

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
(YÜKSEK LİSANS TEZİ)

OTOMATİK ARAÇ PLAKA TANIMA SİSTEMİ

İsmail IRMAKÇI

Elektrik Elektronik Mühendisliği Ana Bilim Dalı

Bilim Dalı Kodu: 609.01.00

Sunuş Tarihi:31 Mart 2008

Tez Danışmanı: Doç. Dr. Aydoğan SAVRAN

Bornova-İZMİR

III

İsmail IRMAKÇI tarafından yüksek lisans tezi olarak sunulan “Otomatik Araç Plaka Tanıma Sistemi” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 31.03.2008 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

-

Jüri Üyeleri:

İmza

Jüri Başkanı :

Raportör Üye:

Üye :

ÖZET

OTOMATİK ARAÇ PLAKA TANIMA SİSTEMİ

IRMAKÇI, ismail

Yüksek Lisans Tezi, Elektrik Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç.Dr.Aydoğan Savran

31 Mart 2008, 88 sayfa

Plaka tanıma sistemleri otoyollarda hız tespiti, kırmızı ışık ihlali tespiti gibi uygulamalarda önemli rol oynamaktadır. Çoğu sistem, sabit ışık ortamı, sabit araç hızı gibi belirli sınırlamalar altında çalışmaktadır; Bu çalışmada bir veya iki satır olan TC sivil araç plakalarından tek satır olanları için bir sistem tasarlanmıştır. Tasarlanan sistemde öncelikle imgeye önışlem uygulanmakta, imgeden plaka çıkarımı için morfolojik işlemler uygulanarak köşeler belirlenmekte ve belirlenen köşelerin kapladığı en küçük dörtgen bölge imgeden çıkarılmaktadır. Çıkarılan bölgeye tekrar önışlemler uygulandıktan sonra bölge karakter ayrıştırma sisteminde yatay ve düşey taramaya sokulmaktadır. Son olarak da karakterlerin tanınması aşamasında 23 harf ve 10 rakamın herbiri için geriye yayılma algoritması ile eğitilmiş çok katmanlı yapay sinir ağı (YSA) kullanılarak plakanın ASCII kodu tespit edilmektedir.

Anahtar kelimeler : morfolojik işlemler, otomatik araç plaka tanıma, karakter ayrıştırma, çok katmanlı yapay sinir ağı

VII

ABSTRACT

AUTOMATIC CAR LICENSE PLATE RECOGNITION SYSTEM

IRMAKÇI, ismail

Master Thesis, Electric Electronics Engineering Department

Supervisor: Assoc. Prof. Dr. Aydoğan Savran

31 March 2008, 88 pages

Plate recognition systems play important roles in applications such as determination of traffic speed and red light violation etc. Most of the systems run under restricted conditions; fixed light condition, constant speed of vehicle etc. TR civil plates would be one or two rows. In this work, the system is designed for one row plates. Firstly, the image is preprocessed, then vertices on the image are determined applying morphologic operations to the image for plate extraction and smallest quadrangle region of the determined vertices which cover region is extracted from the image. After preprocessing the extracted image again, the vertical and horizontal scans are performed in the character decomposition system. Finally, in the characters recognition stage, ASCII code of the plate is determined by using multi-layer artificial neural networks(ANN) which are trained by backpropagation algorithm separately for 23 letters and 10 digits.

Keywords : morfological processes, automatic license plate recognition, character decomposition, multilayer artificial neural networks

TEŞEKKÜR

Bu çalışma süresince değerli görüşlerini ve desteğini esirgemeyen tez danışmanım sayın Doç. Dr. Aydoğan SAVRAN'a teşekkürü bir borç bilirim.

Yine çalışmam sürece desteklerini esirgemeyen sayın Yrd. Doç. Dr. Aybars UĞUR' a minnetimi arz ederim.

Hayatım boyunca maddi ve manevi desteklerini esirgemeyen aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

ÖZET	V
ABSTRACT	VII
TEŞEKKÜR	IX
İÇİNDEKİLER	XI
ŞEKİLLER DİZİNİ	XIII
ÇİZELGELER DİZİNİ	XIX
1. GİRİŞ	1
2. GENEL BİLGİLER	5
2.1 Sayısal İmge	5
2.2 Sayısal İmgede Komşuluk İşlemleri	7
2.3 KYM(RGB) İmge	9
2.4 Gri Seviye	10
2.5 İkili Seviye	11
2.5.1 Normal eşikleme	11
2.5.2 Otsu eşikleme	12
2.6 Sayısal İmgede Yeniden Boyutlandırma	13
2.6.1 En yakın komşuluk	14
2.6.2 Bilineer aradeğerleme yöntemi	15
2.6.3 Bikübik aradeğerleme yöntemi	16
2.6.4 Süper örnekleme	17
2.7 Gürültüleri İmgeden Kaldırma	19
2.7.1 Ortalama(medyan) filtre	19
2.7.2 İmge piramitleri	20
2.8 Kenar Belirleme Yöntemleri	23

İÇİNDEKİLER (devam)

2.9	Morfolojik İşlemler Ve Şekil Analizi	27
2.9.1	4 komşuluğu ve 8 komşuluğu	27
2.9.2	Bağlı bileşenlik	28
2.9.3	Genişletme	29
2.9.4	Şekil analizi.....	31
2.10	Yapay Sinir Ağları	35
2.10.1	Nöron modeli	38
2.10.2	En küçük ortalama kareler yöntemi	43
2.10.3	Tek katmanlı algılayıcı	45
2.10.4	Çok katmanlı algılayıcı.....	46
2.10.5	Geri yayılım algoritması	48
3.	MATERYAL VE YÖNTEM.....	54
3.1	Plaka Ayırıştırma Sistemi	54
3.1.1	Ön işlem	54
3.1.2	Kenarların bulunması ve eşikleme.....	56
3.1.3	Köşelerin bulunması ve biçimlendirilmesi	59
3.2	Karakter Ayırıştırma Sistemi	64
3.2.1	Ön işlem	64
3.2.2	Yatay ve düşey tarama	68
3.3	Optik Karakter Tanıma Sistemi	72
4.	TARTIŞMA VE SONUÇ	75
	KAYNAKLAR	78
	EK 1 AKIŞ DİAGRAMI	83
	ÖZGEÇMİŞ	87

XIII

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 2.1	Sayısal imge piksellerinin ızgara şeklinde gösterilmesi ... 5
Şekil 2.2	Elektromanyetik alanının görülür alanı içeren dalga boyları 6
Şekil 2.3	3x3 lük maske ve kaynak imgede üzerinde yerleşimi 8
Şekil 2.4	KYM renk modeli 9
Şekil 2.5	KYM imge 10
Şekil 2.6	Normal eşikleme 11
Şekil 2.7	1 Boyutlu ara değerlendirme fonksiyonları 14
Şekil 2.8	En yakın komşuluk yöntemiyle yeniden boyutlandırma. 15
Şekil 2.9	Bilineer yöntemiyle yeniden boyutlandırma 15

XIV

ŞEKİLLER DİZİNİ (devam)

Şekil 2.10	Bikübik yöntemiyle yeniden boyutlandırma	17
Şekil 2.11	Süper örnekleme ağırlıkları	18
Şekil 2.12	İmge piramidi.....	21
Şekil 2.13	İmge piramidini oluşturan sistemin blok diagramı	21
Şekil 2.14	Belirli bir standart sapmaya sahip düşük geçirgen gaussian filtresi	22
Şekil 2.15	Kenar belirleme maskeleri	25
Şekil 2.16	(x, y) noktasındaki komşuluklar	28
Şekil 2.17	Genişletme ve yapısal eleman	30
Şekil 2.18	Yapısal elemanın ikili imgedeki ötelenmesi	30

ŞEKİLLER DİZİNİ (devam)

Şekil 2.19	Genişletme çıktı imgesi.....	31
Şekil 2.20	Zincir kodlarının sayı şeması	32
Şekil 2.21	Zincir kod karşılıkları.....	33
Şekil 2.22	ϵ değerine göre en uzak köşe.....	34
Şekil 2.23	Başarılı çok kenarlı yaklaştırma.....	35
Şekil 2.24	Biyolojik nöronlar	38
Şekil 2.25	Nöronun lineer olmayan modeli	39
Şekil 2.26	Doğrusal olmayan nöronun diğer modeli.....	40
Şekil 2.27	Eşikleme fonksiyonu.....	41
Şekil 2.28	Parçalı doğrusal fonksiyonu.....	41

ŞEKİLLER DİZİNİ (devam)

Şekil 2.29	Sigmoid fonksiyonu.....	42
Şekil 2.30	Tek katlı algılayıcının sinyal akış grafiği	46
Şekil 2.31	İki gizli katmanlı çok katmanlı algılayıcı	47
Şekil 2.32	Fonksiyon ve hata sinyalleri	48
Şekil 2.33	Geri yayılım algoritmasının sinyal akış grafiği	50
Şekil 3.1	İmge piramit aşağı örnekleme	55
Şekil 3.2	İmge piramit yukarı örnekleme	55
Şekil 3.3	İmgenin kırmızı kanalı.....	57
Şekil 3.4	Canny metodu çıktısı	57
Şekil 3.5	Genişletilmiş ikili seviye imge	58

XVII

ŞEKİLLER DİZİNİ (devam)

Şekil 3.6	Artan değerlerde eşikleme çıktıları	59
Şekil 3.7	Gri seviye(Kırmızı kanal)	61
Şekil 3.8	Çok kenarlı basitleştirmeden önceki iç ve dış kontörler .	62
Şekil 3.9	Çok kenarlı basitleştirmeden sonraki iç ve dış kontörler	62
Şekil 3.10	Kanalların VEYA'lanması	65
Şekil 3.11	Girdi imgesi.....	66
Şekil 3.12	Kanallar ve eşikleme çıktıları	66
Şekil 3.13	VEYAlama ve medyan çıktısı.....	67
Şekil 3.14	Yatay tarama	68
Şekil 3.15	Yatay tarama da ortalama değer.....	69

XVIII

ŞEKİLLER DİZİNİ (devam)

Şekil 3.16	Yatay tarama sonucu.....	69
Şekil 3.17	Düşey tarama	71
Şekil 3.18	Karakter ayırıştırma sonucu.....	71
Şekil 3.19	35-20-23 topolojiye sahip harf YSA.....	73
Şekil 3.20	35-20-10 topolojiye sahip rakam YSA	73

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
Çizelge 1.1 TC sivil plaka karakter dizilişleri.....	2
Çizelge 4.1 Deneysel sonuçlar	76
Çizelge 4.2 Tasarlanan sistemle literatürdeki benzer sistemlerin karşılaştırılması	77

1. GİRİŞ

Otomatik araç plaka tanıma sistemleri son yıllarda değişik uygulamalarda yer almıştır. Bu uygulamalarda çevrenin getirdiği zorluklar otomatik araç plaka tanıma sistemlerinde plaka yerinin belirlenmesini ve plakanın tanınmasını güçleştirmektedir. Örneğin çok güneşli bir havada plakanın belirlenememesi, karanlık bir ortamda ışığın yetersiz kalması gibi sorunlar otomatik araç plaka sistemlerinde karşılaşılan zorluklardan bazılarıdır. Genel olarak plaka tanıma sistemleri (PTS) üç ana bölümü kapsamaktadır.

- 1) Plaka Ayırıştırma Sistemi(PAS)
- 2) Karakter Ayırıştırma Sistemi (KAS)
- 3) Optik Karakter Tanıma Sistemi(OKTS)

Otomatik araç plaka tanıma sistemlerine ait çalışmalar incelendiğinde plaka ayırıştırma sistemi, karakter ayırıştırma sistemi ve optik karakter tanıma sistemi için değişik yöntemler mevcuttur. Plaka ayırıştırma sistemi için belli boyutlardaki iki pencereyi imgenin üzerinde kaydırarak bu pencerelerin istatistiksel bilgilerine göre plaka bölgesini bulan yöntemler(Anagnostopoulos, Anagnostopoulos, Loumos and Kayafas, 2006), imgenin morfolojik açılması ile imgenin farkı alınarak yatay ve düşey tarama yöntemiyle plaka bölgesini bulan yöntemler (Wu, On, Weng , Kuan and Ng, 2005), imgedeki plaka bölgesinin renk bilgisinden yararlananan yöntemler (Xu, Li and Yu, 2004),kenar belirleme yöntemini olan Sobel ve imgedeki doğruları bulmaya yarayan

Hough dönüşümünü kullanan yöntemler(Yanamura, Goto, Nishiyama, Soga, Nakatani and Saji, 2003), plaka bölgesinin köşe bilgisinden yararlanılan yöntemler (Qin, Shi., Xu and Fu., 2006) ya da genetik algoritmaları kullanan yöntemler(Yohimori, Mitsukura, Fukumi, Akamatsu and Pedrycz, 2004) mevcuttur. Karakter ayrıştırma sistemi için; bulunan plaka bölgesine yatay ve düşey tarama yöntemi (Oral ve Çelik, 2003), optik karakter tanıma sistemi için ise örüntü eşleştirme (Ahmed., Sarfaz, Zidouri and Al-Khatib, 2003) (Ko and Kim, 2003) ve yapay sinir ağları (Abdullah, Khalid, Yusof and Omar, 2006)(Paliyl, Turchenko, Koval , Sachenkol and Markowsky, 2004) yöntemleri mevcuttur. TC(Türkiye Cumhuriyeti) sivil araç plakaları iki ya da bir satır, beyaz zemin üzerine siyah karakterlerden oluşmaktadır. TC sivil araç plaka formatı Çizelge 1.1’de görüldüğü gibi kodlanmıştır(MEB, 2007). Bu formattan * olan plaka dizilişi çok az rastlandığı için bu çalışmanın dışında tutulmuştur. Tasarlanan sistem sadece bir satıra sahip TC sivil araç plakaları için çalışmaktadır.

Çizelge 1.1 TC sivil plaka karakter dizilişleri R:Rakam H:Harf

7 Karakterli	8 Karakterli
RRHRRRR	RRHRRRRR*
RRHHRRR	RRHHRRRR
RRHHHRR	RRHHHRRR

TC sivil araç plaka örneklerinin çoğu standartlara uymasına rağmen bu standartların dışında olan plaka sayısı da azımsanmayacak

derecededir. Bu tür plakaların üzerinde genellikle vida, etiketler, çamur, plakaya işlenmiş TC bayrağı vs. gibi gürültü olarak adlandırabileceğimiz yabancı nesneler vardır. Tasarlanan sistem değişik ışık koşullarında çekilen Türk sivil plaka imgeleri üzerine yoğunlaşmakta algoritma da yapılacak birkaç değişiklik ile diğer plaka çeşitlerine de uyarlanabilmesi amaçlanmıştır.

Yapılan çalışmadaki sistemin genel yapısı PAS, KAS ve OKTS den oluşmaktadır. PAS bölümüne gelen kırmızı-mavi-yeşil (KMY) imgeye öncelikle olabilecek gürültülerden arındırmak için “piramit aşağı ve yukarı örnekleme” yöntemi uygulanır. İmgede ki aday plaka bölgesine ait köşeleri bulmak için öncelikle imgenin her kanalı için Canny kenar bulma yöntemi sonrasında ise (S. Suzuki and K. Abe, 1985) de belirtilen bağlı bileşenler yöntemi uygulanır. Ardından tekrar imgenin her kanalına plaka üzerinde olabilecek ışık etkisini ortadan kaldırmak için artan değerlerde eşikleme yapılmakta ve tekrar bağlı bileşenler yöntemi uygulanmaktadır. Canny ve eşikleme yöntemleri sonrasında bulunan kontörler Douglas-Peucker algoritması (Hershberger and Snoeyink , 1992) ile basitleştirilmektedir. Bu sayede köşelere sahip olan dörtgenler elde edilmektedir. Bulunan köşelerin bir kare ya da dikdörtgene ait olup olmadığı sınılandıktan sonra bulunan aday plaka bölgeleri KAS bölümüne gönderilir. Çıkarılan bölgelerin her biri için artan değerlerde eşiklemeler yapılarak tekrar bir ön işleme uygulanmakta ve bölgeler karakter ayrıştırma sisteminde yatay ve dikey taramaya sokulmaktadır. Plaka da karakter sırasına göre 23 harf ve 10 rakam için ayrı ayrı geri yayılım algoritması ile eğitilmiş çok katmanlı yapay sinir

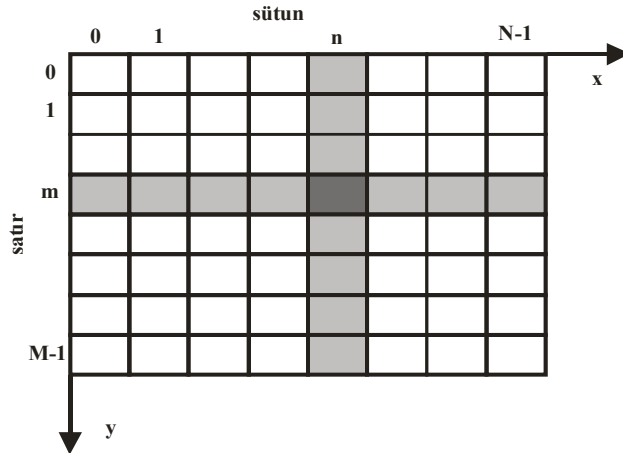
ağları (YSA) sayesinde her bölgenin ASCII kodu tespit edilip bu kodlardan en iyisi seçilerek plaka oluşturulmaktadır.

2. GENEL BİLGİLER

Bu bölümde, tasarlanan sistemin alt yapısını oluşturmak için gerekli temel bilgiler sunulacaktır.

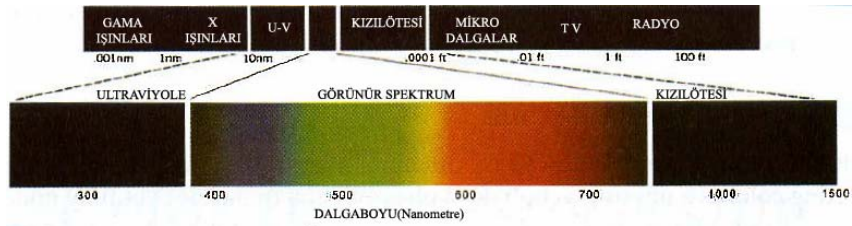
2.1 Sayısal İmge

Bir imge x ve y nin düzlemsel koordinatlar ve bu (x, y) noktasında $f(x, y)$ 'nin genliğinin grilik seviyesi ya da yoğunluk olduğu, (x, y) 'nin iki boyutlu bir fonksiyonudur. $f(x, y)$ 'nin genlik değeri ve x, y noktalarındaki değerler ayrık bir değere ve sonlu bir değere karşılık geliyorsa bu imge bir sayısal imgedir. Sayısal imge sonlu sayıda elemanlardan oluşur bunların her birinin koordinatlarda bir yerleri ve değerleri vardır ve bu elemanlara piksel adı verilir. Pikseller sayısal resmi temsil etmek için ızgara şeklinde dizilirler (Şekil 2.1). Şekil 2.1 de $M \times N$ boyutlu bir sayısal imge için ilk indeks m satırın pozisyonunu, ikinci indeks n sütunun pozisyonunu gösterir. Bu iki noktanın kesiştiği koordinatın değeri ise gri seviye olarak adlandırılır.



