

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

**SÜRÜ ZEKASI YAKLAŞIMLARININ RENKLİ
GÖRÜNTÜ KESİMLEMeye UYARLANMASI VE
TANIMA SİSTEMLERİ ÜZERİNDE
GERÇEKLEŞTİRİMİ**

Doğın AYDIN

Tez Danışmanı : Doç. Dr. Aybars UĞUR

Bilgisayar Mühendisliğı Anabilim Dalı

Bilim Dalı Kodu : 619.01.00

Sunuş Tarihi : 11.02.2011

Bornova-İZMİR

2011

DOĞAN AYDIN tarafından Doktora tezi olarak sunulan “SÜRÜ ZEKASI YAKLAŞIMLARININ RENKLİ GÖRÜNTÜ KESİMLEMeye UYARLANMASI VE TANIMA SİSTEMLERİ ÜZERİNDE GERÇEKLEŞTİRİMİ” 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 11.02.2011 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı	: Doç. Dr. Aybars UĞUR	
Raportör Üye	: Prof. Dr. Serdar KORUKOĞLU	
Üye	: Yrd. Doç. Dr. Muhammed CİNSDİKİCİ	
Üye	: Yrd. Doç. Dr. Korhan KARABULUT	
Üye	: Yrd. Doç. Dr. Vecdi AYTAÇ	

ÖZET**SÜRÜ ZEKASI YAKLAŞIMLARININ RENKLİ GÖRÜNTÜ
KESİMLEMeye UYARLANMASI VE TANIMA SİSTEMLERİ
ÜZERİNDE GERÇEKLEŞTİRİMİ**

AYDIN, Doğan

Doktora Tezi, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Aybars UĞUR

Şubat 2011, 113 sayfa

Kesimleme işlemi tanıma sistemlerinin en önemli ve zor aşamasıdır. Bu tezde, sürü zekasına dayalı karınca kolonisi optimizasyonu (KKO) ve parçacık sürü optimizasyonu (PSO) tabanlı renkli görüntü kesimleme yöntemleri geliştirilmiştir.

IPSOAntK-means kümeleme algoritması, artırılmış parçacık sürü optimizasyonu (IPSO), KKO ve K-means algoritmalarının melez bir şekilde birleştirilmesi ile geliştirilmiştir. CAVIAR ve Oxford-17 çiçek veritabanları üzerinde denenmiş ve K-Means yönteminden daha başarılı sonuçlar üretmiştir. IPSOAntK-Means algoritması ile kesimlenen çiçekleri tanıyan bir modül de yazılarak test edilmiştir. Bu modülde, çiçeklere ait şekilsel ve renksel öznitelikler çıkarılarak farklı YSA modellerine (FFNN, CFNN, ENN, RBNN, PNN) verilmiştir. Modül, çiçek türlerini %90'ın üzerinde başarı ile tespit edebilmektedir.

Geliştirilen diğer yöntem, pikselleri optimum kümelemeyi hedefleyen, bir KKO algoritması olup, genel renkli görüntü kesimlemeye uyarlanmıştır. Berkeley veritabanındaki resimler üzerinde başarılı sonuçlar elde edilmiştir. Ayrıca renkli retina resimlerinde kan damarlarını kesimleyen bir yeni melez algoritma gerçekleştirilmiştir. DRIVE veri kümesi üzerinde test edilerek diğer kan damarı çıkarma algoritmalarıyla karşılaştırılmıştır.

Sürü zekasından esinlenen yöntemlerin ilk defa renkli görüntü kesimlemeye uyarlanması ve ürettikleri başarılı sonuçlar değerlendirilmiş ve tartışılmıştır.

Anahtar sözcükler: Sürü zekâsı, karınca kolonisi optimizasyonu, parçacık sürü optimizasyonu, renkli görüntü kesimleme, tanıma, kümeleme.

ABSTRACT**ADAPTATION OF SWARM INTELLIGENCE APPROACHES INTO
COLOR IMAGE SEGMENTATION AND THEIR
IMPLEMENTATIONS ON RECOGNITION SYSTEMS**

AYDIN, Doğan

Phd in Computer Eng.

Supervisor: Assoc. Prof. Dr. Aybars UĞUR

February 2011, 113 pages

Segmentation task in color images is the most important and difficult stage of image based recognition systems. In this thesis, swarm intelligence based ant colony optimization (ACO) and particle swarm optimization (PSO) oriented color image segmentation methods have been developed.

IPSOAntK-means clustering algorithm was developed by combining incremental particle swarm optimization (IPSO), ACO and K-means in a hybrid manner. It was tested on CAVIAR and Oxford-17 flower datasets, and obtained better results than K-means method. A module which recognizes flowers segmented by IPSOAntK-means was also developed and tested. In this module, shape and color features of flowers were extracted and given as input to the different ANN models (FFNN, CFNN, ENN, RBNN, PNN). The module can recognize flower species correctly with 90% accuracy.

Other developed method is an ACO algorithm, aimed optimum pixel clustering, which was adapted to general color image segmentation. Good results were obtained on Berkeley dataset images. Moreover, a new hybrid algorithm based on ants' behaviour was implemented for blood vessels segmentation in color retinal images. It was tested on DRIVE dataset and compared with other blood vessels segmentation algorithms.

The methods inspired from Swarm Intelligence applied first time to color image segmentation and their successful results were evaluated and discussed.

Keywords: Swarm intelligence, ant colony optimization, particle swarm optimization, color image segmentation, recognition, clustering.

TEŞEKKÜR

Bu çalışma süresince katkılarını ve görüşlerini esirgemeyen başta danışmanım Doç. Dr. Aybars Uğur olmak üzere Prof. Dr. Serdar Korukoğlu ve Yrd. Doç. Dr. Muhammed G. Cinsdikici' ye, maddi ve manevi desteklerini esirgemeyen sevgili eşime ve aileme, tezimde kullandığım bazı algoritmalar konusunda fikir ve katkılarından yararlandığım Thomas Stützle'ye ve IRIDIA Laboratuvarı çalışanlarına teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	v
ABSTRACT	vii
TEŞEKKÜR	ix
ŞEKİLLER DİZİNİ.....	xvii
ÇİZELGELER DİZİNİ	xx
SİMGELER VE KISALTMALAR DİZİNİ.....	xxi
1.GİRİŞ	1
2. SÜRÜ ZEKÂSI VE SÜRÜ ZEKÂSI TABANLI OPTİMİZASYON ALGORİTMALARI	3
2.1 Sürü Davranışları	3
2.1.1 Stigmergy	4
2.1.2 Özörgütleme.....	5
2.1.3 Ortaklaşa davranış çeşitleri	7
2.1.3.1 Eşgüdüm (coordination).....	7
2.1.3.2 Dayanışma (cooperation)	8
2.1.3.3 Tartışma-danışma (deliberation)	9
2.1.3.4 İşbirliği (collaboration)	9

İÇİNDEKİLER (DEVAM)

	<u>Sayfa</u>
2.2 Sürü Tabanlı Optimizasyon Algoritmaları	9
2.2.1 Karınca kolonisi optimizasyonu (KKO).....	10
2.2.1.1 Karıncalarda yön bulma ve yem arama stratejisi.....	11
2.2.1.2 Karınca kolonisi optimizasyonunun temel yapısı.....	12
2.2.1.3 Karınca sistemi (ant system).....	15
2.2.1.4 Elitist karınca sistemi (elitist ant system)	16
2.2.1.5 Karınca kolonisi sistemi (ant colony system).....	16
2.2.1.6 Sıra-tabanlı karınca sistemi (rank-based ant system)	17
2.2.1.7 Eniyi-enkötü karınca sistemi (best-worst ant system).....	18
2.2.1.8 Max-min karınca sistemi (max-min ant system)	19
2.2.2 Parçacık sürü optimizasyonu (PSO).....	20
2.2.2.1 Algoritma davranışını etkileyen ayrıntılar.....	22
2.2.2.2 Parçacık sürü optimizasyonu üzerinde temel değişiklikler	26
3. RENKLİ GÖRÜNTÜ KESİMLEME.....	29
3.1 Görüntü Kesimleme.....	29
3.1.1 Piksel-tabanlı kesimleme.....	31
3.1.1.1 Histogram-tabanlı teknikler.....	31

İÇİNDEKİLER (DEVAM)

	<u>Sayfa</u>
3.1.1.2 Kümeleme-tabanlı teknikler	32
3.1.2 Bölge-tabanlı kesimleme	33
3.1.2.1 Bölge-büyütme teknikleri	33
3.1.2.2 Parçala-ve-birleştir teknikleri	34
3.1.3 Kenar-tabanlı kesimleme	35
3.2 Renk Uzayları.....	36
3.2.1 RGB (kırmızı-yeşil-mavi) uzayı	37
3.2.2 <i>n</i> RGB (normalleştirilmiş kırmızı-yeşil-mavi) uzayı	38
3.2.3 HSI (ton-doygunluk-parlaklık) uzayı	39
3.2.4 YIQ uzayı	40
3.2.5 YUV uzayı	41
3.2.6 CIE uzayları	41
3.2.7 Fizik tabanlı uzaylar	42
4. ÖNCEKİ ÇALIŞMALAR.....	43
5. MF/ANT (MATCH FILTER / ANT COLONY) ALGORİTMASI İLE RETİNA GÖRÜNTÜLERİNDE KAN DAMARLARININ TESPİTİ.....	48
5.1 Match Filtresi (Filter) Algoritması.....	49
5.2 MF/Ant Algoritması.....	50

İÇİNDEKİLER (DEVAM)

	<u>Sayfa</u>
5.2.1 Önışlemler	51
5.2.2 Karınca algoritması	52
5.2.3 MF algoritmasının uygulanması	54
5.2.4 Sonuları birleřtirme ve uzunluk filtreleme	55
5.3 Deneysel alıřmalar	55
5.3.1 Kullanılan veri kümesi ve başarı ölçütleri	55
5.3.2 Parametrelerin belirlenmesi	56
5.3.3 DRIVE veri kümesi üzerinde test sonuları	58
6. IPSOANTK-MEANS ALGORİTMASI İLE İEK KESİMLEME	62
6.1 K-Means Algoritması	63
6.2 Renkli iek Resimlerinde Kesimleme Yaklařımı	64
6.2.1 Renk uzayının seimi	65
6.2.2 IPSOAntK-means kümeleme algoritması ile kesimleme	65
6.3 Deneysel alıřmalar	69
6.3.1 Iterative f-race algoritması	69
6.3.2 Uygun parametrele deėerlerinin seilmesi	72
6.3.3 CAVIAR-flower veri kümesi üzerinde test sonuları	73

İÇİNDEKİLER (DEVAM)

Sayfa

6.3.4 Oxford-17 çiçek veri kümesi üzerinde test sonuçları.....	76
7. KARINCA KOLONİSİ ALGORİTMASI İLE GENEL RENKLİ GÖRÜNTÜ KESİMLEME	79
7.1 Önerilen Kesimleme Yaklaşımı	79
7.1.1 Aday küme merkezlerinin ve sezgisel değerlerinin belirlenmesi	80
7.1.2 Karınca kolonisi optimizasyonu ile en iyi küme merkezinin seçilmesi	81
7.1.3 Kümelerin birleştirilmesi ile piksellerin kümelenmesi	83
7.2 Deneysel Çalışmalar	83
7.2.1 Oxford-17 çiçek veri kümesi üzerinde test sonuçları.....	84
7.2.2 Berkeley veri kümesi üzerinde test sonuçları.....	85
8. ÇİÇEK TANIMA SİSTEMİ	91
8.1 Sistemin Yapısı	91
8.1.1 Çiçeğe ait öznitelikler (öznitelik vektörü).....	92
8.1.1.1 Şekilsel öznitelikler	93
8.1.1.2 Renksel öznitelikler.....	95
8.1.2 Kullanılan YSA modelleri	95
8.2 Deneysel Sonuçlar.....	96

