketlerinin tespiti yapılır
4. Hareket tespit görüntülerinden HOG özellik vektörü çıkarılır ve elde edilen HOG özellik vektörleri ait oldukları harekete göre etiketlenir. Bu şekilde veri seti oluşturulur
5. Oluşturulan veri seti, çapraz-doğrulama yöntemi ile eğitim ve test veri setlerine ayrıştırılır
6. Hareket tanımanın SVM yöntemi ile gerçekleştirilir

Yukarda verilen hareket tanıma yöntemi adımları başlıklar altında detaylıca anlatılmıştır.

8.1.1. Görüntülerin gri renge dönüştürülmesi

Görüntü tanıma ve analiz uygulamalarını çoğu gri görüntüler kullanır (Kanan ve Cottrell, 2012). Kanan ve Cottrell yaptıkları çalışmada renkli görüntülerin gri görüntüye dönüştürülmesinde kullanılan yöntemin görüntü tanıma başarısına etki ettiğini göstermektedir.

Görüntüleri griye dönüştürmek için kullanılan en yaygın yöntemler Denklem (8.1) ve (8.2)'de verilmiştir;

$$gri_{yoğunluk} = \frac{1}{3}(R+G+B) \quad (8.1)$$

$$gri_{Luminance} = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (8.2)$$

Bu çalışmada yukardaki griye dönüştürme yöntemlerinden $gri_{Luminance}$ yöntemi kullanılmıştır.

8.1.2. Hareket tespiti

Video görüntülerindeki hareket tespit GMACA ile bölümlendirilen görüntüler kullanılarak gerçekleştirilmiştir. GMACA ile yapılan hareket tespit yöntemi altıncı bölümde detaylıca açıklanmıştır. Bu kısımda kısaca anlatılmıştır.

GMACA ile yapılan hareket tespit yöntemi, ardışık görüntülerin farkına dayalı hareket tespit etme yöntemleri altında değerlendirilebilir. Geliştirilen hareket tespit yöntemin daha önceki yapılan çalışmalardan en önemli farkı, insan hareketlerinin tespiti GMACA durum güncelleme mantığı kullanılarak yapılmasıdır. Yöntemde ardışık görüntüler arasındaki farka dayalı hareket tespiti yapılmıyor, bunun yerine ardışık iki görüntü arasında karşılık gelen piksel konumlarının aynı cezbediciye ait olup olmamasına göre hareket tespiti yapılıyor. Daha açık bir ifade ile ardışık iki görüntülerin karşılık gelen pikselleri aynı cezbediciye ait ise hareket olmadığı, farklı ise hareket olduğu sonucuna varılıyor.

Geliştirilen hareket tespit yönteminin adımları şöyledir:

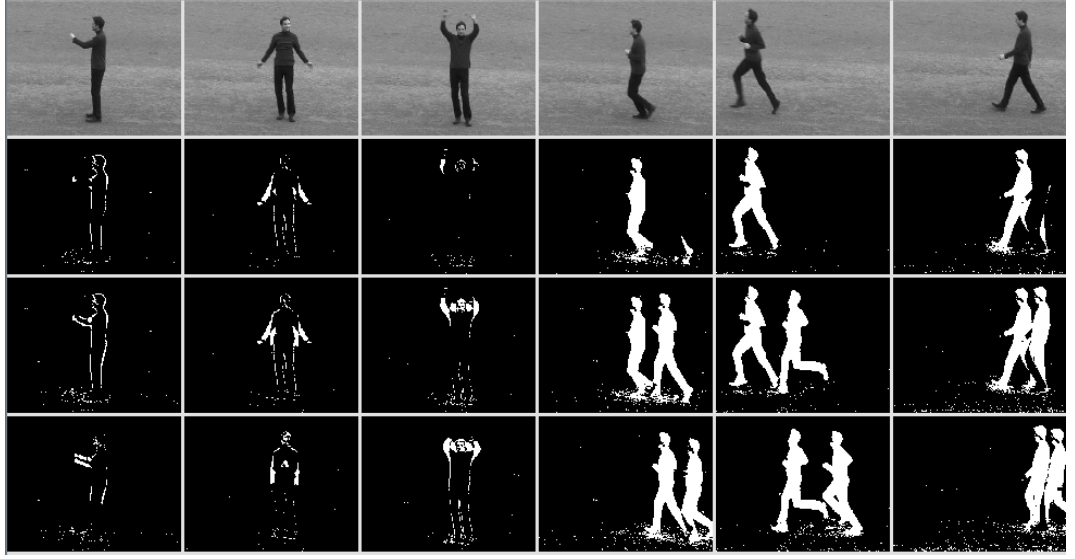
- İstenen sayıda cezbedici üreten kural vektörü belirlenir. Kural vektörünün nasıl belirlendiği ikinci bölümde detaylıca açıklanmıştır.

- O anki görüntü için kullanılacak kural vektörünün cezbedicileri hesaplanır. Görüntüdeki her bir piksel belirlenen kural vektörüne göre bir cezbediciye atanır
- Bir önceki görüntü ile o anki görüntünün cezbedici durumları karşılaştırılır. Eğer görüntülerin aynı konuma karşılık gelen pikselleri aynı cezbediciye sahip ise hareket olmadığı, aynı cezbediciye sahip değil ise hareket olduğu sonucuna varılır.

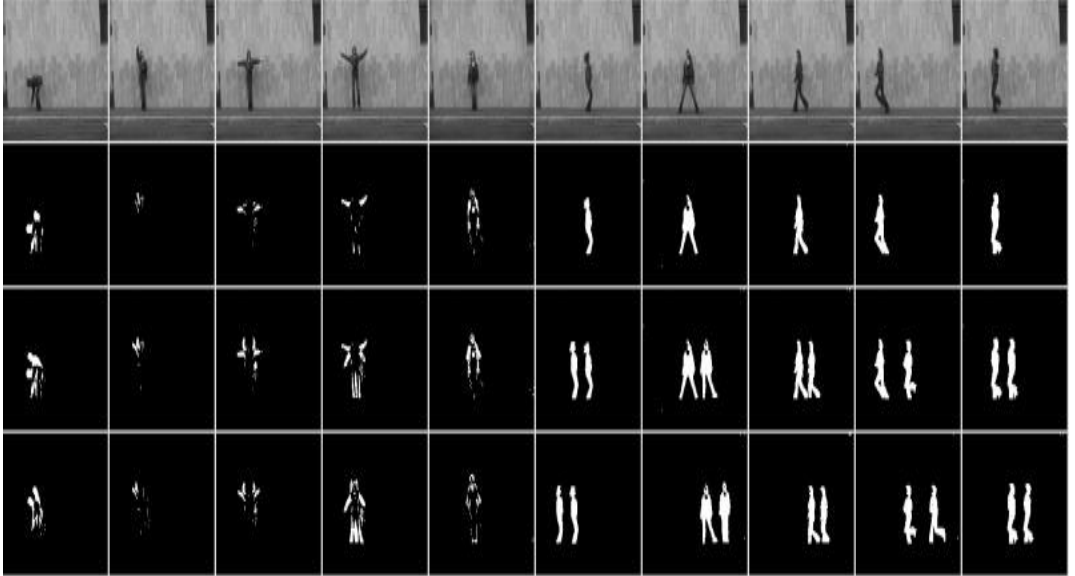
Şekil 8.1 ve 8.2’de GMACA kullanılarak KTH ve WEIZMANN veri tabanları üzerinde elde edilen hareket tespit görüntüleri yer almaktadır. Şekillerde yer alan resimler yatay ve dikey bölümlere ayrılmıştır. Dikey bölümlerin her birisi ayrı bir hareketi göstermektedir.

Yatay bölümler, sırasıyla orijinal hareket görüntülerini ve farklı zamanlardaki hareket tespit görüntülerini göstermektedir. Yatay ve dikey bölümlendirmeyi sırasıyla satır ve sütun olarak da düşünülebilir. Sütunlarda sırasıyla KTH veri tabanı için boks, alkış, el sallama, tempolu koşu, normal koşu ve yürüme hareketlerine ait görüntüler yer almaktadır. WEIZMAN veri tabanı için eğilme, tek el sallama, iki el sallama, jack(iki el ve iki bacağın yerinde zıplayarak beraber açılıp kapanması), yerinde zıplama, ileri doğru zıplama, yan yürüme, yürüme, koşma ve tek ayak üzerinde sekerek yürüme hareketlerine ait görüntüler yer almaktadır.

Satırlarda sırasıyla orijinal görüntü $I(x,y,t)$, t zamanındaki gerçek hareket tespit görüntü $D_g(x,y,t)$ t zamanındaki hareket tespit görüntüsü $D(x,y,t)$ ve $t-1$ zamanındaki hareket tespit görüntüsü; $D(x,y,t-1)$ yer almaktadır.



Şekil 8.1. KTH veri tabanında yer alan altı farklı hareket tespit görüntüleri



Şekil 8.2. WEIZMANN veri tabanında yer alan on farklı hareket tespit görüntüleri

Bir sonraki alt bölümde, HOG özelliklerinin çıkarılması anlatılmıştır.

8.1.3. HOG özelliklerinin çıkarılması

HOG, ilk defa Fransız Ulusal Bilgisayar Bilimi ve Kontrol Enstitüsü (INRIA) araştırmacılarından Navneet Dalal ve Bill Triggs tarafından geliştirilmiştir (Dalal ve Triggs, 2005). HOG görüntülerdeki piksel değerlerinin yerel eğilimlerine veya kenarların yönüne göre görüntülerden öznitelik elde eder. HOG tanımlayıcılarının arkasındaki temel fikir, bir görüntü içindeki yerel nesne görünümünün ve şeklinin

yoğunluk eğilimlerinin veya kenar yönlerinin dağılımı ile tanımlanabilmesidir. HOG bilgisayarla görü ve görüntü işlemede sıkça kullanılmaktadır.

HOG yönteminin görüntülerden öznitelik çıkarımını şöyle özetlenebilir: Görüntü hücre (cell) adı verilen birbiriyle bağlantılı küçük bölümlere ayrıştırılır. Blok olarak adlandırılan hücre bölümlerini içine alan daha geniş yapılar oluşturulur. Her bir blok için piksel değerlerinin eğilimleri hesaplanır. Bu eğilimlerden eğim açıları elde edilir. Elde edilen iki farklı eğilimden eğilim histogramı oluşturulur. Yöntemin ana fikrini bu eğilim histogramının elde edilmesi oluşturur. Bu nedenle yöntem adını bu işlemde alır. Görüntüdeki ışık farklılıklarını en aza indirmek için elde edilen histogram normalleştirme işlemine tabi tutulur. Her bir blok için elde edilen normalleştirilmiş histogramlar birbirine eklenerek görüntüye ait öznitelik vektörü elde edilir (Cerit ve ark., 2016).

Yukarıda genel hatları verilen HOG yöntemi, bir sonraki alt bölümde detaylıca ele alınmıştır. HOG uygulaması temelde üç aşamadan oluşur. Bunlar; her bir piksele ait eğilimin büyüklüğü ve açısının hesaplanması, eğilim histogramının elde edilmesi ve eğilim histogramının normalleştirilmesi. Bu adımları ayrı başlıklar altında ele alınmıştır.

8.1.3.1. Piksel eğilimlerinin hesaplanması

Görüntüdeki her bir pikselin eğilimi Denklem (8.3)'teki gibi hesaplanır:

$$\Delta f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right) \quad (8.3)$$

Formülde yer alan x ve y görüntüdeki pikselin konumunu temsil etmektedir. $f(x,y)$ fonksiyonu (x,y) konumundaki pikselin değerini temsil eder. $\frac{\partial f(x,y)}{\partial x}$, (x,y) konumundaki pikselin x 'e göre türevini ve $\frac{\partial f(x,y)}{\partial y}$, (x,y) konumundaki pikselin y 'ye göre türevini vermektedir. Yapılan işlemler aslında bir filtreleme işlemidir. Filtreleme işleminde (x,y) konumundaki piksele uygulanacak maske x ve y eksenleri için sırasıyla $[-1,0,1]$ ve $[-1,0,1]^T$ maskeleridir. Belirtilen maskeler Dalal ve Tiggs'in çalışmalarında en iyi sonucu veren maskeler olarak belirtilmiştir (Dalal ve Triggs, 2005). Yapılan

türev işlemleri matematiksel olarak x ve y için Denklem (8.4) ve (8.5)'teki gibi de gösterilebilir;

$$\frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x-1,y) \quad (8.4)$$

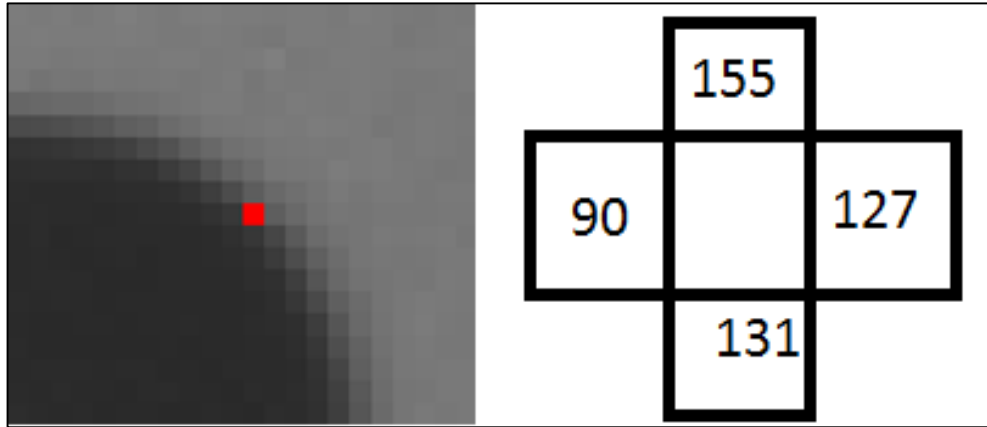
$$\frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y-1) \quad (8.5)$$

$\frac{\partial f(x,y)}{\partial x}$ ve $\frac{\partial f(x,y)}{\partial y}$ notasyonlarına sırasıyla f_x ve f_y olsun, (x,y) konumundaki pikselin büyüklüğünü ve yönünü Denklem (8.6) ve (8.7)'deki gibi bulunur;

$$|\Delta f(x,y)| = \sqrt{f_x^2 + f_y^2} \quad (8.6)$$

$$\theta(x,y) = \tan^{-1} \left(\frac{f_x}{f_y} \right) \quad (8.7)$$

Yukarıda formüllerle gösterilen; eğilim, büyüklük ve yön bulma işlemlerini örnek verilerek aşağıda gösterilmiştir.



Şekil 8.3. Görüntüye ait bir pikselin yatay ve dikey eksenindeki komşu piksellerinin değeri

Şekil 8.3'te kırmızı nokta ile gösterilen bir pikselin yatay ve dikey eksenindeki komşulukları yer almaktadır. Pikselin x ve y eksenlerindeki eğilimleri Denklem (8.8) ve (8.9)'da gösterildiği gibi bulunur;

$$\frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x-1,y) = 155 - 131 = 24 \quad (8.8)$$

$$\frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y-1) = 127 - 90 = 37 \quad (8.9)$$

Pikselin x ve y eksenlerindeki eğilimleri bulunduğundan sonra bu eğilimin bileşke büyüklüğü ve açısı Denklem (8.10) ve (8.11)'deki gibi bulunur;

$$|\Delta f(x,y)| = \sqrt{24^2 + 37^2} = 44.1 \quad (8.10)$$

$$\theta(x,y) = \tan^{-1} \left(\frac{24}{37} \right) = 32.58 \text{ derece} \quad (8.11)$$

HOG yönteminde, eğilim büyüklüğü ve açısı bulunduğundan sonra eğilim histogramı oluşturulur. Bir sonraki bölümde eğilim histogramının elde edilmesi anlatılmıştır.

8.1.3.2. Eğilim histogramının elde edilmesi

Her bir piksel için eğilim büyüklüğünün ve açısının (yönünün) nasıl hesaplanacağını yukarıdaki bölümde anlatılmıştır. Yukarıdaki büyüklük ve açı bulma işlemi görüntüdeki her bir piksel için yapılarak görüntüye ait büyüklük ve açı matrisleri elde edilir. Elde edilen bu matrisleri Şekil 8.4'teki gibi görselleştirilebilir.