Şekil 2.1 Sayısal imge piksellerinin ızgara şeklinde gösterilmesi(Jähne, 2002)

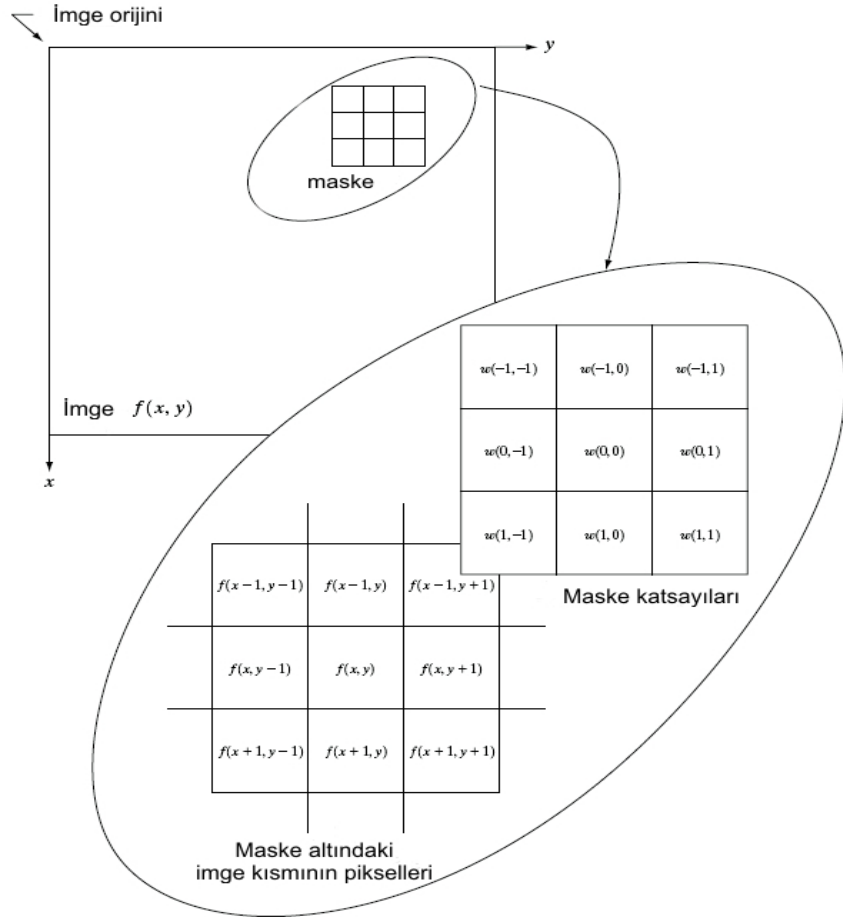
İmgeler renklerden oluşur. Temel olarak insanların algıladığı nesne renkleri nesnelerden yansıyan ışığın doğasından kaynaklanır. Şekil 2.2’den görüldüğü gibi insanların renkleri görebildiği spektrum alanı oldukça düşüktür (400 – 700 nanometre arası). Örneğin insanın yeşil bir nesneyi görebilmesi için bu nesneden yansıyan ışığın dalga boyu yaklaşık olarak şekil den görüldüğü gibi 535nm ye yakın bir değere sahip olması gerekir.



Şekil 2.2 Elektromanyetik alanının görülr alanı içeren dalga boyları (Gonzalez and Woods, 2002)

2.2 Sayısal İmgede Komşuluk İşlemleri

Komşuluk işlemleri, sayısal imge işlemede imgenin özelliklerini değiştirmek için kullanılan en temel yöntemdir. İşlenecek olan imge piksellerinin komşulukları alt imgeyle evrişim(konvolüsyon) işlemine tabi tutulur. Konvolüsyon işleminde kaynak imgenin üzerinde gezdirilmesi için kullanılan alt imgeye genel olarak maske, şablon, çekirdek, filtre yada pencere ismi verilmektedir. Bu maskeler sayısal imge işlemede düzlemsel filtre işleminde kullanılması bakımından önem göstermektedir. Maske de kullanılan değerlere genel olarak piksel değeri değil de katsayı adı verilmektedir. 3 x 3 lük örnek maske ve maskenin kaynak imgede nasıl yerleştirildiği şekilde gösterilmiştir.



Şekil 2.3 3x3 lük maske ve kaynak imgede üzerinde yerleşimi (Gonzalez and Woods, 2002)

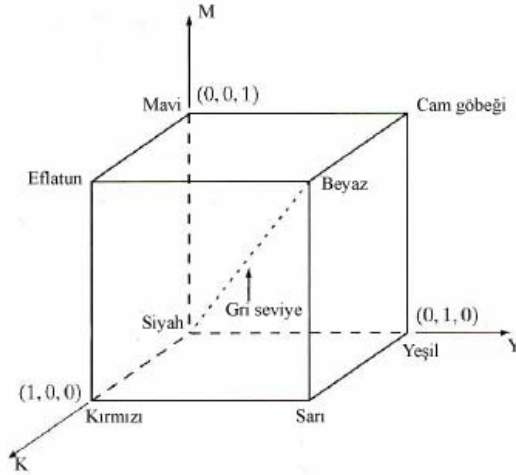
Genel olarak $M \times N$ boyutlu bir imgenin $f(x, y)$ lineer olarak $m \times n$ boyutlu bir maske ile konvolüsyonu işleminde kullanılan ifade

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t) \quad (2.1)$$

$g(x,y)$ şeklinde sonuçlanır (Gonzalez and Woods, 2002). Burada $a=(m-1)/2$, $b=(n-1)/2$ ve maske katsayısı $w(s,t)$ dir. Bu ifade de m ve n nin yerine 3 değeri konduğu zaman Şekil 2.3 'de kolayca gösterilmiş olur.

2.3 KYM(RGB) İmge

KYM renk modelinde her renk ana renk olan kırmızı-yeşil-mavi(red-green-blue) bileşenlerinden oluşmaktadır. Bu model kartezyan koordinatları temel alır. KYM modeline göre koordinat birimleştirilmiş değerleri Şekil 2.4'deki küpten görülebilir.



Şekil 2.4 KYM renk modeli (Gonzalez and Woods, 2002)

Şekil 2.4'e göre KYM değerleri ayrı ayrı üç köşede yerleşmiştir(Kırmızı yeşil ve mavi). Siyah orijindedir, beyaz ise orijinden en uzak noktadadır. Bu modelde gri seviye orijinden başlayıp beyaz a kadar olan doğru boyunca uzanır. Bu modeldeki farklı renkler ya küpün

üzerinde ya da küpün içinde bir noktada yer alır. KYM renk modelinde temsil edilen imgeler her biri ana renkten oluşan üç bileşenle temsil edilir şöyle ki kırmızı yeşil ve mavi. KYM uzayında her pikseli temsil etmek için kullanılan bitlerin sayısı piksel derinliği olarak adlandırılır. Kırmızı, yeşil, mavi imgelerinin her birinin 8 bit olduğu bir KYM imgesi göze alındığında bu KYM imgesinin bit derinliğinin 24 bit olduğu söylenir. Bu KYM imgenin içerisinde kullanılabilecek renk sayısı ise $(2^8)^3 = 16,777,216$ dır.



Şekil 2.5 KYM imge

2.4 Gri Seviye

Gri seviye imgeler grinin tonlarından oluşur. Normal olarak 8 bitlik bir imge için maksimum değeri 255 minimum değeri ise 0 dır. Gri seviyeli imgeler diğer renk modellerindeki imgelerin belirli dönüşümler sayesinde oluşturulurlar. Aslında gri rengi KYM alt uzayındaki K,Y,M bileşenlerinin genliğinin eşit olduğu noktadaki renktir.(Şekil 2.4). KYM

den gri seviye ye dönüşüm yapılırken kullanılan ağırlıklar (2.2) den görülebilir.(OpenCV, 2000)

$$Y = 0.212671 \times R + 0.715160 \times G + 0.072169 \times B \quad (2.2)$$

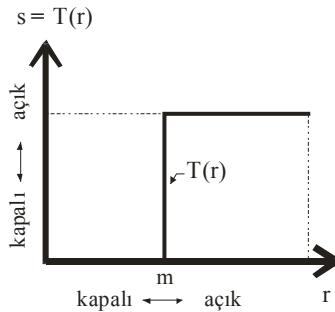
Ancak çalışma da kullanılan gri seviye imge (2.2) ye göre değil KYM imgesinin K, Y ve M kanalının ayrı ayrı ayarlanmasıyla ele alınmaktadır.

2.5 İkili Seviye

Sekiz bitlik bir imgede piksellerin ikili seviye değerleri sadece 255 ya da 0 dır. Sayısal imge çıkartımında değişik eşikleme yöntemleri kullanılarak gri-seviye imgelerden ikili imgeler çıkarılmaktadır. Değişik eşikleme yöntemleri olmasına karşın çalışmada kullanılan normal eşikleme ve OTSU eşikleme yöntemleri anlatılacaktır.

2.5.1 Normal eşikleme

Normal eşikleme yönteminde gri seviye imgelerde herhangi bir m değeri için girdi imgesi $f(x, y)$; T operatörü sonucu $g(x, y)$ ikili çıktı imgesine dönüşmektedir(Şekil 2.6). Burada $s = g(x, y)$ ve $r = f(x, y)$ dir.



Şekil 2.6 Normal eşikleme (Gonzalez and Woods, 2002)

2.5.2 Otsu eşikleme

Otsu metodu gri seviye için eşikleme yaparken eşikleme değerini optimum olarak bulur. Bir kere histogram değeri hesaplandıktan sonra yöntem oldukça basit ve hızlıdır. Otsu metodunun gri seviyeden iki seviyeli imgeye dönüştürürken yaptığı varsayımlar :

- Histogram ve imgenin bimodal (siyah ve beyaz) olduğu
- imgenin sabit istatistik değerlere sahip ve
- değişmeyen parlaklık değerine sahip olduğu

şeklinde. Metot, gri seviye imgenin piksellerini iki tane sınıfa ayırır (arkaplan ve nesne ya da tersi olarak). Ağırlaştırılmış sınıf içi değişimi

$$\sigma^2_w(t) = q_1(t)\sigma^2_1(t) + q_2(t)\sigma^2_2(t) \quad (2.3)$$

ve sınıf olasılıkları

$$q_1(t) = \sum_{i=1}^t P(i) \quad q_2(t) = \sum_{i=t+1}^l P(i) \quad (2.4)$$

(2.3) ve (2.4) deki 1, [1-L] gri seviye değerlerine, t bulunacak optimum eşikleme değerine karşılık gelmektedir. Sınıf ortalamaları

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad \mu_2(t) = \sum_{i=t+1}^l \frac{iP(i)}{q_2(t)} \quad (2.5)$$

ve ayrı ayrı sınıf değişimleri

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \sigma_2^2(t) = \sum_{i=t+1}^l [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \quad (2.6)$$

ve bundan sonra $\sigma^2_w(t)$ yi minimum duruma getirecek optimum t eşikleme değeri (2.7) de aranır. Toplam değişim

$$\sigma^2 = \underbrace{\sigma^2_w(t)}_{\text{sınıf içi}} + \underbrace{q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2}_{\text{sınıflar arası değişim}} \quad (2.7)$$

olarak optimum t yi bulmak için iterasyon başlangıç değerleri

$$q_1(1) = P(1) \quad \mu(1) = 0 \quad (2.8)$$

ve iterasyon adımları ise;

$$q_1(t+1) = q_1(t) + P(t+1) \quad (2.9)$$

$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)} \quad (2.10)$$

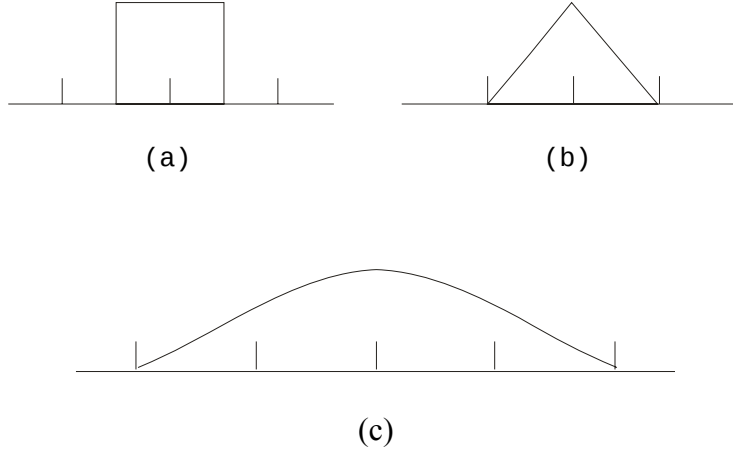
$$\mu_2(t+1) = \frac{\mu + q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)} \quad (2.11)$$

şeklindedir. (Course Handouts, 2004)

2.6 Sayısal İmgede Yeniden Boyutlandırma

Sayısal imgeler sayısal imge işlemede istenen boyutlarda işlenebilmesi için tekrar boyutlanmak zorundadır. Ara değerlendirme yöntemleri bilinen iki nokta arasındaki noktayı bir tahmin etmek üzerine dayalıdır. Yeniden boyutlandırma yaparken imgenin boyutu arttıkça ilgili imge için işlem zamanı da doğrusal olarak artmaktadır. Bu yöntem diğer ara değerlendirme yöntemlerine nazaran hassasiyet derecesi az olmasına

karşın oldukça hızlı bir yöntemdir. Temelde var olan değerlerden yeniden boyutlandırılan sayısal imgeye yeni verilen nasıl yerleştirileceğini bulur.

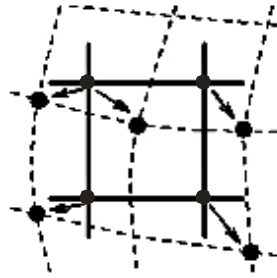


Şekil 2.7 1 Boyutlu ara değerlendirme fonksiyonları (a)en yakın komşuluk (b)lineer (c) kübik

2.6.1 En yakın komşuluk

$$R_0(x) = 1 \quad -\frac{1}{2} \leq x \leq \frac{1}{2} \quad (2.12)$$

Denklem (2.12) 1 boyutlu en yakın komşuluk formülü olmak üzere en yakın komşuluk ara değerlendirme yönteminde kaynak imgenin her bir pikseline en yakın değerler yeniden boyutlandırılan imgede yerini almaktadır (Şekil 2.8).



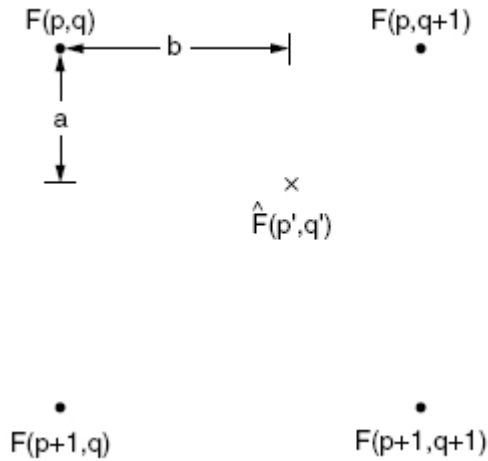
Şekil 2.8 En yakın komşuluk yöntemiyle yeniden boyutlandırma

$F(p', q')$, p, q nokta ikilisinin ara değerlendirme sonucu için $F(p, q)$, kaynak imge ve $F(p', q') = F(\text{yuvarla}(p), \text{yuvarla}(q))$ dir.

2.6.2 Bilineer aradeğerleme yöntemi

$$R_1(x) = \begin{cases} x+1 & -1 \leq x \leq 0 \\ 1-x & 0 \leq x \leq 1 \end{cases} \quad (2.13)$$

(2.13) 1 boyutlu bilinear ara değerlendirme formülü olmak üzere



Şekil 2.9 Bilineer yöntemiyle yeniden boyutlandırma

bilineer yönteminin sonucu (2.14) ve şekil 2.9 'da görülür.

$$\begin{aligned}\hat{F}(p', q') = & F(p, q)R\{-a\}R\{b\} + F(p, q+1)R\{-a\}R\{-(1-b)\} \\ & + F(p+1, q)R\{1-a\}R\{b\} + F(p+1, q+1)R\{1-a\}R\{-(1-b)\}\end{aligned}\quad (2.14)$$

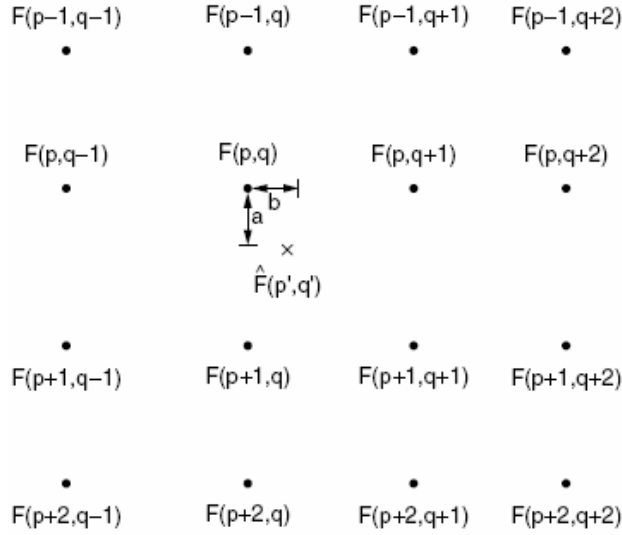
Burada $R\{x\} = R_1\{x\}$ dir.

2.6.3 Bikübik aradeğerleme yöntemi

$$R_c = \begin{cases} A_1|x|^3 + B_1|x|^2 + C_1|x| + D_1 & 0 \leq |x| \leq 1 \\ A_2|x|^3 + B_2|x|^2 + C_2|x| + D_2 & 1 \leq |x| \leq 2 \end{cases} \quad (2.15)$$

$$R_c = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \leq |x| \leq 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \leq |x| \leq 2 \end{cases} \quad (2.16)$$

(2.15) ve (2.16) da 1 boyutlu kübik aradeğerleme fonksiyonu olmak üzere (Pratt, 2001) de belirtildiği gibi, a kullanıcı bağımlı aradeğerleme iyileştirme parametresidir. Yöntem genel olarak zaman karmaşıklığı yüzünden 4x4 komşuluğu ile işleme tabi tutulur (Şekil 2.10).



Şekil 2.10 Bikübik yöntemiyle yeniden boyutlandırma

Bikübik aradeğerleme işleminin formülü (2.17) de gösterildiği gibidir.

$$F(p',q') = \sum_{m=-1}^2 \sum_{n=-1}^2 F(p+m, q+n) R_c\{(m-a)\} R_c\{-(n-b)\} \quad (2.17)$$

2.6.4 Süper örnekleme

Bu yöntemde eğer yeniden boyutlandırılan imge kaynak imgeden küçükse yukarıda tanımlanan aradeğerleme algoritmaları kaynak imgedeki bazı pikselleri gözardı edebilmektedir. Bu nedenle kaynak imge eşit dikdörtgenlere bölünür. Her dikdörtgen çıktı imgesinde bir bölgeye karşılık gelmekte ve kaynak imgedeki pikseller 1x1 lik karelerle temsil edilmektedir. Eşit dikdörtgenlerin içerisindeki piksellerin ya da bu dikdörtgenlerle kesişen ve 0 olmayan piksellerin ağırlık toplamı bulunur.