9. SONUÇ.....	98
KAYNAKLAR.....	101
ÖZGEÇMİŞ.....	113

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 Bir stigmergy örneği	5
2.2 Karınca kolonilerinde (a) eşgüdüm (yuva inşası), (b) dayanışma (yem taşıma), (c) danışma (yön bulma) ve (d) işbirliği örnekleri (görev paylaşımı).....	8
2.3 PSO algoritmalarında önerilen bazı sürü ağ yapıları	25
3.1 (Solda) kesimleme yapılacak resim, (sağda) kesimleme sonucu elde edilen resim.....	34
3.2 (Solda) kesimlenmiş resim, (sağda) bölümlerin hiyerarşik yapısı.....	35
3.3 RGB uzayı.....	38
3.4 HSI uzayının geometrik temsili	39
5.1 MF/Ant algoritmasının yöntemi	51
5.2 (a) orjinal retina resmi, (b) yeşil kanalın gri tonajda gösterilmiş resmi, (c) yeşil kanalın doğrusal olmayan yoğunluk dönüşümü sonrası görüntüsü.....	52
5.3 Karıncanın dolaşımı (ortada), karıncanın ölmesi (sol altta).....	53
5.4 50 derecelik açı için oluşan ve uygulanan match filtresi	55
5.5 Farklı q_0 ve ρ parametreleri için kesimleme sonuçları.....	57
5.6 Bir retina resmi için ortaya çıkan ROC eğrisi ve 2. gözlemciye göre karşılaştırılması	59

ŞEKİLLER DİZİNİ (DEVAM)

<u>Şekil</u>	<u>Sayfa</u>
6.1 Önerilen yöntem	64
6.2 IPSOAntK-means algoritmasının akış diyagramı	66
6.3 Uygulanan problem örneğine göre aday sayısının durumu	70
6.4 CAVIAR-Flower veri kümesi üzerinde bazı sonuç örnekleri	75
6.5 ROC eğrisi üzerinde IPSOAntK-means ile K-means algoritmalarının karşılaştırılması	76
6.6 Oxford-17 veri kümesine ait bir resim ve ilgili trimap resmi.....	77
7.1 Bir resme ait histogramlar ve her birindeki çubukların orta noktaları ...	81
7.2 Adım sayısı (hafıza büyüklüğü) üç olan bir karıncanın gerçekleştirdiği bir turun görünümü.....	82
7.3 Örnek bir resim üzerinde KKO yaklaşımıyla elde edilen kesimleme sonuçları	84
7.4 Oxford-17 çiçek veri kümesi üzerinde bazı test sonuçları	85
7.5 Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları – F-ölçütü ..	87
7.6 Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları – Precision-Recall eğrileri	88
7.7 Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları – Bulunan kenarlar	89
7.8 Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları – Kesimlenen bölgeler	90

ŞEKİLLER DİZİNİ (DEVAM)

8.1	Çiçek tanıma sisteminin yapısı (Solda), tanıma sisteminin bir örnek resim için işleyişi (Sağda)	92
8.2	Çiçeklere ait merkezden uzaklık hatları ve yaprak başlangıç-bitiş noktaları (kırmızı noktalar)	94
8.3	Tanıma sisteminde test edilen bazı çiçek türlerine ait resimler.....	96
8.4	Karşılaştırma sonuçları: Başarı yüzdeleri.....	97

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
5.1 Seçilen parametre değerleri	57
5.2 Blok boyutuna göre karınca sayısı ve adım sayısının etkisi	58
5.3 20 resim için elde edilen ROC eğrisi altında kalan alan değerleri	59
5.4 DRIVE veri kümesindeki resimler için maksimum başarı değerleri.....	60
5.5 Karşılaştırma sonuçları	61
6.1 Iterative F-Race algoritmasına ait parametreler, tanımları ve değerleri	71
6.2 IPSOAntK-means parametreleri ve bunların aralık veya değerleri.....	72
6.3 Iterative F-Race ile IPSOAntK-means için seçilen parametre değerleri.....	73
7.1 Berkeley veri kümesi üzerinde karşılaştırma sonuçları: F-ölçütü	87
8.1 Karşılaştırma sonuçları: Başarı yüzdeleri	97

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
τ_{ij}	Feromon yoğunluğu.
η_{ij}	Sezgisel fonksiyon.
v_i, v_j	Şehirler.
i, j	Düğüm indeksleri.
α, β	Ayar parametreleri.
ρ	Buharlaştırma katsayısı.
m	Karınca sayısı.
k	Karınca indeksi.
$T_k(t)$	k indeksli karıncanın attığı tur.
p_{ij}	Bir durumun seçilme olasılığı.
e	Elitist karınca sayısı.
L	Tur uzunluğu, Çiçeğin çevresinin uzunluğu.
$\vec{v}_i$	Parçacığın hız vektörü.
$\vec{p}_i$	Parçacığın en iyi konumu.
$\vec{p}_g$	Komşularının en iyi pozisyonu.

SİMGELELER VE KISALTMALAR DİZİNİ (DEVAM)

<u>Simgeler</u>	<u>Açıklama</u>
$\vec{x}_i$	Parçacığın o anki pozisyonu.
ϕ_1, ϕ_2	İvme katsayıları.
$V_{\max,j}, V_{\min,j}$	j boyutunda maksimum ve minimum hız değerleri.
w	Eylemsizlik ağırlığı.
χ	Büzme katsayısı.
r	Döndürme matrisi.
$p_{\max}$	Resimdeki maksimum piksel değeri.
ψ, ϕ	Eşik değerleri.
μ	Ortalama.
σ	Standart sapma.
Ω	Ziyaret edilmemiş durumlar kümesi.
C	Küme merkezi.
B	Bütçe.
E, F, P	Başarı ölçütleri.
δ	Aday çiçek bölgesindeki toplam piksel sayısı.

SİMGELER VE KISALTMALAR DİZİNİ (DEVAM)

<u>Simgeler</u>	<u>Açıklama</u>
M	Karınca hafızasının boyutu.
n	Histogramdaki çubuk sayısı.
f	Uygunluk fonksiyonu.
γ_i	C_i küme merkezine en yakın piksellerin sayısı.
A	Çiçeğin kapladığı alan
<u>Kısaltmalar</u>	
CFNN	Kademeli-ilerleyen YSA.
CNN	Hücrel sinir ağı.
EKS	Elitist karınca sistemi.
ENN	Elman YSA.
FFNN	İleri Beslemeli YSA.
GAMF	Genetik algoritma ile optimize edilmiş match filtresi.
GSP	Gezgin satıcı problemi.
HSI	Ton-doygunluk-parlaklık uzayı
IPSO	Arttırımlı parçacık sürü optimizasyonu.
İKKS	Eniyi-enkötü karınca sistemi.
KKO	Karınca kolonisi optimizasyonu.

SİMGELER VE KISALTMALAR DİZİNİ (DEVAM)

<u>Kısaltmalar</u>	<u>Açıklama</u>
KKS	Karınca kolonisi sistemi.
KS	Karınca sistemi.
MF	Match filtresi.
MMKS	Max-min karınca sistemi.
n RGB	Normalleştirilmiş Kırmızı-yeşil-mavi uzayı
OGMF	Optimize edilmiş match filtresi.
PNN	Olasılıklı YSA.
PSO	Parçacık sürü optimizasyonu.
RBNN	Radyal-tabanlı YSA.
RGB	Kırmızı-yeşil-mavi uzayı
ROC	Alıcı işletme özellikleri.
STKS	Sıra-tabanlı karınca sistemi
YSA	Yapay sinir ağı.

1. GİRİŞ

Sürü zekasına dayalı optimizasyon algoritmaları sürülerin zeki davranışlarından ilham alınarak gerçekleştirilen çok etmenli, esnek ve sağlam yaklaşımlardır. Bu algoritmalarından en çok kullanılanları karınca sürülerinin yemek arama stratejilerinden ilham alan Karınca Kolonisi Optimizasyonu (KKO) ve kuş sürülerinin hareket stratejilerinden ilham alan Parçacık Sürü Optimizasyonu (PSO) algoritmalarıdır.

Karınca Kolonisi Optimizasyonu algoritmaları genellikle kombinasyon optimizasyon problemlerinin çözümünde kullanılırken Parçacık Sürü Optimizasyonu algoritmaları genellikle sürekli optimizasyon problemlerinin çözümünde kullanılmaktadır. Örneğin rotalama problemleri (gezgin satıcı, araç rotalama, ardışık sıralama vb.), atama problemleri (kareli atama, atama, çizge boyama vb.), planlama problemleri (proje planlama, açık dükkan vb.) gibi bir çok kombinasyon optimizasyon problemlerine KKO algoritmaları ile çözüm getirilebildiği gibi fonksiyon optimizasyonu gerektiren bir çok mühendislik biliminde, kontrol tasarımında, sinyal işlemede, robotik bilimi ve daha bir çok alanda PSO ile çözüm üretilebilmektedir.

Görüntü kesimleme resmin homojen parçalara ayrılması veya resim içindeki nesnelerin birbirinden ayrıştırılması işlemidir. Kesimleme işlemi görüntü tabanlı algılama ve sorgulama sistemlerinin en zor ve önemli adımıdır. Çünkü kesimlemenin kalitesi tanıma sisteminin kalitesini doğrudan etkilemekte ve görüntülerdeki ışık değişimi, parlaklık etkisi, görüntü kalitesi, farklı desen çeşitleri vb. birçok etken kesimlenin başarısını olumsuz etkilemektedir. Bu yüzden şu ana kadar gerçekleştirilen kesimleme yaklaşımlarından hiç birisi her görüntü için çok iyi sonuç verememektedir.

Gerçekleştirilen birçok kesimleme yaklaşımı gri-tonajlı resimler üzerinde denense de renkli resimlerin gri-tonajlı resimlerden çok daha fazla bilgi içermesi ve bilgisayar teknolojisinin gelişmesiyle, bilgisayarların renkli resimler üzerindeki işlemleri artık hızlıca yapabiliyor güçte olmasından dolayı renkli görüntü kesimleme son yıllarda araştırmacıların dikkatini çekmektedir (Chen et al. 2001). Renk bilgisinin birçok bilgi taşıması bir avantaj olsa da renk bilgilerinin üç boyutlu renk uzaylarıyla temsil edilmesi ve renkler arasındaki farkın tespit edilme zorluğu ile diğer birçok etki (doygunluk, renk geçişi, parlaklık vb.) etkiler renkli görüntülerin kesimlenmesini oldukça güçleştirmektedir.

Renklerin homojen kümelerle ayrıştırılarak kesimlenmenin gerçekleşmesi işlemi bir optimizasyon problemi olarak tanımlanabileceği gibi sürü davranışlarının modellenmesi yolu ile piksellerin birbirlerinden ayrıştırılması ve kesimlenmek istenilen bölgenin ortaya çıkartılması sağlanabilir. Literatürde hem pikselleri optimum kümelerle ayırarak kesimleme yapmaya çalışan hem de sürü hareketlerine dayalı nesne kesimleme yapmaya çalışan bazı çalışmalar vardır. Fakat bu çalışmaların hemen hemen tamamı gri-tonajlı resimler üzerinde gerçekleştirilen çalışmalardır.

Bu tezde, sadece renkli resimler üzerinde odaklanılmış ve üç farklı alanda, renkli resimlerdeki piksellerin kümelenmesi ve karıncaların hareket stratejilerine dayalı üç yeni kesimleme yöntemi geliştirilmiştir.