Görüntü	Eğilimin Büyüklüğü	Eğilimin Açısı																																																																																																																																
	<table><tr><td>2</td><td>3</td><td>4</td><td>4</td><td>3</td><td>4</td><td>2</td><td>2</td></tr><tr><td>5</td><td>11</td><td>17</td><td>13</td><td>7</td><td>9</td><td>3</td><td>4</td></tr><tr><td>11</td><td>21</td><td>23</td><td>27</td><td>22</td><td>17</td><td>4</td><td>6</td></tr><tr><td>23</td><td>99</td><td>165</td><td>135</td><td>85</td><td>32</td><td>26</td><td>2</td></tr><tr><td>91</td><td>155</td><td>133</td><td>136</td><td>144</td><td>152</td><td>57</td><td>28</td></tr><tr><td>98</td><td>196</td><td>76</td><td>38</td><td>26</td><td>60</td><td>170</td><td>51</td></tr><tr><td>165</td><td>60</td><td>60</td><td>27</td><td>77</td><td>85</td><td>43</td><td>136</td></tr><tr><td>71</td><td>13</td><td>34</td><td>23</td><td>108</td><td>27</td><td>48</td><td>110</td></tr></table>	2	3	4	4	3	4	2	2	5	11	17	13	7	9	3	4	11	21	23	27	22	17	4	6	23	99	165	135	85	32	26	2	91	155	133	136	144	152	57	28	98	196	76	38	26	60	170	51	165	60	60	27	77	85	43	136	71	13	34	23	108	27	48	110	<table><tr><td>80</td><td>36</td><td>5</td><td>10</td><td>0</td><td>64</td><td>90</td><td>73</td></tr><tr><td>37</td><td>9</td><td>9</td><td>179</td><td>78</td><td>27</td><td>169</td><td>166</td></tr><tr><td>87</td><td>136</td><td>173</td><td>39</td><td>102</td><td>163</td><td>152</td><td>176</td></tr><tr><td>76</td><td>13</td><td>1</td><td>168</td><td>159</td><td>22</td><td>125</td><td>143</td></tr><tr><td>120</td><td>70</td><td>14</td><td>150</td><td>145</td><td>144</td><td>145</td><td>143</td></tr><tr><td>58</td><td>86</td><td>119</td><td>98</td><td>100</td><td>101</td><td>133</td><td>113</td></tr><tr><td>30</td><td>65</td><td>157</td><td>75</td><td>78</td><td>165</td><td>145</td><td>124</td></tr><tr><td>11</td><td>170</td><td>91</td><td>4</td><td>110</td><td>17</td><td>133</td><td>110</td></tr></table>	80	36	5	10	0	64	90	73	37	9	9	179	78	27	169	166	87	136	173	39	102	163	152	176	76	13	1	168	159	22	125	143	120	70	14	150	145	144	145	143	58	86	119	98	100	101	133	113	30	65	157	75	78	165	145	124	11	170	91	4	110	17	133	110
2	3	4	4	3	4	2	2																																																																																																																											
5	11	17	13	7	9	3	4																																																																																																																											
11	21	23	27	22	17	4	6																																																																																																																											
23	99	165	135	85	32	26	2																																																																																																																											
91	155	133	136	144	152	57	28																																																																																																																											
98	196	76	38	26	60	170	51																																																																																																																											
165	60	60	27	77	85	43	136																																																																																																																											
71	13	34	23	108	27	48	110																																																																																																																											
80	36	5	10	0	64	90	73																																																																																																																											
37	9	9	179	78	27	169	166																																																																																																																											
87	136	173	39	102	163	152	176																																																																																																																											
76	13	1	168	159	22	125	143																																																																																																																											
120	70	14	150	145	144	145	143																																																																																																																											
58	86	119	98	100	101	133	113																																																																																																																											
30	65	157	75	78	165	145	124																																																																																																																											
11	170	91	4	110	17	133	110																																																																																																																											

Şekil 8.4. Bir görüntüye ait Eğilim Büyüklüğü ve Eğilim Açısı matrisleri

Yukarıdaki şekilde 8x8 ebatlarında bir görüntüye ait Eğilim Büyüklüğü ve Eğilim Açısı matrisleri verilmiştir. Bu matrisler HOG öznitelik çıkarma yönteminin temelini oluşturmaktadır. Bu matrisler kullanılarak eğilim histogramı elde edilecektir.

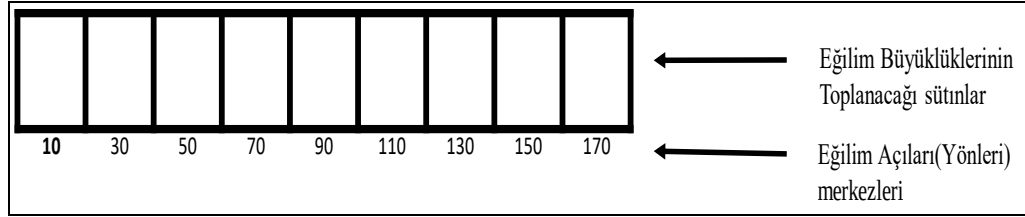
Eğilim histogramının elde edilmesine geçmeden önce Eğilimin Açısı matrisini tekrardan incelemek faydalı olacaktır. Eğilim açılarını veren matristeki değerler 0-180 arasında açı değerleridir. 0 ile 360 derece arasında değer alabilen açılar 0 ile 180 derece arasına çekilmiştir. Bunun nedeni açıların işaretli tamsayı olarak elde edilmesidir. Diğer bir değişle zıt yöndeki aynı derecedeki açılar bir kabul edilmiştir. Açılar 0 ile 180 derece arasına çekilmesinin sebebi yöntemi öneren araştırmacıların deneysel çalışmasının sonuçlarına dayanmaktadır. Dalal ve Tiggs çalışmalarında 0 ile 180 derece arası açıların kullanılmasının insan tespitinde daha iyi sonuç verdiğini belirtmişlerdir (Dalal ve Triggs, 2005).

Eğilim histogramını elde etmek için elimizde iki matris var; Eğilimin Büyüklüğü ve Eğilimin Açısı(yönü) matrisleri. Bu matrisleri kullanarak eğilim histogramını elde edilecektir. İlk önce histogramın tanımını hatırlayalım: Histogram, belirli aralıktaki değerlerin sıklığının sütunlarla gösteren grafik türüdür. Diğer bir ifadeyle, aynı aralıktaki sayıların sıklığını bulup, bu sıklığı sütun yüksekliğiyle göstererek verileri gruplara ayırıştırarak dikdörtgen sütunlar halinde gösterilmesidir. Görüntü işleme bağlamında, bir görüntünün histogramı piksel değerlerinin bir histogramına karşılık gelir. Bu histogram, bir görüntüdeki her farklı yoğunluk değerindeki piksel sayısını gösteren bir sıklık vektörü olarak düşünülebilir. 8 bitli gri tonlamalı bir görüntü için 256 farklı olası yoğunluk vardır ve bu nedenle histogram grafiksel olarak bu gri tonlama değerleri arasındaki piksel dağılımını gösteren 256 sayı görüntüler.

Şimdi, iki tane iki boyutlu matrisi tek boyutlu bir vektörle nasıl ifade edileceğine değinelim. HOG için önerilen sütun sayısı 9'dur. Yani iki boyutlu, iki matris bir boyutlu dokuz uzunluklu bir vektör ile temsil edilecektir. Eğilimin Açısı matrisini, histogramın sütun aralıklarını temsil edecek şekilde, Eğilimin Büyüklüğü matrisini histogram aralıklarına denk gelecek sıklık şeklinde temsil edilecektir.

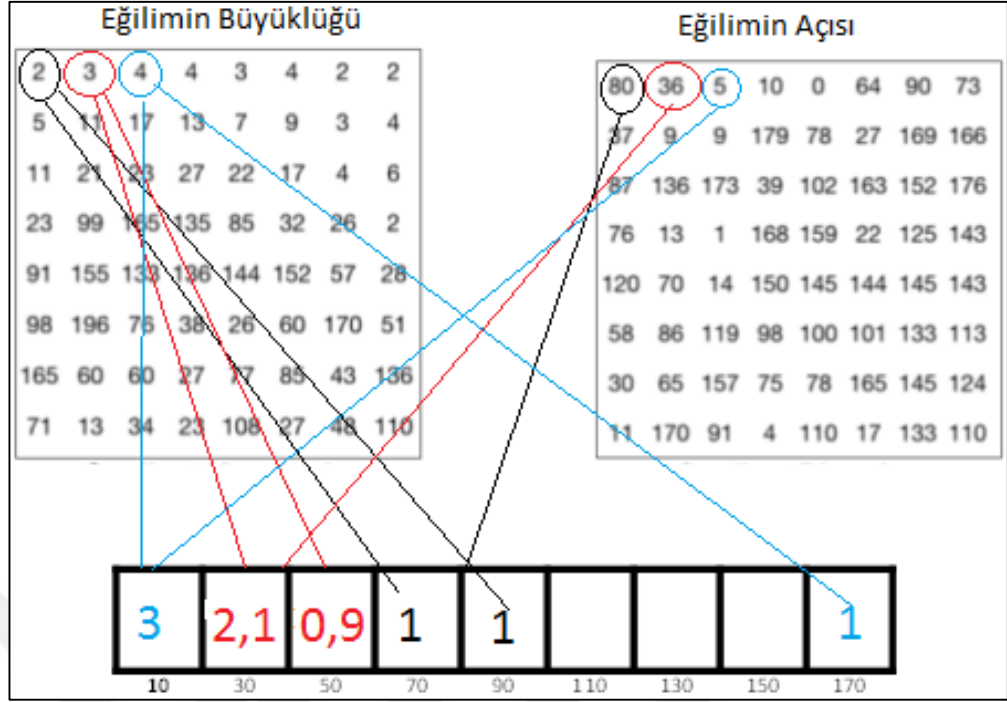
Sayısal değerlerle eğilim histogramının oluşturulmasını anlatmak faydalı olacaktır. Eğilimin Açısı matrisinde yer alan değerlerin 0 ile 180 derece arasında olacağı yukarıda belirtilmişti. 0-180 değer aralığını dokuz sütünde temsil etmek için dokuz eşit parçaya ayrılması gerekiyor. Şekil 8.5'te eğilim histogramının yapısı görülmektedir. Histogramın sütunları dokuz gruptan oluşmaktadır. Grupların merkezlerini 10, 30, 50, 70, 90, 110, 130, 150 ve 170 değerleri işaret etmektedir. Histogramın sütunları aynı

açı aralığındaki eğilim büyüklüklerinin toplamından oluşacaktır. Bir piksele ait eğilim açısı ve büyüklüğü sırasıyla 50 ve 14 olsun. Bu pikselin histogramdaki temsili; eğilim açı merkez 50 olan sütuna 14 değerinin eklenmesiyle gerçekleşecektir. Eğilim açısı ve büyüklüğü sırasıyla 130 ve 122 olan bir pikselin histogramdaki temsili; açı merkez 130 olan sütuna 122 değerinin eklenmesiyle gerçekleşecektir.



Şekil 8.5. Eğilim histogramında Eğilim Büyüklükleri ve Eğilim Açılarının temsili

Dikkat edilmesi gereken bir diğer konu eğilim açılarının histogramdaki açı merkezlerine gelmediği durumlardır. Örneğin bir piksele ait eğilim açısı ve büyüklüğü sırasıyla 20 ve 90 olsun. Bu pikselin eğilim açısı 20 olduğu için histogramdaki herhangi bir eğilim açısı merkezine denk gelmemektedir. Bu nedenle bu pikselin eğilim büyüklüğünün tamamını bir eğilim açısı merkezine atmıyor; bunun yerine iki eğilim merkezi uzaklığına göre eğilim büyüklüğünü bu iki merkeze paylaştırılıyor. Eğilim açısı 20 olan pikselin 90 olan eğilim büyüklüğünü 45 ve 45 olmak üzere iki parçaya ayırarak eğilim açı merkez 10 ve 30 olan sütunlara paylaştırılır. Bu paylaştırma işlemini Şekil 8.6'yı incelenerek daha kolay anlaşılabilir.



Şekil 8.6. Eğilim büyüklüklerinin eğilim açı merkezlerine dağılımı

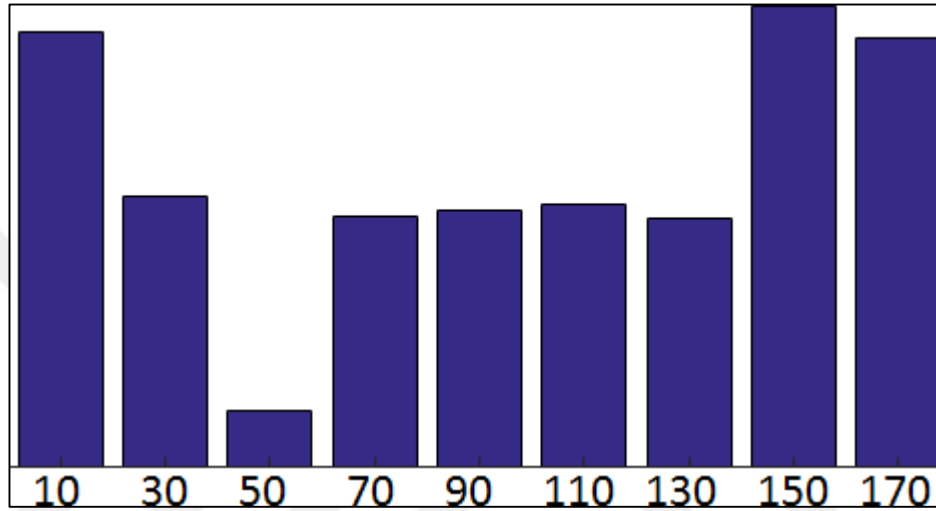
Şekil 8.6’da 8x8 ebatlarındaki bir görüntü parçasına ait eğilim büyüklükleri ve eğilim açısı matrisleri yer almaktadır. Bu matrislerin aynı konumdaki değerlerinin eğilim histogramına dağılımı gösterilmektedir. Eğilim açısı 80 ve eğilim büyüklüğü 2 olan pikselin eğilim büyüklüğü iki ayrı eğilim açı merkezlerine dağıtılmıştır. 2 olan eğilim büyüklüğünün yarısı açı merkezi 70 olan sütuna diğer yarısı da açı merkezi 90 olan sütuna eklenmiştir.

Dikkat edilmesi gereken bir diğer durum eğilim açısının iki ayrı eğilim açı merkezine aynı uzaklıkta olmamasıdır. Bu durumda eğilim açısının iki eğilim açı merkezine yakınlığı bulunur. Bulunan bu yakınlığa göre eğilim büyüklüğü eğilim açı merkezlerine paylaştırılır. Şekil 8.6’daki ikinci konumdaki eğilim büyüklüğü ve açıları sırasıyla 3 ve 36’dır. Eğilim açısı 36 her hangi bir eğilim merkezine denk gelmediği gibi iki eğilim merkezine eşit uzaklıkta değildir. 36 açısı 30 eğilim açısı merkezine 6, 50 eğilim açısı merkezine 14 birim uzaklıktadır. 36 açısına karşılık gelen eğilim büyüklüğü 3 bu oranlar dikkate alınarak paylaştırılacaktır. Paylaştırma oranı yakınlıkla doğru orantılı olarak daha fazla olacaktır. Bunun için aşağıdaki denklem kullanılacaktır:

$$\text{Paylaştırma oranı} = \text{büyüklük} \left(1 - \frac{\text{yakınlıkMesafesi}}{20}\right) \quad (8.12)$$

30 eğilim açısı merkezi için paylaştırma oranı $3 \left(1 - \frac{6}{20}\right) = 2,1$ ve 50 eğilim açısı merkezi için paylaştırma oranı $3 \left(1 - \frac{14}{20}\right) = 0,9$ elde edilir.

Sonuçta elde edilen eğilim histogramı Şekil 8.7’deki gibi olacaktır



Şekil 8.7. Elde edilecek eğilim histogramının temsili görünümü.

Histogram sütunlarının altındaki değerler eğilim açısı merkezlerini ve sütunlarda açısı merkezlerine gelen eğilim büyüklüklerinin toplamını göstermektedir.