Eğer kaynak imgedeki piksel tam olarak dikdörtgenin içerisindeyse ağırlığı 1 olarak alınır. Eğer dikdörtgen ile pikselin kare bölgeleri $a < 1$ lik bir alanla kesişiyorsa bu pikselin ağırlığı a şeklinde alınır (Şekil 2.11).

	Δ_2	Δ_2	Δ_2	$\Delta_2 \times \Delta_3$
	1	1	1	Δ_3
	1	1	1	Δ_3
	Δ_1	Δ_1	Δ_1	$\Delta_1 \times \Delta_3$

Şekil 2.11 Süper örnekleme ağırlıkları(OpenCv, 2006)

Kaynak imgedeki pikselin yeniden boyutlandırılan imgedeki değeri dikdörtgen bölgelerinde hesaplanan ağırlık değerinin boyut(kaynak) / boyut(çıktı)' a bölünmesiyle bulunmaktadır.

2.7 Gürültüleri İmgeden Kaldırma

Sayısal imgelerde genellikle gürültünün ana kaynakları dış ortamdan imgeler alınırken ya da imgelerin iletim esnasında ortaya çıkmaktadır. İmgeleme sensörlerinin performansı o anki çevre koşulları ya da sensörün kendi içerisindeki algılayıcıların kalitesinden çok fazla etkilenmektedir. Örneğin sensörün sıcaklığı, ışık koşulları elde edilen imgedeki gürültüyü çok fazla etkilemektedir.

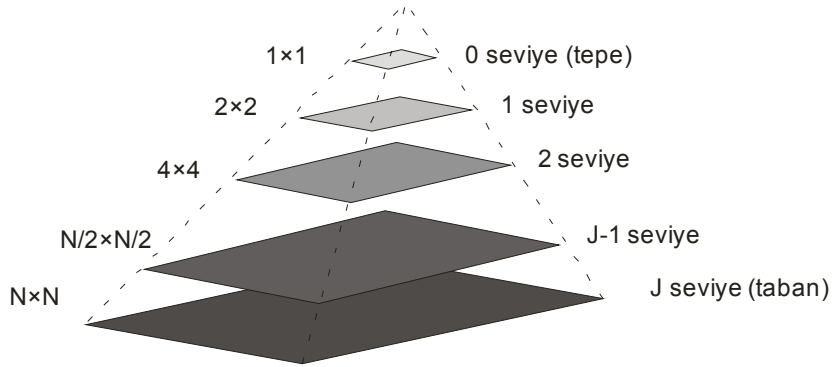
2.7.1 Ortalama(medyan) filtre

Ortalama filtresi ilgili pikselin komşuluğundaki piksellerin (kendi dahil) gri seviye değerlerini sıralayarak, ortalanmış değerle yer değiştirir. Bu filtreler lineer olan filtrelerle göre imgeyi daha az yumuşattıkları için genelde fazla kullanılmaktadır. Ortalama filtreleri özellikle dürtü gürültülerini (tuz biber etkisi) kaldırmakta etkilidir (Gonzalez and Woods, 2002).

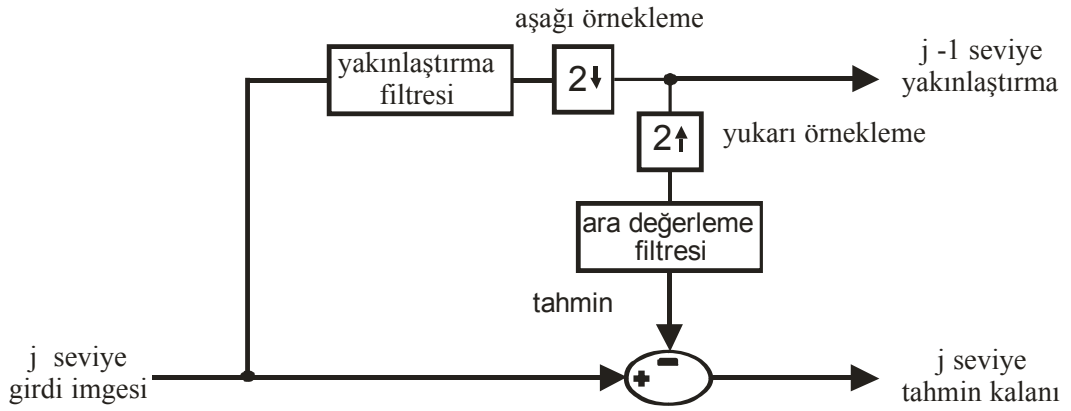
Örneğin 3x3 komşuluğundaki piksellerin gri seviye değerleri 10, 20, 20, 20, 15, 20, 20, 25, 100 olsun. Bu değerler sıralandığında 10, 15, 20, 20, 20, 20, 20, 25, 100 değerlerinde 15 ile 20 değerinin değiştiği görülür.

2.7.2 İmge piramitleri

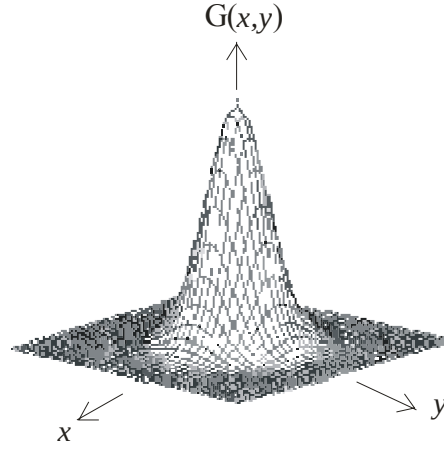
Yapısal olarak sayısal imgeleri bir çözünürlükten daha fazla çözünürlüklerde temsil etmek için imge piramitleri kullanılır (Gonzalez and Woods, 2002). Şekil 2.12'den görüldüğü gibi piramidin tabanı imgenin en yüksek çözünürlüğünü piramidin tepesi ise imgenin yaklaşılmış düşük çözünürlüğünü temsil etmektedir. Piramitden yukarı doğru gidildikçe çözünürlük azalmaktadır. Tabandaki J seviye $2^J \times 2^J$ ya da $N \times N$ olduğundan ara j seviye ise $2^j \times 2^j$ dir. Burada $J = \log_2 N$ ve $0 \leq j \leq J$ dir. Tam olarak temsil edilen piramitlerin çözünürlük seviyeleri $J+1$ olan $2^J \times 2^J$ den $2^0 \times 2^0$ ye uzanmaktadır. Çoğu piramitler ise $P+1$ seviyesine yuvarlanırlar. Bu yüzden $j = J - P, \dots, J - 2, J - 1, J$ ve $1 \leq P \leq J$ dir. Şekil 2.13'de gösterilen blok diagrama göre $j-1$ seviye yakınlaştırma çıktısı kaynak imgenin bir ya da daha fazla yakınlaştırma çözünürlüğünü içeren yakınlaştırma piramitlerini oluşturmak için kullanılır. j seviye tahmin kalanı çıktısı ise tahmin kalanı piramitlerini oluşturmak için kullanılır.



Şekil 2.12 İmge piramidi



Şekil 2.13 İmge piramidini oluşturan sistemin blok diagramı



Şekil 2.14 Belirli bir standart sapmaya sahip düşük geçiren gaussian filtresi

Şekil 2.13 blok diagram da gösterildiği gibi yakınlaştırma ve tahmin kalanı piramitleri yinelemeli biçimde hesaplanır. $P+1$ seviye piramidi Şekil 2.13'deki blok diagramın P kere işlenmesi ile sonuçlanır. Buna göre birinci yineleme de $j = J$ ve $2^J \times 2^J$ boyutundaki gerçek imge J seviye girdi imgesi olarak uygulanır. Bu işlem $J-1$ seviye yakınlaştırma ve J seviye tahmin kalanı olarak sonuçlanır. $j = J-1, J-2, \dots, J-P+1$ (sırasıyla) seviye yinelemeleri için bir önceki $j-1$ seviye yakınlaştırma çıktısı tekrar girdi olarak sisteme uygulanır. Her yineleme üç adımdan oluşmaktadır.

1.adım

Girdi imgesinin düşük çözünürlüğe sahip yakınlaştırma piramidi hesaplanır. Bu genel olarak imgenin Şekil 2.14 deki düşük geçiren Gaussian filtresi ile konvolüsyonu, yani Gaussian piramidinin

oluşturulması ardından da sonucun 2 çarpanı ile aşağı örneklenmesi olarak tanımlanır.

2.adım

1.adımın çıktısı tekrar 2 çarpanı ile yukarı örneklenir ve ara değerlendirme filtresi ile filtrelenir.

3.adım

2.adımın çıktısı ile kaynak imgenin farkı alınır ve j seviye tahmini kalan olarak tanımlanır. Bu j seviye tahmini kalan ileride kaynak imgenin tekrar oluşturmak için kullanılır.

Bu aşamaya kadar olan adımlar kaynak imgenin çözünürlüğünü düşürmekle ilgilidir. Kaynak imgeyi tekrar eski boyutlarına getirmek için yaklaştırma piramidi sonucu Şekil2.13'deki blok diagramda geri yönde gidilerek 2 çarpanı ile Gaussian piramidi oluşturulur ve bu piramitle tahmini kalan piramidi toplanarak kaynak imge tekrar elde edilir.

2.8 Kenar Belirleme Yöntemleri

Kenar bulma işlemi sayısal imge işlemede en çok kullanılan yöntemlerden birisidir. Kenar, iki bölge arasındaki sınır üzerinde uzanan bağlı noktalar olarak tanımlanmaktadır.(Gonzalez and Woods,2002). Kenar belirleme yöntemleri nesne belirlemede nesnenin arka plandan ayrılmasında oldukça belirleyici rol oynamaktadırlar. Sayısal imge işleme algoritmalarında kenar belirleme yöntemleri birbirlerine karşı avantaj ve dezavantajlarla aşağıda açıklanmıştır (Gonzalez, Woods, and Eddins, 2004). Kenar belirleme yöntemleri sabit gri seviye değerlerinin bastırılmasına ve değişiklere duyarlı olmayı gerektirir (Jähne, 2002).

Burada iki boyutlu imgenin (x, y) noktasındaki gradyan fonksiyonu $f(x, y)$ vektörel olarak (2.18) de gösterildiği gibidir.

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} \quad (2.18)$$

Kenar belirleme de önemli diğer bir özellik ise gradyan vektörünün büyüklüğüdür.

$$|\nabla f| = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \quad (2.19)$$

Gradyan vektörünün doğrultusu ise açısal olarak x eksenine göre

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (2.20)$$

şeklinde tanımlanır. Genel olarak (2.19) fazla işlem zamanı aldığı için pek tercih edilmemektedir. Bunun yerine benzer sonucu üreten (2.21) kullanılır.

$$|\nabla f| \approx |G_x| + |G_y| \quad (2.21)$$

Şekil 2.15’de her bir yönteme ait olan maskeler gösterilmiştir Sobel kenar belirleme yöntemi birinci derece türevleri olan G_x ve G_y i kullanarak Şekil 2.15 (c)’de gösterilen maskeyle imgeyi filtreler. Sobel filtresi sadece bir doğrultudaki üç pikseli grupların ortalama toplamına baktığı için süreksizliklere karşı daha az duyarlıdır.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

a)

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \quad G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

b)

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

c)

-1	0
0	1

0	-1
1	0

$$G_x = z_9 - z_5 \quad G_y = z_8 - z_6$$

d)

Şekil 2.15 Kenar belirleme maskeleri a) imge komşulukları b) Prewitt c) Sobel d) Roberts (Gonzalez, Woods and Eddins., 2004)

Prewitt kenar belirleme yöntemi birinci derece türevleri olan G_x ve G_y i kullanarak Şekil 2.15 (b)'de gösterilen maskeyle imgeyi filtreler. Prewitt filtresi imge üzerinde filtre gezdirilmesi bakımından daha az zaman harcar. Sobel e göre daha gürültülü sonuçlar verme eğilimindedir.

Roberts kenar belirleme yöntemi birinci derece türevleri olan G_x ve G_y i kullanarak Şekil 2.15 (d)'de gösterilen maskeyle imgeyi filtreler. Bilinen en eski kenar belirleme yöntemlerinden biridir. Fakat diğerlerine göre simetrik olmaması ve 45° nin katları olan kenarları bulamaması bakımından daha az tercih edilir. Fakat basit bir maskeye sahip olması ve daha az zaman harcaması yüzünden gömülü sistemlerde tercih edilir.

Canny, yukarıda tanımlanan kenar belirleme yöntemleri arasında en etkili olan yöntemdir. Canny yönteminde bir imge için kenar çıkarımında izlenen yol aşağıdaki gibidir (Gonzalez, Woods and Eddins., 2004).

- imge, gürültüyü azaltmak için şekil 2.14 deki belirli bir standart sapmaya sahip olan Gaussian filtresi ile yumuşatılır.
- imge deki her nokta üzerinde yerel gradyan $\nabla f = [G_x^2 + G_y^2]^{1/2}$ ve kenar doğrultusu $\alpha(x, y)$ hesaplanır. Gradyan doğrultusundaki yerel olarak büyüklüğü en büyük olan piksel kenar noktası olarak işaretlenir.

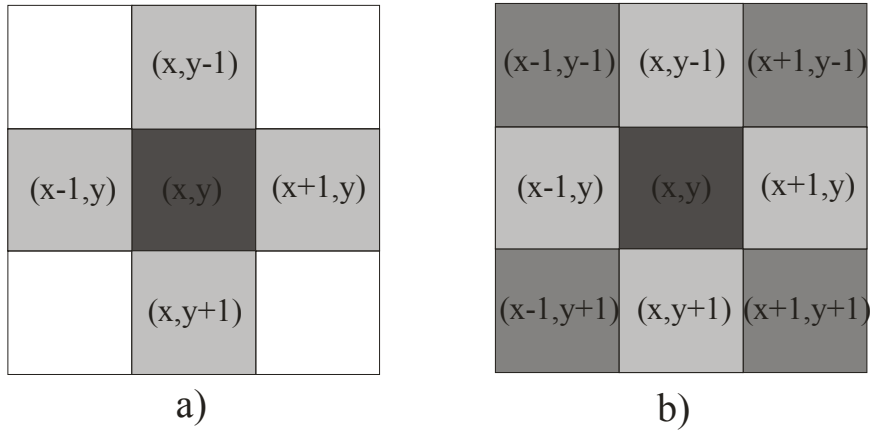
- kenar noktalarına dik olan ve yerel olarak gradyan büyüklüğü maksimum olmayan noktalar bastırılır (non-maximum suppression).
- yöntem eşikleme değerlerini histerisis olarak kullanır. metodun iki eşik değerinden yüksek eşik değerinin üstünde olan noktalar kenara ait nokta, düşük eşik değerinin altında olanlar kenara ait olmayan nokta, iki eşik değerinin arasında olan ve bir kenar noktası ile komşuluğu olan noktalar ise kenar noktası olarak adlandırılır.

2.9 Morfolojik İşlemler ve Şekil Analizi

2.9.1 4 komşuluğu ve 8 komşuluğu

Piksellerin komşuluk özellikleri göze alındığında imge üzerinde $(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$ koordinatlarına sahip (x, y) noktasındaki bir p pikselin dört tane yatay ve dikey komşuluğu vardır (Gonzalez and Woods, 2002). Bu piksel kümesine p nin 4 komşulu adı verilir ve $N_4(p)$ olarak gösterilir. Her piksel (x, y) noktasına birim uzaklıktadır ve eğer (x, y) noktası imgenin sınırındaysa p nin bazı komşulukları imgenin dışına karşılık gelir. p nin dört köşegen komşuluğu $(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$ koordinatlarına sahiptir ve $N_D(p)$ ile gösterilir. Bu noktalar 4 komşuluğu ile birlikte p nin

8 komşuğu olarak adlandırılır ve $N_8(p)$ şeklinde gösterilir. Aynı şekilde eğer (x, y) noktası imgenin sınırındaysa p nin bazı komşulukları imgenin dışına karşılık gelmektedir. Şekil 2.16 (a) ve (b) den bu komşuluklar görülebilir. Şekil 2.16 (a) ve (b) de grinin tonları 1 e karşılık gelmektedir.



Şekil 2.16 (x, y) noktasındaki komşuluklar a)4 b)8 (Gonzalez and Woods, 2002)

2.9.2 Bağlı bileşenlik

Bir pikselin ikili bir imgede 4 komşuluğunda veya 8 komşuluğunda diğer bir pikselle bağlı olması koşulu bu iki pikselin bu komşuluklarda aynı değere sahip olmasını gerektirmektedir (Gonzalez and Woods, 2002). V iki pikselin bitişikliğini göstermek üzere ikili imgeler için 1 değerine sahip olsun $V = \{1\}$.

- 4 bitişikliği V kümesinden den p ve q piksellerinin değerleri eğer $q \in N_4(p)$ nun kümesindeyse ya da

- 8 bitişikliği V kümesinden den p ve q piksellerinin değerleri eğer $q \in N_8(p)$ nun kümesindeyse bu iki piksel bağlı olarak adlandırılır.

2.9.3 Genişletme

Genişletme sayısal bir imge üzerinde yapılacak morfolojik(biçimsel) işlemlerin temelini oluşturmaktadır. Genişletme işlemi ikili bir imgedeki nesneleri büyütme ya da kalınlaştırma işlemidir. Nesnelerin genişletme işleminde nasıl bir işlem göreceği genişletme işlemine sokulan yapısal elemanın şekline bağlıdır. Matematiksel olarak Z^2 de tanımlanan A kümesi ve B yapısal elemanı için genişletme işlemi

$$A \otimes B = \left\{ z \mid \left(\hat{B} \right) \cap A \neq \emptyset \right\} \quad (2.22)$$

olarak tanımlanır (Gonzalez and Woods, 2002).

(2.22) de genişletme işleminde B yapısal elemanının orijinine göre simetrisi alınarak z kadar ötelendikten sonra A la genişletme işlemine tabi tutulmaktadır. Aşağıdaki Şekil 2.17 (b)'de köşegen elemanları 1 lerden oluşan yapısal elemanla 1 lerden oluşan dikdörtgen bölgeye sahip Şekil 2.17 (a) daki ikili imgenin nasıl genişlediği görülebilir (Şekil 2.17). Şekil 2.17'de 1 ler beyaz a 0 lar siyah a karşılık gelmektedir. Şekil 2.18'de yapısal elemanın kaynak imge üzerinde nasıl ötelendiği görülebilir. Aşağıda kırmızı çerçeve ile işaretlenmiş yapısal elemanın merkezi ikili imgede 1 değerine sahip kısımlarla çakıştığı zaman çakışan kısım genişletilmektedir. Şekil 2.19'da çıktı imgesi görülmektedir.