Birinci yöntem renkli retina görüntülerinde tespiti oldukça zor olan kan damarı ağlarının ortaya çıkarılmasını sağlayan karıncaların hareket stratejilerinden yararlanılarak geliştirilmiş bir melez yöntemdir.

İkinci yöntemde KKO, PSO ve K-means algoritmalarının birleştirilmesiyle geliştirilen melez bir algoritma vasıtasıyla renkli resimlerde çiçekli bölgelerin kesimlenmesi sağlanmıştır.

Son olarak renkli resimlerde piksellerin optimum kümelenmesini sağlayacak Karınca Kolonisi Optimizasyonuna kullanılarak gerçekleştirilen genel bir kesimleme algoritması üzerinde çalışılmıştır. Gerçekleştirilen bu algoritma hem doğal resimlerde renklerin ve nesnelerin kesimlenmesinde hem de yine renkli resimlerdeki çiçekli bölgelerin kesimlenmesinde kullanılmıştır.

Tüm yaklaşımlar ilgili alanlarında kesimleme performansını ölçmek için oluşturulan veri setleri üzerinde denenmiş ve literatürdeki o alanlara ait diğer kesimleme yaklaşımlarıyla karşılaştırılmıştır.

Sonraki bölümlerde öncelikle sürü zekâsı kavramına ve özellikle tezde kullanılan temel sürü zekâsı algoritmalarına değinilmiştir. Daha sonra genel görüntü kesimleme algoritmalarından ve renk uzaylarından bahsedilmiştir. Son olarak tez kapsamında gerçekleştirilen üç farklı sürü zekâsı tabanlı kesimleme yaklaşımlarından bahsedilmiş, elde edilen karşılaştırma ve test sonuçları gösterilmiştir.

2. SÜRÜ ZEKÂSI VE SÜRÜ ZEKÂSI TABANLI OPTİMİZASYON ALGORİTMALARI

Uzun zaman önce, insanlar doğada bulunan hayvan ve böcek sürülerinin davranışlarını keşfettiler. Kuş sürülerinin havada süzülmesi ve farklı şekil alması, karıncaların yiyecek araştırması, balık sürülerinin beraberce yüzmesi ve kaçışması bu sürü davranışlarından sadece birkaçıdır. Son yıllarda biyologlar ve bilgisayar bilimciler bu sosyal hayvanların nasıl iletişim kurup hareket ettiklerini ve hedefe ulaştıklarını anlayabilmek için biyolojik sürüleri matematiksel olarak modellemeye çalıştılar. Öte yandan, bu sürü davranışların “sürü zekâsı” kapsamı altında optimizasyondan robot-bilime, askeri uygulamalardan iletişim ve taşımacılık sistemlerine kadar birçok alanda uygulanabilir olması mühendislerin de bu konuya dikkatlerini çekmesine neden oldu.

Bu bölümde ilk olarak sürü davranışlarının temel özelliklerine ve sürü zekâsı yaklaşımlarına ilham kaynağı olan bazı sürü davranışlarına yer verilmiştir. İkinci olarak kullanılan sürü zekası tabanlı optimizasyon algoritmalarından bahsedilmiştir.

2.1 Sürü Davranışları

Uzun yıllardan beri sürü davranışları biyologların en çok ilgisini çeken konulardan birisi olmuştur. Sürülerin kollektif davranışları üzerindeki ilk tahminler yaklaşık yüz sene öncesine dayanır. Eski hipotezlerde her bir böceğin küresel yapıyı bildiği ve bu küresel yapı üzerinden böceklerin kendi hareketlerini düzenlediği düşünülmüştür (Thorpe 1963). Diğer bir deyişle, insanlar karar verme karmaşıklığı ile kolonide görülen örüntüler arasında direk bir ilişki olduğunu düşünmüş ve bu karmaşık karar verme mekanizmalarının ve örüntülerinin bireysel seviyede her bir böceğin bildiği ve buna göre hareket ettiği tahmin edilmiştir. Örneğin, kraliçe arının koloni ile ilgili tüm bilgileri topladığı ve koloniyi idare ettiği ve tüm işlerin yapılmasında işçileri yönlendirdiği sanılırdı.

Son 40 senedir yapılan çalışmalar ise sürülerin tamamen farklı bir organizasyon yapısına sahip olduğunu gösterdi (Theraulaz et al. 1998). Şu an biliyoruz ki böceklerin her birisi küresel bir bilgiye, bir gösterime ya da haritaya ihtiyaç duymamaktadırlar. Diğer bir deyişle, bir böcek tüm küresel duruma hakim olacak, tüm bilgileri toplayabilecek ve kontrol edebilecek durumda değildir. Yani bu kolonilerde herhangi bir kontrol merkezi veya denetçi söz konusu değildir.

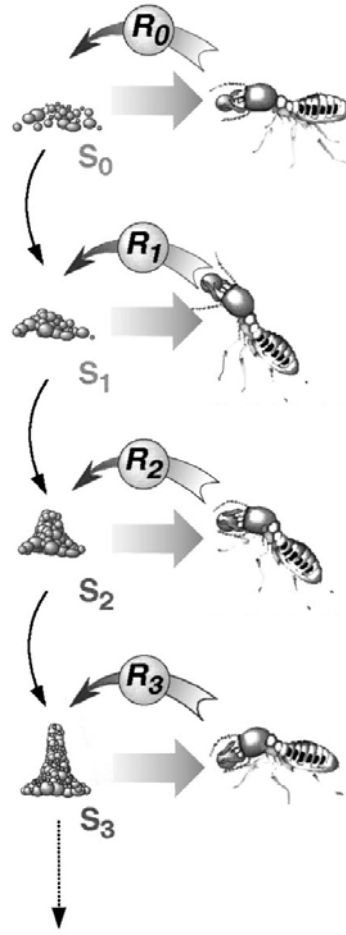
Bunun yerine, böcekler merkezi olmayan bir sistem ile iş görürler. Her bir birey birbirinden bağımsız olarak ortamda iş görür. Bu davranışları, basit olasılıklı uyarıcı-tepki davranış kurallarıyla açıklanır. Bu kurallar ortamdan elde edilen yerel bilgilere göre işletilir ve küresel herhangi bir bilgi söz konusu değildir. Ayrıca, her bir birey sadece bazı basit kuralları takip eder (Deneubourg et al. 1983). Örneğin, karıncalarda her bir bireyin ortalama 20 tane basit kurala göre hareket ettiği tespit edilmiştir.

Bu basit kurallar ve olasılıklı uyarıcı-tepki davranışları sayesinde yapılan iletişim koloni seviyesinde bir organizasyonun oluşmasına neden olur. Karşılıklı etkileşimler hem bilginin tüm koloni boyunca yayılmasını garantiler hem de her bir bireyin eylemlerini kontrol eder. Bu karmaşık etkileşim ağı sayesinde, sürüler birçok farklı problemi çözebilirler ve dışsal zorluklara karşı esnek ve sağlam bir yapı oluştururlar.

2.1.1 Stigmergy

Sosyal böceklerin hareketleri üzerinde ilk ciddi teorik açıklama 1959 yılında Fransız biyolog Grassé tarafından yapılmıştır. Grassé termitlerin yuva inşasındaki aktivitelerini stigmergy adını verdiği bir kavram ile açıklamıştır. Bu kavrama göre, yuva inşasındaki aktivitelerdeki eşgüdüm ve düzenleme işi karıncaların kendilerine bağlı değil yuvanın o anki şekline bağlıdır. Diğer bir ifadeyle, bilgi yerel ortamdan gelmektedir ve süregelen işin kendisi her bir bireyin aktivitesinde rehber rol oynamaktadır. Örneğin, her an bir işçi yuva inşası konusunda bir eylem yapmaktadır ve bu eylemler yuvanın şeklinde değişikliklere neden olur. Yuvanın şeklinin değişimi eylemin değişmesini de tetiklemiş olur (Bkz Şekil 2.1). Ortamda değişiklik yaparak ortaya çıkan bu süreç ortak çalışmanın eşgüdümünün mükemmel bir şekilde sağlanmasına olanak tanır. Bu eşgüdüm o kadar güzel bir şekilde işletilir ki, koloninin çok iyi tanımlanmış bir planı takip ettiği sanılabilir.

Grassé'nin ilk olarak termitlerin aktiviteleri üzerine açıklamada kullandığı stigmergy terimi daha sonra diğer böcek türlerinde de gözlemlenen ortamda değişiklik yaparak gerçekleştirdikleri dolaylı iletişim yöntemi yerine kullanılmaya başlanmıştır (Theraulaz, E. Bonabeau, 1999).



Şekil 2.1. Bir stigmergy örneği: Termitlerin yuva inşasında ortaya çıkan dolaylı iletişim. Yuva yapısının ilk olarak S_0 durumunda olduğunu varsayalım. S_0 durumu karıncanın R_0 aktivitesini tetikleyecektir (ör., büyük toprak tanecikleri koy). Yuva durumu $S_1, S_2, S_3, \dots, S_n$ e geldiğinde karıncaların $R_1, R_2, R_3, \dots, R_n$ aktiviteleri tetiklenecektir. Böylece hem her bir birey ortama göre tepkilerini değiştirecek hem de her bir birey ortamda değişikliklere neden olarak diğer bireylerin tepkilerinin değişmesinde dolaylı olarak etkide bulunacaktır (Bonabeau et al, 1999).

2.1.2 Özörgütlenme (self-organization)

Özörgütlenme alt-seviyedeki bileşenlerin etkileşimleriyle sistemin küresel seviyesinde yapıları ortaya çıkaran dinamik mekanizmalar kümesini ifade eder. Sistemin alt-bileşenlerinde gerçekleştirilen faaliyetler tamamen yerel bilgiler çerçevesinde gerçekleştirilir ve küresel bilgi kullanılmaz. Böylece dışarıdan bir etki ve yönlendirme olmaksızın tüm grup bir eşgüdümle hareket edebilir. Karıncaların yol bulma yöntemlerinden, bal arılarının kovan yapmalarına ve termitlerin yuvalarını inşa etmelerine kadar birçok sosyal böcek türünde görülen ortaklaşa kararlar bu özörgütlenme kuramı içinde ifade edilmektedir.

Özörgütlenme dört temel içeriğe sahiptir;

- Pozitif Geri-besleme: Genel yapının ortaya çıkmasına katkıda bulunan basit davranışların sonucudur. Örneğin, yiyecek kaynağını gösteren feromon izini güçlendirmek (o yol üzerinde feromon bırakmak) bir çeşit pozitif geri-beslemedir. Çünkü bu davranış küresel seviyede iz ağının ortaya çıkmasını sağlayan koşulların ortaya çıkmasına neden olur.
- Negatif Geri-besleme: Pozitif geri-beslemeyi dengeler ve ortaklaşa oluşan örüntünün dengelenmesini sağlar. Örneğin yine karınca yem arayışında, negatif geri-besleme birçok şeyden kaynaklanabilir. Bunlardan bazıları belli yollarda kısıtlı sayıda araştırmacı karıncaların bulunması, feromon maddesinin buharlaşmasıdır.
- Kararsızlığın Güçlendirilmesi: Sosyal böcekler hareketlerini rastlantısal olarak gerçekleştirirler. Bu davranışların bazıları rastgele yürüyüşler, hatlar, rastgele iş-değişimi vb. olabilir. Rastgelelik özörgütlemeye oldukça kritik bir rol oynamaktadır. Çünkü rastgelelik koloninin yeni çözümler bulmasına olanak tanır. Örneğin, kaybolan araştırmacı karıncalar, yeni ve daha önce bulunmamış bir yiyecek kaynağını veya aynı yiyecek kaynağına ulaşan daha iyi bir yolu keşfedip bunu pozitif geri-besleme ile diğer bireylere bildirebilirler.
- Çoklu Etkileşim: Özörgütleme, bireyler arasında çoklu, direk veya stigmergy vasıtasıyla dolaylı etkileşimi gerektirir. Böylece her bir birey hissedilir bir sonuç üretir ve bu sonuç büyük ve genel yapının oluşmasında katkıda bulunur.