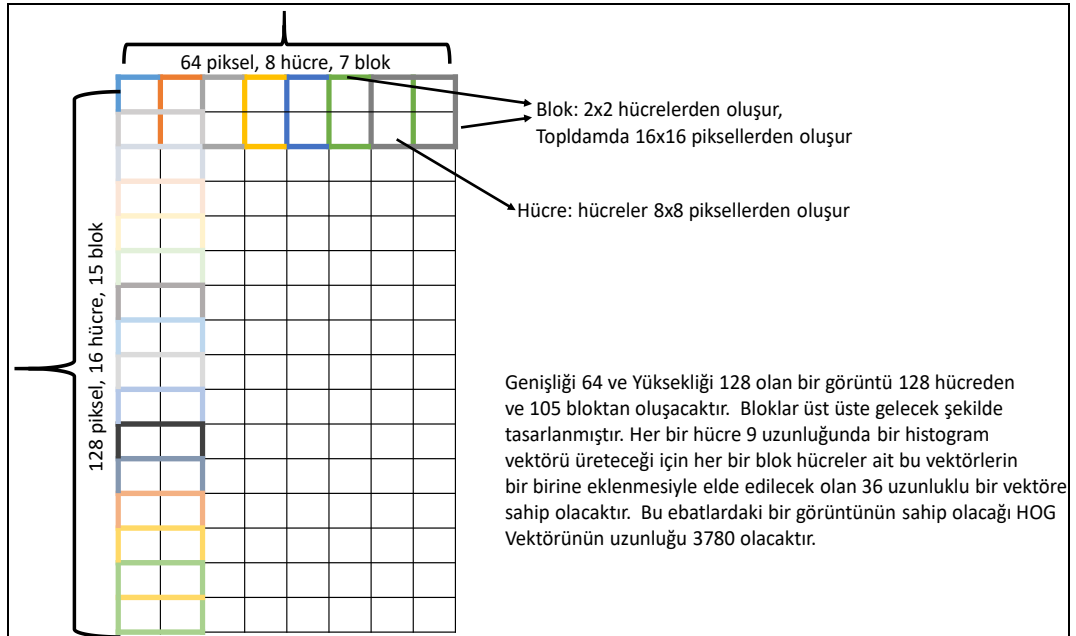
Eğilim büyüklüklerinin paylaşımında dikkat edilmesi gereken bir diğer durum 10’dan küçük ve 170’den büyük açılara sahip piksellerin eğilim büyüklüklerinin bu iki açısı merkezi arasında paylaştırılmasıdır. Örneğin eğilim büyüklüğü ve eğilim açısı sırasıyla 4 ve 5 olan pikselin eğilim histogramına eklenmesini inceleyelim. 5 açısı 10 açısı merkezine 5, 170 açısı merkezine 15 birim olacak biçimde hesaplanır. Bu açılara denk gelen eğilim büyüklüğünün 10 açısı merkezi için paylaştırma oranı $4 \times \left(1 - \frac{5}{20}\right) = 3$ olacaktır. 170 açısı merkezi için paylaştırma oranı $4 \times \left(1 - \frac{15}{20}\right) = 1$ olacaktır. Bu paylaşımın görseli Şekil 8.6’da yer almaktadır.

8.1.3.3. Hücreslerden blok oluřturma

HOG 6znitelik ıkarma yaklařımı, 8x8 ebatlarındaki hcrelerden eęilim histogramı elde eder. Hcrelerin 6znitelik vekt6rne katkısını artırmak iin hcreleri ierisine alan blok yapıları oluřturulmuřtur. Bloklar d6rt hcrenden oluřan daha byk yapılarıdır. řekil 8.8’de HOG y6ntemine ait hcre-blok yapısı g6sterilmektedir. Bloklar st ste gelecek řekilde tasarlanmıřtır. řekil 8.8’de bu yapıyı g6rmek mmkndr. Blokların yarısı dięer blokların yarısıyla st ste gelecek řekilde oluřturulur.

8x8 ebatlarındaki hcrelerden 9 uzunluklu bir histogram vekt6r elde edildięini yukarıda belirtilmiřti. Blokların sahip olacaęı vekt6r uzunluęu hcrelerden elde edilen vekt6rlerin birleřtirilmesiyle olacaktır. Birleřtirme iřlemi vekt6rlerin u uca eklenmesiyle elde edilir. Bu ekleme sonucunda elde edilecek histogram vekt6rnn uzunluęu 36 olacaktır.

HOG y6nteminin g6rntden ıkaracaęı 6znitelik vekt6r bloklardan elde edilen 6znitelik vekt6rlerinin u uca eklenmesiyle elde edilecektir. řekil 8.8.’de verilen 64x128 bir g6rnt iin elde edilecek HOG 6znitelik vekt6rnn uzunluęu 36 arpı blok sayısı kadar olacaktır. Bu g6rntdeki blok sayısı 105 olduęu iin elde edilecek 6znitelik vekt6rnn uzunluęu $36 \times 105 = 3780$ olacaktır.



řekil 8.8. 64x128 ebatlarına sahip bir g6rnt iin hcre-blok yapısı

Tüm görüntü için eğilim histogramı elde edildikten sonra yapılması gerekli son bir işlem; eğilim histogramının normalleştirilmesidir. Normalleştirme makine öğrenmesi modellerinin başarısını etkilediği için gerçekleştirilmesi gereken önemli bir işlemdir. Bir sonraki bölümde eğilim histogramının normalleştirilmesi anlatılmıştır.

8.1.3.4. Eğilim histogramının normalleştirilmesi

Bir önceki bölümde HOG yöntemine ait eğilim histogramının elde edilmesini anlatılmıştır. HOG öznitelik vektörü bu eğilim histogramlarının uç uca eklenmesinden elde edilmektedir. Elde edilen histogramın ışık farklılıklarına karşı daha iyi sonuç vermesi için normalleştirme yapmak gereklidir.

Normalleştirmenin nasıl işe yaradığını sayısal bir örnek vererek açıklayalım. [50, 100, 120, 90, 100] piksel değerlerine sahip vektör olsun. Bu piksellerin normalleştirilmesini L2 norm yöntemine göre Denklem (8.13)'teki gibi elde edilebilir;

$$L2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad (8.13)$$

İlk önce bu vektörün L2 norm formu elde edilir:

$$L2 = \sqrt{50^2 + 100^2 + 120^2 + 90^2 + 100^2} = 212.132.$$

Vektörün normalleştirilmesi vektördeki değerlerin L2 formuna bölünmesiyle elde edilir: $\left[\frac{50}{212.132}, \frac{100}{212.132}, \frac{120}{212.132}, \frac{90}{212.132}, \frac{100}{212.132} \right] = [0.23570226, 0.47140452, 0.56568542, 0.42426407, 0.47140452]$

Şimdi bu piksellerin ışık değerini artırmak için her birini 2 ile çarpalım. Yeni değerler şöyle olacaktır; [100, 200, 240, 180, 200]. Şimdi bu vektörün normalleştirilmiş halini bulalım. İlk önce L2 formunu elde ediyoruz:

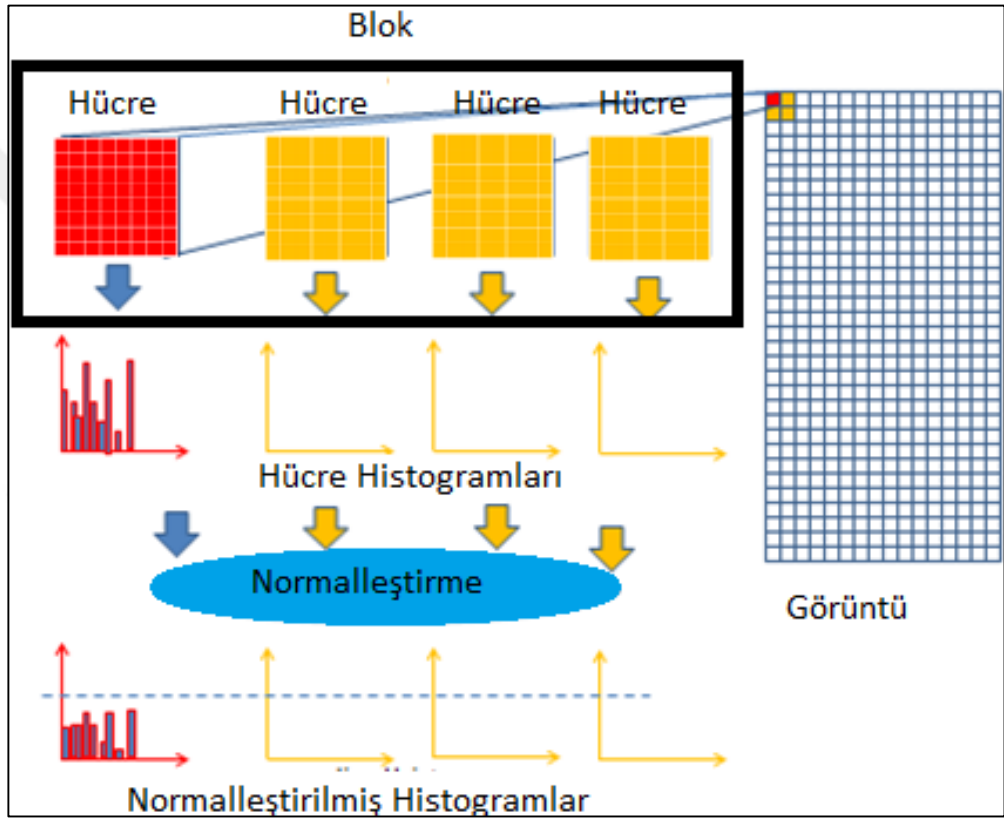
$$L2 = \sqrt{100^2 + 200^2 + 240^2 + 180^2 + 200^2} = 424.264$$

Yeni vektörün normalleştirilmiş hali: $\left[\frac{100}{424.264}, \frac{200}{424.264}, \frac{240}{424.264}, \frac{180}{424.264}, \frac{200}{424.264} \right] =$

[0.23570226, 0.47140452, 0.56568542, 0.42426407, 0.47140452]

İki farklı piksel değerlerine sahip vektör için aynı normalleştirme vektörünü elde edilmiştir. Bu, normalleştirme işleminin çarpan etkisini ortandan kaldırdığını gösteriyor.

HOG özniteliklerini elde ederken normalleştirme işlemini bloklardan elde edilen 36 uzunluklu histogram eğilim vektörleri üzerine uygulanır. Normalleştirme işlemini hücre üzerine uygulamak yerine blok histogramları üzerine uygulamak daha iyi bir fikirdir.



Şekil 8.9. HOG yönteminin özet gösterimi

Şekil 8.9.'da HOG yönteminin öznitelik vektörü elde etmesinin özet gösterimi yer almaktadır. Birinci adımda blok içerisindeki hücelere ait eğilim histogram vektörleri elde edilir. Elde edilen histogram vektörleri uç uca birleştirilerek 36 uzunluklu blok histogram eğilim vektörü elde edilir. Elde edilen bu vektörün normalleştirme işlemi yapılır. Her bir bloğun normalleştirilmiş vektörü elde edildikten sonra uç uca eklenerek HOG öznitelik vektörü elde edilir.

HOG ilk defa, Fransız Ulusal Bilgisayar Bilimi ve Kontrol Enstitüsü (INRIA) araştırmacılarından Navneet Dalal ve Bill Triggs tarafından geliştirilmiştir (Dalal ve

Triggs, 2005). HOG görüntülerdeki piksel değerlerinin yerel eğilimlerine veya kenarların yönüne göre görüntülerden öznitelik elde eder. HOG tanımlayıcılarının arkasındaki temel fikir, bir görüntü içindeki yerel nesne görünümünün ve şeklinin yoğunluk eğilimlerinin veya kenar yönlerinin dağılımı ile tanımlanabilmesidir. HOG bilgisayarla görü ve görüntü işlemede sıkça kullanılmaktadır.

HOG yönteminin görüntülerden öznitelik çıkarımını şöyle özetlenebilir: Görüntü hücre (cell) adı verilen birbiriyle bağlantılı küçük bölümlere ayrıştırılır. Blok olarak adlandırılan hücre bölümlerini içine alan daha geniş yapılar oluşturulur. Her bir blok için piksel değerlerinin yönelimi hesaplanır. Bu yönelimlerden eğim açıları elde edilir. Elde edilen iki farklı eğilimden eğilim histogramı oluşturulur. Yöntemin ana fikrini bu eğilim histogramının elde edilmesi oluşturur. Bu nedenle yöntem adını bu işlemden alır. Görüntüdeki ışık farklılıklarını en aza indirmek için elde edilen histogram normalleştirme işlemine tabi tutulur. Her bir blok için elde edilen normalleştirilmiş histogramlar birbirine eklenerek görüntüye ait öznitelik vektörü elde edilir (Cerit ve ark., 2016).

Bu çalışmada HOG öz niteliklerini hareket tespiti yapılan görüntülerden çıkartılmıştır. Özniteliklerin çıkarma işlemini dört saniyede bir gerçekleştirilmiştir. KTH veri tabanını oluşturan araştırmacıların belirttiği gibi veri setindeki hareketler dört saniyede bir gerçekleşmektedir.

Makine öğrenme modelleri veriler üzerinde tanıma işlemi yapmak için iki şeye ihtiyaç duyar; birincisi özellik vektörüne, ikincisi özellik vektörünün ait olduğu sınıfın etiketine. Elde edilen HOG özellik vektörüne ilgili hareketin etiketi atanır. Etiket atama işlemi doğrudan sınıf adının atanması yerine sınıflara verilen numaraların atanmasıyla gerçekleştirilir. Örneğin bu uygulamada var olan altı harekete 0'da 5'e kadar numaralar atanır.

8.1.4. Çapraz doğrulama (cross validation)

Veri biliminde birçok modelleme tekniği vardır. Asıl soru hangisinin daha iyi olduğudur. İyi bir modelin iki özelliği vardır: yüksek tahmin oranına sahiptir ve görmediği veriler üzerinde de iyi sonuçlar verir. Bunu başarmak için bir hata ölçütü

(modelin ne kadar yanlış olduğunu) ve bir doğrulama stratejisi tanımlanması gerekir (Cielen ve ark., 2016).

Çapraz doğrulama, genelleme performansını eldeki veriyi eğitim ve test setine ayırmadan daha istikrarlı ve kapsamlı bir şekilde artırmak için geliştirilmiş istatistiksel bir yöntemdir. Çapraz doğrulamada, veriyi eğitim ve test setine bölmek yerine, veriler tekrar tekrar bölünür ve model birden çok defa test edilir. (Mueller ve Guido, 2016).

Çapraz doğrulama için çeşitli yöntemler önerilmesine rağmen, K katlı(K-fold) çapraz doğrulama yaygın şekilde kullanılmaktadır. Bu yöntemde, eğitim seti K eşit parçaya bölünür ve bunlardan biri doğrulama seti olarak kullanılır. Kalan K-1 kısımları eğitim setini oluşturmaktadır. Her seferinde doğrulama seti olarak farklı bir bölüm kullanılır. Bu işlemi K kez tekrar edilir ve doğrulama seti üzerindeki başarısı her defasında hesaplanır. Böylece, K tane başarı değeri elde edilir. Elde edilen k tane başarı değerinin ortalaması modelin başarısı kabul edilir (Murty ve Devi, 2015).

Doğrulama son derece önemlidir çünkü modelinizin gerçek yaşam koşullarında çalışıp çalışmadığını belirler. Açıkça söylemek gerekirse, önerilen modelin bir kuruş değeri olup olmadığını gösterir (Cielen ve ark., 2016).

Eldeki veri setini bir eğitim ve test setine bölme yerine çapraz doğrulamayı kullanmanın birçok avantajı vardır. Verimizi eğitim ve test setlerine ayırdığımızda modellemesi zor verilerin eğitimde modellenmesi kolay verilerin test setinde ya da tam tersi gelme ihtimali vardır. Bu önerilen modelin gerçek performansının çok altında veya üstünde sonuçlar üretmesine yol açacaktır.

Çapraz doğrulama, veri setindeki her örneğin eğitim setinde ve test setinde kullanılmasını garanti edecektir. Dolayısıyla, modelin tüm çapraz doğrulama skorlarının (ve ortalamalarının) yüksek olması için veri kümesindeki tüm örnekler üzerinde iyi sonuç vermesi gerekir. Ayrıca çapraz doğrulama önerilen modelin vereceği başarı sonucunun aralığını belirler (Mueller ve Guido, 2016). Örneğin geliştirilen öğrenme modeli 5 farklı eğitim ve test setlerini kullanarak en kötü %85 en iyi %95 başarı oranı ortaya koydu. Bu model için kullanılan veriye göre %10'luk bir başarı değişikliği olduğu çıkarımında bulunulabilir.

Çapraz doğrulama yaklaşımları, hesaplama açısından maliyetli olabilir, çünkü eğitim verilerinin farklı alt kümelerini kullanarak aynı makine öğrenmesi yöntemi birden çok kez test edilir. Bununla birlikte, hesaplama gücünün geldiği nokta, yeniden örnekleme yöntemlerinin hesaplama gereksinimlerini genelde karşılayabilecek seviyededir (James ve ark., 2013).