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

a)

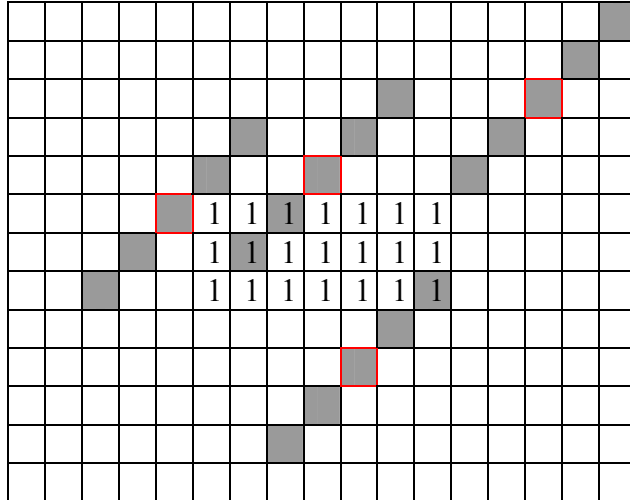
```

      1
     1
    1
   1
  1
 1
1
1

```

b)

Şekil 2.17 Genişletme ve yapısal eleman a) genişleştirilecek ikili imge b) yapısal eleman (Gonzalez and Woods, 2002)



Şekil 2.18 Yapısal elemanın ikili imgedeki ötelenmesi (Gonzalez and Woods, 2002)

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

Şekil 2.19 Genişletme çıktı imgesi (Gonzalez and Woods, 2002)

2.9.4 Şekil analizi

Eşikleme ve kenar bulma yöntemleriyle KYM ya da gri seviye imgelerden ikili imgelere dönüştürülen imgelerden nesneler ayrıştırıldıktan sonra bu nesnelerin genellikle daha ileri işlemlerde yürütülebilmesi için daha uygun bir şekilde temsil edilebilmeleri ve tanımlanabilmeleri gerekmektedir. Temel olarak nesneleri temsil etmek iki seçenek vardır(Gonzalez and Woods, 2002);

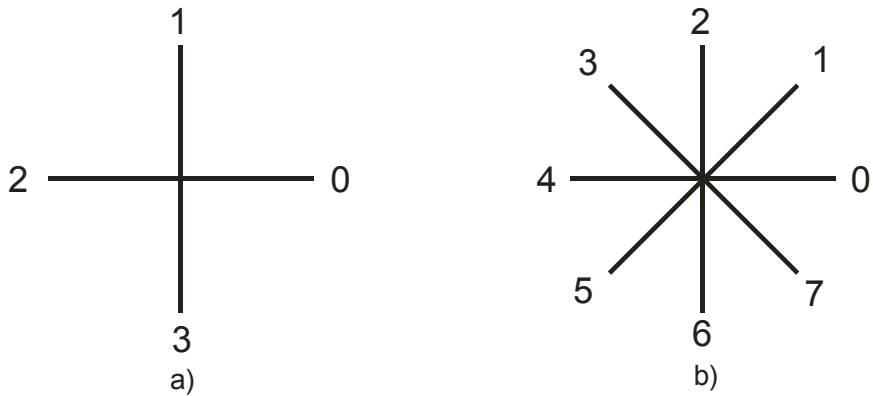
- nesnelerin dış özellikleriyle(kontör),
- nesnelerin iç özellikleri(bölgeyi temsil eden pikseller)

Temsil etme şeklinin seçilmesi bilgisayarda işlenecek verilerin kolaylaştırılmasının sadece ilk kısmıdır. Diğer kısmı ise seçilen temsil etme şekline dayalı nesnenin tanımlanmasıdır. Örneğin bir nesne

kontörleri ile temsil edilebilir ve bu kontörlerde uzunluklarıyla ya da köşelerinin sayısı ile tanımlanabilir.

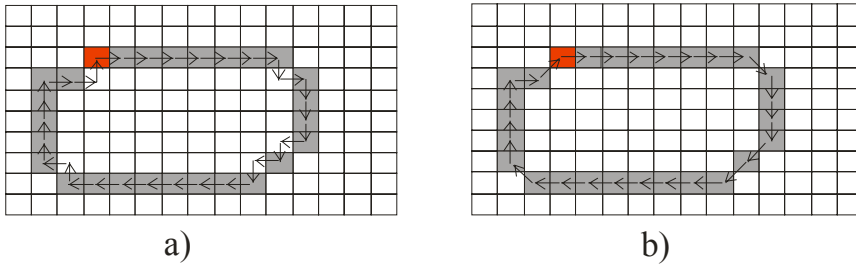
2.9.4.1 Zincir kodu

Zincir kodu ayrık ızgara üzerinde ikili imgeleri temsil etmek için etkili bir yöntemdir. Sayısal imgeler çoğunlukla eşit aralığa sahip x ve y doğrultusundaki ızgara şeklinde alınır ve işlenir. Zincir kodları bir sınırı belirli uzunlukta ve doğrultuda olan bağlanmış sıralı düz doğru parçalarıyla temsil eder (Gonzalez and Woods, 2002). Genellikle bu temsil parçaların 4 ya da 8 komşuluğuna bağlıdır (Bkz 2.9.1). Her parçanın doğrultusu Şekil 2.20 de gösterilen sayı şemasına göre kodlanır. Şekil 2.20 (a) da zincir kodunun 4 komşuluğundaki doğrultusu (b) de ise 8 komşuluğundaki doğrultusu gösterilmektedir. İkili imgeden nesnelere ait sınır koordinatlarının noktalar şeklinde depolanması yerine başlangıç pikseli alınır ve sadece bu pikselin koordinatı depolanır.



Şekil 2.20 Zincir kodlarının sayı şeması a) 4 komşuluğu b) 8 komşuluğu

4 komşuluğunda ve 8 komşuluğunda kontörlerin çıkartılması ve kodlanması Şekil 2.21 (a) ve (b)'den görülebilir. Kırmızı ile işaretlenmiş piksel başlangıç noktasına karşılık gelmektedir. Şekil 2.21 (a) nın zincir kodu karşılığı 00000003033323232222222121111001 şeklinde ve Şekil 2.21(b) nin zincir kodu karşılığı ise 000000076665544444443222201 şeklindedir.

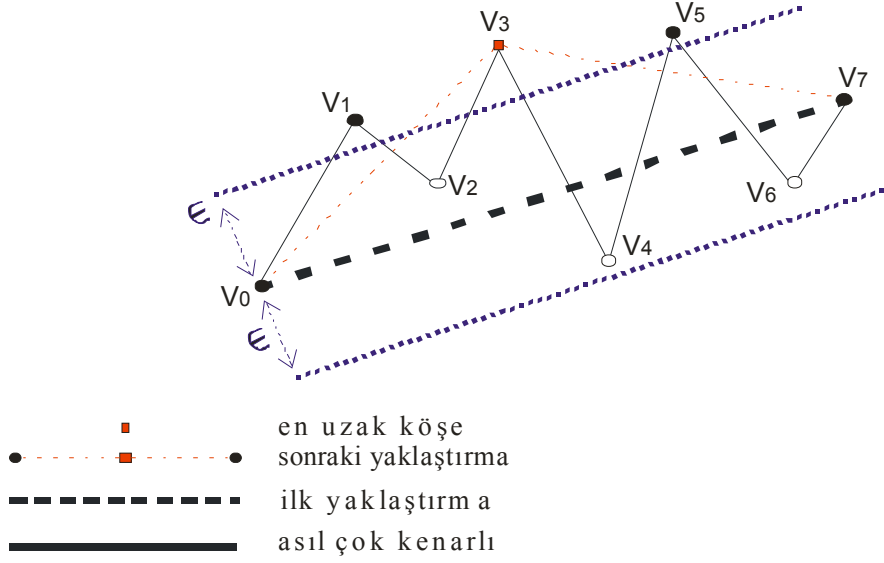


Şekil 2.21 Zincir kod karşılıkları a) 4 b) 8 komşuluğu

2.9.4.2 Çok kenarlı basitleştirme

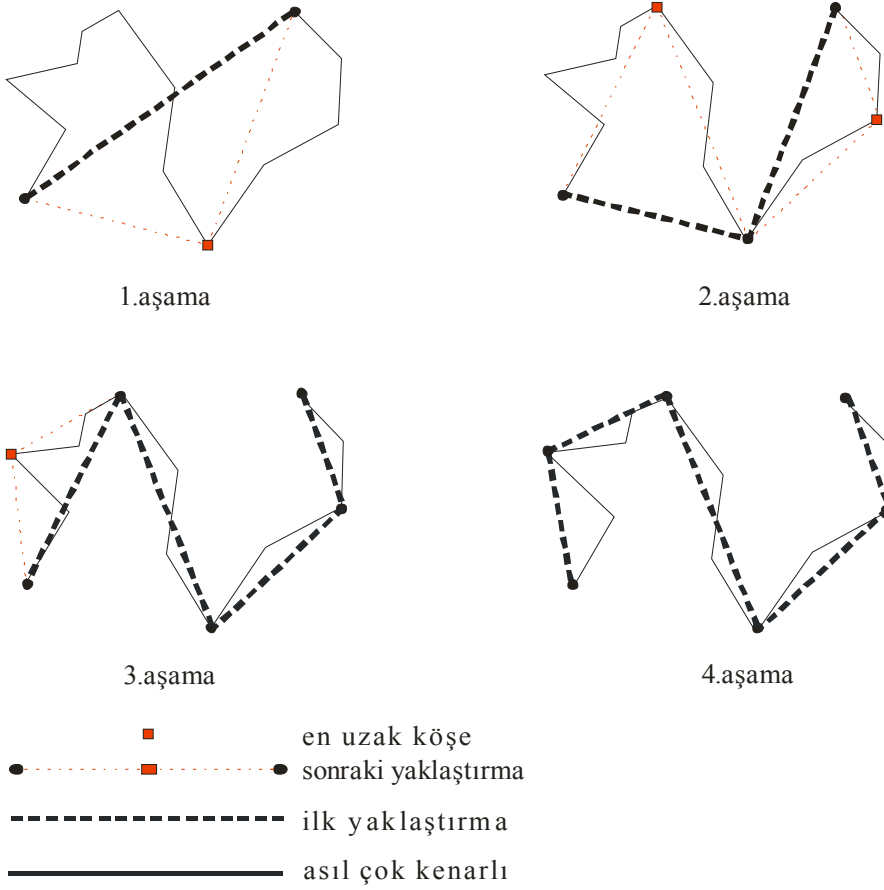
(Hershberger and Snoeyink., 1992)'de belirtilen algoritma az sayıdaki köşeler üzerinde düz bir kontör almak için kullanılır. Bu algoritma bir köşenin kenara yakınlığını kullanır. Algoritma ayrıntılı olarak ele alındığında çok kenarlılığın en uzak iki köşesi olan son noktalar kabaca bir doğruyla basitleştirmenin ilk adımı için birleştirilir. Çok kenarlılığın tam olarak ne kadar yaklaştırılacağı tüm ara köşelerin bu doğruya belirli bir tolerans ϵ değerinden büyük olup olmayacağına bağlıdır. Bu kalan köşeler doğruya belirli tolerans değerinden daha az ise yaklaştırma iyidir ve son noktalar korunur. Eğer tolerans değerinden fazlaysa yaklaştırma yeterince iyi değildir. Bu durumda asıl doğrudan en uzaktaki nokta yeni bir köşe olarak belirlenir, yaklaştırılacak olan çok

kenarlıya eklenir ve son noktalardan bu noktaya asıl doğrudan daha kısa olan iki doğru Şekil 2.22'deki gibi çizilir.(Sunday, 2007). Bu yöntem bu iki kısa doğru üzerinde yinelemeli şekilde devam eder.



Şekil 2.22 ϵ değerine göre en uzak köşe

Başarılı bir yaklaştırma Şekil 2.23 den görülebilir.



Şekil 2.23 Başarılı çok kenarlı yaklaştırma

2.10 Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninden esinlenerek oluşturulmuş bilgi işleme araçlarıdır. Beyin, doğrusal olmayan, bilgi işleme bakımından paralel, oldukça karmaşık bir yapıya sahiptir. Beyin günümüzdeki klasik bilgisayarlardan çok daha hızlı bir şekilde kendisini oluşturan nöronları organize ederek imge tanıma işlerini

gerçekleştirebilmektedir. Örneğin, bozuculu bir ortamda yüz tanıma işleminde klasik bilgisayarların işlemi günlerce sürerken beyin bu işlemi yaklaşık 100-200 ms de gerçekleştirmektedir. Yapay sinir ağı deneyimsel bilgiyi depolamaya doğal eğilimi olan ve bunun kullanımını sağlayan basit işlem birimlerinden oluşan, oldukça büyük paralel dağıtık bir işlemcidir(Haykin, 1999). YSA'nın tarihine bakıldığında; YSA araştırmaları üç periyotluk geniş bir aktiviteye maruz kalmıştır (Jain, Mao and Mohiuddin, 1996). 1940 ların ilk zirvesi McCulloch ve Pitts in öncülüğünde olmuştur. İkincisi Rosenblatt'ın algılayıcı yakınsama teoremi ve basit algılayıcının sınırlı olduğunu gösteren Minsky ve Papert in çalışmalarıdır. Minsky ve Papert in sonuçları bilim adamlarının özellikle bilgisayar bilimi topluluğunun heyecanını azaltmıştır. YSA çalışmalarındaki sükunet 20 yıl kadar devam etmiştir. 1980 lerin başından itibaren, YSA larına olan ilgi tekrar artmıştır. Bu ilgi artışının başlıca etkenleri, 1982'deki Hopfield'in enerji yaklaşımı ve ilk defa Werbos tarafından önerilen geri yayılım algoritmasının geliştirilmesi olmuştur. 1986'da geri yayılım algoritması Rumelhart tarafından popüler hale getirilmiştir. YSA nın özellikleri incelendiğinde;

- Uyarlanabilir öğrenme

YSA ları güçlü öğrenme yeteneğine sahiptirler. Etraflarını çevreleyen ortamlara kendilerini uyarlayabilirler.

- Genelleştirme

Genelleştirme, YSA nın eğitim esnasında karşılaşmadığı girişlere makul sonuçlar üretmesidir. İyi eğitilmiş bir YSA üstün genelleştirme yeteneğine sahiptir.

- Doğrusal olmama

Nöron doğası gereği lineer olmama özelliğine sahiptir. YSA lar problemleri evrensel yaklaştırma yeteneğini kullanarak çözer. YSA lar siyah kutu gibi çalışmaktadır. Bu yapıyı analiz etme ve problem bilgisini arama zorunluluğunu ortadan kaldırır. Sadece girdi ve çıktı verilerin verilmesi yeterlidir.

- Paralellik

YSA ları paralel işlem yapısına sahiptirler. Her basit işlem birimi(nöron) genellikle basit toplama, çarpma, bölme ve eşikleme işlemlerine sahiptir. Bu paralel yapı paralel yazılım ve donanım uygulamalarına izin verir.

- Kendi kendini organize etme

Bazı YSA lar mimarisi gereği kendi kendini organize etme özelliğine sahiptir. Bu mimarilerin eğitilmesi eğiticişiz öğrenme şeklinde olmaktadır.

- Gürbüzlük ve hata toleransı

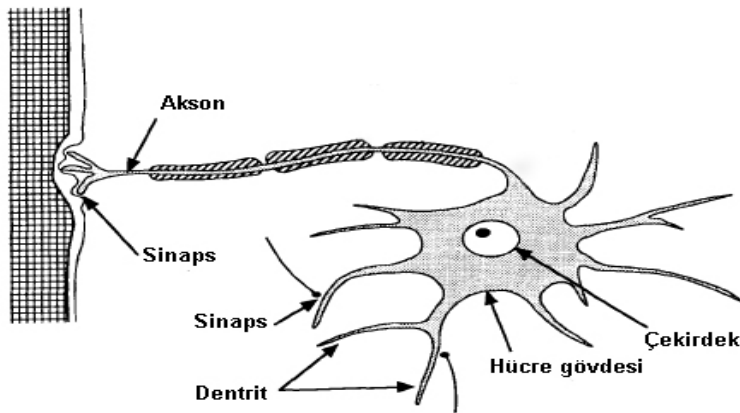
YSA lar gürbüz ve hata toleransı yeteneğine sahiptirler. Bir YSA kolaylıkla belirsiz, bulanık, gürültülü ve istatiksel bilgileri ele alabilir. YSA bilginin tüm ağ yapısı içerisinde dağıtık bir şekilde tutulduğu dağıtık bir bilgi sistemidir. Bundan dolayı tüm performans bazı düğümlerdeki kayıplar ya da ağ daki bazı bağlantıların kopması ile önemli bir şekilde değişmez. Ağ öğrenme

kuralı ve o anki sonuçları kullanarak ağırlıkların güncelleşmesi ile hemen bu sorunu telafi edecektir.

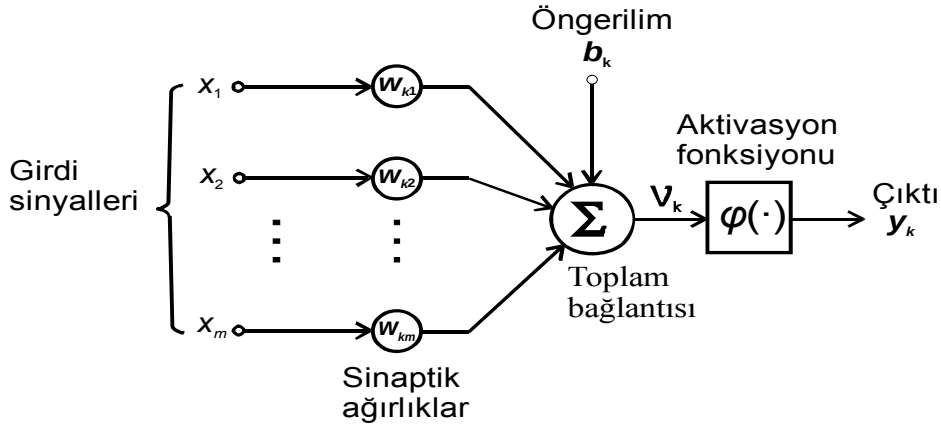
olduğu görülür (Du and Swamy, 2006).