Bunun yanında, özörgütleme bazı anahtar özelliklere göre karakterize edilmiştir:

- *Özörgütlemeli sistemler dinamiklidir.* Yapıların ortaya çıkması için bireylerin devamlı birbirleriyle ve ortamla iletişim halinde olmaları gereklidir.
- *Özörgütlemeli sistemler genel seviyede ortaya çıkan davranışları sergilerler.* Yani bu sistemler basit bireylerin özelliklerinden/davranışlarından çok daha karmaşık bir özelliği resmederler. Bu davranışlar sürü/koloni üyelerinin birbiri aralarında yaptıkları iletişimin doğrusal olmayan (nonlinear) bileşimlerinden ortaya çıkarlar.
- *Ortaya çıkan özelliklerle birlikte doğrusal olmayan etkileşimler çatalanmalara neden olur.* Bir çatalanma sistemin parametrelerinde olabilecek değişiklikler sonucunda yeni kararlı çözümlerin

görünümüdür. Diğer bir deyişle, etkileşimlerin farklılaşması farklı ortaklaşa özelliklerin değişmesine de neden olur.

- Son olarak, *özörgütlemeli sistemler çoklu-kararlı olabilir*. Çoklu-kararlıdan kasıt sistem verilen farklı parametrelere, ilk duruma ve rastgele dalgalanmaların durumuna göre farklı kararlı durumlara ulaşabilir.

2.1.3 Ortaklaşa davranış çeşitleri

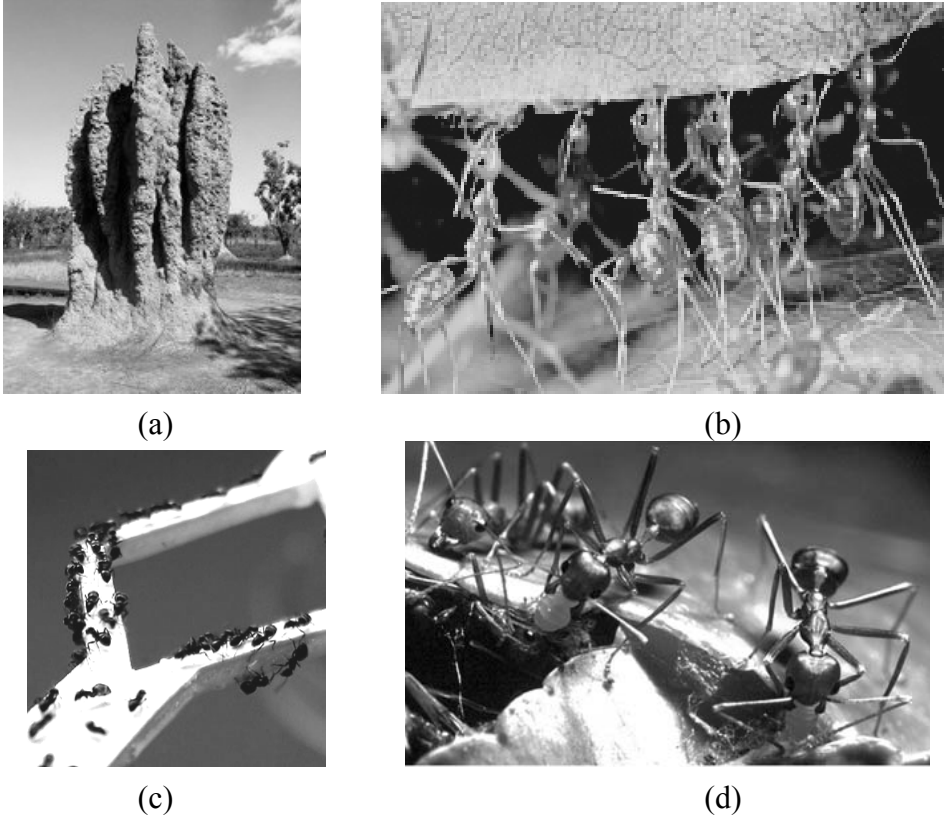
Sürülerdeki ortaklaşa davranışlar, yaptıkları işlerin farklılığına göre çeşitlilik gösterir. Genel olarak bakıldığında dört farklı fonksiyonu içlerinde gerçekleştirirler. Bunlar; eşgüdüm, dayanışma, tartışma (danışma), işbirliğidir (Bkz Şekil 2.2) (Garnier et al., 2007). Aşağıdaki alt bölümlerde önce bu fonksiyonlar açıklanacak daha sonra sürüler üzerindeki örneklerine yer verilecektir.

2.1.3.1 Eşgüdüm (coordination)

Eşgüdüm, bir problemin çözümünde gerekli işlerin zamanlanmasını ve belli bir alandaki uygun organizasyonlar bütününe ifade eder. Bu fonksiyon bireylerin alan üzerinde geçici dağılımlarına, aktivitelerine ve aktivite sonuçlarına neden olur. Bu da çözüme ulaşmalarını sağlar.

Örneğin, eşgüdüm balarılarının yer değiştirmelerinde ve çekirge sürülerinde görülür (Buhl et al. 2006; Janson et al. 2005). Bu sürülerde, karşılıklı etkileşim, bireylerin hedeflerine ulaşmak için senkronize (geçici organizasyonlar şeklinde) ve yönelimli (yüzeysel organizasyonlar şeklinde) hareketlerini ortaya çıkartır.

Diğer bir örnek, birçok böcek kolonisinin yuva inşasındaki aktivitelerindeki eşgüdümlü işleridir. Eşek arıları (Karsai and Theraulaz, 1995) ve termitler (Grassé, 1959) yuvalarını inşa ederken daha önce bahsi geçen stigmergy tabanlı süreç, bireylerin zaman ve mekân içinde farklı organizasyonları ve görevleri almalarına neden olur.



Şekil 2.2. Karınca kolonilerinde (a) eşgüdüm (yuva inşası), (b) dayanışma (yem taşıma), (c) danişma (yön bulma) ve (d) işbirliği örnekleri (görev paylaşımı)

2.1.3.2 Dayanışma (cooperation)

Dayanışma, bireylerin tek başına kaldıklarında yapamadıkları işleri birlikte yapmak zorunda olduğu durumlarda ortaya çıkar. Bireyler problemi çözebilmek için kendi güçlerini birleştirmek zorundadırlar.

Dayanışma en açık bir şekilde büyük avların taşınması işleminde görülür. Bu durumda genellikle tek bir bireyin gücü avı taşımada yeterli olmaz. Birçok karınca türünde (dokumacı, savaşçı ve ağaç karıncaları gibi) farklı ortak taşıma dayanışması yöntemi gözlenmiştir. Dayanışmalı taşımacılık ile yiyeceklerin taşınmasının oldukça verimli olduğu gözlenmiştir. Örneğin bir karınca türünde, dayanışma ile yapılan taşımacılıkla bir tek taşıyıcının taşıyabileceği ağırlığın on katı kadar ağır avlar taşınabilmektedir.

Dayanışma örnekleri, av taşımacılığından başka görevlerde de gözlenebilmektedir. Örneğin, dokumacı karıncaların birbirlerine tutunarak köprü vaziyetini almaları ve böylece iki farklı dalı veya yaprağı birleştirebilmeleri (Deneubourg et al., 2002) yine dayanışmanın bir örneğidir.

2.1.3.3 Tartışma-danışma (deliberation)

Bu davranış mekanizması koloni birçok ihtimalle (fırsatla) karşılaştığında ortaya çıkar ve koloninin en iyi ihtimale karar vermesine yardımcı olur.

Örneğin bal arılarının yeni yuvalarının tespitinde veya nektar toplanacak çiçekli alanlarının tespitinde kullandıkları sallanma dansları bir danışma-tartışma örneğidir. Danslarında yapmış oldukları sallanmaların kuvveti tarif ettikleri bölgenin kalitesiyle doğru orantılıdır ve diğer arıların karar verme aşamasında farklı yerlerden gelen arıların bu danslarının şiddetleri önemli bir rol oynamaktadır.

Bu mekanizma karıncaların yem arayışlarında da gözlenmektedir. Birçok yiyecek ve bu yiyeceklere giden farklı yollar olmasına rağmen karıncalar sadece bir yolu tercih etmektedirler. Bu seçim, yol üstünde bırakılan kimyasal maddelerin gücüyle doğru orantılıdır. Aslında bu bir rekabettir. Seçilen ihtimallerden güçlü olanı yarış kazanır ve sürü bu ihtimali kendilerine seçim olarak belirler.

2.1.3.4 İşbirliği (collaboration)

İşbirliği farklı aktivitelerin aynı koloni içinde farklı gruplar tarafından eşzamanlı gerçekleştirilmesidir; karınca kolonilerinde aynı anda av taşıma veya yuva içinde yavrularla ilgilenme gibi. Bu özelleşme tamamen morfolojik ya da tamamıyla davranışsal farklılıklardan kaynaklanabileceği gibi bireylerin yaşıyla da ilgili olabilir.

Bu iş paylaşımını en iyi açıklayan örnekler farklı sınıftan grupları barındıran sürülerde gözlenmektedir. Örneğin, yaprak kesen karıncalar dört farklı sınıfı içinde barındırır. Bu sınıflardaki bireylerinin boyutu, yaptıkları işlerle orantılıdır. Sadece, kafa büyüklüğü 1,6 mm den büyük olan karıncalar yaprak kesebilmektedir. Kesilen bu yapraklar, üzerlerinde mantar yetiştirilmek üzere yuvaya taşınır. Öte yandan, sadece kafa büyüklüğü 0,5 mm civarında olan küçük işçi karıncalar mantar yetiştirme görevini üstlenirler.

2.2 Sürü-Tabanlı Optimizasyon Algoritmaları

Hiçbir kontrol mekanizması olmadan sürüdeki bireylerin basit yapıda ve özellikte olup basit iletişim yolları ile az maliyet harcayarak esnek, sağlam bir

yapı oluşturup zekice hareket etmeleri ve kendi karşılaştıkları problemlere çözüm getirmeleri biyologların ilgisini çektiği gibi bilgisayar bilimcilerin ve mühendislerin de ilgisini çekmektedir. Çünkü iletişim, taşımacılık, endüstriyel üretim, robot-bilim gibi birçok alanda çözümler getirmeye çalışan bilgisayar ve mühendislik bilimleri bu alanların birçoğunda çözülmesi oldukça zor olan veya zaman veya maliyet olarak masraflı olan problemlerle karşı karşıya kalırlar. Sürülerin karşılaştıkları ve çözümledikleri problemlerin birçoğu bu problemlerle paralellik göstermektedir. Bu da sürü zekâsı kavramının geliştirilmesinde önemli bir etken olmuştur.

Bilgisayar bilimlerinde ve mühendislikte Sürü Zekâsının tarifi Bonabeau ve arkadaşları tarafından (1999) “Sosyal böcek kolonilerinin veya hayvan sürülerin zeki ortaklaşa çalışma davranışlarından ilham alarak algoritmalar veya problem-çözücü makineler tasarlama girişimi” olarak tanımlanmışlardır. Doğada birçok sürü veya koloni ve bu birlikteliklerde birçok farklı davranış modelleri bulunduğundan bu davranışlardan ilham alınarak tasarlanan birçok algoritma ve aygıt bulunmaktadır ve halen yeni yaklaşımlar öne sürülmektedir. Tezimizde sürü tabanlı bir aygıt kullanmadığımızdan robot-bilimde gerçekleştirilen sürü zekâsı yöntemlerine değinilmemiştir. Ayrıca, tezimizde sadece sürü zekâsı tabanlı optimizasyon algoritmaları kullanıldığından bu algoritmalar üzerinde ağırlık verilmiş, diğer sürü zekâsı tabanlı algoritmalara değinilmemiştir. Diğer sürü zekâsı yaklaşımları için (Bonabeau et al., 1999) kaynağına bakılabilir.