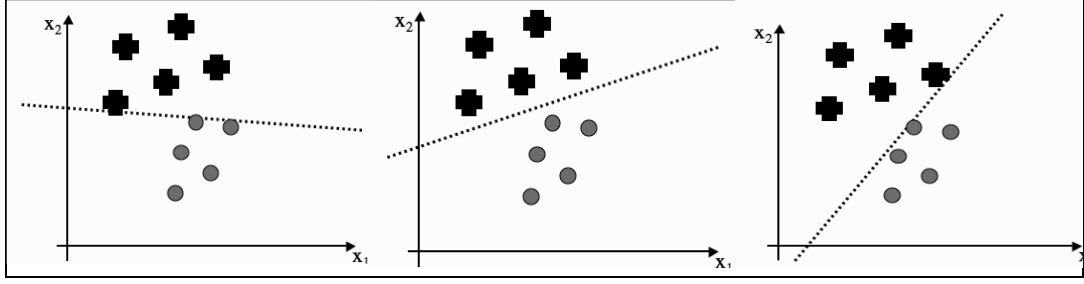
8.1.5. SVM İle hareket tanıma

Destek Vektör Makineleri (Support Vector Machine SVM), en popüler gözetimli öğrenme modellerinden biridir. SVM ortaya çıkışı Vapnik 1960'ların başında yaptığı çalışmalara dayanmaktadır. Araştırmacıların dikkatini çekmesi Bell laboratuvarlarında Cortes ve Vapnik'in beraber yaptıkları çalışmayla başlamıştır. Yapılan çalışmada SVM iki gruplu sınıflandırma problemlerinin çözümü için önerilmiştir. Yapılan çalışmada yöntemin başarısı karakter tanıma üzerinde test edilmiştir (Cortes ve Vapnik, 1995).

SVM eğitim verileri üzerinde sınıfları ayıran bir hiperdüzlem bulmayı amaçlar. Bulunan bu hiperdüzlem test verileri üzerinde sınıflandırmada kullanılır. Hiperdüzlem terimi, eğitim aşamasında elde edilen ve yanlış sınıflandırmayı en aza indiren karar sınırını belirtmek için kullanılır. Öğrenme, eğitim örneklerini ayırmak için en uygun karar sınırına sahip bir sınıflandırıcı bulma ile sağlanır. Ardından bulunan sınıflandırıcı(hiperdüzlem) test örneklerini ayırmak için kullanılır (Cortes ve Vapnik, 1995). SVM'ler son derece büyük veri kümelerinde iyi çalışmaz, çünkü hesaplamalar eğitim örneklerinin sayısı ile iyi ölçeklenmediğinden hesaplama açısından çok maliyetli olur (Marsland, 2015).

8.1.5.1. En iyi ayrıştırma

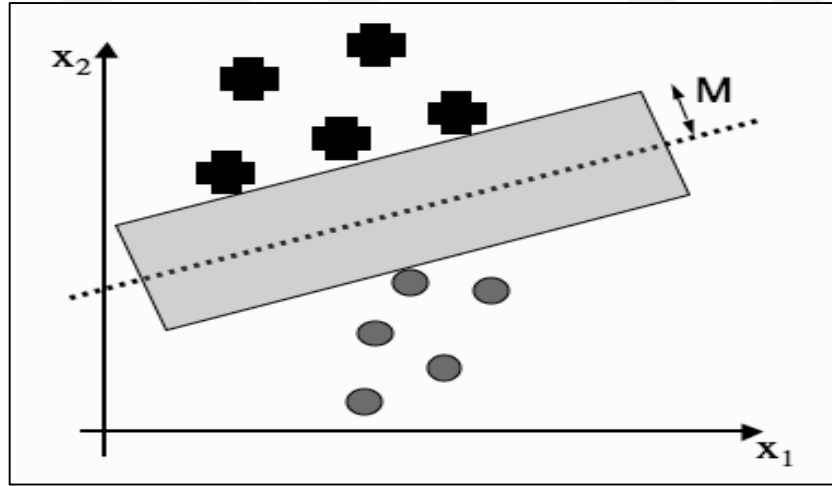
Şekil 8.10.'da doğrusal ayrılabilen iki sınıf yer almaktadır. Bu sınıflardan biri “artı” sembolü ile diğeri “daire” sembolü ile temsil edilmektedir. Bu iki sınıfı birbirinden ayıran sonsuz sayıda doğru olabilir. Şekil 8.10'da sadece sınıfları birbirinden ayıran üç tane doğru yer almaktadır. Bu doğrulardan bir tanesi diğerlerine göre daha iyi ayrıştırma yapmaktadır. Bu doğru ortadaki resimde yer alan doğrudur. Bu doğru diğer doğrulardan daha iyi yapan özellik nedir? Bu sorunun cevabı sezgisel olarak şöyle olabilir: iki sınıfı birbirinden tam ortandan ayıran doğru olduğu için.



Şekil 8.10. İki sınıfı birbirinden ayıran farklı doğrular(Marsland, 2015)

Şekil 8.10’da iki sınıf birbirinden ayıran en iyi doğru ortadaki şekilde yer almaktadır. SVM’nin amacı iki sınıfı birbirinden en iyi şekilde ayıran hiperdüzlemi bulmaktır. Şekil 8.10’daki örnek iki boyutlu veriler üzerinde gösterildiği için iki sınıfın birbirinden ayrılmasını sağlayan düzleme hiperdüzlem yerine doğru denebilir.

SVM anlatılırken sıkça kullanılan “marjın” ve “destek vektör” kavramlarını anlamak faydalı olacaktır. Marjın, sınıfları birbirinden ayıran doğrunun sınıflara olan uzaklığını ifade etmektedir. Sınıfları ayıran doğrunun “artı” sınıfına ve “daire” sınıfına olan uzaklığı veya yakınlığı eşit olmalıdır. Şekil 8.11.’de sınıfları ayıran marj bölgesi görülmektedir. Bu bölgede iki sınıfa ait herhangi bir örnek yer alamaz.



Şekil 8.11. Sınıfların, sınıfları ayıran doğruya olan uzaklığını gösteren marj bölgesi(Marsland, 2015)

Sınıflandırma doğrusuna en yakın olan her sınıftaki veri noktalarına destek vektörleri denir. En iyi sınıflandırıcının, sınıflara ait örneğin yer almadığı bölgenin ortasından geçmektedir. Bu bilgiye dayanarak şunları söylenebilir: Marj bölgesi mümkün olduğu kadar geniş tutulmalıdır ve destek vektörlerinin işimize yarayacak noktalar olduğunu anlaşılmalıdır (Marsland, 2015). SVM eğitim aşamasından sonra, sadece destek

vektörlerine ve sınıfları birbirinden ayıran doğruya ihtiyaç duyar. Bu açıdan SVM'nin hafızayı verimli kullandığı söylenebilir.

SVM yönteminde amaç iki sınıfı birbirinden ayıran doğruyu bulmaktır. Bu doğru çok boyutlu sınıfları birbirinden ayırdığı için bir hiperdüzlem olmalıdır. Bu hiperdüzlemin doğrusal ayrılabilen örnekler üzerinde tanımlanmasını yapalım. N tane örneğe sahip bir eğitim kümesi Denklem (8.14)'teki gibi tanımlansın;

$$X = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \mid x_i \in \mathbb{R}^d, y_i \in \{+1, -1\}\} \quad (8.14)$$

Denklemde d , x vektörünün özellik sayısı yani vektörün uzunluğunu, y ise özellik vektörünün ait olduğu sınıfı göstermektedir. SVM iki sınıflı bir sınıflandırma yöntemi olduğu için y 'nin sınıf etiketleri $+1$ ve -1 'dir.

Bu iki sınıfı birbirinden ayıran hiperdüzleme karar fonksiyonu dersek, tanımlaması Denklem (8.15)'teki gibi olacaktır;

$$f(x) = w^T x_i + b \quad (8.15)$$

Denklemde w^T ağırlık vektörüdür ve b bias değeridir. Sınıflandırılması istenen x vektörü, $f(x)$ fonksiyonuna sokulur ve fonksiyon ağırlık vektörü w^T ile x vektörünün çarpımının sonucu bias değeri ile toplanır ve bir değere döner. Bu değer 0'dan büyük ise x vektörünün $+$ sınıfına ait olduğu, sıfırdan küçük ise eksi sınıfına ait olduğu kabul edilir. Bu sınıflandırma işlemi Denklem (8.16)'daki gibi gösterilebilir;

$$\text{sign}(f(x)) = \begin{cases} +, & \text{Eğer } f(x) > 0 \\ -, & \text{Eğer } f(x) < 0 \end{cases} \quad (8.16)$$

Eğer $w^T x_i + b$ işleminin sonucu 0'a eşit ise iki sınıfı birbirinden ayıran hiperdüzlemin bulunduğunu gösterir. Bu gerçekten yola çıkarak destek vektörleri olarak adlandırılan hiperdüzleme en yakın örnekleri tanımlayalım. Eğer $w^T x_i + b = 1$ ise x_i pozitif destek vektörüdür. Eğer $w^T x_i + b = -1$ ise x_i negatif destek vektörüdür. Destek vektörleri en iyi hiperdüzlemin bulunmasında kullanılacaktır.

Sınıflandırma işlemi yapılabilmesi için, w ağırlık vektörünün ve b değerinin elde edilmesi gerekiyor. Bu ağırlık vektörünü ve bias değerini elde ederken dikkat edilmesi gereken kısıtlar var. Birinci kısıt; iki sınıfı ayıran marjın mümkün olduğu kadar fazla

olmasıdır. Marj ağırlık vektörünün büyüklüğüne bağlıdır. Bir vektörün büyüklüğü o vektörün normu demektir ve Denklem (8.17)'deki gibi gösterilir;

$$\|w\| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2} \quad (8.17)$$

İki sınıfı birbirinden ayıran sonsuz sayıda hiperdüzlem olabilir. Bu hiperdüzlemler arasında iki sınıf arasında maksimum marjı veren bir hiperdüzlem vardır. En iyi hiperdüzlem marjı maksimize eder. Bunu için birbirine paralel olan iki hiperdüzleme dik olması gerekir. Hiperdüzlemlere dik olan doğru normu en küçük olan doğrudur. Yani marjı maksimize ederken ağırlık vektör büyüklüğünü minimize edilir.

Lineer hiperdüzlemin Denklem (8.18) ve (8.19)'daki kısıtları da karşılaması gerekir;

$$y_i = 1 \quad w^T x_i + b \geq 1 \quad (8.18)$$

$$y_i = -1 \quad w^T x_i + b \leq -1 \quad (8.19)$$

Denklemlerinde $i=1,2,\dots,n$. Yukardaki denklemleri tek bir denklem ile temsil edilebilir. Her iki denklemi, karşılık gelen y_i ile çarpılırsa yeni denklem, Denklem (8.20)'deki gibi olacaktır;

$$y_i(w^T x_i + b) \geq 1 \quad (8.20)$$

8.2. Deneysel Çalışma

Bu bölümde yapılan hareket tanıma çalışmasının sonuçlarına yer verilecektir. Sonuçlar sınıflandırma performansını ölçen yaklaşımlarla ortaya konacaktır. İlk önce sınıflandırma modelinin performansını gösteren Karışıklık Matrisinin ne olduğunu anlamak faydalı olacaktır.

8.2.1. Karışıklık matrisi

Karışıklık matrisi, sınıflandırma algoritmasının performansını özetlemek için kullanılan bir tablolama tekniğidir. Karışıklık matrisi neden ihtiyaç duyulur, sorusunun cevabı: Yapılan sınıflandırma değerlendirmesinin sadece başarı oranına göre yapılması yanıltıcı olabilir. Başarı oranı, doğru yapılan sınıflandırma sayısının toplam

sınıflandırma sayısına bölümünden elde edilir. Tablo 8.1’deki örnek karışıklık matrisini inceleyerek konuyu netleştirelim.

Tablo 8.1’de “Elmalar” ve “Armutlar” adında iki sınıfın sınıflandırılmasına ait karışıklık matrisi yer almaktadır. Tabloya göre sınıflandırmanın başarı oranı= $15/20=0,75$. Sınıflandırmanın başarı oranından tasarlanan sınıflandırma modelinin hata oranı elde edilebilir. Tablo 8.1 için hata oranı $1-0,75=0,25$. Her sınıfta eşit olmayan sayıda örnek varsa veya veri kümemizde ikiden fazla sınıf varsa yalnızca başarı oranı sınıflandırma modelini değerlendirmek için yeterli olmayabilir.

Tablo 8.1. Örnek bir Karışıklık Matrisi

N=20		Tahmin Edilen	
		Elmalar	Armutlar
Gerçek	Elmalar	5	1
	Armutlar	4	10

Sınıflandırma başarı ölçütündeki asıl problem, sınıflandırma modelinin performansını daha iyi anlamak için gereken ayrıntıları gizlemesidir. Elimizdeki verilerde ikiden fazla sınıf varsa başarı oranı hangi sınıfın ne oranda başarı oranına sahip olduğu bilgisini veremez. Ayrıca veri setindeki sınıflar eşit sayıda örneğe sahip değilse başarı oranı sınıflandırma modelinin başarısını tam yansıtamaz. Sınıflandırma başarı oranı, modelin performansını teşhis etmek için ihtiyaç duyulan ayrıntıları gizleyebilir. Karışıklık matrisi kullanarak bu ayrıntıyı ayırt edebiliriz. Karışıklık matrisi, sınıflandırma modelinin haklı çıktığı durumları ve hangi hataların oluştuğu hakkında daha iyi bir fikir verir.

Karışıklık matrisi geliştirilen sınıflandırma modelinin tahmin sonuçlarını özetleyen bir tablodur. Doğru ve yanlış tahminler her bir sınıf için gösterilir. Sınıflandırma modelinin yaptığı yanlış tahminlerin detayını verir (Kuyumcu, 2016). Sadece, sınıflandırma modelinin hata yaptığını söylemek yerine hatayı nasıl yaptığını söyler. Böylece “başarı oranı sonucunda” gizli kalan detaylar karışıklık matrisi kullanılarak ortaya çıkar.

8.2.2. Karışıklık matrisinin oluşturulması

Karışıklık matrisini oluşturma için yapılması için gereken adımlar şöyledir:

1. Her sınıf için doğru tahmin sayısı bulunur

2. Her bir sınıf için yapılan yanlış tahminler sınıflara göre sayılır
3. Sınıf sayısı n olana bir model için nxn bir matris oluşturulur
4. Matrisin satırları gerçek sınıfları, sütunları tahmin edilen sınıfları temsil eder.
5. Yanlış ve doğru sınıflandırma sayıları bu tabloya doldurulur

Tablo 8.1.'de yer alan karışıklık matrisinde 20 adet sınıflandırma işlemi yapılmış. Elmalar sınıfında yer alan 6 örnekten 5'i doğru 1'i yanlış sınıflandırılmış. Armutlar sınıfında yer alan 14 örnekten 10'u doğru 4'ü yanlış sınıflandırılmış. Sınıflandırma modeli 15 kez doğru tahminde 5 kez yanlış tahminde bulunmuştur.

Karışıklık matrisinde yer alan hücrelerin genel bir adlandırması vardır. Bu adlandırma her bir sınıf için ayrı bir anlam ifade eder. Ayrıca, bu adlandırmalar her problem için aynı terimleri kullanma imkânı verir. Tablo 8.1'i Tablo 8.2'deki gibi genelleştirilebilir.

Tablo 8.2. Elmalar sınıfının genelleştirilmiş karışıklık matrisi

N=20		Tahmin Edilen	
		Elmalar	Armutlar
Gerçek	Elmalar	True Positive(TP)=5	False Positive(FP)=1
	Armutlar	False Negative(FN)=4	True Negative(TN)=10

- TP: Doğru tahmin edilen elmalar.
- TN: Doğru tahmin edilen elma olmayanlar.
- FP: Yanlış tahmin edilen elmalar.
- FN: Yanlış tahmin edilen elma olmayanlar.