2.10.1 Nöron modeli

Nöronlar insan beyninin en temel işlem birimleridir. Nöronlar temel olarak hücre gövdesi, dendrit ve aksondan oluşurlar. Hücre gövdesi nöronların ihtiyacı olan maddeyi tutan plazma ve kalıtsal özellikleri içeren çekirdeğe sahiptir. Nöronlar diğer nöronlardan bilgiyi sinapslara yakın olan dendritler sayesinde alır ve hücre gövdesinin ürettiği bilgileri sinapslara yakın olan aksonlar sayesinde iletir. Sinapslar iki nöron arasındaki temel ve işlevsel birimdir. YSA mimarisini gerçekleştirirken bu birimin modellenmesi gerekir. Biyolojik nöronların iç yapısı şekil 2.24 de, yapay sinir ağındaki modeli Şekil 2.25 de gösterilmiştir.(Haykin, 1999)



Şekil 2.24 Biyolojik nöronlar (Freeman and Skapura, 1991)



Şekil 2.25 Nöronun lineer olmayan modeli (Haykin, 1999)

Şekil 2.25 deki modele göre sinapslar ağırlıklarla temsil edilir. Girdi sinyalleri sinaptik ağırlıklarla çarpıldıktan sonra öngerilimle toplanır, aktivasyon fonksiyonundan geçirilir ve çıktı elde edilir. Model matematiksel terimler ile ifade edilecek olursa

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.23)$$

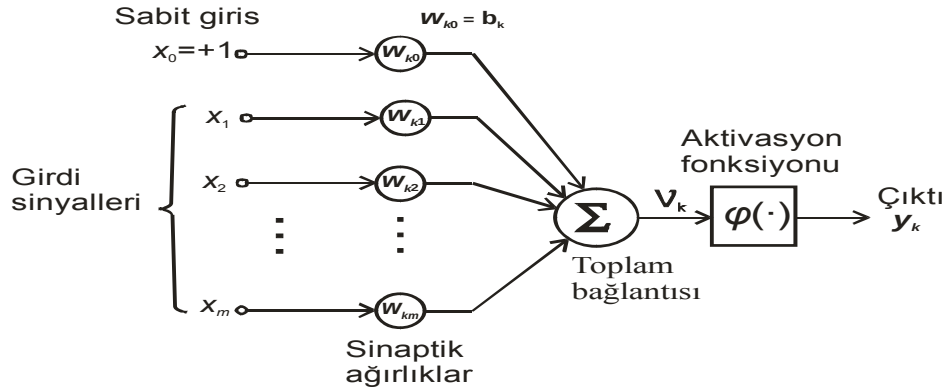
$$v_k = u_k + b_k \quad (2.24)$$

$$y_k = \varphi(u_k + b_k) \quad (2.25)$$

ile temsil edilir. Burada $x_1, x_2, \dots, x_m$ girdi sinyalleri, $w_{k1}, w_{k2}, \dots, w_{km}$ nöron k nın sinaptik ağırlıkları, u_k girdi sinyallerinden dolayı lineer birleştirici çıktısı, b_k öngerilim, $\varphi(\cdot)$ aktivasyon fonksiyonu, v_k uyarılmış yerel alan ve y_k ise nöronun çıktı sinyalidir. Denklemler daha da sadeleştirildiğinde u_k ve b_k

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (2.26) \quad \text{ve} \quad y_k = \varphi(v_k) \quad (2.27)$$

ortaya çıkmaktadır. (2.27) ye yeni bir sinaps eklenmiştir. Girişi $x_0 = +1$ ve ağırlığı $w_{k0} = b_k$ dır. Şekil 2.26'da doğrusal olmayan nöronun diğer modeli görülmektedir.



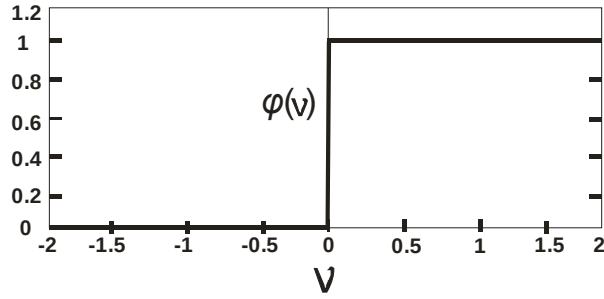
Şekil 2.26 Doğrusal olmayan nöronun diğer modeli

YSA larında değişik aktivasyon fonksiyonları kullanılmaktadır. Bu fonksiyonlar aşağıda özetlenmiştir. $\varphi(v)$ ile gösterilen aktivasyon fonksiyonu nöronun çıktısını uyarılmış yerel alan v_k cinsinden tanımlar.

- Eşikleme fonksiyonu

Eşikleme fonksiyonun ifadesi (2.28)'de ve grafiksel gösterimi Şekil 2.27'de verilmektedir.

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (2.28)$$

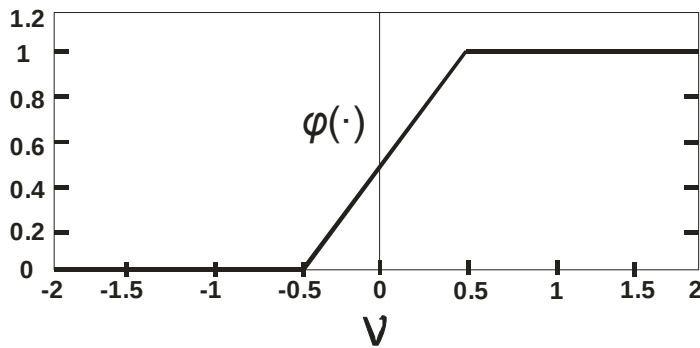


Şekil 2.27 Eşikleme fonksiyonu

- Parçalı doğrusal fonksiyonu

Parçalı doğrusal fonksiyonun ifadesi (2.29) de ve grafiksel gösterimi Şekil 2.28’de verilmektedir.

$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \quad (2.29)$$

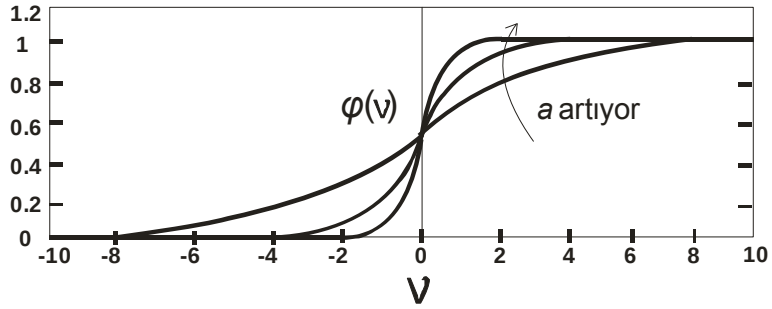


Şekil 2.28 Parçalı doğrusal fonksiyonu

- Sigmoid fonksiyonu

Sigmoid fonksiyonun ifadesi (2.30) da ve grafiksel gösterimi Şekil 2.29'da verilmektedir. Sigmoid fonksiyonu YSA da en çok kullanılan fonksiyondur. (2.30) da a sigmoid fonksiyonun eğim parametresini göstermektedir.

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (2.30)$$



Şekil 2.29 Sigmoid fonksiyonu

Yukarıda verilen fonksiyonların hepsinin değeri 0 ile 1 arasında değişmektedir. Bazen aktivasyon fonksiyonlarının değerinin -1 ile 1 arasında değişmesi gerekebilir. Özel olarak eşikleme fonksiyonu

$$\varphi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \quad (2.31)$$

şeklinde tanımlanabilir. Bu fonksiyona signum adı verilir. Değeri -1 ile 1 arasında değişen yaygın kullanılan diğer fonksiyonda

$$\varphi(v) = \tanh(v) \quad (2.32)$$

fonksiyonudur.

2.10.2 En küçük ortalama kareler yöntemi

YSAların başlıca özelliği ağırlık çevresinden öğrenme özelliği ve performansını öğrenme ile geliştirebilmesidir. Bu öğrenme işlemini gerçekleştirirken belirli öğrenme kuralları uygulanmaktadır. İlerde kullanılacak olan geri yayılım algoritması için öğrenme kuralı olan hata düzeltme temelli en küçük ortalama kareler algoritması kullanılmaktadır. Bu algoritma sınırlamasız eniyileştirme tekniği olan endik iniş yöntemine dayalıdır (Haykin, 1999). Endik iniş yönteminde, ağırlık vektörü w ye uygulanan ardışık ayarlamalar endik iniş yönünde, yani gradyan vektörü $\nabla \mathcal{E}(w)$ nin tersi yönündedir. Basitlik için, gradyan vektörü $g = \nabla \mathcal{E}(w)$ şeklinde yazılır. Endik iniş yönteminde ağırlık vektörünün güncellenmesi $w(n+1) = w(n) - \eta g(n)$ (2.33)

şeklindedir. Yukarıdaki denklemlerde $\mathbf{\epsilon}(w)$ bilinmeyen ağırlık vektörleri w nin sürekli türevi alınabilir maliyet fonksiyonudur, η pozitif olarak tanımlanan adım boyutu ya da öğrenme hızı parametresi, $g(n)$ $w(n)$ noktasında hesaplanan gradyan vektörüdür. n . iterasyondan den $n+1$. iterasyona kadar yöntem

$$\begin{aligned}\Delta w(n) &= w(n+1) - w(n) \\ &= -\eta g(n)\end{aligned}\quad (2.34)$$

olan düzeltmeyi yapar. En küçük ortalama kareler algoritması maliyet fonksiyonu $\mathbf{\epsilon}(w)$ nun anlık değerlerine bağlıdır. Şöyle ki

$$\mathbf{\epsilon}(w) = \frac{1}{2} e^2(n) \quad (2.35)$$

burada $e(n) = d(n) - y(n)$ n zamanındaki hata sinyali, $d(n)$ istenen çıktı, $y(n)$ ise sistemin çıktısıdır. $\mathbf{\epsilon}(w)$ nin ağırlık vektörü w ye göre türevi

$$\frac{\partial \mathbf{\epsilon}(w)}{\partial w} = e(n) \frac{\partial e(n)}{\partial w} \quad (2.36)$$

şeklindedir. En küçük ortalama kareler algoritmasında, doğrusal nöron durumunda hata sinyali

$$e(n) = d(n) - x^T(n)w(n) \quad (2.37)$$

şeklinde tanımlanabilir. Böylece

$$\frac{\partial e(n)}{\partial w} = -x(n) \quad (2.38)$$

ve

$$\frac{\partial \mathcal{E}(n)}{\partial w} = -x(n)e(n) \quad (2.39)$$

denklemleri ortaya çıkar. En son denklemden gradyan vektörünün tahmin denklemi

$$\hat{g}(n) = -x(n)e(n) \quad (2.40)$$

şeklinde yazılabilir. Endik iniş yöntemindeki (2.33) e göre en küçük ortalama kareler algoritması

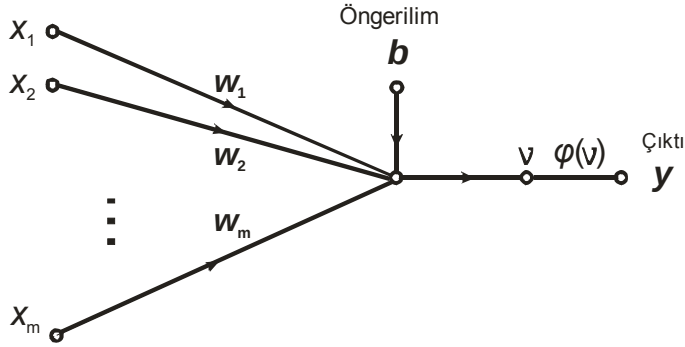
$$\hat{w}(n+1) = \hat{w}(n) + \eta x(n)e(n) \quad (2.41)$$

şeklinde ifade edilir. (2.41) de w lerin yerine $\hat{w}$ olarak kullanılması en küçük ortalama kareler yönteminin endik iniş yöntemi kullanarak ağırlıkların kestirim değerlerini bulmaya yönelik olduğunu göstermek içindir.

2.10.3 Tek katmanlı algılayıcı

Çok katmanlı YSA ları daha iyi anlamak için tek katmanlı algılayıcıları incelemek gerekir. Bu bölümde Rosenblatt ın algılayıcı modeli anlatılacaktır. En küçük ortalama kareler yöntemi doğrusal nöronun etrafında inşa edilmesine rağmen, algılayıcı doğrusal olmayan nöronun etrafında inşa edilir. Şekil 2.30'da ki sinyal akış grafiğine göre

$x_1, x_2, \dots, x_m$ girdi sinyalleri, $w_1, w_2, \dots, w_m$ nöronun sinaptik ağırlıkları, b dışardan uygulanan öngerilim, y ise nöronun çıktı sinyalidir. Algılayıcının görevi dışarıdan uygulanan uyarıcı sinyallerin $x_1, x_2, \dots, x_m$ iki sınıftan birine $\mathfrak{T}_1$ ya da $\mathfrak{T}_2$ e doğru bir şekilde sınıflandırmaktır. Sınıflandırmanın görevi algılayıcı çıktısı y 1 olduğunda girdileri $\mathfrak{T}_1$ e, -1 olduğunda ise $\mathfrak{T}_2$ e atamaktır.

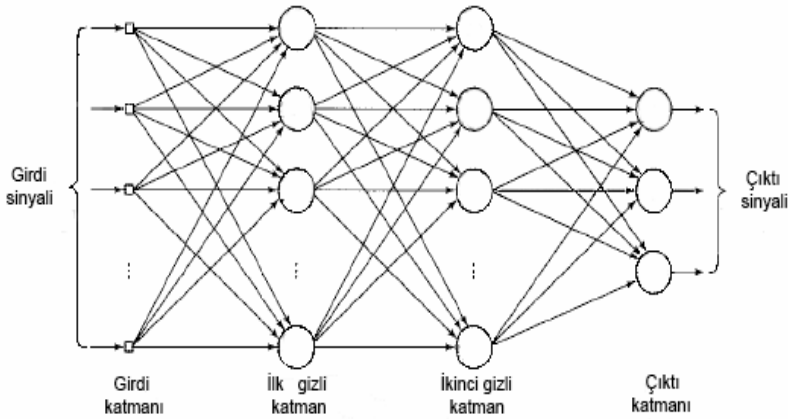


Şekil 2.30 Tek katlı algılayıcının sinyal akış grafiği

2.10.4 Çok katmanlı algılayıcı

Genel olarak ağ girdi katmanı, bir ya da daha fazla olan gizli katmanlardan, ve çıktı katmanından oluşur. Girdi sinyali ağın içerisinde ileri yönde katman katman ilerler. Bu YSA lara çok katmanlı algılayıcılar adı verilir. Oldukça popüler olan hatanın geri yayılımı algoritması ile eğitilmiş çok katmanlı algılayıcılar çok çeşitli problemlere uygulanmıştır.

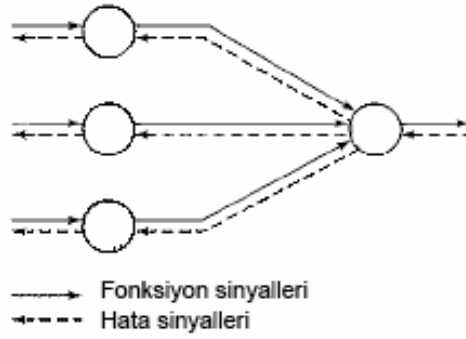
Hata geri yayılım algoritması en küçük ortalama kareler yöntemi temellidir. Temel olarak hatanın geri yayılımında öğrenme, ileri ve geri geçiş olarak adlandırılan iki geçişten oluşmaktadır. İleri geçişte, girdi vektörü ağına girdi katmanına uygulanır ve katman katman ağı içerisinde yayılır. Sonuçta, çıktı ağına gerçek yanıtı olarak alınır. İleri geçiş esnasında tüm sinaptik ağırlıklar sabittir. Geri geçiş esnasında, tüm sinaptik ağırlıklar en küçük ortalama kareler yöntemine uygun olarak ayarlanır. Ağına gerçek yanıtı istenen yanıtı çıkartılarak hata sinyali bulunur ve bu hata sinyali ağı içerisinde geriye doğru katman katman dağıtılır. Bu nedenle bu algoritmaya hatanın geri yayılımı algoritması denmektedir. Örnek bir iki gizli katmana sahip YSA Şekil 2.31’de gösterilmiştir.



Şekil 2.31 İki gizli katmanlı çok katmanlı algılayıcı

2.10.5 Geri yayılım algoritması

Bu bölümde çok katmanlı algılayıcıların eğitiminde kullanılan geriye yayılım algoritmasının adımları anlatılacaktır. Çok katmanlı algılayıcıda Şekil 2.32 de görüldüğü gibi iki temel sinyal akışı vardır. Bu sinyaller fonksiyon ve hata sinyalleridir.



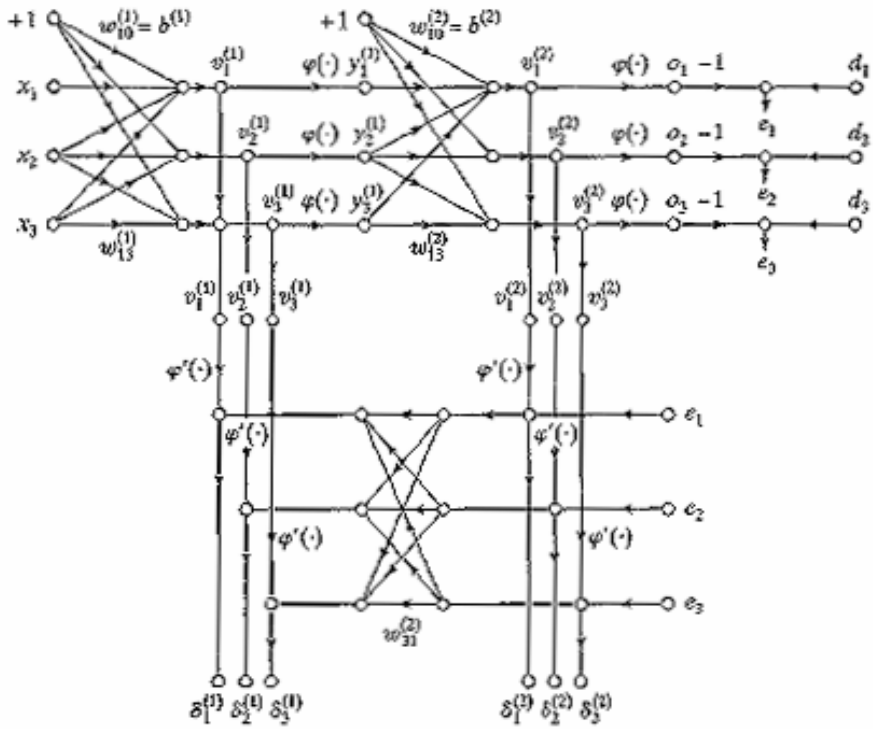
Şekil 2.32 Fonksiyon ve hata sinyalleri(Haykin, 1999)

Fonksiyon sinyali ağın girdi kısmından giren ve ağ boyuca ileri doğru yayılan sinyaldir. Fonksiyon sinyali denmesinin nedeni, fonksiyon sinyalinin ağın içerisinde geçerken her bir nöronda bu nörona uygulanan ağırlıkların ve girdilerin fonksiyonu şeklinde olmasıdır. Hata sinyali ise ağın çıkışında oluşur ve ağın içerisinde katman katman geriye doğru yayılır. Geriye yayılım algoritmasında ağırlıkların ayarlanması için YSA na girdi vektörlerinin tamamının sırasıyla sokulmasına bir devir bu girişlerin sadece bir tanesinin sokulmasına iterasyon denmektedir. Algoritmada kullanılan simgeler(Haykin, 1999);

- i, j ve k indisleri ağdaki farklı nöronları göstermektedir, katmandaki nöron j nöron i nin sağında yer alır, nöron j gizli birim olduğunda nöron k nöron j nin sağında yer almaktadır.
- $\mathcal{E}(n)$ simgesi n . iterasyondaki hata karelerinin toplamı ya da hata enerjisidir. Tüm n lerin üzerinden $\mathcal{E}(n)$ nin ortalaması ortalama enerjiyi $\mathcal{E}_{ort}$ göstermektedir
- $w_{ji}(n)$ simgesi n . iterasyonda nöron i nin çıkışını nöron j nin girişine bağlayan sinaptik ağırlıktır. Bu ağırlığa uygulanan düzeltme ise $\Delta w_{ji}(n)$ ile gösterilmektedir.
- Tüm çıktı vektörünün k . elemanı $o_k(n)$ ile gösterilmektedir.
- m_l simgesi çok katmanlı algılayıcının l katmanındaki boyutu göstermektedir (Düğümlerin sayısı gibi). $l = 0, 1, \dots, L$ ve L ağın derinliğidir. Bundan dolayı m_0 girdi katmanının boyutunu, m_1 ilk

gizli katmanının boyutunu ve m_L çıktı katmanının boyutunu göstermektedir. $m_L = M$ ayrıca kullanılabilir.