2.2.1 Karınca kolonisi optimizasyonu (KKO)

Karınca kolonisi optimizasyonu (KKO) ilk olarak Dorigo ve arkadaşları tarafından (1996) karınca kolonilerinin yem arama yöntemlerinden ilham alınarak tasarlanan popülasyon-tabanlı bir sürü zekâsı optimizasyon algoritmasıdır. Çeşitli birçok türü ortaya atılmış ve birçok NP-Zor problemlerin çözümünde başarıyla uygulanmıştır (Uğur and Aydın, 2009).

Aşağıdaki alt bölümlerde KKO algoritmalarına ilham kaynağı olan karıncaların yem arama stratejilerine değinilmiş, daha sonra temel KKO algoritmalarından bahsedilmiştir.

2.2.1.1 Karıncalarda yön bulma ve yem arama stratejisi

Karıncalar birbirleriyle feromon adı verilen bir kimyasal madde yoluyla iletişim kurarlar. Bir karınca, bir yiyecek bulduğunda hemen yuvasına geri döner ve bu dönüş yolunda ardında feromon maddesi salgılar ve bir iz oluşturur. Bu iz diğer karıncaların yiyecek kaynağına doğru yönelmelerinde rehberlik eder. Diğer karıncalar da yiyecekleri bulup yuvaya dönüş yolunda bıraktıkları kendi feromon salgılarıyla daha önceden geçmiş olunan yolu (izi) güçlendirirler. Bu iz bilgisi pozitif geri besleme örneğidir. Yani ne kadar karınca o yolu kullanırsa, o yol (iz) o kadar çekici hale gelir. Öte yandan, bir zaman sonra izi güçlendirilme ve izi takip etme seyrekleştikçe iz kaybolmaya başlar. Çünkü feromon maddesi buharlaşma özelliğine sahiptir. Bu tip bir durum, yiyeceğin kaynakta tükenmesi veya daha iyi yolların keşfedilmesi durumlarında ortaya çıkabilir. Garip olan, bu iz güçlendirme ve takip sistemi sadece birçok karıncanın yiyecek kaynağı etrafında toplanmalarında değil, yiyecek kaynağına varacak en verimli ve kısa yolların seçilmesinde de önemli bir rol oynar.

1990'ların başında Deneubourg ve arkadaşları iki köprü deneyi adını verdikleri basit fakat etkili bir deney yaparak feromon bilgisinin nasıl iyi yolların seçiminde etkili olduğunu ispat etmişlerdir (Deneubourg and Goss, 1989). Bu deneyde, yuva ile yiyecek kaynağı arasında uzunlukları aynı olan iki köprü yerleştirilmiştir. Normalde her bir yolun seçim oranı %50 olmasına rağmen bir zaman sonra karıncaların birçoğunun sadece bir yoldan gittiği gözlemlenmiştir. Dikkatle incelendiğinde, ilk aşamada karıncalar her hangi bir bilgiye sahip olmadıklarından tamamıyla rastgele %50 olasılıkla yollara dallanmışlar fakat zamanın geçmesiyle rastgele seçimler bazı karıncaların daha çok bir tarafa dallanmalarına sebep olmuştur. Daha fazla karınca, bir tarafa az da olsa fazla yönelme yaptığından o yol üzerinde daha çok feromon maddesi depolanmış ve bu da arkadan gelen karıncaların o yolu daha çok seçmelerini tetiklemiştir. Sonuçta, birçok karınca artık bir tarafı daha çok seçer olmuştur.

Aynı bilim adamları, diğer bir çalışmalarında bu deneyi bir parça ilerleterek deney düzeneğini değiştirmişlerdir. Yeni deneyde, iki köprüden biri diğerinden daha uzun tutulmuştur. Bu durumda, karıncaların, çoğu defa aynı mekanizmayı kullanarak kısa yolu tercih ettiği gözlenmiştir. Bu sonuç şaşırtıcı değildir. Çünkü kısa yoldan dönüş yapan karıncalar yuvalarına daha kısa zamanda döndüklerinden feromon maddesini yol üzerinde bırakıp diğer işçilere rehberlik etmesi çok daha hızlı olacaktır. Bu da diğer karıncaların kısa yollara yönelmesinde etkin rol

oynayacağından bir zaman sonra karıncaların büyük çoğunluğunun kısa yolu tercih etmesi kaçınılmaz olacaktır.

2.2.1.2 Karınca kolonisi optimizasyonunun temel yapısı

KKO karıncaların yem arama yöntemlerinden ilham alılarak tasarlanan algoritmalar bütünüdür. KKO gerçek karıncaların doğrudan modellenmesi ve benzetimi üzerinde uğraşmaz. Fakat sanal karıncaları bir optimizasyon tekniği olarak kullanır. KKO’nda sanal karıncalar gerçek karıncalar gibi kendi yolları üzerinde çözümlerinin kalitesine göre feromon maddesi bırakmalarına rağmen sanal karıncaları gerçek karıncalardan ayıran bazı özellikler vardır. Bunlar (Aydın, 2009);

- Sanal karıncalar bir hafızaya sahiptir. Dolaştıkları çözüm yollarını hafızalarında kaydederler.
- Tamamıyla kör değillerdir. Yani, ilk çözüm aramaya başladıklarında dahi onları çözüme hızlı bir şekilde ulaştıracak yerel ve sezgisel bilgilerle donanmış olarak hareket ederler.
- Zamanın ayrışsal olduğu bir çevrede yaşarlar. Her karınca bir zamanda bir yerden diğer bir yere hareket eder. Zaman normal doğadakinin aksine sürekli değildir.

KKO’nda yapay karıncalar birbirlerinden bağımsız olarak ortamda hareket ederek bireysel olarak çözüm üretmeye çalışırlar. Öte yandan aralarında iz bırak-iz takip et yöntemiyle sağladıkları dolaylı iletişim tekniğiyle daha iyi çözümlerin üretilmesinde birbirlerinden yardım alırlar. KKO algoritmalarının temel iskelet yapısı aşağıdaki gibidir.

```

Algoritma KarıncaKolonisiOptimizasyonuAlgoritması
    Parametreleri ayarla, feromon izlerini ilkle
    Sonlandırma durumuna ulaşılan dek
        ÇözümleriÜret
        YerelAramaUygula %İsteğe bağlı
        FeromonİzleriniGüncelle
    Son
Son
  
```

Burada belirtilen KKO algoritmasının her bir adımını ayrıntılı bir şekilde inceleyecek olursak;

Parametrelerin Ayarlanması ve Feromon İzlerinin İlklenmesi: Bu aşama algoritmadaki kullanılan parametrelerin ve problem uzayı üzerindeki feromon izlerinin ilklendiği kısımdır. Karınca kolonisi algoritması kendine özgü birkaç parametreyi içinde barındırır. Ayrıca problem, kendine özgü bazı parametreleri barındırabilir. Bu parametrelerin doğru bir şekilde ayarlanması oldukça önemlidir. Çünkü parametre değerleri algoritmanın performansını direk olarak etkiler. Doğru parametrenin seçilmemesi durumunda algoritma optimum sonuçlardan uzaklaşabilir. Parametre değerlerinin, performansa etkileri çok kritik olmasına karşın doğru parametre seçimi oldukça zor bir işlemdir. Çünkü her bir problem için farklı bir uygun parametre değer kümesi mevcuttur. Tezimizde kullandığımız algoritmaların parametrelerinin nasıl seçildiği ileriki bölümlerde ayrıntılı bir şekilde incelenecektir.

Öte yandan, feromon izleri program süreci esnasında küresel bilgiyi ifade eder. Sürecin başında herhangi bir küresel bilgiye ulaşılmadığından genellikle problem üzerindeki feromon izleri her bölgede eşit ve küçük değerler alacak şekilde ilklenir.

Çözümleri Üret: Algoritmanın her adımında karınca kolonisindeki her bir birey kendine ait çözümleri üretmeye çalışır. Bu çözümleri üretirken, karıncalar arama üzerindeki durumları birer birer dolaşırlar. Bir durumdan diğerine geçerken, olasılıklı bir seçim kuralını uygularlar. Bu olasılıklı seçim kuralı *random proportional transition rule* olarak adlandırılır ve iki değer bileşimine bağlıdır:

- *Hareket etme çekiciliği* olup probleme uygun sezgisel fonksiyonu ifade eder. Sezgisel fonksiyon hareket seçiminin birincil etkenini oluşturur.
- *Yapay feromonun iz seviyesi* olup öğrenilmiş seçimliliği ifade eder. Bu bilgi süreç boyunca bulunan konumdan diğerlerine geçme durumunun sıklığına ve bu vasıta ile elde edilen çözümlerin kalitesine göre değerler alır. Dolayısıyla durumlar arasındaki feromon izleri çözüme ulaşmadaki küresel bilgiyi arama uzayı üzerinde depolar.

Birçok KKO algoritma çeşidi aynı seçim kuralını kullanmakla birlikte bazı algoritmalar bu kural üzerinde bazı oynamalar yaparak performans artırımı

sağlamışlardır. Bu kuralın ayrıntılarına her bir KKO algoritma çeşidini bahsettiğimiz bölümlerde değinilmiştir.

YerelAramaUygula: KKO algoritmaları birçok zor problemin çözümünde her örnek için kendi başına iyi çözümler üretemeyebilir. Bu tip koşullarda isteğe bağlı olarak yerel arama yöntemlerinden faydalanılabilir. Yerel arama algoritmaları temel olarak verilen bir çözümü alarak onu iyileştirmeye çalışan algoritmalar ve her probleme özgü birçok yerel arama algoritması önerilmiştir. KKO ile birlikte kullanıldığında genellikle her bir karıncanın ürettiği çözümleri iyileştirme tarafında görev alırlar. Böylece, feromon güncelleştirilmesi yerel arama ile elde edilmiş daha iyi bir sonuç üzerinde gerçekleştirilmiş olur. Doğru yerel arama algoritması doğru bir şekilde kullanıldığında KKO algoritmalarının performansı iki kattan fazla artırılabilir (Stützle and Hoos, 2000). Bu yüzden problemlerin çözümünde güncel birçok KKO algoritması yerel arama yöntemleri ile birlikte çoklukla kullanılmaktadır. Fakat her probleme özgü bir yerel arama algoritması bulunmadığından yerel arama algoritmalarının seçime bağlı olarak kullanılması öngörülmüştür.

FeromonİzleriniGüncelleme: Çözümler üretildikten ve hesaplandıktan sonra arama uzayı üzerindeki feromon izlerinin seviyesi güncellenir. Güncelleme işlemi feromon depolama ve buharlaşma olmak üzere iki farklı şekilde gerçekleştirilir. Feromon depolama işlemi, her karınca veya sadece bazı özelliğe sahip karıncalar tarafından gerçekleştirilir ve feromonlar sadece oluşturulan çözüm üzerindeki yollar üzerinde depolanır. Buharlaştırma işlemi ise negatif geri-beslemeyi sağlar ve genellikle tüm olası yollar üzerinden belli oranda feromonun azaltılmasıyla sağlanır.

Her döngü sonunda o ana kadar atılan en iyi tur kaydedilir (güncellenir). Programın bitirilmesi için farklı yaklaşımlar kullanılabilir. Bunlar belli bir döngü sayısı, zaman veya optimum sonuca ulaşılması olabilir. Program bitirilme koşuluna geldiğinde atılan en iyi tur, algoritmanın ürettiği çözüm yolunu verir.