Tablo 8.2'ye göre dört farklı istatistiksel metrik hesaplanmıştır. Başarı oranı, doğru pozitif oranı, kesinlik ve f1skoru sırasıyla Denklem (8.21), (8.22), (8.23) ve (8.24)'deki gibi elde edilir;

$$\text{Başarı Oranı(Accuracy Rate)} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8.21)$$

$$\text{Doğru Pozitif Oranı(Recall)} = \frac{TP}{TP+FN} \quad (8.22)$$

$$\text{Kesinlik(Precision)} = \frac{TP}{TP+FP} \quad (8.23)$$

$$\text{F1 Skoru} = \frac{2TP}{2TP+FP+FN} \quad (8.24)$$

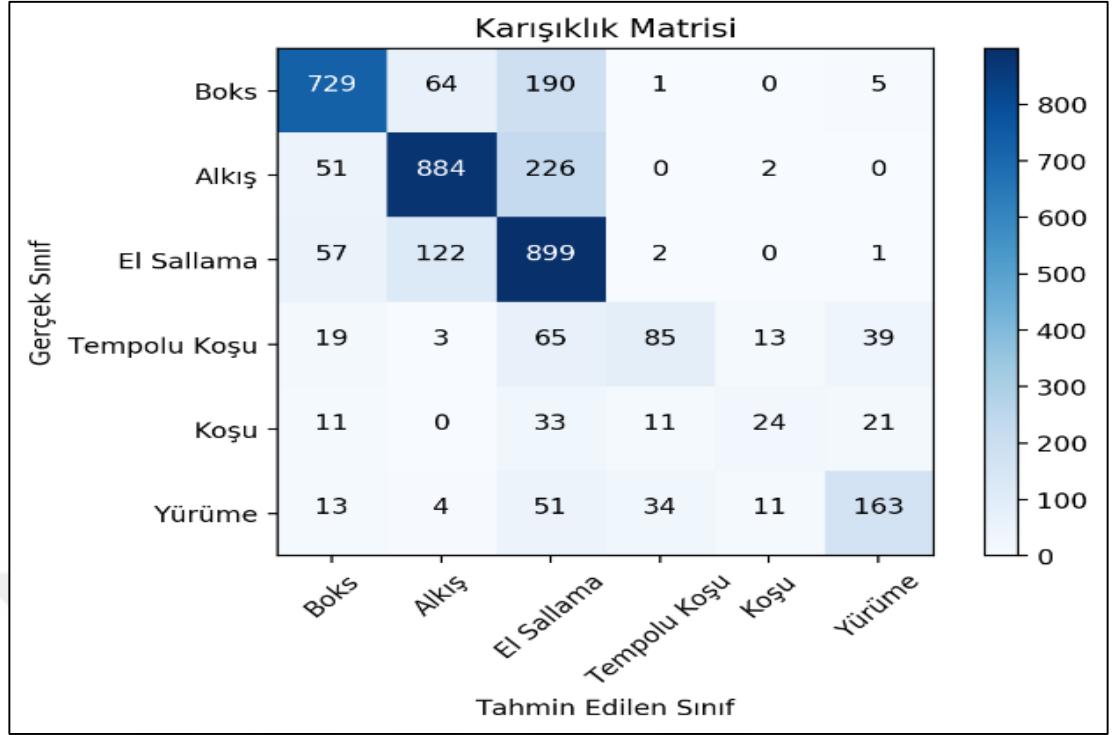
8.2.3. Deneysel sonuçlar

Bu bölümde GMACA ile yapılan hareket tespit görüntülerinden insan hareketlerinin tanınması yapılmıştır. Tanıma işleme, insan hareketlerinin video görüntülerden tanınması için yapılan çalışmalarda kullanılması amacıyla hazırlanan KTH ve WEIZMANN veri setlerindeki video görüntülerde yer alan insan hareketleri üzerinde gerçekleştirilmiştir.

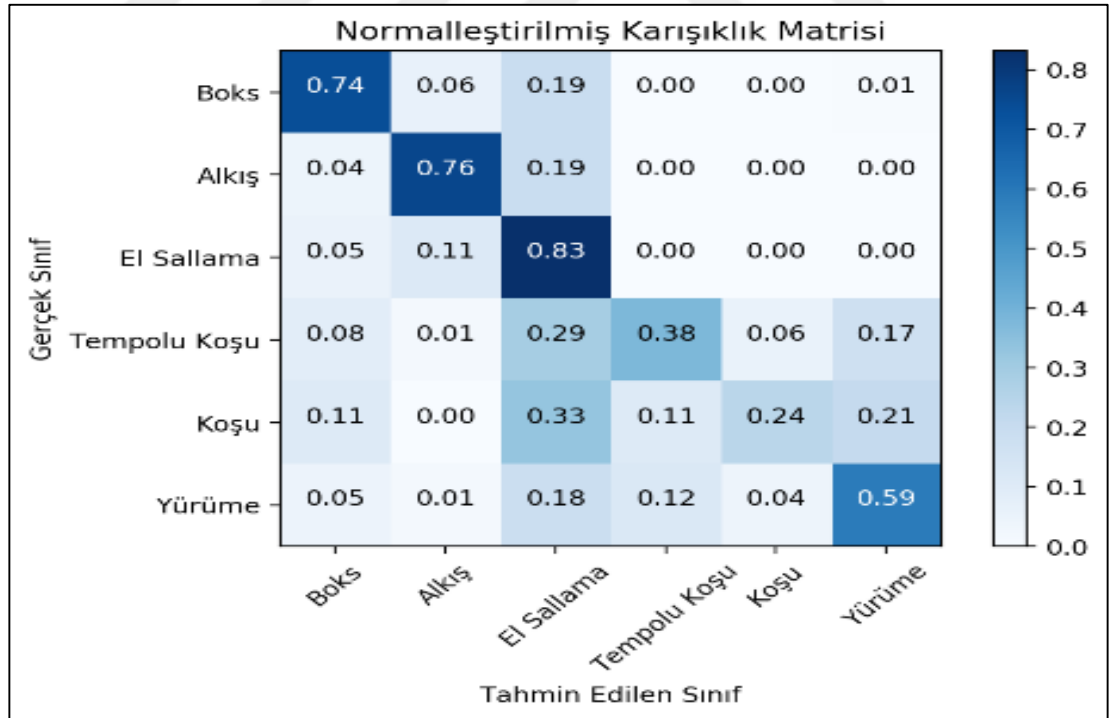
KTH veri tabanında altı farklı hareket biçimine ait videolar vardır. Geliştirilen yöntem birbirinden farklı bu hareketler üzerinde iyi sonuçlar vermiştir. KTH veri tabanında yer alan hareketler şunlardır; boks, alkış, el sallama, tempolu koşu, normal koşu ve yürüme.

WEIZMANN veri tabanında on farklı hareket biçimine ait videolar vardır. WEIZMANN veri tabanında yer alan hareketler şunlardır; eğilme, tek el sallama, iki el sallama, jack(iki el ve iki bacağın yerinde zıplayarak beraber açılıp kapanması), yerinde zıplama, ileri doğru zıplama, yan yürüme, yürüme, koşma ve tek ayak üzerinde sekerek yürüme.

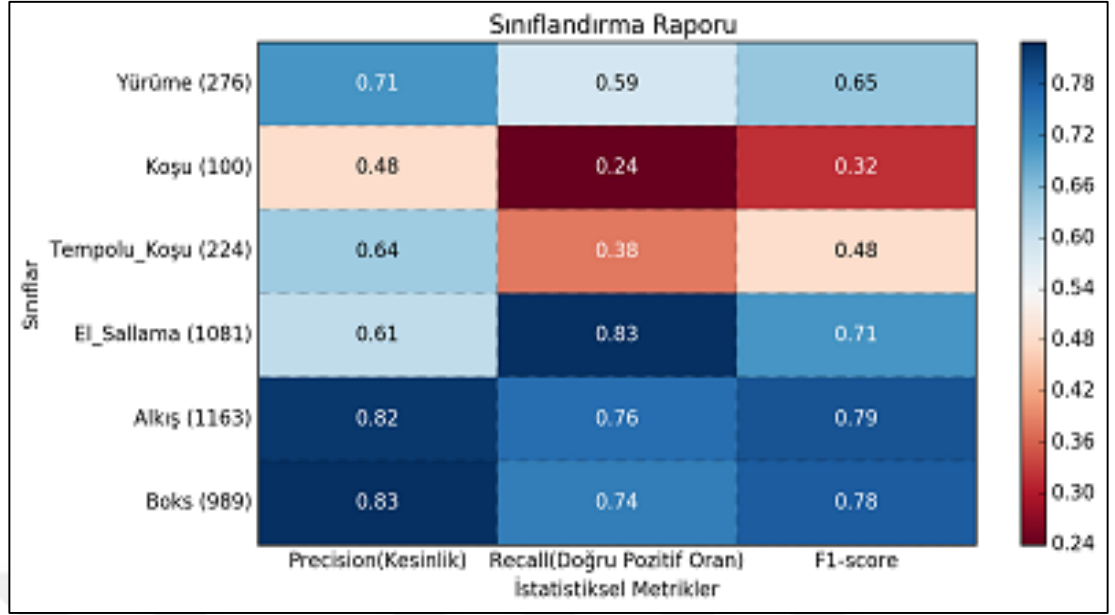
Elde edilen tanıma sonuçları, istatistiksel sınıflandırma metrikleri olan; karışıklık matrisi, normalleştirilmiş karışıklık matrisi ve sınıflandırma raporuyla verilmiştir. Aşağıdaki şekillerde KTH ve WEIZMANN veri setleri üzerinde elde edilen sınıflandırma sonuçlarına ait karışıklık matrisleri, normalleştirilmiş karışıklık matrisleri ve sınıflandırma raporları verilmiştir. Şekil 8.12, 8.13 ve 8.14'te KTH veri seti üzerinde elde edilen sınıflandırma sonuçlarına ait, sırasıyla; karışıklık matrisi, normalleştirilmiş karışıklık matrisi ve sınıflandırma raporu yer almaktadır. Şekil 8.15, 8.16 ve 8.17'de WEIZMANN veri seti üzerinde elde edilen sınıflandırma sonuçlarına ait, sırasıyla; karışıklık matrisi, normalleştirilmiş karışıklık matrisi ve sınıflandırma raporu yer almaktadır



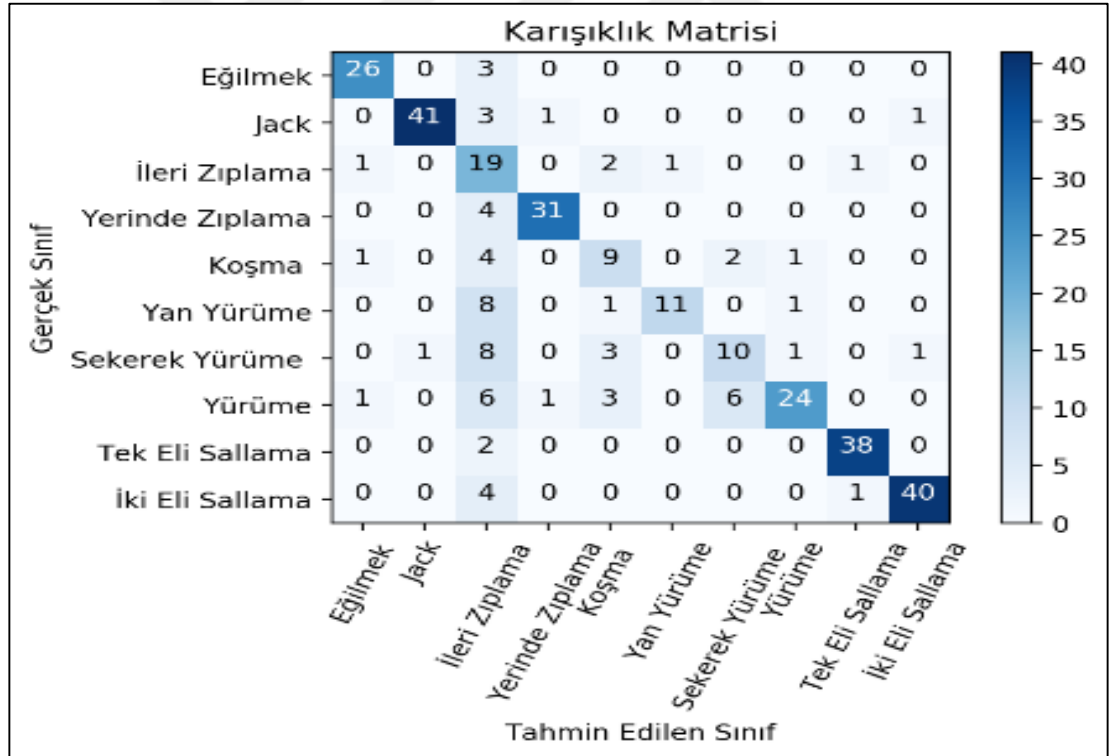
Şekil 8.12. KTH veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait karışıklık matrisi.



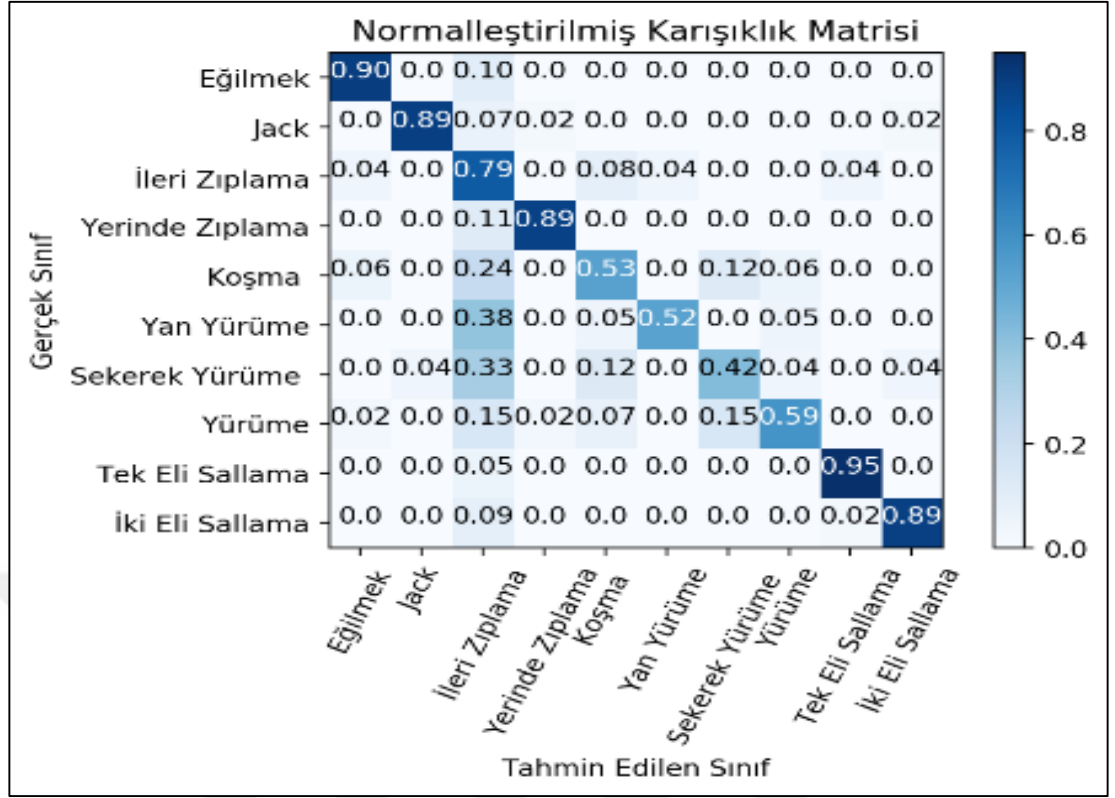
Şekil 8.13. KTH veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait normalleştirilmiş karışıklık matrisi.



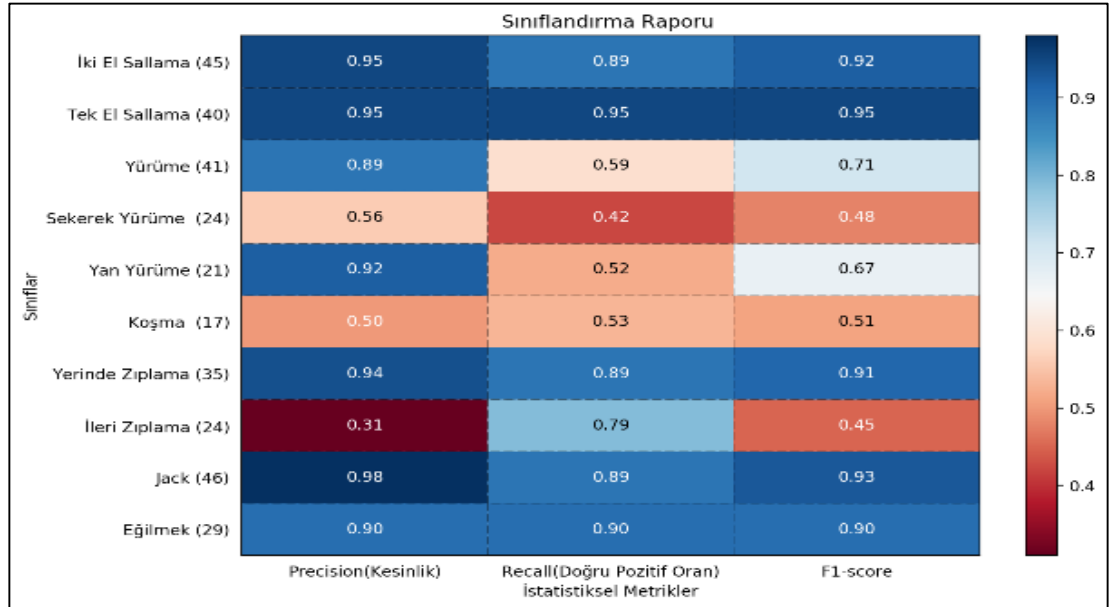
Şekil 8.14. KTH veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait sınıflandırma raporu.