Algoritma da kullanılan diğer semboller önceki bölümlerde bahsedilmişti. Öğrenme işleminde hesaplamaları ileri ve geri fazlar şeklinde içeren $L = 2$ ve $m_0 = m_1 = m_2 = 3$ olduğu geri yayılım algoritmasının sinyal akış grafiği Şekil 2.33 de gösterilmektedir.



Şekil 2.33 Geri yayılım algoritmasının sinyal akış grafiği

Şekil 2.33' de sinyal akış grafiğinde üst kısım ileri geçişi, alt kısmı ise geri geçişi göstermektedir. Geri yayılım algoritmasındaki adımlar (Haykin, 1999);

- Başlangıç; Sinaptik ağırlıklara 0 ya da belirli sınırlar içerisinde rastgele değerler atanır
- Eğitim örneklerinin sunulması; Bir devirlik eğitim (girdi vektörlerin hepsi) seti ağı uygulanır. Eğitim setindeki her örnek için 3. ve 4. maddelerde tanımlanan ileri ve geri hesaplamalar sırası ile işletilir.
- İleri hesaplama; Bir devirdeki eğitim seti örneği $(x(n), d(n))$ olsun. $x(n)$ ağı sunulan girdi vektörü, $d(n)$ ise arzu edilen çıktıdır. Ağı tümü boyunca katman katman ilerleyerek fonksiyon sinyalleri ve uyarılmış yerel alan hesaplanır. l katmanındaki nöron j için uyarılmış yerel alan $v_j^{(l)}(n)$

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} w_{ji}^{(l)} y_i^{(l-1)}(n) \quad (2.42)$$

denklemleri ile hesaplanır. Burada $y_i^{(l-1)}(n)$ n . iterasyondaki $l-1$ katmanındaki nöron i nin fonksiyon sinyali çıktısıdır ve $w_{ji}^{(l)}$, $l-1$ katmanındaki nöron i den beslenen l katmanındaki nöron j nin sinaptik ağırlığıdır. $i=0$ için $y_0^{(l-1)}(n) = +1$ ve

$w_{j0}^{(l)} = b_j^{(l)}(n)$ ise l katmanındaki nöron j ye uygulanmaktadır. l katmanındaki nöron j nin çıktısı

$$y_j^{(l)} = \varphi_j(\nu_j(n)) \quad (2.43)$$

dir. Eğer nöron j ilk gizli katmanda ise (örneğin $l=1$)

$$y_j^{(0)}(n) = x_j(n) \quad (2.44)$$

dir. Burada $x_j(n)$ girdi vektörünü $\mathbf{x}(n)$ nin j . elemanıdır. Eğer nöron j çıktı katmanında ise (örneğin $l=L$, burada L ağız derinliğidir.), o zaman

$$y_j^{(L)}(n) = o_j(n) \quad (2.45)$$

dir. Bu adımda son olarak hata sinyali

$$e_j = d_j(n) - o_j(n) \quad (2.46)$$

hesaplanır. Burada $d_j(n)$ arzu edilen $\mathbf{d}(n)$ vektörün j . elemanıdır.

- Geri hesaplama; (2.47) ile tanımlanan yerel gradyanlar

δ hesaplanır.

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n) \varphi_j'(\nu_j^{(L)}(n)) & L \text{ çıktı katmanındaki nöron } j \text{ için} \\ \varphi_j'(\nu_j^{(l)}(n)) \left[\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \right] & l \text{ gizli katmanındaki nöron } j \text{ için} \end{cases} \quad (2.47)$$

Yukarıdaki denklem de $\delta_k = e_k(n) \varphi'_k(\nu_k(n))$ ve $\varphi'_j(\cdot)$ argümanına göre türevi göstermektedir. (2.48) e göre l katmanındaki ağırlık sinaptik ağırlıkları ayarlanır.

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha \left[w_{ji}^{(l)}(n-1) \right] + \eta \delta_j^{(l)} y_i^{(l-1)}(n) \quad (2.48)$$

(2.48) de η öğrenme oranı parametresi ya da adım boyutu ve α ise momentum sabitidir.

- İterasyon; Ağa durma kriterleri karşılanana kadar eğitim setinin yeni devirleri 3. ve 4. maddeler altında ileri ve geri hesaplamalar uygulanmaktadır. Buradaki durma kriteri ortalama hatanın ϵ_{ort} yeterince küçük bir değerden az olup olmadığıdır.

Geri yayılım algoritmasında öğrenme işlemi sıralı ve yığın öğrenme olarak ikiye ayrılmaktadır. Sıralı öğrenme de ağırlıklar eğitim setinin her bir girdi vektörü için güncellenir. Yığın öğrenme de ise eğitim setinin tamamı ağa uygulandıktan sonra yani her bir devirde güncellenir.

3. MATERYAL VE YÖNTEM

TC sivil araç plakalarını bulmak için yapılan çalışmada Intel firmasının açık kaynak bilgisayar görü kütüphanesi (OpenCV, 2006) kullanılarak C/C++ da kodlanmıştır. Geliştirme ortamı olarak Visual Studio 2005 ve kullanıcı ara birim olarak Visual C++ dan yararlanılmıştır. Çalışmada kullanılan görüntüler günün çeşitli saatlerinde otopark ve yollarda duran 3-5 metre arasında çekilen araçlardan elde edilmiştir. Çekim sırasında 4 mega piksellik NIKON E4800 model fotoğraf makinesi kullanılmıştır. Görüntüler 640*480 piksel boyutundadır. Plaka çıkarımı için önerilen yöntem plaka ayrıştırma sistemi, karakter ayrıştırma sistemi ve son olarak da optik karakter tanıma sisteminden oluşmaktadır. Aşağıda bu yöntemler ayrıntılı olarak ele alınmıştır.

3.1 Plaka Ayrıştırma Sistemi

Bu bölümde plaka çıkarımı için kullanılan yöntem (OpenCv, 2006) da karelerin ve dikdörtgenlerin bulunmasında kullanılan yöntemle dayanmaktadır.

3.1.1 Ön işlem

Önişleme de aday plaka bölgelerinin olduğu ya da olabileceği imgelerde gürültülerin kaldırılması amaçlanmıştır. Çalışmada imge de olabilecek gürültüleri kaldırmak için imge piramit aşağı örnekleme ve piramit yukarı örnekleme yöntemleri kullanılmaktadır (Bkz 2.7.2). Şekil 3.1’de görüldüğü gibi imgeye piramit aşağı örnekleme uygulanması

için 5×5 lik bir Gaussian alçak geçirgen filtresi ile filtrelenir ve imgenin çift sayı olan tüm satır ve sütunları imgeden kaldırılır. Ardından bu kez Şekil 3.2’ de görüldüğü gibi piramit aşağı örnekleme sonucunda alınan imge, piramit yukarı örnekleme ye girdi olarak verilir ve tekrar 5×5 lik bir Gaussian alt geçirgen filtresi ile filtrelenir ve aradeğerleme uygulanarak imgenin çift sayı ve sıfır olan tüm satır ve sütunları imgeye eklenir. Gürültüler Gaussian filtresini içerisinde kullanan imge piramitleri sayesinde araç görüntüsünün netliği değişmeden gürültüler kaldırılmıştır.



a)



b)

Şekil 3.1 İmge piramit aşağı örnekleme a)asıl imge b)piramit aşağı örnekleme sonucu



a)



b)

Şekil 3.2 İmge piramit yukarı örnekleme a)asıl imge b)piramit yukarı örnekleme sonucu

Bulunan sonuc yine üç kanallı bir imgedir. Bu KYM imgenin her bir kanalı ayrıştırılarak kırmızı, yeşil ve mavi kanalların her bir için aşağıdaki yöntemler uygulanmaktadır.

3.1.2 Kenarların bulunması ve eşikleme

Bu bölümde ise aday plakaya ait olan kenarları belirlemek için üç kanallı imgenin her bir kanalı için ayrı ayrı Canny kenar belirleme yöntemi(Bkz. 2.8) ve artan değerlerde eşikleme yöntemi kullanılmaktadır. Canny ve eşikleme yönteminin amacı gri seviyedeki (kırmızı, mavi ya da yeşil kanallar) imgelerde aday plaka bölgesine ait kenar bilgisinin ortaya çıkarılmasıdır. Eşikleme ve Canny yöntemlerinin çıktıları ikili seviye imgelerdir ve bu sayede bu imgelerin içerisindeki nesnelerin kontör bilgisinden yararlanılabilmektedir. Önışlemeden gelen imgenin kırmızı kanalı Şekil 3.3 'de ve Canny metodunun çıktısı Şekil 3.4 ' de görölmektedir.

Bu metot da kenarların çıkartılması işleminde dikkat edilmesi gereken nokta Canny metodunda kenarlar çıkartılırken iki eşik değerinden küçük olanın sıfır yapılması ve büyük olan eşik değerinin düşük bir değere sahip olmasıdır. Metotta kullanılan büyük eşik değeri 50 dir. İki eşik değerinden düşük olanın 0 yapılması bulunan kenarların birleştirilmesiyle sonuçlanır. Bu da köşelerin daha belirgin hale gelmesini neden olmaktadır.

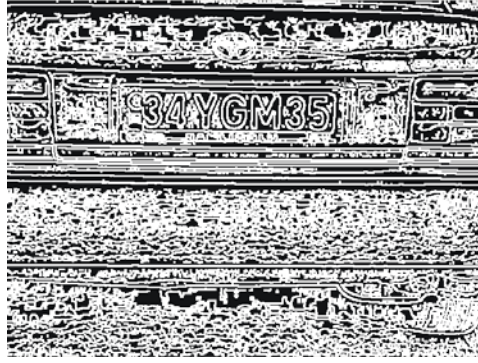


Şekil 3.3 İmgenin kırmızı kanalı



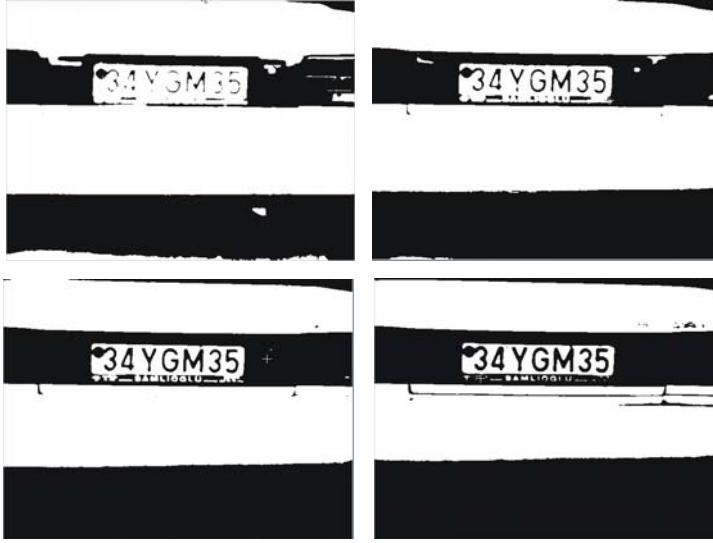
Şekil 3.4 Canny metodu çıktısı

Canny metodu ile gri seviye imgeden kenarlar çıkarıldıktan sonra Şekil 3.5 den görüldüğü gibi siyah zemin üzerinde beyaz pikseller genişletilerek (Bkz. 2.9.3) plaka kenarlarında olası boşluklar doldurulmaktadır.



Şekil 3.5 Genişletilmiş ikili seviye imge

Canny metodu uygulandıktan sonra tekrar baştan aynı kanala artan değerlerde eşikleme yöntemi uygulanmaktadır (Şekil 3.6). Yöntemde her bir eşikleme değeri 25'er aralıklarla 255'e kadar artırılmaktadır. Bu sayede değişik ışık koşullarındaki plaka görüntüleri ortaya çıkarılmakta ve imgedeki olası aday plaka bölgesi sayısı artırılmaktadır.



Şekil 3.6 Artan değerlerde eşikleme çıktıları

3.1.3 Köşelerin bulunması ve biçimlendirilmesi

İmge deki köşelerin bulunması işlemi Canny kenar bulma ve eşikleme yöntemlerinin çıktısı üzerinde kurulmaktadır. Bu yöntemlerden gelen çıktılar (S. Suzuki and K. Abe, 1985) de belirtilen bağlı bileşenler yöntemiyle bir listeye aktarılmaktadır. Bağlı bileşenler yöntemi siyah zemin üzerinde beyaz nesneleri bulmak üzerine kuruludur. Kullanılan bağlı bileşenler yönteminde dört tane algoritma vardır. Çalışmada bunlardan ikincisi kullanılmıştır. İkinci algoritma sekiz komşuluğundaki ikili seviyedeki tüm kontörleri yani dış kontörler ve bunların içerisindeki kontörleri bulmaktadır. Bulunan kontörler zincir kodunun basitleştirilmesiyle listeye aktarılır. Basitleştirme işlemi zincir kodunun çok kenarlılarla temsil edilmesi esasına dayanır. Bu işlemde zincir kodunun yatay, dikey ve köşegen parçalarına bunlara ait son noktalar

koyulmaktadır. Kontörler listeye aktarıldıktan sonra çok kenarlılar üzerinde basitleştirme yöntemi olan Douglas-Peucker algoritması (Bkz 2.9.4.2) ile tekrar bir basitleştirme işlemi uygulanmaktadır. Bu işlemde de çok kenarlı olarak basitleştirilen zincir kodunun temsil ettiği her bir kontör, kontör çevresinin %2 lik bir tolerans değerine göre basitleştirilmesi ile sonuçlanır. Örnek olarak Şekil 3.7 'de girdi görüntüsünün kırmızı kanalı için 139 değerine göre eşikleme sonucunda imgenin ikili seviye çıktısı görülmektedir.

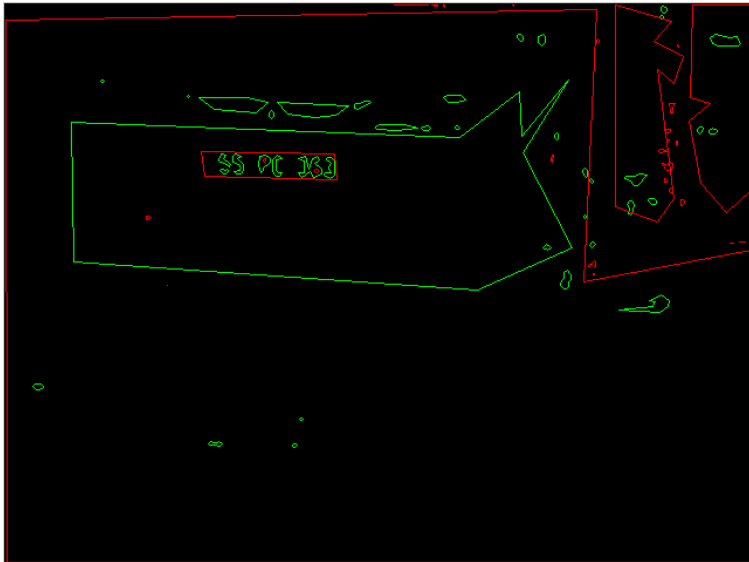


Şekil 3.7 Gri seviye(Kırmızı kanal)

Şekil 3.8’de Şekil 3.7’de gösterilen imgenin kontörleri bulunduğundan sonraki basitleştirilmiş durumu görülmektedir. Şekil 3.8’de kırmızı ile gösterilen doğrular dış kontörlere yeşil ile gösterilenler ise iç kontörlere karşılık gelmektedir. Şekil 3.9’da ise Şekil 3.8’de bulunan kontörlerin Douglas-Peucker algoritması ile basitleştirilmiş hali görülmektedir.



Şekil 3.8 Çok kenarlı basitleştirmeden önceki iç ve dış kontörler



Şekil 3.9 Çok kenarlı basitleştirmeden sonraki iç ve dış kontörler

Douglas-Peucker algoritmasıyla basitleştirme işleminden sonra bu doğruların köşeleri artık listededir. Bulunan köşelerin bir dörtgene ait olup olmadığı, kapladığı alanın 1000 pikselden fazla olup olmadığı ve çokgen olup olmadığına bakılır. Eğer bir dörtgen ise bulunan köşelerin sayısı 4 ve bu köşelerin sırasıyla birbirine 90° lik bir açı ile dik olması gerekmektedir. Köşe çiftlerinin arasındaki açının dik açı olup olmadığı kosinüs teoremi ile bulunur. Kullanılan açı gürültülerden oluşabilecek sapmaları hesaba katmak için tam 90° değil 90° ye yakın açılar alınmıştır. Bu işlemten sonra dörtgeni ya da dörtgeni oluşturan köşeleri çevreleyen bir dikdörtgen oluşturulmaktadır. Oluşturulan dikdörtgenlerin aşağıdaki özelliklere sahip olanları karakter ayırıştırma sistemine sokulmaktadır.

- eni ve boyu 400 pikselden az
- eni ve boyu 20 pikselden fazla
- en ve boy oranı 2 ile 7 arasında olanlar

aday plaka bölgesi olarak sınıflandırılırlar.

3.2 Karakter Ayırıştırma Sistemi

Karakterter ayırıştırma sistemine giren aday plaka bölgelerine gürültüyü kaldırmak için önışlem uygulanmakta ve önışleminden sonra yatay ve düşey taramaya sokulmaktadır.