KKO algoritmasının bilinen tüm çeşitleri Gezgin Satıcı Problemi üzerinde denenerek sunulmuştur. Bu şaşırtıcı değildir. Çünkü doğadaki karıncalar da yem arama yöntemlerinde en kısa yol problemi üzerinde çalışmaktadırlar. Bu da Gezgin Satıcı Problemine KKO algoritmalarının kolay adaptasyonunu sağlamıştır. En çok bilinen KKO çeşitleri Karınca Sistemi (Ant System) (Dorigo et al., 1996), Karınca Kolonisi Sistemi (Ant Colony System) (Dorigo and Gambardella, 1997),

Elitist Karınca Sistemi (Elitist Ant System) (Dorigo et al., 1996), Sıra-Tabanlı Karınca Sistemi (Rank-Based Ant System) (Bullnheimer et al., 1999), Eniyi-Enkötü Karınca Sistemi (Best-Worst Ant System) (Cordon et al., 2000) ve Max-Min Karınca Sistemi (Max-Min Ant System) (Stützle and Hoos, 2000) algoritmalarıdır. Aşağıda bu algoritmalar ve arasındaki farklar üzerinde durulmuştur.

2.2.1.3 Karınca sistemi (ant system)

Karınca Sistemi (KS), Gezgin Satıcı Problemini (GSP) çözmek için gerçekleştirilen ilk karınca kolonisi algoritmasıdır. KS algoritmasında her turda koloninin her bireyi kendi olasılıklı çözümünü oluşturur. Feromon, turlar tamamlandıktan sonra atılan turlar üzerinde ilgili çözümle orantılı olarak her karınca tarafından depolanır.

Karınca Sistem’inde GSP için çözümü şu şekilde bulunur: her bir adımda, bir karınca k , v_i şehrinden v_j şehrine

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in \Omega} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} & \text{eğer } j \in \Omega \\ 0 & \text{aksi halde} \end{cases} \quad 2.1$$

olasılığına göre hareket eder. Burada τ_{ij} ; v_i ve v_j şehirleri arasındaki feromon yoğunluğu, α ; τ_{ij} için dengeleme parametresi, η_{ij} ; v_i ’ye olan görünürlük ($\eta_{ij} = 1 / d_{ij}$), β ; η_{ij} etkisini dengeleme parametresi, Ω ziyaret edilmemiş şehirler kümesi, d_{ij} ise v_i ile v_j arasındaki uzaklık anlamındadır. Buradaki α ve β parametreleri, algoritmanın davranışını büyük ölçüde etkilemektedir. $\alpha = 1$ olduğunda feromon bilgisi devre dışı kalmakta ve algoritma olasılıklı bir açgözlü arama algoritmasına dönüşmektedir ve bu da algoritmanın performansını düşürür. Eğer $\beta = 0$ olursa, sezgisel bilgi devreden çıkmış olacaktır. Sadece feromon bilgisine dayalı bir arama ise tamamen rastlantısal turların oluşmasına neden olur ve üretilen sonuçlar optimuma yaklaşamaz. Dolayısıyla α ve β parametreleri belli oranda katkıda bulunmalıdır. Fakat bu katkı oranının yani parametre değerlerinin tespiti daha önce de bahsettiğimiz gibi oldukça zordur.

Her bir karınca kendi turunu tamamladıktan sonra feromon güncelleme işlemi yapılır. KS algoritmasında her karıncanın feromon bırakmasına izin verilir.

Ayrıca feromon buharlaştırma işlemi, depolama işlemi sırasında ortak bir formülle gerçekleştirilir. İşte bu feromon bırakma ve buharlaştırma işlemi aşağıdaki kurala göre gerçekleştirilir.

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad 2.2$$

Burada ρ buharlaşma katsayısını ifade eder. Ayrıca $\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k$ ve

$\Delta \tau_{ij}^k = \begin{cases} 1/L_k, & \text{eğer } (i,j) \in T_k(t) \\ 0, & \text{eğer } (i,j) \notin T_k(t) \end{cases}$ olup m kolonideki toplam karınca sayısını, L_k k . karıncanın attığı turun uzunluğunu, $T_k(t)$ k . karıncanın t zamanında (döngüsünde) attığı turu göstermektedir.

2.2.1.4 Elitist karınca sistemi (elitist ant system)

Karınca Sistemi algoritması ilk olarak küçük ölçekli problemler üzerinde denenmiş ve iyi sonuçlar vermiştir. Fakat orta ve büyük ölçekli problemlerde aynı başarıyı gösterememiştir. Karınca Sisteminin bu özelliği, farklı KKO algoritmalarının ortaya çıkmasına neden olmuştur. Yani diğer KKO metotlarından çoğu KS algoritmasının dezavantajlarını ortadan kaldırmak için sunulan yöntemlerdir. Bu amaçla ortaya atılan ilk yöntem Elitist Karınca Sistemi (EKS) algoritmasıdır.

KS sistemi algoritmasında tüm karıncaların feromon depolamasına izin vermektedir. Bu davranış, sistemde o ana kadar üretilen en iyi sonuçları üreten yollara olan ilginin azalmasına neden olur. Çünkü her turda yeni rastgele turlar üretilmektedir ve her kenar üzerinde bir sonraki karıncaları etkileyecek belli oranda feromon depolanır. EKS bu dezavantajı ortadan kaldırmak için o ana kadar bulunan en iyi sonuç turu üzerinde fazladan e/L^* (L^* o ana kadar bulunan en iyi karınca turunun uzunluğu) kadar feromon depolanır. Böylece o ana kadar bulunan en iyi tur güçlendirilmiş olur. Bu sayede, karıncaların bir sonraki turlarda en iyi turun etrafında yeni iyi sonuçları üretmeleri istenerek daha hızlı bir şekilde iyi sonuçlar üretilmesi hedeflenir.

2.2.1.5 Karınca kolonisi sistemi (ant colony system)

Karınca Kolonisi Sistemi (KKS) algoritması, KS algoritmasının ilk ciddi anlamdaki geliştirilmiş çeşididir.

Bu algoritmada biri yerel diğeri de küresel olmak üzere iki tip feromon güncelleme mekanizması işletilir. Yerel feromon güncelleme işlemi, (her karınca çözümlerini ürettikleri aşamasında) bir şehirden diğerine geçerken aşağıdaki formüle göre güncellenir:

$$\tau_{ij} = (1 - \phi) \cdot \tau_{ij} + \phi \cdot \tau_0 \quad 2.3$$

Burada ϕ feromon azaltma katsayısıdır ($0 < \phi \leq 1$). τ_0 en yakın komşu (nearest neighbor) algoritmasına göre üretilen bir çözümün değerini ifade eder ve bu değer aynı zamanda feromon ilkleme işleminde her kenar üzerinde eklenen ilk feromon seviyesini ifade eder. Global feromon güncelleştirme işlemi ise tüm karıncalar turlarını tamamladıktan sonra uygulanır. Karınca Sistemi farklı olarak küresel feromon güncelleme işleminde sadece o ana kadar ki en iyi tur üzerinde feromon biriktirilmesine izin verilir.

KKS ile birlikte gelen bir değişiklik de olasılıklı seçim kuralı üzerindedir. KKS'a göre, i şehirdeki bir karınca için ziyaret edilecek yeni j şehri *pseudo-random proportional rule* adı verilen şu olasılıklı seçim kuralına göre belirlenir;

$$j = \begin{cases} \arg \max_{j \in \Omega} \{ [\tau(i, j)] \cdot [\eta(i, j)]^\beta \} & \text{eğer } q \leq q_0 \\ J & \text{aksi durumda} \end{cases} \quad 2.4$$

Burada q bir rastgele sayı ($0 \leq q \leq 1$), q_0 bir parametre ($0 \leq q_0 \leq 1$), J Karınca Sisteminde kullanılan olasılıklı seçim kuralına göre seçilen şehri ifade eder. Her bir i şehirden j şehrine geçişte yeni bir q değeri üretilir ve bu kurala göre şehir belirlenir. Bu kural vasıtasıyla algoritmanın çokça rastlantısal çalışması önlenerek genellikle en iyi ihtimalli olan şehirlerin seçilmesine zorlanmıştır. Bu da çözümler arasındaki varyansın azalmasına neden olur. Böylece her bir çözümün en iyi çözümlerin etrafında aranmasını güçlendirmiştir.

2.2.1.6 Sıra-tabanlı karınca sistemi (rank-based ant system)

Sıra-Tabanlı Karınca Sistemi (STKS) algoritması, yine sadece en iyi turlar üzerinde odaklanmayı hedef edinerek sunulan bir KKO çeşitidir. STKS her iterasyon sonunda atılan turları en iyiden en küçüğe sıralar. Sadece, içlerinden en iyi r tane turu atan karıncaların feromon depolamasına izin verilir. Buna ek olarak tüm turlar boyunca bulunan en iyi tur üzerinde de feromon depolanır. Bu

depolama işlemi her karınca için aynı oranda tutulmaz. Sıralamadaki yerleriyle orantılı bir ağırlığa göre gerçekleştirilir. O ana kadarki en iyi tur üzerine w ağırlıklı (bu değer genelde popülasyon sayısının dörtte biri olarak alınır) feromon depolanır. O anki iterasyon sonucunda elde edilen turlardan da en iyi $w-1$ tane en iyi tur feromon güncellenmesi için seçilir. Sonuçta, her iterasyon sonunda en iyi tur üzerinde w/L^* kadar feromon bırakırken, sıralama sonucunda en iyi $w-1$ kadar tur üzerindeki güncelleme aşağıdaki formülle ifade edilir:

$$\sum_{r=1, \dots, w-1} (w-r) \Delta \tau_{ij,r}(t) \quad 2.5$$

Burada $\Delta \tau_{ij,r}(t)$ r sırasındaki karıncanın feromonları olup attığı turun uzunluğuyla ters orantılıdır ($\Delta \tau_{ij,r}(t) = 1/L^*$ eğer $(i,j) \in T^r(t)$).

Bu feromon güncelleme mekanizması ile en iyi turlar ve o ana kadar atılmış en iyi tur belli bir ağırlıkta güçlendirilmiş olur. En iyi yolları güçlendirmek karıncaların sadece iyi turlar etrafında dolaşmalarını ve hızlı bir şekilde yeni iyi çözümleri üretme olasılığını güçlendirir.

2.2.1.7 Eniyi-enkötü karınca sistemi (best-worst ant system)

Eniyi-Enkötü Karınca Sistemi (İKKS) algoritması 1999 yılında Cordon ve arkadaşları tarafından geliştirilmiştir. İKKS algoritması iki aşamalı feromon güncelleme mekanizmasına sahiptir.

Birinci aşamada her koloni turu sonunda sadece o ana kadar ki en iyi ve en kötü turlar üzerinde feromon değerleri güncelleştirilir. Bunun için KKS algoritmasında olduğu gibi en iyi turun uzunluğuna ters orantılı olarak feromon güncellenir. Daha sonra o ana kadar ki en kötü turun üzerinde feromon buharlaştırma işlemi gerçekleştirilir. Bu işlem gerçekleştirilirken en iyi yol ile en kötü yolların çakıştığı kenarlar üzerinde feromon buharlaştırılması yapılmamaya dikkat edilir. Böylece kesişmeden kaynaklanacak yan etkiler önlenmiş olur.