Şekil 8.15. WEIZMANN veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait karışıklık matrisi.



Şekil 8.16. WEIZMANN veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait normalleştirilmiş karışıklık matrisi.



Şekil 8.17. WEIZMANN veri seti üzerinde GMACA hareket tespit görüntüleri kullanılarak elde edilen hareket tanıma sonuçlarına ait sınıflandırma raporu.

Aşağıdaki tablolarda aynı veri setleri üzerinde diğer araştırmacılar tarafından elde edilmiş başarı oranı ve geliştirilen yönetime ait başarı oranları verilmiştir. Tablo 8.3'te

KTH veri seti üzerinde yapılan diğer çalışmalarla karşılaştırma yapılırken, Tablo 8.4'te WEIZMANN veri seti üzerinde yapılan diğer çalışmalarla karşılaştırma yapılmıştır.

Tablo 8.3. Geliştirilen yönteme ve literatürde KTH veri seti üzerinde yapılan çalışmalara ait başarı oranları

Yöntem	Başarı Oranı (%)
Schüldt ve diğ (Schuldt ve ark., 2004)	71,63
Schindler ve Gool (Schindler ve Gool, 2008)	87,98
Baumann (Baumann, 2013)	94,31
Laptev ve diğ. (Laptev ve ark., 2008)	91,8
O'Hara and Draper (O'Hara ve Draper, 2012)	97,9
Geliştirilen Yöntem	72,70

Tablo 8.4. Geliştirilen yönteme ve literatürde WEIZMANN veri seti üzerinde yapılan çalışmalara ait başarı oranları

Yöntem	Başarı Oranı (%)
Niebles ve Fei-Fei (Niebles ve Fei-Fei, 2007)	72,80
Dollar ve diğ. (Dollar ve ark., 2005)	86,70
Ali ve diğ. (Ali ve ark., 2007)	92,60
Wang ve Suter (Wang ve Suter, 2007)	97,80
Geliştirilen Yöntem	77,3

8.2.4. Sonuç

GMACA ile elde edilen hareket tespit görüntüleri ile yapılan hareket tanıma uygulaması mevcut arka plan çıkarma çalışmaları kadar iyi sonuç vermiştir. Elde edilen sonuçlar GMACA'nın hareket tespitinde ve hareket tanıma çalışmalarında kullanılabileceğini göstermektedir. Daha önceki çalışmamızda GMACA'yı görüntü bölümlendirmede ve hareket tespitinde kullanmıştık. Yapılan çalışmalar, GMACA kullanılarak yapılan ilk görüntü bölümlendirme ve hareket tespit çalışmalarıydı. GMACA kullanılarak yapılan hareket tespit yönteminin, gerçek zamanlı uygulamalarda kullanılabileceğini göstermektedir. Hareket tanıma çalışması, önceki bölümde önerilen hareket tespit yönteminin, insan hareketlerini tanımada da kullanılabileceğini göstermektedir.

9. SONUÇLAR VE ÖNERİLER

Tezin katkıları, her bir bölümde detaylıca anlatılmıştır. Bu bölümde tezin genel değerlendirmesi ve ana bölümlerde yapılan çalışmaların katkıları özet olarak sunulmuştur. Ayrıca, hücrel otomata tabanlı yapılabilecek bilimsel çalışmalar için önerilerde bulunulmuştur.

9.1. Tezin Bilimsel Literatüre Katkıları

Öncelikle şunu belirtmek gerekir: Hücrel otomata hesaplama modeli üzerine çalışılmış Türkiye merkezli ve Türkçe bilimsel çalışmalar oldukça azdır. Hatta bu tez kapsamında üzerinde çalışılan örüntü sınıflandırma için kullanılan hücrel otomatlar üzerine herhangi bir Türkiye merkezli ve Türkçe çalışmaya rastlanmamıştır. Bu açıdan tez kapsamında yapılan ve rapor edilen çalışmalar önemli bir boşluğu dolduracaktır.

Bu tezin amacı, GMACA'ların ürettiği ve çekim merkezi formunda davranan cezbedicileri örüntü sınıflandırma ve görüntü işlemlerde kullanmaktır. İkinci bölümde hücrel otomata durum uzayındaki durumu geçişlerinin nitelenmesi için gerekli teorik alt yapı yer almaktadır. Hücrel otomatların nitelenmesi için lineer olmayan yöntemlere ilgi duyan araştırmacıların faydalanabileceği, kapsamlı bilgiler ve anlatımlar vardır. Erişilebilirlik ağacının oluşturulması, ulaşılabilir ve ulaşılmaz durumların belirlenmesi, cezbediciler için erişilebilirlik ağacı ve kapsayıcı bitler için erişilebilirlik ağacının kullanılması konuları detaylıca anlatılmıştır.

Üçüncü bölümde hücrel otomatların uygulamasında kullanılabilecek, var olan bir veri organizasyon yöntemi önerilmiştir; birleştir/bul veri yapısı. Hücrel otomatların dinamik ve ayrık yapısı birleştir/bul veri yapısı ile kolay ve verimli bir şekilde uyum sağlamıştır. Kural vektöründeki kuralların özelliklerine göre oluşan ters döndürülebilir ve ters döndürülemez hücrel otomatların kümelere ayrıştırılması gerçekleştirilmiştir. Ayrıca, girdi olarak verilen bir elamanın hangi kümeye ait olduğu bulunması işlemi, yine birleştir/bul veri yapısı kullanılarak gerçekleştirilmiştir. Birleştir/bul veri yapısı için önerilen; hızlı-bul, hızlı-birleştir, ağırlıklı-hızlı-birleştir ve yol-sıkıştırılmalı-hızlı-

birleřtir yöntemleri üçüncü bölümde anlatılmıřtır. Performansı en yüksek olan yol-sıkıřtırmalı-ağırlıklı-hızlı-birleřtir yöntemi görüntü verileri üzerinde GMACA durum geçiř diyagramlarının ayrıřtırılmasında kullanılmıřtır. Kullanılan görüntü verileri yüz veri tabanından alınmıřtır. Veri tabanında 40 kiřiye ait 10 farklı yüz görüntüsü vardır. Veri tabanındaki toplam yüz görüntüsü sayısı 400'dür. Birleřtir/bul veri yapısıyla GMACA durum geçiř diyagramlarının ayrıřtırılması 400 görüntü verisi için çok hızlı sürede gerekleřtirilmiřtir.

Dördüncü bölümde GMACA'ların örüntü çekim gücü ilk defa görüntü bölümlendirmede kullanılmıřtır. Yapılan literatür arařtırmasında GMACA tabanlı görüntü iřleme alıřmasına rastlanmamıřtır. alıřma, görüntü iřleme için GMACA modelleme kabiliyetini kullanmaya yönelik ilk giriřimdir. Yöntem, kümeleme görüntü bölümleme yöntemleri altında deęerlendirilebilir. Temel varsayım, bir bölgeye ait piksellerin yakın deęeri olacak olmasıdır. Daha önceki temel CA tabanlı görüntü iřleme alıřmalarının çoęu, ikili görüntüler üzerinde alıřmaktadır. Bu bölümde geliřtirilen görüntü bölümlendirme yöntemi, gri görüntüler üzerinde gerekleřtirilmiřtir. Gri görüntüler üzerinde alıřabilen CA tabanlı yeni bir yöntem arařtırmacıların dikkatini çekecektir. Görüntü bölümlendirme için Berkeley Bölümlendirme Veri seti(Berkeley Segmentation Dataset) kullanılmıřtır. Elde edilen sonuçlar literatürde yer alan sonuçlarla hem görsel hem de istatistiksel metriklerle karřılařtırılmıřtır.

Beřinci bölümde GMACA'ların örüntü çekim gücü boyut azaltmada kullanılmıřtır. Yapılan alıřma GMACA kullanılarak yapılan ilk boyut azaltma alıřmasıdır. Yapılan alıřmada, el yazması rakam görüntülerindeki her bir satır 8 bit uzunluklu ikilik tabandaki deęerlere dönüřtürüldükten sonra GMACA boyut azaltmada kullanılmıřtır. 64 uzunluęundaki özellik vektörü 16 ve 8 uzunluklu özellik vektörlerine dönüřtürülmüřtür. Boyutu azaltılan özellik vektörleri KNN algoritması kullanılarak tanıma iřlemi yapılmıřtır. KNN ile yapılan tanıma iřleminde farklı uzaklık metrikleri kullanılmıřtır. Kullanılan uzaklık metrikleri Öklid, Manhattan ve Hamming uzaklık metrikleridir. En iyi tanıma oranları Hamming ve Manhattan uzaklık metrikleri ile elde edilmiřtir. Özellik vektörünün boyutu 8 uzunluklu vektöre dönüřtürüldüğünde % 81 tanıma oranı Hamming uzaklıęı metrięi kullanan KNN algoritmasıyla elde edilmiřtir. Özellik vektörünün boyutu 16 uzunluklu vektöre dönüřtürüldüğünde % 86 tanıma

oranı Manhattan uzaklığı metriği kullanan KNN algoritmasıyla elde edilmiştir.Özellik vektörünün boyutu azaldıkça tanıma oranında azalmalar olmuştur.

Altıncı bölümde GMACA'lar ilk defa video görüntülerindeki hareket tespitinde kullanılmıştır. CA'nın paralel hesaplama imkanı sağlayan hesaplama modeli, önerilen yöntemi ilgi çekici hale getirmektedir. Önerilen yöntem, GMACA'nın ileri görüntü işlemede kullanılabilirliğini göstermesi açısından, ilham vericidir. Ardışık görüntülerin karşılaştırmasına göre yapılan GMACA tabanlı hareket tespit çalışması gayet iyi sonuçlar vermiştir. Çalışma, GMACA kullanılarak yapılan ilk hareket tespit çalışmasıdır. Elde edilen sonuçlar GMACA'nın hareket tespitinde kullanılabileceğini göstermektedir. Bir önceki bölümde GMACA görüntü bölümlendirmede kullanmıştır. Yapılan görüntü bölümlendirme, GMACA kullanılarak yapılan ilk görüntü bölümlendirme çalışmasıydı. Bu bölümdeki hareket tespit çalışması, GMACA'nın gerçek zamanlı uygulamalarda da kullanılabileceğini göstermektedir. Önerilen hareket tespit yönteminin önceki hareket tespit çalışmalarından ayıran en önemli özelliği ardışık görüntülerin farkından yararlanmamasıdır. Bunun yerine ardışık görüntülerin ait olduğu cezbediciye göre hareket tespiti yapmasıdır.

Yedinci bölümde, insan hareketlerinin video görüntülerden tanınması üzerine yapılan bilimsel çalışmaları kapsamlı taraması yapılmıştır. İnsan hareketlerinin tanınması üzerine çalışmak isteyen araştırmacıların faydalanabileceği, inceleme çalışması niteliğindedir. Video görüntülerdeki insan hareketlerinin tanınması, birçok görevin yerine getirilmesiyle gerçekleştirildiği için literatür değerlendirmesi bu görevler dikkate alınarak oluşturulmuştur. İlk önce hareketin farklı formlarının, araştırmacılar tarafından nasıl adlandırıldıkları ele alınmıştır. Hareket tanımda temel işlemler olan; hareket tespiti ve takibi üzerine yapılan çalışmalar, ayrı başlıklar altında incelenmiştir. Makine öğrenmesi modeli için kullanılan özellik çıkarma yöntemleri değerlendirilmiştir. Son olarak hareketlerin sınıflandırılması için kullanılan makine öğrenmesi yöntemleri listelenmiştir

Sekizinci bölümde GMACA ile yapılan hareket tespit görüntüleri kullanılarak, insan hareketlerinin tanınması gerçekleştirilmiştir. GMACA ile elde edilen hareket tespit görüntüleri ile yapılan hareket tanıma uygulaması mevcut arka plan çıkarma çalışmaları kadar iyi sonuç vermiştir. Hareket tanıma çalışması, önceki bölümde

önerilen hareket tespit yönteminin, hareket tanımada da kullanılabileceğini göstermektedir.

9.2. Gelecekte Yapılacak Çalışmalar İçin Öneriler

Tez kapsamında gerçekleştirilen çalışmaların, birçok bilimsel çalışmaya ilham verecek önemli yönleri vardır. Yapılan hücresel otomata tabanlı çalışmalara ilgi duyan araştırmacılar için öneriler aşağıda listelenmiştir.

Öncelikle şunu belirtmek gerekir ki; Wolfram'ın vurguladığı gibi, hücresel otomatların hesaplama modelinin dokunmadığı çok az bilimsel alan vardır. Bu bağlamda araştırmacıların, hücresel otomatların hesaplama modelini yeni alanlara uygulamak için çabada bulunmaları önerilmektedir.

Üçüncü bölümde durum geçiş diyagramlarının ayrıştırılması için kullanılan birleştir/bul veri yapısı, görüntü verileri üzerinde kullanılmıştır. Aynı yöntemi, başka tür verilerde kullanılması, uygulama meraklıları için açık bir alan olarak durmaktadır.

İkinci bölümde tek uzunluk döngülü cezbedicilerin nitelenmesi için, erişilebilirlik ağacı tabanlı yöntemler kullanılmıştır. Genetik algoritmalar, cezbedicilerin nitelenmesi için kullanılan diğer bir yaklaşımdır. Genetik algoritmalar kullanılarak yapılan cezbedici nitelenmesi yöntemlerinin araştırılması Türkçe literatürüne önemli katkıları olacaktır.

Veri setlerinin üzerinde GMACA kullanılarak sınıflandırma yapmak için veri setinin GMACA'ya uygun hale getirecek ön işlemlerden geçirilmesi gerekir. Farklı ön işlem stratejileriyle GMACA sınıflandırıcılarının performansı artırılabilir.

Bu tez kapsamında GMACA sınıflandırıcıları, ilk defa çoklu sınıflara sahip veri setlerinin sınıflandırılmasında kullanılmıştır. Kullanılan yöntemlerle elde edilen başarı oranları, GMACA'ların çoklu sınıflandırmada kullanılabileceğini gösterse de, daha iyi başarı sonuçları için yeni yöntemler araştırılabilir. Daha iyi çoklu sınıf stratejilerinin geliştirilmesi GMACA sınıflandırıcılarının geniş çapta kullanımının yolunu açacaktır.

Hücresel otomatların durum geçiş diyagramlarının davranışlarına göre ikiye ayrılır; ters döndürülebilir ve test döndürülemez hücresel otomatlar. Sınıflandırma için ters

döndürülemez hücresele otomatlar kullanılmıştır. Ters döndürülebilir hücresele otomatlar, kriptoloji alanında kullanım potansiyeline sahiptir. Bu potansiyel, farklı uygulama alanlarında kullanılabilir.

Dördüncü bölümde önerilen görüntü bölümlendirme yöntemi, GMACA kullanılarak yapılan ilk bölümlendirme çalışmasıdır. Önerilen yöntemde görüntü bölümlendirme yapılırken, görüntü piksellerinin değerlerine odaklanılmıştır. Birbirine yakın değere sahip piksellerin aynı bölgeye ait olduğu varsayımıyla bölümlendirme işlemi yapılmıştır. Piksellerin konum yakınlıkları dikkate alınarak yapılacak çalışmalar, açık bir alan olarak araştırmacıları beklemektedir.

Beşinci bölümde GMACA'lar boyut azaltmada kullanılmıştır. Boyut azaltmanın uygulandığı veri seti, 64 uzunluğunda özellik vektörüne sahiptir. GMACA'ların boyut azaltma yeteneklerinin, daha büyük boyutlarda özellik vektörüne sahip veriler üzerinde gerçekleştirilmesi araştırmacılar için cazip bir alandır. Ayrıca, yapılan boyut azaltma işlemi görüntü verileri üzerinde gerçekleştirilmiştir. Sinyal, finans, metin ve tıbbi verileri üzerinde kullanılabilir. Boyut azaltma gözetimsiz öğrenmenin önemli bir alanıdır. GMACA'ları farklı gözetimsiz öğrenme alanlarında kullanmak, doktora seviyesinde yapılabilecek çalışmalara odak noktası olabilir.