3.2.1 Ön işlemler

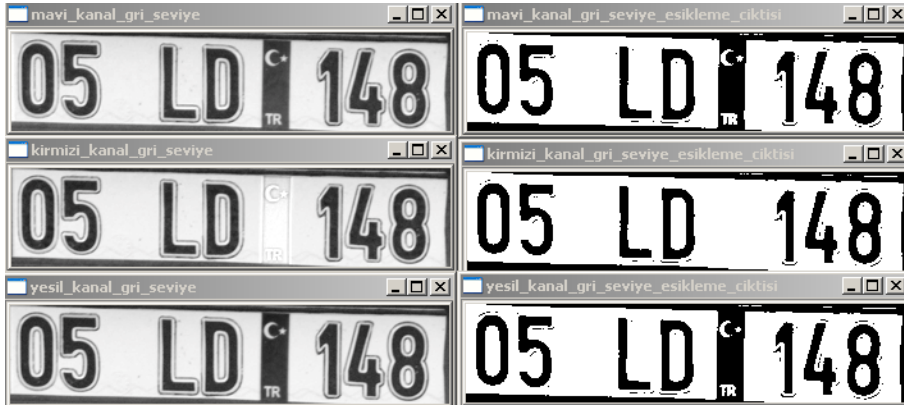
Bulunan aday plaka bölgeleri KYM formatında karakter ayırıştırma sistemine girdi olarak verilir. Aday plaka bölgesinin KYM formatında girdi olarak verilmesinin nedeni önışlemlerde bölgenin siyah ve beyaz olmayan tüm kısımlarının aday bölgeden kaldırılmak istenmesidir. Veritabanında, ışıklı ortamdaki imgeler için çoğu gürültünün renkli olduğu görülmüştür. Karakter ayırıştırma da bu gürültülerden bazılarının karakter olarak algılanma şansı vardır. Bundan dolayı basit ama etkili bir yöntemle üç kanalda da artan değerlerle eşiklemeler yapılmaktadır. Böylece plaka üzerine yapıştırılan renkli etiketlerin ya da zaten plaka üzerinde var olan sol kısımdaki mavi bölge kaldırılmış olur. Birinci eşikleme için bu ön işlemlerde her bir kanalın piksel değerlerine bakılır. Şekil 3.10 da yöntemin daha kolay anlaşılabilmesi için KYM imgesindeki 5 pikselin değeri verilmektedir. Şekilde piksellere ait kanalların birinci eşiklemeden önce ve sonraki değerleri eski ve yeni şeklinde gösterilmektedir. Örneğin sarı renge yakın 1.pikselin 130 değerine göre birinci eşiklenmeden önceki kırmızı kanalı 243, yeşil kanalı 249 ve mavi kanalı 14 değerine sahiptir. Birinci eşiklenmeden sonraki değerleri sırasıyla 255, 255, ve 0 dır. Şekilde diğer piksel değerlerinin bu şekilde oluşturulduğu görülebilir. Bu şekilde her piksel

Aşağıda bu yöntemin plaka üzerinde karakter olmayan renkli bir gürültü üzerinde nasıl uygulandığı görülmektedir. Şekil 3.11 de önışleme giren girdi imgesi görülmektedir.



Şekil 3.11 Girdi imgesi

Şekil 3.12 de girdi imgesinin 80 değerine göre birinci eşikleme çıktıları sırasıyla karşılıklı olarak gösterilmektedir.



Şekil 3.12 Kanallar ve eşikleme çıktıları



Şekil 3.13 VEYAlama ve medyan çıktısı

Şekil 3.13 de kanalların VEYAlanması sonucu plaka üzerinde tuz – biber etkisi olarak adlandırılan gürültüler ortaya çıkmıştır. Bu da yine Şekil 3.13’den görüleceği gibi medyan filtresi kullanılarak ortadan kaldırılmaktadır. Burada her defasında eşiklemelerin sonucu yatay ve düşey taramaya sokulmakta ve ardından optik karakter tanıma sistemine gönderilmektedir. OKTS de tanınan karakter katarları karakter katarı dizisine aktarılmakta ve en son olarak bu karakter katarı dizilerinden en iyisi plaka sonucu olarak seçilmektedir.

3.2.2 Yatay ve dişey tarama

Önişlemden gelen aday plaka bölgelerinden karakterlerin ayrıştırılması için yatay ve dişey tarama algoritmaları uygulanmaktadır.

3.2.2.1 Yatay tarama

Yatay tarama da siyah-beyaz olarak algoritmaya giren aday plaka bölgelerine normalizasyon işlemleri uygulanmaktadır (Zhang and Zhang, 2003) Normalizasyon işleminde yatay taramaya giren tüm aday plaka bölgeleri süper örnekleme yöntemiyle 160*40 (40 satır 160 sütun) boyutuna getirilmektedir. Bu sayede yatay tarama da kullanılan algoritma tüm aday plaka bölgeleri için uygulanabilmektedir. Bu işlemlerde plakanın ortasından hayali yatay bir çizgi alınır ve yaklaşık olarak tüm karakterlerden geçen bir çizgiye sahip olunur (Oral M. and Çelik U.,2003).



Şekil 3.14 Yatay tarama

Aday plakası üzerindeki hayali çizgi 4 piksel yukarı ve aşağı kaydırılarak ortalama bir değeri elde edilir(Şekil 3.14). Burada tek bir değeri değil de ortalama bir değeri kullanılması nedeni aday plaka bölgesinin ortasından alınan hayali çizginin her zaman orta noktaya denk gelmemesidir. Şöyle ki; Şekil 3.15 dan görüldüğü gibi 3 rakamı için 1 doğrusunun geçtiği çizgideki toplam piksel değeri 2 den geçen çizgideki toplam piksel değeriyle aynı olmamaktadır. Alınan ortalama değeri

algoritmada kullanılan sabit değer böylece uyumlu hale getirilmiş olur. Böylece daha gürbüz bir işlem gerçekleştirilmiş olur.



Şekil 3.15 Yatay tarama da ortalama değer

Artık aday plaka bölgesinin net bir ortalama değeri vardır. Bu ortalama değerden yararlanılarak karakterlerin boyu bulunabilir. Karakterlerin boyunu bulmak için, alınan çizgi önce plaka da yukarı doğru kaydırılır ne zaman ki bu ortalama değerden 6000 piksel fazla bir değerle karşılaşırsa o anki koordinatlar saklanır. Aynı şekilde bu seferde bu çizgi aşağıya doğru kaydırılır ve ne zaman ki bu ortalama değerden 6000 piksel fazla bir değerle karşılaşırsa tarama kesilir ve koordinatlar kaydedilir. Eğer tarama işlemi plakanın sınır noktalarına kadar işlerse o zamanda sınır noktalar alınan koordinatlara dahil edilmektedir. Kullanılan 6000 piksel değeri çalışmalar sonucunda ortaya çıkan ortalama bir değerdir. Yatay tarama sonucunda artık karakterlerin boyunu içeren aday plaka bölgesi dikey taramaya gönderilmektedir. (Şekil 3.16).



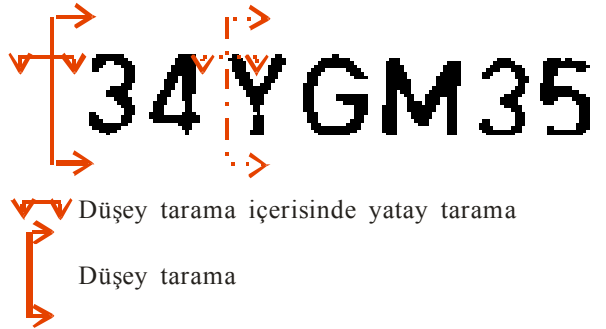
Şekil 3.16 Yatay tarama sonucu

3.2.2.2 Düşey tarama

Düşey tarama da yatay tarama sonucu alınan aday plaka bölgesi üzerinde işlem yapılmaktadır. Bu yöntemde plakanın sol tarafından başlanarak düşeydeki piksel değerlerinin değişimi sayesinde her bir karaktere ait boy bilgisi bulunmaktadır. Bu aşamada önceden tanımlanan kriterlere göre plakadaki karakter olabilecek gürültülerde kaldırılmış olur. Düşey tarama işleminde bir aday plaka üzerindeki karakterlerin sistem tarafından kabul görmesi için o karakterin eni 30 pikselden az 5 piksel den fazla olması, boyununda yatay tarama sonucu algoritmaya giren bölgenin yüksekliğinin yarısından fazla olması gereklidir. Karakterlerin ayrıştırılması işleminde olabilecek gürültülerden daha az etkilenmek için aday plaka bölgesinin solundan 2 piksel içerisinden taramaya başlanır. Tarama işleminde düşey hayali bir çizgi alınır. Bu hayali çizginin geçtiği her sutunda yukarıdan aşağıya gidilerek satır değerine bakılır. Taranan pikselin değeri 0(siyah) ise o anki koordinat aday karakterin başlangıç noktası olarak kaydedilir. Bundan sonraki sütunlarda 255(beyaz) pikseli aranır ve ne zamanki 255 le karşılaşılırsa o anki koordinat aday karakterin son noktası olarak kaydedilir. Ya da aranan değerin 0 dan 255 e geçtiği görülüyorsa gidilen son nokta karakterin son noktası olarak kaydedilir. Bu esnada bulunan aday karakterin eninin kriterlere uyup uymadığı kontrol edilir. Uymayanlar sisteme dahil edilmez. Tarama tüm aday karakterler için yapıldıktan sonra bu seferde her bir aday karakter için bilinen başlangıç ve son noktalardan başlayarak yatay tarama yapılır. Yatay tarama da düşey tarama için anlatılanlar geçerlidir. Yatay tarama esnasında aday

karakterin boyunun kritere uyup uymadığı kontrol edilir. Uymayanlar sisteme dahil edilmemektedir. Düşey tarama ve düşey tarama içerisinde yapılan yatay taramadan sonra her bir karakterin en ve boy özellikleri sistemde saklanmaktadır.

Düşey taramanın ardından karakterlerin sayısı sınılanmakta ve sayısı 7 ya da 8 olmayan bölgeler optik karakter tanıma sistemine gönderilmemektedir. Düşey tarama sonucu Şekil 3.17 deki imgenin ayrıştırılmış hali Şekil 3.18’de görülmektedir.



Şekil 3.17 Düşey tarama



Şekil 3.18 Karakter ayrıştırma sonucu

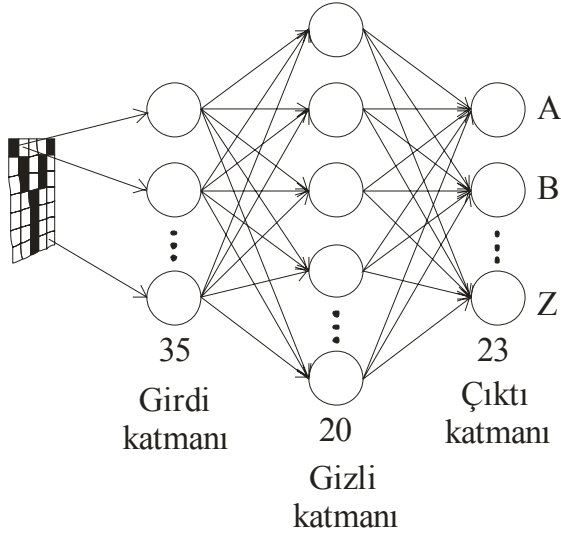
Ayrıştırılan karakterler süper örnekleme yöntemiyle 7*5 lik imge boyutlarına getirilir ve OTSU metoduyla eşiklenir. Buradan da optik karakter tanıma sistemine girdi olarak verilir.

3.3 Optik Karakter Tanıma Sistemi

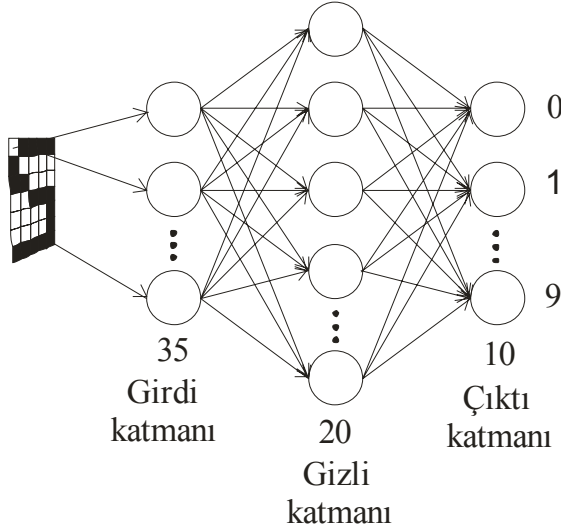
Sistemin bu bölümünde geri yayılım algoritmasıyla eğitilmiş ileri beslemeli çok katmanlı YSA'ları kullanılmaktadır. TC sivil araç plakalarının rakamları için rakam YSA ve harfleri için harf YSA olmak üzere iki tane YSA oluşturulmuştur. Farklı ışık koşulları için harf YSA 8831 eğitim setiyle, rakam YSA ise 23075 eğitim setiyle eğitilmiştir. Eğitim seti gerçek görüntülerden alınan karakterlerden oluşmaktadır. Eğitim esnasında geri yayılım algoritmasını için kullanılan parametreler devir sayısı 1000, momentum 0.1 ve her nörondaki signum fonksiyonun a sı 1 şeklindedir. Geri yayılım algoritmasında ağırlıklar sıralı yöntemle güncellenmektedir. (Bkz 2.10).

Harf YSA 35 giriş, 20 gizli nöron ve 23 çıkış nöronuna(Şekil 3.19), rakam YSA ise 35 giriş, 20 gizli nöron ve 10 çıkış nöronuna sahiptir(Şekil 3.20). Hangi karakterin hangi yapay sinir ağına gideceği ise karakterlerin dizilişleri belirler. Tek satırlı TC sivil plakaları için Çizelge 1.1 den görülebileceği gibi aday plaka bölgesinin 7 karakterli olanları için ilk 2 ve son 2 karakteri her zaman rakam ve 3. karakteri her zaman harf olduğu için sırasıyla bu karakterler rakam ve harf YSA larına gönderilir. 4. ve 5. sıradaki karakterler ise hem harf hem rakam yapay sinir ağına gönderilerek yapay sinir ağlarından alınan sonuca göre hangisinin değeri daha fazlaysa o yapay sinir ağı için sınıflandırılır. Aday

plaka bölgelerinin 8 karakterli olanlarının ise ilk 2 ve son 3 karakteri her zaman rakam, 3. ve 4. sü her zaman harf olduğu için sırasıyla bu karakterlerde rakam ve harf YSA na gönderilmektedir. 5. karakter ise her iki YSA na gönderilerek alınan sonuca göre sınıflandırılmaktadır.



Şekil 3.19 35-20-23 topolojiye sahip harf YSA



Şekil 3.20 35-20-10 topolojiye sahip rakam YSA

Sınıflandırılan karakterlerin ASCII karşılığı oluşturularak karakter katarı dizisine aktarılmaktadır. Buraya kadar olan yöntemlerin sonucunda oluşturulan karakter katarı dizisinde elman sayısı ikiden fazlaysa bu karakter katarlarının en iyisi seçilmektedir. En iyisini seçmek için 7 ve 8 karaktere sahip olan katarlardan hangisi karakter katarı dizisinde daha fazla ise o elemanlara bakılır. Örneğin 8 elemanlı olan katarlar daha fazlaysa sadece 8 karaktere sahip olanlara bakılır. Seçme işleminde karakter katarlarının ilk elemanlarından başlanarak son elemanlarına kadar her bir karakter katarının sütun hizasında ilgili karakterden hangisinden daha fazla ise o karakter sonuca yazılmaktadır. Örneğin 55PC232 , 55PC233 ve 53PD233 aday plaka bölgeleri sistemin sonunda bulunsun. Aday bölgelerde karakter katarlarının ilk sütununda 5, ikinci sütununda 5, üçüncü sütununda P, dördüncü sütununda C, beşinci sütununda 2, altıncı sütununda 3 ve yedinci sütununda 3 karakterleri baskındır. Bundan dolayı sonuç 55PC233 olarak sistemin çıktısı oluşturulmaktadır.

4. TARTIŞMA VE SONUÇ

Tasarlanan sistemde günün farklı saatlerinde farklı mesafelerden çekilen araç görüntüleri kullanılmıştır. Çalışma Intel® Pentium M 1.6GHz işlemci 512Mb belleğe sahip bir dizüstü bilgisayarda test edilmiştir. Girdi olarak verilen bir imge için ortalama işlem süresi 0.492 saniyedir. Boyutları 640*480 olan 145 araç imgesinden 126 tanesinin plakası tanınmıştır. Çizelge 4.1’den görüleceği gibi PAS bölümünde bu 145 araç imgesinden 140 tanesinde plakanın yeri bulunmuştur. KAS bölümünde 1033 karakterden 998 tanesi ayrıştırılmıştır. OKTS bölümünde ise 1033 karakterden 984 si tanınmıştır. Sistemin bölümlerinden bağımsız tüm başarısına baktığımızda %86.89 lik bir başarı elde edilmiştir. PAS da karşılaşılan zorluklardan en önemlisi ortam ışığının çok fazla ya da çok az olması olarak gösterilebilir. Işığın değişken olması PAS daki eşikleme ve kenar bulma yöntemi için kenar bilgisini azaltmaktadır. PAS için kenar bulma yöntemine sokulan parametrenin 50 den fazla bir değerle ayarlanması da aday plaka bölge sayısını artırmasına rağmen işlenen zamanında ayrıca artırmıştır.

KAS bölümünde karşılaşılan zorluklardan bir tanesi karakter fontlarının çok büyük olduğu için karakterlerin birbirlerine çok yakın olmasıdır,diğer bir zorluk ise gürültü olarak adlandırabileceğimiz renkli etiketlerin karakterlerin üzerine yapıştırılmasıdır. Bu, üç kanalda eşikleme yaparken kısmen karakter bilgisini yok etmektedir.

OKTS de karşılaşılan zorluklar ise 0-D,8-B gibi karakterlerin birbirine çok benzemesidir ve bu bir nebze kullanılan iki tane YSA ile

çözülse bile kullanılan YSA na girdi olarak verilen karakter boyutlarında artırımı gidilerek bir iyileştirme yapılabilir. Fakat bu da PAS sonucu dönen bölgeler için kullanılan iki eşik değerinin iterasyon değerleri yüzünden çok zaman almaktadır.

Çizelge 4.1 Deneysel sonuçlar

	Sayı	Yüzde
PAS	140 / 145	%96.55
KAS	998 / 1033	%96.61
OKTS	984 / 1033	%95.25

Sistemde kullanılan OKTS sistemdeki diğer bölümlerin (PAS, KAS) çıktısıyla doğrudan ilgilidir. PAS bölümünde ya da KAS bölümündeki başarısızlık direk olarak sistemin başarısını etkilemektedir.

Tasarlanan sistem literatürdeki benzer çalışmalarla subjektif olarak karşılaştırıldığında her bir sistemin performansı Çizelge 4.2 den görülebilmektedir.

PAS	OKTS	Platform/İşlemci	Zaman(s)	İmge
%96.55	%95.25	C/C++ Pentium M 1.6GHz	0.492	145 [1]
%97.6	%95.6	PIV 1.6GHz	~2.5	1065 [2]
%100	%92.5	Matlab6 PII 300MHz	4	40 [3]
%98.8	%95.2	Visual C++ PIV 1.4	0.65(plaka) 0.1(karakter başına)	? [4]

Çizelge 4.2 Tasarlanan sistemle literatürdeki benzer sistemlerin karşılaştırılması [1] Tasarlanan sistem, [2] (S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, 2004), [3] (Anagnostopoulos C., Kayafas E., and Loumos. V., 2000), [4] (Duan T. D., Hong Du T. L., Phuoc T. V., and Hoang N. V., 2005)

KAYNAKLAR

Abdullah S. N. H. S., Khalid M., Yusof R. and Omar K., April 2006, License plate recognition using multi-cluster and multilayer neural networks, Information and Communication Technologies ICTTA '06. 2nd, vol.1, pp.1818 - 1823

Ahmed M J., Sarfraz M., Zidouri A. and Al-Khatib W. G., December 2003, License plate recognition system, Proceedings of the 2003 10th IEEE International Conference, vol 2, pp.898 - 901

Anagnostopoulos C. N. E., Anagnostopoulos I. E., Loumos V., Kayafas E., September 2006, A license plate-recognition algorithm for intelligent transportation system applications, Intelligent Transportation Systems, IEEE Transactions, vol. 7, no. 3, pp.377-392

Anagnostopoulos C., Kayafas E. and Loumos. V., 2000, Digital image processing and neural networks for vehicle license plate identification, Journal of Electrical Engineering, vol. 1, no. 2, pp. 2-7

Course Handouts, 2004, Signal Analysis and Machine Perceptron Laboratory <http://sampl.ece.ohio-state.edu/EE86>

KAYNAKLAR (DEVAM)

Du K. L. and Swamy M.N.S., 2006, Neural Networks In A Softcomputing Framework, p.609

Duan T. D., Hong Du T. L., Phuoc T. V., and Hoang N. V., February 2005, Building an automatic vehicle license plate recognition system, Intl. Conf. in Computer Science RIVF, pp. 59–63.