Sadece en iyi yollar üzerinde feromon güncelleme gerçekleştirme algoritmanın hırçınlığını arttırmakla birlikte algoritmadaki çeşitlilik oranını azaltmakta ve çözümler arası varyasyonu gittikçe düşürmektedir. Bunu önlemek için ikinci aşamada Genetik Algoritmalarda kullanılan mutasyon yönteminden

ilham alınarak tüm yollar üzerindeki feromon değerleri bir olasılıkla mutasyona tabi tutar. Bir i ile j şehri arasındaki feromon seviyesi üzerindeki mutasyon işlemi:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{ij}(t) + \Delta\tau_{mut} & \text{if } a = 0 \\ \tau_{ij}(t) - \Delta\tau_{mut} & \text{if } a \neq 0, \end{cases} \quad 2.6$$

şekilde formüle edilir. Burada, $\Delta\tau_{mut}$ değeri mutasyon miktarı olup iterasyon sayısına ve en iyi tur üzerindeki ortalama feromon miktarına göre değerler alır. $a \in \{0,1\}$ olup rastgele bir sayıyı temsil eder.

Ayrıca İKKS algoritması, KKS'nde kullanılan ve daha hırçın davranan olasılıklı seçim kuralını kullanmaktadır. Feromon güncelleştirme mekanizmaları ile birlikte bu seçim kuralının kullanılması hem en iyi turlar üzerinde yeni sonuçları aramaya hem de sonuçlar üretilirken yerel minimumlardan kaçınılmasına neden olmaktadır.

2.2.1.8 Max-min karınca sistemi (max-min ant system)

Diğer birçok KKO çeşidinde görüldüğü gibi sadece en iyi tur üzerinde feromon depolamaya izin vermek KKO algoritmasının performansını arttırmada önemli bir etkidir. Fakat bu strateji programın hızlı bir şekilde durağanlık pozisyonuna geçmesine neden olur. Hep aynı çözüm üzerindeki kenarlara feromon depolanacağından iterasyonların çoğunda defalarca aynı turlar çözüm olarak üretilir.

Max-Min Karınca Sistemi (MMKS) algoritması da yine sadece en iyi tur üzerinde feromon depolarken bahsini ettiğimiz dezavantajı kenarlar üzerinde depolanabilecek maksimum, minimum feromon seviyelerini kısıtlayarak halleder. Maksimum (τ_{max}) ve minimum (τ_{min}) değerler problem bağımlı olup Gezgin Satıcı Problemi için kenarın ortalama uzunluğu ile orantılı olarak belirlenir. Bu yolla feromon izi üzerindeki etki azaltılmış (kısıtlanmış) olur.

MMKS algoritmasında diğer algoritmalarından farklı olarak feromon değerleri programın başında küçük değerlerle değil feromon değerlerinin alabileceği en büyük değer olan τ_{max} değerine eşlenir. Problem ilerledikçe tercih edilmeyen kenarlar üzerindeki feromon seviyesi buharlaştırma yolu ile yavaş yavaş azalır. Buna karşın en iyi yollar üzerinde feromon depolanmaya devam

edildiğinden en iyi turların feromon seviyesi $\tau_{\max}$ değerine yaklaşmaya çalışır. Bu yolla üretilen çözümün çeşitliliği artırılmış olur.

Feromon seviyesinin kısıtlanması çeşitliliği arttırsa da uzun süren çalıştırmalarda MMKS algoritması zamanla durağanlığa geçtiği gözlemlenmiştir. Bunu ortadan kaldırmak için feromon yumuşatma adı verilen bir mekanizma algoritmaya eklenmiştir. Feromon yumuşatma, her bir kenar için kendi üzerindeki feromon miktarı ile $\tau_{\max}$ arasındaki farkla orantılı olarak artırılması işlemidir. Bu yolla, öncelerden tespit edilen iyi turlar zaman geçtikçe çabuk unutulmaz ve o kenarlar üzerindeki feromon değerleri yüksek tutulmaya çalışır. Böylece bir durumdan (şehirden) diğer duruma geçişte seçilebilecek kenarların olasılıkları belli miktarda artacağında çözümler çok uzun çalıştırmalarda bile farklı sonuçları üretebilir (durağanlıktan kurtulabilir).

2.2.2 Parçacık sürü optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (Particle Swarm Optimization) ilk olarak Kennedy ve Eberhart (1995) tarafından ortaya atılmıştır. Yapılan ilk benzetimler Heppner ve Grenander (1990) tarafından kuş sürülerinin hareket modeli ve kuş sürülerinin yiyecek araştırma yöntemlerinden ilham alınarak gerçekleştirilmiştir. Daha sonra bu benzetim güçlü bir optimizasyon metoduna dönüştürülmüş ve bu yöntem Parçacık Sürü Optimizasyonu (PSO) olarak adlandırılmıştır.

PSO’ da, parçacık adı verilen bireyler topluluğu, bir problemin veya fonksiyonun arama uzayı üzerinde dağıtılır ve her birey kendi konumunun uygunluğunu hesaplar. Bu uygunluk fonksiyonu objektif fonksiyon olarak adlandırılır. Daha sonra her parçacık, arama uzayındaki hareketini kendi geçmişindeki en iyi konumu ve çevresindeki parçacıkların en iyi konumlarıyla bağlantılı bir şekilde rastgele belirler. Tüm parçacıklar bu yeni konumlarını belirledikten sonra bir sonraki iterasyona geçilir. Sistem bu haliyle yiyecek bulmak için ortaklaşa hareket eden kuş sürülerinin hareket mekanizmasını andırır ve aynen kuşların hedefe beraber yaklaştıkları gibi parçacıklar da hep birlikte optimum noktaya doğru hareket ederler.

Her bir parçacık bünyesinde üç farklı vektör bilgisi barındırır. Bunların her birisi D -Boyutludur (D , problemin arama uzayının boyut sayısını ifade eder). Bu üç vektörden ikisi parçacığın şu anki konumunu ve parçacığın o ana kadar ki en iyi konumunu ifade eden vektörlerdir. Üçüncü vektör ise hız vektörüdür.

Parçacığın o anki konum vektörü arama uzayındaki bir koordinat noktasını temsil eder. Bu konum her iterasyonda değişir (güncellenir) ve problemin bir çözümü olarak kabul edilerek objektif fonksiyon değeri hesaplanır. Eğer yeni konum, parçacığın o ana kadar bulunduğu diğer tüm konumlardan iyi bir pozisyonda ise o parçacığa ait en iyi konum vektörü güncellenir. En iyi konum vektörü p_{best} değişkeni olarak isimlendirilir. Daha sonra parçacığın yeni konumu hız vektörünün o anki konum vektörüne eklenmesi yolu ile tespit edilir. Hız vektörü de her iterasyonda, parçacığın o anki konumunun kendi en iyi konumuna ve diğer parçacıkların en iyi konumlarının uzaklıklarına bağlantılı olarak bir değer alır. Böylece parçacık hareketini tamamladığında kendi en iyi konumuna ve diğer iyi konumlara yakın bir yerde konumlanmış olur. Bu da, parçacıkların program ilerledikçe optimum noktaya doğru beraberce hareket etmelerini sağlar. Temel PSO algoritmasına ait yalancı-kod parçası aşağıdaki gibidir:

Algoritma ParçacıkSürüOptimizasyonu

Parçacıkları D -Boyutlu arama uzayında konumlandır ve hızlarını ilkle

Sonlandırma durumuna ulaşılan dek

Her parçacık için;

Uygunluk (objektif) fonksiyonu hesapla

Daha iyisi bulunduğuyorsa en iyi konumunu güncelle

Tüm parçacıklar içinde en iyi konumda ise;

En iyi konumu güncelle

Hız vektörünü belirle ve yeniden konumlandır.

Son

Son

Koddan da anlaşıldığı gibi algorithmada öncelikle probleme uygun olarak parçacık konumları arama uzayında ilklenmekte, daha sonra kendi uygunluk fonksiyonlarına göre her bir iterasyon sonunda yeni bir yere konumlanmaktadırlar. İlk PSO çeşitlerinde bir parçacığın yeni konumu aşağıdaki formüle göre belirlenmektedir:

$$\begin{cases} \vec{v}_i = \vec{v}_i + \vec{U}(0, \phi_1)(\vec{p}_i - \vec{x}_i) + \vec{U}(0, \phi_2)(\vec{p}_g - \vec{x}_i) \\ \vec{x}_i = \vec{x}_i + \vec{v}_i \end{cases} \quad 2.7$$

Burada $\vec{v}_i$ parçacığın hız vektörüdür ve bu parçacığın uçuş yönünü hafızada tutmayı sağlar. Bu hafıza bir momentum olarak görülebilir. Yani her defasında hız

vektörüne eklenerek elde edilen hız vektörlerinde parçacığın yönü aniden değiştirilmemiş olur.

$\vec{U}(0, \phi_1)(\vec{p}_i - \vec{x}_i)$ parçacığın kendi en iyi konumuna göre olan performansını ifade eder. $\vec{U}(0, \phi_1)$ 0 ile ϕ_1 arasında bir rastgele değeri, $\vec{p}_i$ ise parçacığın en iyi konumu gösterir. Bu bileşen parçacığın kendi en iyi konumuna doğru yönelmesinde ve bu konum etrafında yeni iyi konumlar araştırmasında yardımcı olur. Son bileşen $\vec{U}(0, \phi_2)(\vec{p}_g - \vec{x}_i)$ ise sosyal bileşen olarak adlandırılır. Burada $\vec{p}_g$ komşularının en iyi pozisyonunu ifade eder. Bu bileşen de parçacığın konumun kendi komşularına veya tüm parçacıkların en iyi konumuna göre performansını ifade eder. Dolayısıyla, bu bileşen tüm parçacıkların sistemde ulaşılan en iyi konumlara doğru yönelmelerinde etkili olur.

Aşağıdaki alt başlıkta algoritmanın davranışını etkileyen diğer önemli etkenlerden bahsedilecektir. Ayrıca, algoritmanın yapısı üzerindeki önerilen ve performans artışı sağlayan bazı önemli PSO çeşitlerinden bahsedilecektir.

2.2.2.1 Algoritma davranışını etkileyen ayrıntılar

PSO' da parçacık ilklenmesi, bitirme koşulu, parçacıklar arasındaki komşuluk ağı (topolojisi) ve kendine özgü parametreleri algoritmanın ilerleyişinde etkileri olduğu görülmüştür.

Parçacıkların programın başlangıcında rastgele konumlandırılır. Bu ilklenme durumu önemsiz görünse de, algoritmanın ilerleyişinde direk olarak etkide bulunur. Çünkü bu konumlandırmalar parçacık sürüsünün ne kadar genişlikte arama uzayını kapsadığını ve parçacıkların arama uzayında ne kadar iyi dağıtıldığını ifade eder. Eğer arama uzayındaki alanlar ilk sürü tarafından iyi kapsamazsa, algoritmanın kapsamayan alanda bulunan bir optimum noktayı bulması oldukça güçleşir. Dolayısıyla parçacıklarının her bir bileşeni o boyutta tanımlanan maksimum ve minimum değerler arasında rastgele ilklenmesi gereklidir. Bu ilklenme durumu aşağıdaki şekilde ifade edilir;

$$x(0) = x_{\min,j} + r_j(x_{\max,j} - x_{\min,j}), \quad \forall j = 1, \dots, n_x, \quad \forall j = 1, \dots, n_s \quad 2.8$$

Burada $x_{\max,j}$ ve $x_{\min,j}$ arama uzayındaki her bir j boyutun maksimum ve minimum aralıklarını, r_j ise j boyutu için rastgele üretilen bir değeri ifade eder. Böylece parçacık sürüsünün tüm arama uzayını olabildiğince fazla kapsamaları sağlanır.