Altıncı bölümde yer alan hareket tespit çalışması, GMACA kullanılarak yapılan ilk hareket tespit çalışmasıdır. Yapılan çalışma, yoğunluk görüntüleri üzerinde hareket tespiti yapma yeteneğine sahiptir. Hareket tespiti, uygulama alanı çok geniş bir görüntü işleme görevidir. Önerilen hareket tespit çalışmasını, farklı uygulama alanlarında var olan hareket tespit çalışmalarından daha iyi sonuçlar elde etmek için bilimsel araştırmalar yapılabilir. Önerilen hareket tespit çalışması, ardışık görüntü çerçevelerin durum değişimine göre hareket tespiti yapmaktadır. Sabit arka plan çıkarımına göre hareket tespiti yapan yöntemler gibi sabit arka plan değişimine göre hareket tespiti yapan, GMACA tabanlı hareket tespiti uygulaması gerçekleştirilebilir.

Sekizinci bölümde GMACA kullanılarak yapılan hareket tespit görüntüleri kullanılarak, insan hareketlerinin tanınması gerçekleştirilmiştir. Hareket tanıma çalışmasında, HOG öznitelik çıkarımı yöntemi kullanılmıştır. Aynı hareket tespit yöntemiyle; SURF, SIFT, NWFE, LKT, LESH ve PCA-SIFT gibi farklı öznitelik çıkarma yöntemleri kullanılabilir. Geliştirilen hareket tanıma çalışması, insan

hareketlerinin global örüntülerini kullanarak sınıflandırma yapmaktadır. İnsan vücudunun lokal davranışlarına odaklanan yeni yöntemler geliştirmek, araştırmacıları zorlayacak önemli bir görev olarak tanımlanabilir.



KAYNAKLAR

Aggarwal J., Ryoo M., Human activity analysis: A review, *ACM Computing Surveys (CSUR)*, 2011, **43**(3), 16-65.

Ahad M., *Computer Vision and Action Recognition*. 1st ed., Atlantis Press, 2011.

Ahad M., *Motion History Images for Action Recognition and Understanding*, 1st ed., Springer , London, 2013.

Ali S., Basharat A., Shah M., Chaotic Invariants for Human Action Recognition, *IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brezilya, 14-21 Ekim 2007.

Alpaydm E., *Introduction to Machine Learning*, 3rd ed., The MIT Press, Cambridge Massachusetts ,2014.

Amrogowicza S., Zhaob Y., Zhao Y., An Edge Detection Method Using Outer Totalistic Cellular, *Neurocomputing*, 2016, **214**(1), 643-653.

Arbelaez P., Maire M., Fowlkes C., Malik J., Contour Detection and Hierarchical Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, **33**(5), 898 - 916.

Artan Y., Supervised and Interactive Image Segmentation Techniques with an Application to Prostate Cancer, Doktora Tezi, Illinois Institute of Technology, Illinois, 2013.

Barnich O., Droogenbroeck M., ViBe: A Universal Background Subtraction Algorithm for Video Sequences, *IEEE Transactions on Image Processing*, 2011, **20**(6), 1709 - 1724.

Baumann F., Action Recognition with HOG-OF Features, *German Conference on Pattern Recognition*, Berlin, Almanya, 13-14 Eylül 2013.

Bi Y., Zhang Y., Chen Y., Image Classification Method Based on Cellular Automata Transforms, *The Sixth World Congress on Intelligent Control and Automation*, Dalian, Çin, 21-23 Haziran 2006.

Blank M., Gorelick L., Shechtman E., Irani M., Basri R., Actions as Space-Time Shapes, *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, Beijing, Çin, 17 Ekim 2005.

Bozkurt M. R., Yurtay N., Yılmaz Z., Sertkaya C., Comparison of Different Methods for Determining Diabetes, *Turkish Journal of Electrical Engineering & Computer Sciences*, 2014, **22**(4), 1044-1055.

Brejl M., Sonka M., Object Localization and Border Detection Criteria Design in Edge-Based Image Segmentation: Automated Learning from Examples, *IEEE Transactions on Medical Imaging*, 2000, **19**(10), 973-985.

Cerit B., Bölük S., Demirci M., Analysis of The Effect of Image Resolution on Automatic Face Gender and Age Classification, *Signal Processing and Communication Application Conference (SIU2016)*, Zonguldak, Türkiye, 16-19 Mayıs 2016.

Chandran S., Navina S. N., Survey on Various Image Denoising Techniques Using Cellular Automata, *International Journal of Advanced Research in Electronics and Communication Engineering*, 2012, **1**(6), 11-14.

Chaudhuri P., Chowdhury D., Nandi S., Chattopadhyay S., Additive Cellular Automata, 1st ed., *IEEE Computer Society Press*, Los Alamitos, 1997.

Chen L., Wei H., Ferryman J., A Survey of Human Motion Analysis Using Depth Imagery, *Pattern Recognition Letters*, 2013, **34**(15), 1995-2006.

Choudhury P. P., Nayak B. K., Sahoo S., Rath, S. P., Theory and Applications of Two-dimensional, Null-boundary, Nine-Neighborhood, Cellular Automata Linear Rules, arXiv, <https://arxiv.org/abs/0804.2346> (Ziyaret tarihi: 10 Ekim 2018).

Chung D., Sapiro G., Segmenting Skin Lesions with Partial-Differential-Equations-Based Image Processing Algorithms, *IEEE Transactions on Medical Imaging*, 2000, **19**(7), 763-767.

Cielen D., Meysman A., Ali M., *Introducing Data Science*, Manning Publications, Shelter Island, 2016.

Codd E., *Cellular Automata*, Academic Press, New York, 1968.

Cortes C., Vapnik, Support-Vector Networks, *Machine Learning*, 1995, **20**(3), 273-297.

Çalıkoğlu D., On the Computational Properties of A Class of Cellular Automata, Doktora Tezi, The University of Iowa, Iowa, 1973.

Dabbaghian V., Spicer, V., Singh, S., Borwein, P., Brantingham, P., The Social Impact in A High-Risk Community: A Cellular Automata Model, *Journal of Computational Science*, 2011, **2**(3), 238-246.

Dalal N., ve Triggs, B., Histograms of Oriented Gradients for Human Detection, *Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, Amerika, 20-26 Haziran 2005.

Das S., Theory and Applications of Nonlinear Cellular Automata In VLSI Design, Doktora Tezi, The Bengal Engineering and Science University, Bengal, 2006.

Das S., Sikdar, B. K., Classification of CA Rules Targeting Synthesis of Reversible Cellular Automata, *International Conference on Cellular Automata*, Perpignan, Fransa, 20-23 Eylül 2006.

Das S., Kundu A., Sikdar B. K., Chaudhuri P. P., Design of Nonlinear CA Based TPG Without Prohibited Pattern Set in Linear Time, *Journal of Electronic Testing*, 2005, **21**(1), 95-107.

Das S., Mukherjee S., Naskar N., Sikdar B., Characterization of Single Cycle CA and its Application in Pattern Classification, *Electronic Notes in Theoretical Computer Science*, 2009, **252**(1), 181-203.

Das S., Sikdar B. K., Chaudhuri P. P., Characterization of Reachable/Nonreachable Cellular Automata States, Editors: P. Soot, B. Chopard, H. A.G., *Cellular Automata ACRI 2004*, 1st ed., Springer, Berlin, Heidelberg, 2004.

Deutsch E. S., Thinning Algorithms on Rectangular, Hexagonal, and Triangular Arrays, *Communications of the ACM*, 1970, **15**(9), 827-837.

Dhanachandra N., Manglem K., Chanu Y., Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm, *Procedia Computer Science*, 2015, **54**(1), 764-771.

Dioşan L., Andreica, Boros I., Voiculescu I., Avenues for the Use of Cellular Automata in Image Segmentation, Editors: Squillero G., Sim K., *Applications of Evolutionary Computation*, Springer, Cham, 2017.

Dollar, P., Tu Z., Belongie S., Supervised Learning of Edges and Object Boundaries, *IEEE Computer Vision and Pattern Recognition (CVPR)*, New York, Amerika, 17-22 Haziran 2006.

Dollar P., Rabaud V., Cottrell G., Belongie S., Behavior Recognition Via Sparse Spatiotemporal Feature, *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, Çin, 15-16 Ekim 2005.

Dong G., Xie M., Color Clustering and Learning For Image Segmentation Based on Neural Networks, *IEEE Transactions on Neural Networks*, 2005, **16**(4), 925-936.

Duch W., Grudziński K., Ensembles of Similarity-Based Models, Editors: Kłopotek M., Michalewicz M., Wierzchoń S., *Advances in Intelligent and Soft Computing: Physica*, Heidelberg, 2001.

Duff M., Watson D., Fountain, T., Shaw, G., A Cellular Logic Array for Image Processing, *Pattern Recognition*, 1973, **5**(3), 241-247.

Egger J., Kapur T., Fedorov A., Pieper S., Miller J. V., Veeraraghavan H., . . . Kikinis R., GBM Volumetry using the 3D Slicer Medical Image Computing Platform, *Nature:Scientific Reports*, 2013, **3**(1), 1364.

Fanti C., Towards Automatic Discovery of Human Movemes, Doktora Tezi, California Institute of Technology, California, 2008.

Gaidon A., Harchaoui Z., Schmid C., Actom Sequence Models for Efficient Action Detection, *Computer Vision and Pattern Recognition (CVPR)*, Providence, Amerika, 20-25 Haziran 2011.

Ganguly N., Cellular Automata Evolution : Theory and Applications in Pattern Recognition and Classification. Doktora Tezi, Bengal Engineering College (A Deemed University), Bengal, 2003.

Ganguly N., Maji P., Sikdar B. K., Chaudhuri P. P., Generalized Multiple Attractor Cellular Automata (Gmaca) Model For Associative Memory, *International Journal of Pattern Recognition and Artificial Intelligence*, 2002, **16**(7), 781-793.

Ganguly N., Sikdar B. K., Deutsch A., Canright G., Chaudhuri P. P., A Survey on Cellular Automata, CiteSeer, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.7729> (Ziyaret tarihi: 20 Kasım 2018).

Gao Y., Yang J., The Application of Cellular Automaton in Medical Semiautomatic Segmentation, Editors: Rosin P., Adamatzky A., Sun X., *Cellular Automata in Image Processing and Geometry*, 1st ed., Springer, Cham, 163-182, 2014.

Gardner M., Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game "Life", *Scientific American*, 1970, **223**(4), 120-123.

Gardner M., Cellular Automata, Self-Reproduction, Garden of Eden and Game Life, *Scientific American*, 1971, **224**(2), 112-122.

Ghosh P. K., Antani S., Long L. R., Thoma G. R., Unsupervised Grow-Cut: Cellular Automata-based Medical Image Segmentation, *Healthcare Informatics, Imaging and Systems Biology (HISB)*, San Jose, Amerika, 26-29 Haziran 2011.

González J., Varona J., Roca F., Villanueva J., aSpaces: Action Spaces for Recognition and Synthesis of Human Actions, Editors: Perales F., Hancock E., *Articulated Motion and Deformable Objects*, Springer, Berlin, 189-200, 2002.

Goodrich M., Tamassia R., *Algorithm Design and Applications*, 1st ed., John Wiley & Sons, 2015.

Green D., Cellular Automata Models in Biology, *Mathematical and Computer Modelling*, 1990, **13**(6), 69-74.

Gross R., Shi J., The CMU Motion of Body (MoBo) Database, CiteSeer, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.5015> (Ziyaret tarihi: 10 Ekim 2017).

Guo G., Lai A., A Survey on Still Image Based Human Action Recognition, *Pattern Recognition*, 2014, **47**(10), 3343-3361.

Ikenaga T., Ogura T., Real-Time Morphology Processing Using Highly Parallel 2-D Cellular Automata CAM/sup 2/, *IEEE Transactions on Image Processing*, 2000, **9**(12), 2018-2026.

İkizler N., Duygulu P., Histogram of Oriented Rectangles: a New Pose Descriptor for Human Action Recognition, *Image and Vision Computing*, 2009, **27**(10), 1515-1526.

James G., Witten D., Hastie T., Tibshirani R., *An Introduction to Statistical Learning with Applications in R*, Springer, New York, 2013.

Ji X., Liu H., Advances in View-Invariant Human Motion Analysis: A Review, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2010, **40**(1), 13 - 24.

Johansson G., Visual Perception of Biological Motion and A Model for Its Analysis, *Perception & Psychophysics*, 1973, **14**(2), 201-211.

Julian D., *Designing Machine Learning Systems with Python*, Packt Publishing, Birmingham, 2016.

Kanan C., Cottrell, G., Color-to-Grayscale: Does the Method Matter in Image Recognition?, *PLoS ONE*, 2012, **7**(1), e29740-29750.

Kauffmann C., Piché N., Seeded ND Medical Image Segmentation by Cellular Automaton on GPU, *International Journal of Computer Assisted Radiology and Surgery*, 2010, **5**(3), 251-262.

Ke S.-R., Thuc H. L., Lee Y.-J., Hwang J.-N., Yoo J.-H., Choi K.-H., A Review on Video-Based Human Activity Recognition, *Computers*, 2013, **2**(2), 88-131.

Krüger, V., Kragic D., Ude A., Geib C., The Meaning of Action: A Review on Action Recognition and Mapping, *Advanced Robotics*, 2007, **21**(13), 1473-1501.

Ku C.-A., Incorporating Spatial Regression Model into Cellular Automata for Simulating Land Use Change, *Applied Geography*, 2016, **69**(1), 1-9.

Kumar S., John M., Human Activity Recognition Using Optical Flow Based Feature Set, *IEEE International Carnahan Conference on Security Technology (ICCST)*, Orlando, Amerika, 24-27 Ekim 2016.

Kumar T., Sahoo G., A Novel Method of Edge Detection using Cellular Automata, *International Journal of Computer Applications*, 2010, **9**(4), 38-44.

Kuyumcu B., *OpenCV Görüntü İşleme ve Yapay Öğrenme*, Level, Istanbul, 2016

Langton C., Self-Reproduction in Cellular Automata, *Physica D: Nonlinear Phenomena*, 1984, **10**(1), 135-144.

Laptev I., Lindeberg T., Space-Time Interest Points, *International Journal of Computer Vision*, 2005, **64**(2), 107-123.

Laptev I., Marszalek M., Schmid C., Rozenfeld B., Learning Realistic Human Actions From Movies, *Computer Vision and Pattern Recognition(CVPR)*, Anchorage, Amerika, 23-28 Haziran 2008.

Li W., Packard N. H., Langton C. G., Transition Phenomena in Cellular Automata Rule Space, *Physica D: Nonlinear Phenomena*, 1990, **45**(1), 77-94.

Lichman M., UCI Machine Learning Repository, California: University of California, Irvine, School of Information and Computer Sciences, <http://archive.ics.uci.edu/ml> (Ziyaret tarihi 10 Eylül 2016).

Maddalena L., Petrosino A., A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications, *IEEE Transactions on Image Processing*, 2008, **17**(7), 1168 - 1177.

Maire, M., Arbelaez P., Fowlkes C., Malik J., Using Contours to Detect and Localize Junctions in Natural Images, *Computer Vision and Pattern Recognition(CVPR)*, Anchorage, Amerika, 23-28 Haziran 2008.

Marsland S., *Machine Learning, An Algorithmic Perspective*, CRC Press, Boca Raton, 2015.

Martin D. R., Fowlkes C. C., Malik J., Learning to Detect Natural Image Boundaries Using Brightness and Texture, *Advances in Neural Information Processing Systems Conference*, Vancouver, Kanada, 2002.

Martin D. R., Fowlke, C. C., Malik J., Learning to detect natural image boundaries using local brightness, color, and texture cues, *IEEE Trans Pattern Anal Mach Intell*, 2004, **26**(5), 530-49.

Martin D., Fowlkes C., Tal D., Malik J., A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, *International Conference Computer Vision(ICCVC)*, Vancouver, Kanada, 7-14 Temmuz 2001.

Martin O., Odlyzko A. M., Wolfram S., Algebraic Properties of Cellular Automata, *Communications in Mathematical Physics*, 1984, **93**(2), 219-258.

Mitchell M., Hraber P. T., Crutchfield J. P., Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations, *Complex Systems*, 1993, **7**(2), 89-130.

Moore E. F., Machine Models of Self-Reproduction, *Proc, Symp. Appl. Math.*, 1962, **14**(1), 17-33.

Moore E. F., Machine Models of Self Reproduction, Editor: Burks A. W., *Essays on Cellular Automata*, University of Illinois Press, Urbana, 1970.