Freeman J. A. and Skapura D. M., 1991, Neural Networks Algorithms, Applications, and Programming Techniques, p.444

Gonzalez, R. C. and Woods, R. E. , 2002, Digital Image Processing, p.888

Gonzalez, R. C. , Woods, R. E. and Eddins, L. S., 2004, Digital Image Processing Using Matlab, p.624

Haykin, S., 1999, Neural Networks A Comprehensive Foundation, 875p

Hershberger J. and Snoeyink J., 1992, Speeding Up The Douglas-Peucker Line Simplification Algorithm, Technical Report, www.cs.ubc.ca/cgi-bin/tr/1992/TR-92-07.ps

Jähne, B., 2002, Digital Image Processing, p.598

Jain A. K., Mao J. and Mohiuddin K. M., March 1996, Artificial neural networks:a tutorial, Computer, vol. 29 issue 3, pp. 31-44

KAYNAKLAR (DEVAM)

Ko M.A. and Kim Y.M., October 2003, License plate surveillance system using weighted template matching, Applied Imagery Pattern Recognition Workshop, pp. 269-274

MEB müfredatına uygun sürücü eğitim rehberi, 2007, 512s.

OpenCv, Open Source Computer Vision Library Reference Manual, 2001, p.436
<http://sourceforge.net/projects/opencvlibrary>

OpenCv, Intel® Image Processing Library Reference Manual, 2000,p.319
<http://sourceforge.net/projects/opencvlibrary>

OpenCv, Intel® Open Source Computer Vision Library, 2006,
<http://sourceforge.net/projects/opencvlibrary>

Oral M. and Çelik U.: September 2003, Motorlu araç plaka görüntülerinden karakter ayrıştırma ve tanıma, Proceedings of International Conference on Signal Processing, vol.1, no:2

Paliyl I., Turchenko V., Koval V., Sachenkol A. and Markowsky G., July 2004, Approach to recognition of license plate numbers using neural networks, Neural Networks Proceedings IEEE International Joint Conference, vol.4, pp.2965 - 2970

KAYNAKLAR (DEVAM)

Pratt W. K., 2001, Digital Image Processing, 738p

S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, March 2004, Automatic license plate recognition, Intelligent Transportation System, vol. 5, no. 1, pp. 42–53

Qin Z., Shi S., Xu J. and Fu H., June 2006, Method of license plate location based on corner feature, Proceedings of the 6th World Congress on Intelligent Control and Automation, pp. 8645 - 8649

Sunday D., 2007, Polyline simplification

http://geometryalgorithms.com/Archive/algorithm_0205/algorithm_0205.htm

S. Suzuki and K. Abe., April 1985, Topological Structural Analysis of Digital Binary Images by Border Following., CVGIP, vol..30, no.1., pp. 32-46

Wu C., On L. C., Weng C. H., Kuan T. S. and Ng K., August 2005, A macao license plate recognition system, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4506 - 4510

KAYNAKLAR (DEVAM)

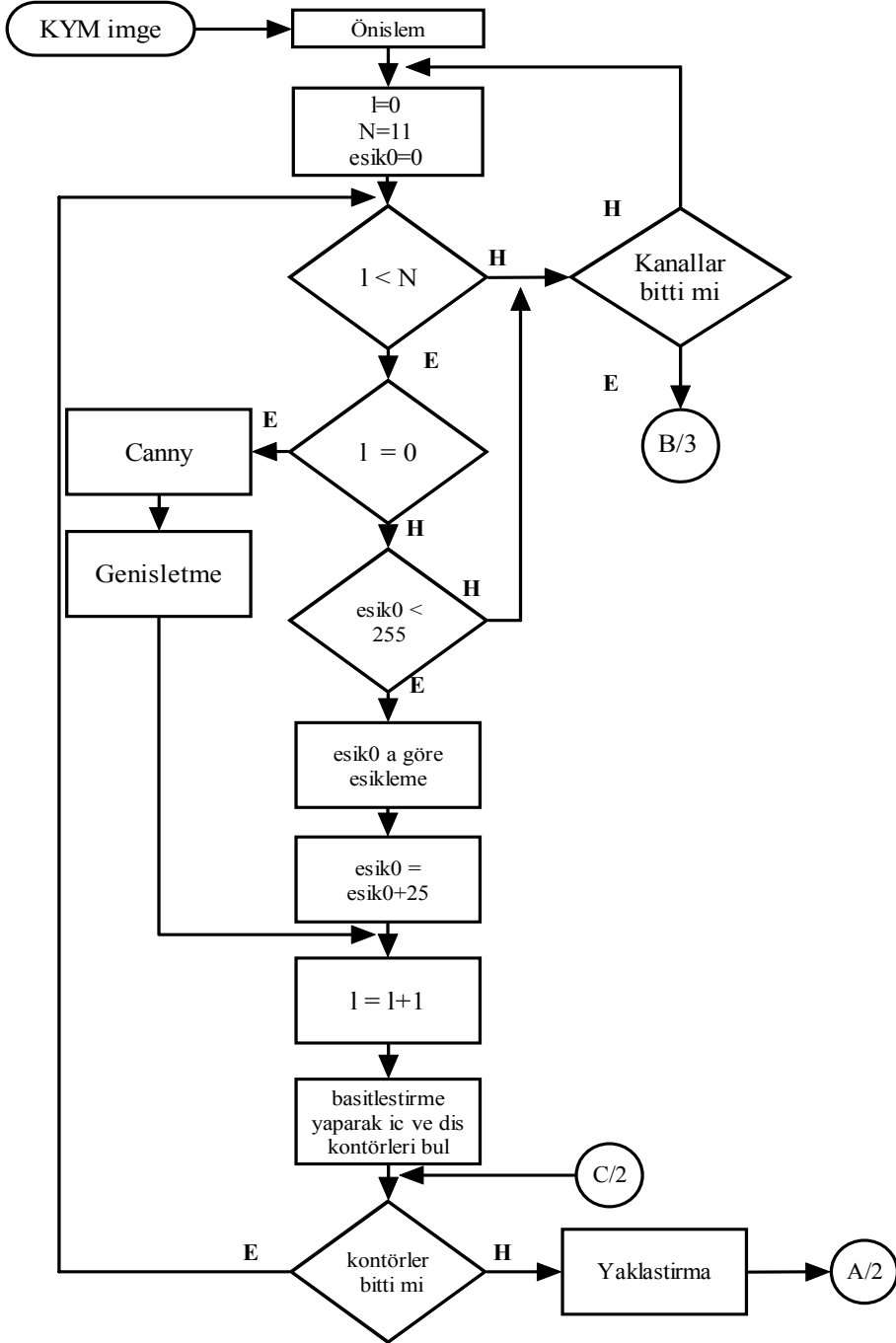
Xu J.-F., Li S.-F. and Yu M.-S., August 2004, Car license plate extraction using color and edge information, Proceedings of the Third International Conference on Machine Learning and Cybernetics, vol.6, pp.3904 - 3907

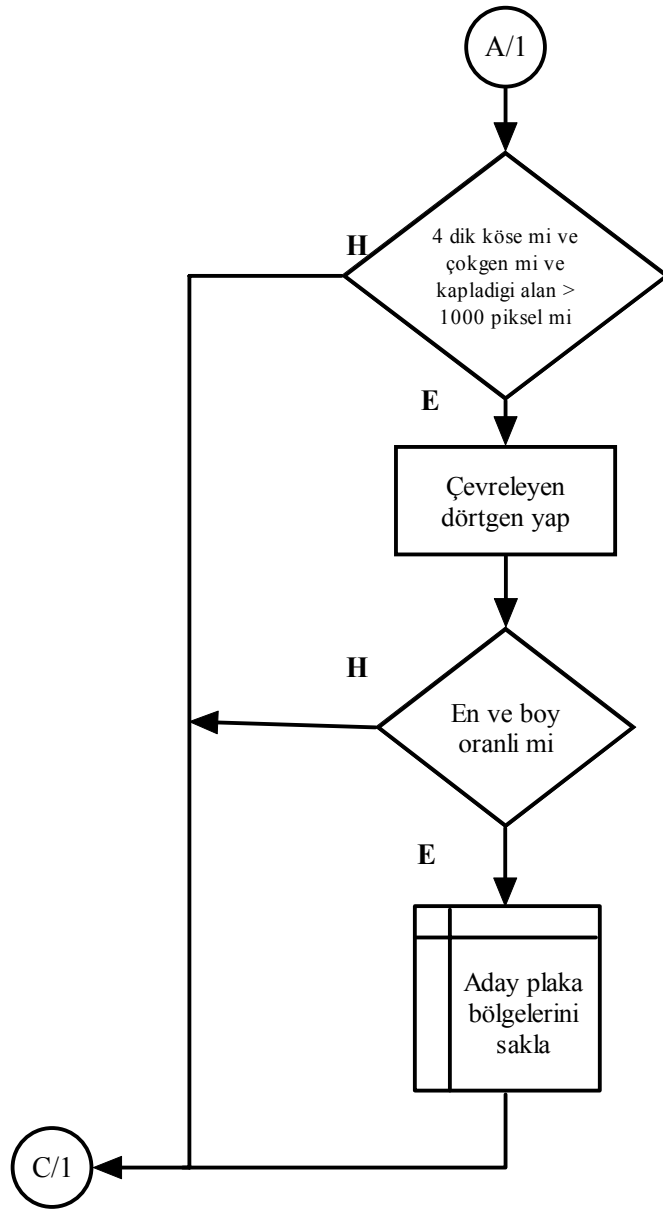
Yanamura Y., Goto M., Nishiyama D., Soga M., Nakatani H. and Saji H., June 2003, Extraction and tracking of the license plate using hough transform and voted block matching, Intelligent Vehicles Symposium Proceedings. IEEE pp. 243 - 246

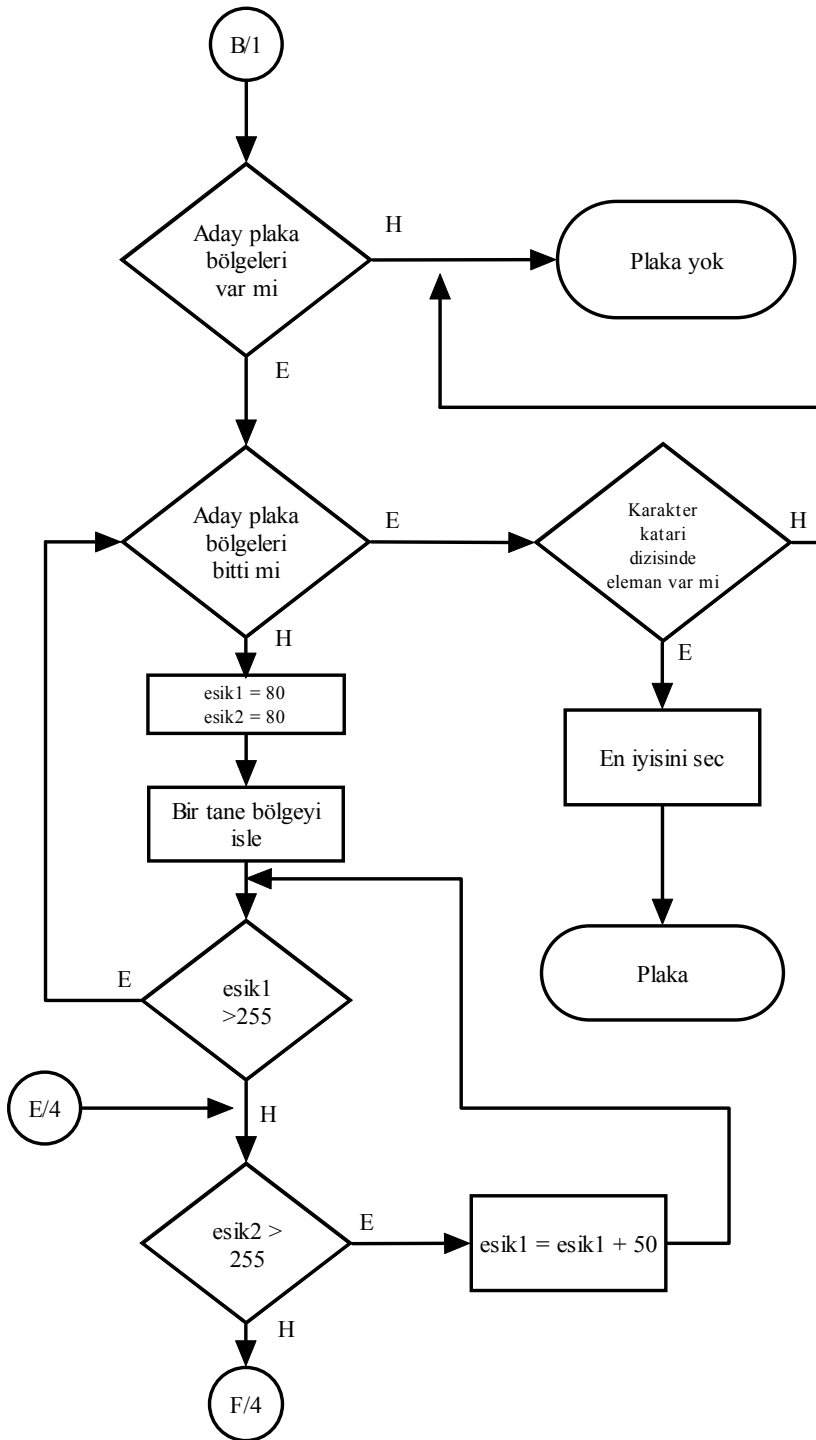
Yohimori S., Mitsukura Y., Fukumi M., Akamatsu N. and Pedrycz N., June 2004, License plate detection system by using threshold function and improved template matching method, Fuzzy Information IEEE, vol.1, pp.357 - 362

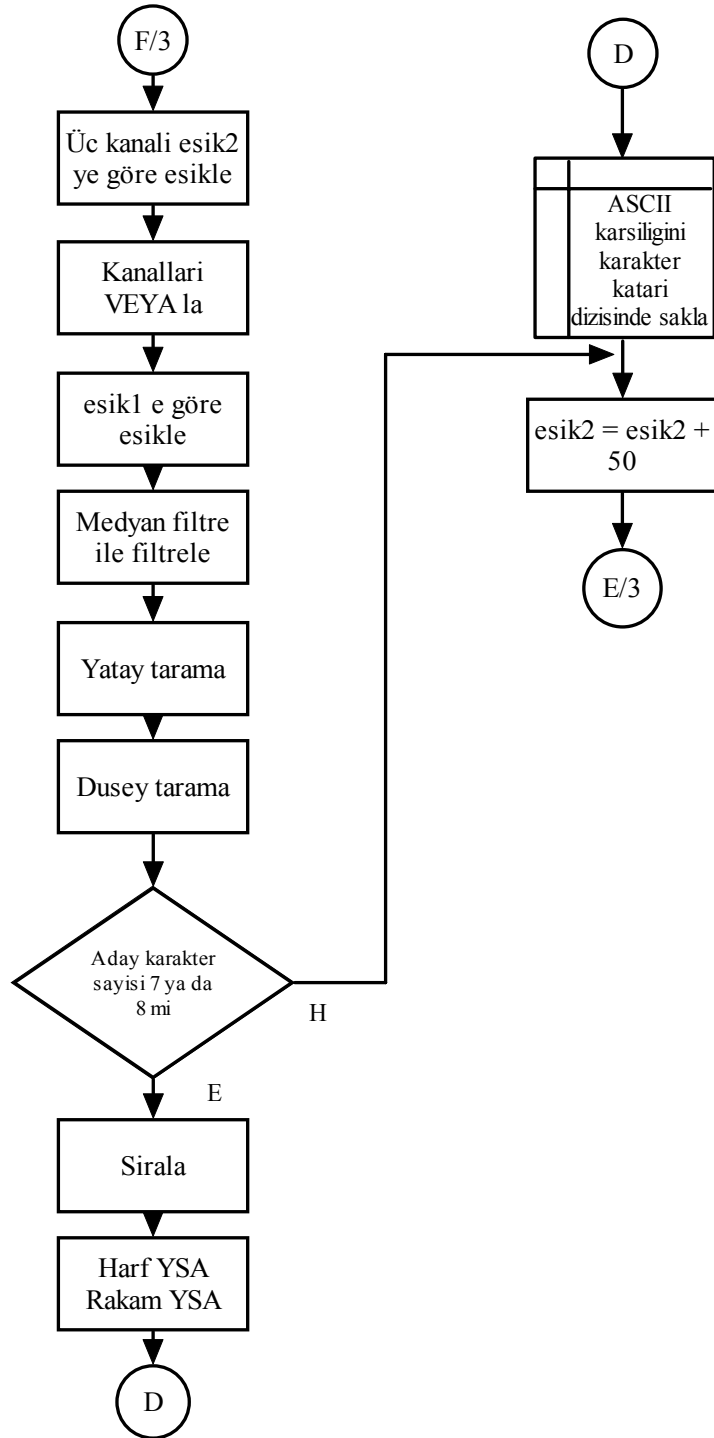
Zhang Y. and Zhang C., June 2003, A new algorithm for character segmentation of license plate, Intelligent Vehicles Symposium Proceedings IEEE, pp. 106 – 109

Ek 1 Akış Diagramı









ÖZGEÇMİŞ

Adı Soyadı : İsmail IRMAKÇI

Doğum Yeri : Offenburg / Almanya

Doğum Tarihi : 19.12.1981

Medeni Hali : Bekar

Yabancı Dili : İngilizce

Eğitim Durumu :

2006 – 2008 Ege Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği, Ort: 87

2005 - 2006 Bilimsel Hazırlık(Yüksek lisans)

2000 - 2005 Dokuz Eylül Üniversitesi Elektrik Elektronik Mühendisliği, Ort: 2.41 (74,1)

1996 - 1999 Merzifon Lisesi, Ort: 3,5

İş tecrübeleri :

Kentkart Ege Elektronik, İzmir, Şubat 2007 - Ağustos 2007

TAGEM, İzmir, Kasım 2006 - Şubat 2007

ERMAŞ, İzmir, Kasım 2005 - Temmuz 2006