Mueller A., Guido S., *Introduction to Machine Learning with Python*, O'Reilly Media Inc, Sebastopol, 2016.

Mukherjee S., Theory and Applications of Nonlinear Cellular Automata In Pattern Recognition, Yüksek Lisans Tezi, Bengal Engineering and Science University, Shibpur, 2009.

Murty M., Devi V., *Introduction to Pattern Recognition and Machine Learning*, World Scientific Publishing, Singapore, 2015.

Myhill J., The Converse of Moore's Garden-of-Eden Theorem, *Proceedings of The American Mathematical Society*, 1963, **14**(4), 685-686.

Nabiyev V., *Algoritmalar Teoriden Uygulamalara*, Seçkin Yayıncılık, Ankara, 2013

Nagel H., From Image Sequences Towards Conceptual Descriptions, *Image and Vision Computing*, 1988, **6**(2), 59-74.

Naiditch D., *A New Kind of Science-Or A Not-So-New Kind of Computer Program?*, Skeptic, Altadena, 2003.

Naskar N., Theory and Applications of Nonlinear Cellular Automata in Pattern Classification, Yüksek Lisans Tezi, The Bengal Engineering and Science University, Shibpur, 2009.

Nayak D. R., Patra P. K., Mahapatra A., A Survey on Two Dimensional Cellular Automata and Its Application in Image Processing, arXiv, <https://arxiv.org/abs/1407.7626>(Ziyaret tarihi: 24 Kasım 2018).

Neumann J., *The Theory of Self-Reproducing Automata*, Editor: Burks, A., Univ. of Illinois Press, Urbana ve London, 1966.

Niebles J. C., Fei-Fei, A Hierarchical Model of Shape and Appearance for Human Action Classification, *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Amerika, 17-22 Haziran 2007.

Nilsson N., *Introduction To Machine Learning*, Stanford University, Stanford, 1998.

O'Hara S., Draper B. A., Scalable Action Recognition with A Subspace Forest, *Computer Vision and Pattern Recognition (CVPR)*, Providence, Amerika, 16-21 Haziran 2012.

Osborne I., Applied Physics: Latching onto Cellular Automata, *Science*, 2001, **291**(5512), 2277e-2279.

Patel D., More S., Edge Detection Technique By Fuzzy Logic and Cellular Learning Automata Using Fuzzy Image Processing, *International Conference Computer Communication and Informatics (ICCCI)*, Coimbatore, Hindistan, 4-6 Ocak 2013.

Peldek S., Hücresel Hareketli Tekniği İle Yüz Tanıyış, Yüksek Lisans Tez, Karabük Üniversitesi, Fen Bilimleri Enstitüsü, Karabük, 2012, 321486.

Peldek S., Becerikli Y., Çoklu Cezbedici Hücresel Otomatların Boyut Azaltmada Kullanımı, *International Conference on Computer Science and Engineering (UBMK)*, Antalya, Türkiye, 5-8 Ekim 2017.

Peldek, S., & Becerikli, Y., GMACA ile Hareket Tespiti Yapılan Video Görüntülerde İnsan Hareketlerinin Tanınması, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, DOI:10.17341/gazimmfd.460500.

Peldek S., Becerikli Y., Hücresel Otomata Tabanlı İkili Sınıflandırma, *Bilişim Teknolojileri Dergisi*, 2019, **12**(1), 43-57.

Popovici A., Popovici, D., Cellular Automata in Image Processing, *Fifteenth International Symposium on Mathematical Theory of Networks and System*, Indiana, Amerika, 12-16 Ağustos 2002.

Preston K., Duff M., Levialdi S., Basics of Cellular Logic with Some Applications in Medical Image Processing, *Proceedings of the IEEE*, 1979, **67**(5), 826 - 856.

Raschka S., *Python Machine Learning*, Packt Publishing, Birmingham, 2015.

Ren X., Multi-scale Improves Boundary Detection in Natural Images, Editors: Forsyth D., Torr P., Zisserman A., *Computer Vision – ECCV 2008*, Springer, Heidelberg, 2008.

Ren X., Bo L., Discriminatively Trained Sparse Code Gradients for Contour Detection, *Advances in Neural Information Processing Systems 25*, Nevada, Amerika, 3-8 Eylül 2012.

Richert W., Coelho L. P., *Building Machine Learning Systems with Python*, Packt Publishing, Birmingham, 2013.

Rosin P., Training Cellular Automata for Image Processing, *IEEE Transactions on Image Processing*, 2006, **15**(7) 2076 - 2087.

Rosin P., Image Processing Using 3-State Cellular Automata, *Computer Vision And Image Understanding*, 2010, **114**(7), 790-802.

Rosin P., Adamatzky A., Sun X., *Cellular Automata in Image Processing and Geometry*, Springer, London ,2014.

Safia D., Oussama D., Chawki B. M., Image Segmentation Using Continuous Cellular Automata, *10th International Symposium on Programming and Systems*, Algiers, Cezayir, 25-27 Nisan 2011.

Saha P., Udupa J., Odhner D., Scale-Based Fuzzy Connected Image Segmentation: Theory, Algorithms, and Validation, *Computer Vision and Image Understanding*, 2000, **77**(22), 145-174.

Samsudin W., Ghazali K., *Motion Detection Using Horn-Schunck Optical Flow*, *Computer Applications for Communication, Networking, and Digital Contents*, 1st ed., Springer, Berlin Heidelberg, 2012.

Sanko M., Hlavac V., Boyle R., *Image Processing, Analysis and Machine Vision*, 3rd ed., Thomson Learning, Toronto, 2008.

Sarkar P., A Brief History of Cellular Automata, *ACM Computing Surveys (CSUR)*, 2000, **32**(1), 80-107.

Schindler K., Gool L.V., Action Snippets: How Many Frames Does Human Action Recognition Require?, *IEEE Computer Vision and Pattern Recognition(CVPR)*, Anchorage, Amerika, 23-28 Haziran 2008.

Schuldt C., Laptev I., Caputo B., Recognizing Human Actions: A Local SVM Approach, *ICPR'04*, Cambridge, İngiltere, 23-26 Ağustos 2004.

Sedgewick R., Wayne K., *Algorithms*, Addison-Wesley Professional, Boston, 2011.

Segaran T., *Programming Collective Intelligence Building Smart Web 2.0 Application*, O'Reilly Media, Sebastopol, 2007.

Sheikh Y., Shah M., Bayesian Modeling of Dynamic Scenes for Object Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, **27**(11), 1778 - 1792.

Sikdar B., Ganguly N., Majumder P., Chauhuri P., Design of Multiple Attractor GF(2/sup p/) Cellular Automata for Diagnosis of VLSI Circuits, *Fourteenth International Conference on VLSI Design*, Bangalore, Hindistan, 7 Ocak 2001.

Smith A., Two-Dimensional Formal Languages and Pattern Recognition by Cellular Automata, *12th Annual Symposium on Switching and Automata Theory (SWAT)*, East Lansing, Amerika, 13-15 Ekim 1971.

Sree P. K., Babu I. R., Nedunuri S. U., Multiple Attractor Cellular Automata (MACA) for Addressing Major Problems in Bioinformatics, *Review of Bioinformatics and Biometrics (RBB)*, 2013, **2**(3), 70-77.

Stanbrough J. L., Physics of BHS, <http://www.batesville.k12.in.us/physics/Discrete/ca/totalistic.html> (Ziyaret tarihi: 3 Ekim 2017).

Stauffer C., Grimson W., Adaptive Background Mixture Models for Real-Time Tracking, *IEEE Computer Vision and Pattern Recognition*, Fort Collins, Amerika, 23-25 Haziran 1999.

Suk H.-I., Sin B.-K., Lee S.-W., Hand Gesture Recognition Based on Dynamic Bayesian Network Framework, *Pattern Recognition*, 2010, **43**(9), 3059-3072.

Triendl E. E., Skeletonization of Noisy Handdrawn Symbols Using Parallel Operations, *Pattern Recognition*, 1970, **2**(3), 215-226.

Turing A. M., On Computable Numbers, with an Application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, 1937, **2**(1), 230-265.

URL-1: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/gray/gm_2_4/main.html, (Ziyaret tarihi: 29 Ekim 2017).

URL-2: <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/algorithms.html>, (Ziyaret tarihi: 29 Ekim 2017).

URL-3: http://www.conwaylife.com/wiki/Main_Page, (Ziyaret tarihi: 11 Haziran 2017).

URL-4: https://en.wikipedia.org/wiki/Hamming_distance, (Ziyaret tarihi: 29 Aralık 2016).

URL-5: <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/algorithms.html>, (Ziyaret tarihi: 11 Haziran 2017).

Vezhnevets V., Konouchine V., “GrowCut” - Interactive Multi-Label N-D Image Segmentation By Cellular Automata. *Proc. of Graphicon*, 2005, **1**(4), 150-156.

Vilarino D., Cabello D., Pardo X., Brea V., Cellular Neural Networks and Active Contours: A Tool for Image Segmentation, *Image And Vision Computing*, 2003, **21**(2), 189-204.

Wang L., Suter D., Recognizing Human Activities from Silhouettes: Motion Subspace and Factorial Discriminative Graphical Model, *IEEE Computer Vision and Pattern Recognition*, Minneapolis, Amerika, 17-22 Haziran 2007.

Wang X.-Y., Wu Z.-F., Chen L., Zheng H.-L., Yang H.-Y., Pixel Classification Based Color Image Segmentation Using Quaternion Exponent Moments, *Neural Networks*, 2016, **74**(1), 1-13.

Waseem K., Image Segmentation Techniques: A Survey, *Journal of Image and Graphics*, 2013, **1**(4), 166-170.

Weinland D., Ronfard R., Boyer E., A Survey of Vision-Based Methods for Action Representation, Segmentation and Recognition, *Computer Vision and Image Understanding*, 2011, **115**(2), 224-241.

Weiss M., *Data Structures and Algorithm Analysis in Java*, Pearson, New Jersey, 2012.

Wilson A. D., Bobick A. F., Parametric Hidden Markov Models for Gesture Recognition, *Transactions on Pattern Analysis and Machine Intelligence*, 1999, **21**(9), 884-900.

Wolfram S., Statistical Mechanics of Cellular Automata, *Reviews Of Modern Physics*, 1983, **55**(3), 601-644.

Wolfram S., Computation Theory of Cellular Automata, *Communications in Mathematical Physics*, 1984, **96**(1), 15-57.

Wolfram S., Universality and Complexity in Cellular Automata, *Physica D: Nonlinear Phenomena*, 1984, **10**(1), 1-35.

Wolfram S., Undecidability and Intractability in Theoretical Physics, *Phys. Rev. Lett.*, 1985, **54**(8), 735-738.

Wolfram S., Random Sequence Generation by Cellular Automata, *Advances in Applied Mathematics*, 1986, **7**(2), 123-169.

Wolfram S., *A New Kind of Science*, Wolfram Media, Champaign, 2002.

Wongthanavas S., Tangvoraphonkchai V., Cellular Automata-Based Algorithm and its Application in Medical Image Processing, *IEEE International Conference Image Processing*, San Antonio, Amerika, 16-19 Eylül 2007.

Yu S. X., Segmentation Induced by Scale Invariance, *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, Amerika, 20-25 Haziran 2005.

Yu Z., Chen Y., A Real-Time Motion Detection Algorithm for Traffic Monitoring Systems Based on Consecutive Temporal Difference, *Asian Control Conference(ASCC)*, Hong Kong, Çin, 27-29 Ağustos 2009.

Zaitoun N., Aqelb, M., Survey on Image Segmentation Techniques, *Procedia Computer Science*, 2015, **65**(1), 797-806.

Zhang H., Fritts J., Goldman S., Image Segmentation Evaluation: A Survey of Unsupervised Methods, *Computer Vision and Image Understanding*, 2008, **110**(2), 260-280.

Zhao Y., Guo H., Billings S., Application of Totalistic CellularAutomata for Noise Filtering in Image Processing, *Journal of Cellular Automata*, 2012, **7**(1), 207-221.

Zhou H., Yuan Y., Shi C., Object Tracking Using SIFT Features and Mean Shift, *Computer Vision and Image Understanding*, 2009, **113**(3), 345-352.

Ziaefard M. T., Bergevin R., Semantic Human Activity Recognition: A Literature Review, *Pattern Recognition*, 2015, **48**(8), 2329-2345.

KİŞİSEL YAYINLAR VE ESERLER

Bardak T., Aydemir D., Tankut N., Sözen E., Tankut A. N., **Peldek S.**, The Effect of Mixing Method on Bonding Performance of PVAc Nanocomposites, *COSTAction FP1004 Prague ESR Conference - Experimental Research*, Prague, Çekya, 21-23 Mayıs 2014.

Gültepe Y., **Peldek S.**, Semantic Web and Cloud Computing for Higher Education, *International Conference on Theoretical and Applied Computer Science and Engineering*, Ankara, Türkiye, 10-11 Kasım 2017

Gültepe Y., **Peldek S.**, *Ontoloji Tabanlı Makale Tavsiye Edici Sistem*, Editors: Özturan M., Kutlu B., *Yönetim Bilişim Sistemleri Kongresi 2014 Seçilmiş Bildiriler Serisi*, 1st ed., Boğaziçi Üniversitesi Matbaası, İstanbul, 2016

Gültepe Y., **Peldek S.**, *Ontoloji Tabanlı Makale Tavsiye Edici Sistem*, *Yönetim Bilişim Sistemleri Kongresi*, İstanbul, Türkiye, 16-17 Ekim 2014

Karacı A., **Peldek S.**, Erdemir M., Implementation and Evaluation of Robot Hand Control with Wearable Gloves, *8th International Advanced Technologies Symposium*, Elazığ, Türkiye, 19-22 Ekim 2017

Peldek S., Cellular Automata Approaches for Image Processing, *International Conference On Applied Analysis And Algebra*, İstanbul, Türkiye, 20-24 Haziran 2012.

Peldek S., Becerikli Y., GMACA ile Hareket Tespiti Yapılan Video Görüntülerde İnsan Hareketlerinin Tanınması, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, DOI:10.17341/gazimmfd.460500

Peldek S., Becerikli Y., Hücresel Otomata Tabanlı İkili Sınıflandırma, *Bilişim Teknolojileri Dergisi*, 2019, **12**(1), 43-57.

Peldek S., Becerikli Y., Çoklu Cezbedici Hücresel Otomatların Boyut Azaltmada Kullanımı, *International Conference on Computer Science and Engineering (UBMK)*, Antalya, Türkiye, 5-8 Ekim 2017

Peldek S., Becerikli Y., Çoklu Cezbedici Hücresel Otomatların Görüntülerde Örüntü Üretici Olarak Kullanımı, *8. Savunma Teknolojileri Kongresi*, Ankara, Türkiye, 12-14 Ekim 2016.

Peldek S., Çınar A., Union/Find Based Image Segmentation and Performance Analysis, *International Conference on Engineering and Natural Science*, Saray Bosna, Bosna Hersek, 24-28 Mayıs 2016.

Peldek S., Durdu P. O., TÜBİTAK Web Sitesinin Kullanılabilirliğinin Farklı Yöntemler İle Değerlendirilmesi, *International Conference on Education in Mathematics, Science and Technology*, Konya, Türkiye, 16-18 Mayıs 2014.

Peldek S., Gültepe Y., Akademik Spinoff Firma Değerlendirilmesi: Ar-Ge Faaliyetlerinin Ticarileştirilmesi, *Bartın Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 2017, **8**(15), 64-81.

Peldek S., Gültepe Y., Akademik Girişimcilikte Değer Üretimi Spinoff Firma Değerlendirmesi, *İnovasyon Ekosistemleri Aracılığı İle Değer Yaratmak: Disiplinler Arası Yaklaşımlar Kongresi*, İzmir, Türkiye, 26-27 Kasım 2015.

Peldek S., Karacı A., Application and Performance Analysis of Maze Construction based on Union/Find Data Structure, *8th International Advanced Technologies Symposium*, Elazığ, Türkiye, 19-22 Ekim 2017



ÖZGEÇMİŞ

1981 yılında Erzincan’da doğdu. İlk, orta ve lise öğrenimini Erzincan’da tamamladı. Lisans eğitimini Doğu Akdeniz Üniversitesi Bilgisayar Mühendisliği bölümünde 2007 yılında ve yüksek lisans eğitimini Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği ana bilim dalında 2012 yılında tamamladı. 2009-2014 yılları arasında Bartın Üniversitesi Yönetim Bilişim Sistemleri bölümünde Araştırma Görevlisi olarak görev yaptı. 2014 yılından beri Kastamonu Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak görev yapmaktadır.