PSO'daki diğer önemli etken, algoritmanın sonlandırılma koşullarıdır. Sonlandırılma koşulunun seçimi, çözümün performansında veya zamanın iyi kullanımında etkilidir. Yanlış sonlandırma koşulu seçildiğinde PSO yarı-optimum sonuçlara ulaştığında sonlandırılabilir veya uygun çözümler bulunduğu halde algoritma devam ettirilebilir. Bu da zaman kaybına ve defalarca aynı uygunluk fonksiyonlarının üretilmesine neden olur. Dolayısıyla, sonlandırma koşulu problemin doğasıyla doğru orantılıdır. Buna göre bazı sonlandırma koşulları aşağıdaki şekilde olabilmektedir;

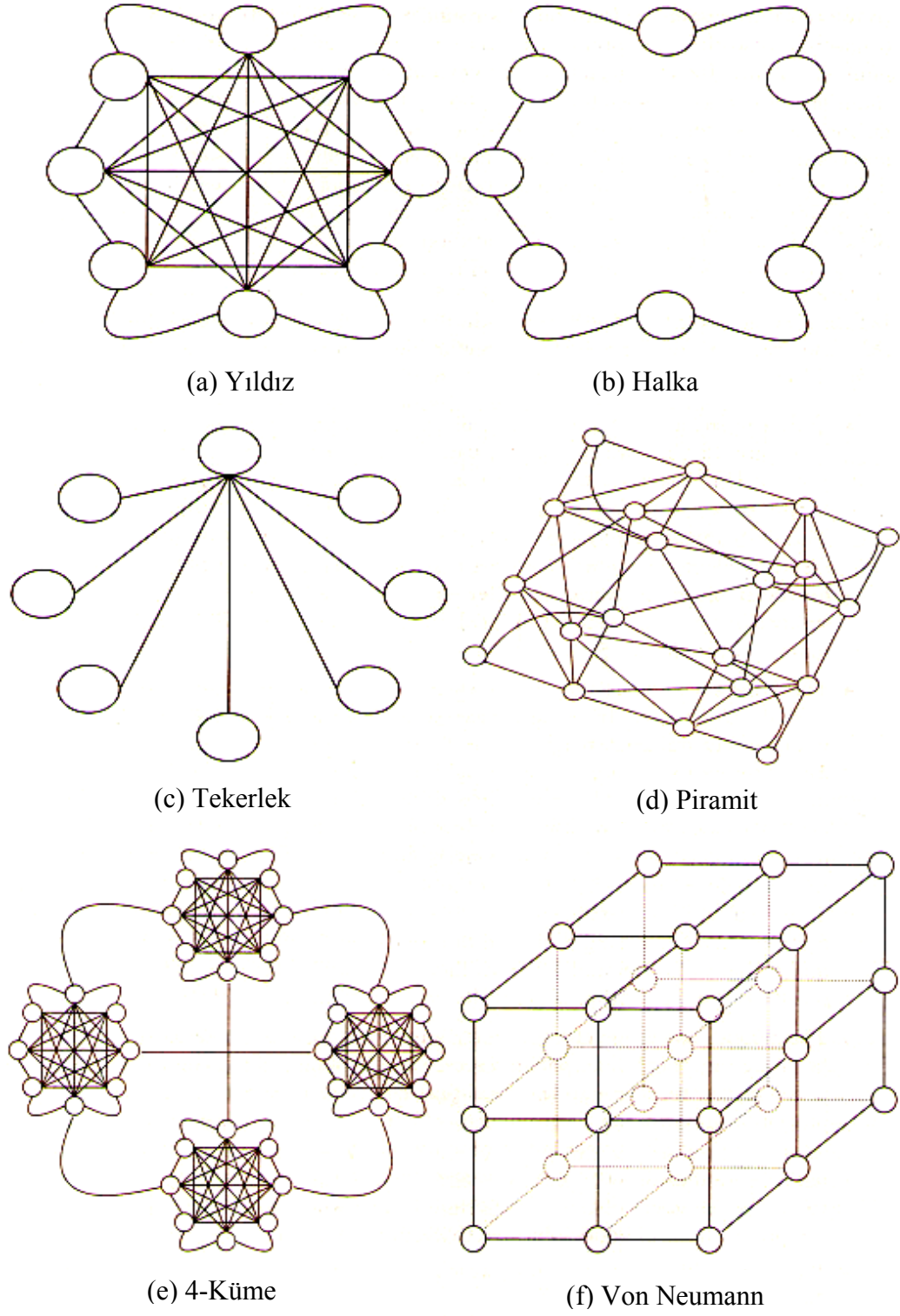
- ***Maksimum iterasyon veya uygun fonksiyonu hesaplama sayısına gelindiğinde sonlandır***: Maksimum sayı az tutulduğunda iyi çözümler bulunmadan algoritmanın sonlandırılacağı rahatça görülür. Fakat bu ölçüt algoritmanın iyi sonuçlara yeterince ulaşmadığı anlaşıldığında sonlandırmak için ideal olabilir.
- ***Kabul edilebilir bir çözüme ulaşıldığında sonlandır***: Bu koşulla, her hangi bir parçacığın optimum noktaya önemsenmeyecek derecede az bir farkla yaklaştığı durumda süreç sonlandırılır. Burada önemsenmeyecek derecede olan hata oranı *epsilon* ile ifade edilir. Eğer bu hata oranı çok küçük olursa, algoritma hiç sonlanma koşulunu sağlayamayabilir ve sonsuza dek devam edebilir. Eğer çok büyük belirlenirse, problem yarı-optimal sonuçları bulduktan sonra sonlanabilir.
- ***Belli iterasyon sonunda bir iyileştirme gözlenemezse durdur***: İyileştirmeyi gözlemlemenin birçok yolu vardır. Örneğin, parçacıkların konumları arasındaki fark ortalama çok küçük olduğu zaman, birkaç iterasyon sonunda ortalama hız değeri sıfıra çok yaklaştığı gözlemlendiğinde veya belli iterasyon sonunda çözümde herhangi bir iyileşme olmadığı durumlarda sonlandırma gerçekleştirilebilir. Bu koşulların dezavantajı, maksimum durağan iterasyon sayısının veya performansı artışı olup olmadığına karar veren eşik değerinin tespitinin zorluğudur.

PSO tamamen sosyal iletişim üzerine kurulu bir yaklaşımdır. Dolayısıyla bu sosyal iletişimin şekli algoritmanın performansını doğrudan etkileyecektir. Bir sosyal ağda bilgi akışı (1) ağ içindeki elemanların bağılılık derecesine (2) ağdaki kümelenme oranına ve (3) elemanların komşularına olan ortalama uzaklığına bağlıdır. Dolayısıyla, sürü içinde oluşturulan farklı şekillerdeki komşuluk ağları

(topolojileri) bilgi akışının seyrini değiştirmekte bu da algoritmanın davranışında değişikliklere neden olmaktadır.

PSO için birçok sosyal ağ yapı modeli önerilmiştir (Engelbrecht, 2007). Fakat bu ağ yapılarından hiç birisi tüm problemler için en iyi sonucu üretemez. Yani yine seçim probleme bağlıdır. PSO için önerilen başlıca ağ yapıları aşağıdaki gibidir;

- **Yıldız Topolojisi:** Tüm parçacıkların birbirleriyle iletişim kurabildikleri ağ yapısıdır (Şekil 2.3-a). Bu modelde, en iyi komşu konumu, tüm parçacıklarla iletişime geçilerek bulunacağından en iyi komşu konumu aynı zamanda en iyi parçacık konumunu ifade eder. Dolayısıyla parçacıklar o ana kadarki en iyi parçacık konumuna doğru hareket ederler. Bu model PSO çeşitlerinin çoğunda kullanılmaktadır ve hızlı bir şekilde yakınsama sağlar. Fakat bu yapıyla yerel optimum noktalara takılma riski büyüktür.
- **Halka Topolojisi:** Bu sosyal ağda her parçacık belli bir sayıda parçacıkla iletişim kurabilir. Her bir parçacık yalnız iki parçacık ile iletişim kurduğu durumda topoloji Şekil 2.3-b deki gibi olacaktır. Bu yapıda her parçacık kendi komşulukları içinde bulunan en iyi çözüme doğru hareketini sürdürür. Bu yapıda, yakınsama yavaş gerçekleşmesine rağmen yerel minimumlardan kaçma olasılığı yüksektir. Çünkü süreç içerisinde arama uzayının kapsamı daha az daralır.
- **Tekerlek topolojisi:** Her bir parçacığın birbirinden izole olduğu fakat sadece bir parçacığın tüm diğer parçacıklar ile iletişim kurabildiği bir yapıdır (Şekil 2.3-c). Bu yapıda bir parçacık diğer tüm parçacıklardan bilgi toplayarak performanslarını karşılaştırır ve kendi pozisyonunu en iyi komşusuna göre değiştirir. Eğer bu parçacığın yeni konumu daha iyi bir sonuç üretirse, bu diğer üyelere iletilir. Yapıdan da anlaşılacağı üzere, bu topoloji kullanıldığında iyi sonuçlar bulma oranı diğerlerine oranla daha düşüktür.
- **Piramit topolojisi:** Üç boyutlu tel-kafesi modeli olup Şekil 2.3-d de gösterilmiştir.



Şekil 2.3. PSO algoritmalarında önerilen bazı sürü ağ yapıları (Engelbrecht, 2007).

- **4-Küme topolojisi:** Şekil 2.3-e de gösterilen bu ağ yapısında, dört küme diğer kümelerle ikili bağlantılarla bağlıdır. Ayrıca küme içindeki her bir parçacık beş komşu ile bağlanmıştır.

- **Von Neumann topolojisi:** Tüm parçacıkların bir ızgara (grid) yapısı oluşturacak şekilde birbirlerine bağlı olduğu yapıdır (Şekil 2.3-f). Birçok problemin çözümünde bu yapı kullanıldığında diğer ağ yapıları ile elde edilen sonuçlardan çok daha iyi sonuçlar elde edildiği görülmüştür (Kennedy and Mendes, 2002; Peer et al., 2003).

2.2.2.2 Parçacık sürü optimizasyon üzerinde temel değişiklikler

Temel PSO algoritması birçok problem üzerinde denenmiş ve başarılı sonuçlar elde edilmiştir. Fakat aynı sonuçlar, temel PSO algoritmasının her zaman için iyi sonuçlar verme konusunda problemler yaşadığını da ortaya koymuştur. Bu durumun ortadan kalkması için, temel PSO algoritması üzerinde birçok değişiklikler önerilmiştir. Bu değişiklikler algoritmanın her aşamasındaki ayrıntılarda gerçekleşebilmektedir. Dolayısıyla tezimizde hepsinden bahsetmek mümkün değildir. Burada çokça bilinen bazı değişikliklere yer verilmiştir.

Hız Sıkıştırma: Parçacıkların kendi en iyi ve komşu en iyi konumlara çok uzak olduğu durumlarda, parçacıklar çok uzun mesafeli yer değişimleri gerçekleşir. Sonuçta, parçacıklar arama alanı dışında bazı bölgelerde konumlanabilirler. Bu durumda parçacıkların arayış kabiliyetleri sınır kısıtları içinde sıkıştırılmalıdır. Bunun için kullanılan yöntem hızın sıkıştırılmasıdır. Eğer parçacık hızı o boyuta ait olarak belirlenen maksimum hızı $V_{\max,j}$ geçtiği tespit edildiğinde en fazla maksimum hıza sahip olmasına izin verilir. $V_{\max,j}$ nin değeri önemlidir. Büyük $V_{\max,j}$ değerleri küresel keşif kabiliyetini artırırken, küçük $V_{\max,j}$ değerleri yerel konumlar etrafında hızlı odaklanmayı sağlar. Diğer yandan, hız sıkıştırması bir parçacığın atabileceği maksimum adım boyutunun değiştirdiği gibi parçacığın hareket yönünün değişmesine de neden olur. Bu da parçacığın istenilen doğrultudan daha farklı bir yöne doğru hareket etmesini sağlar ve optimum sonuçlara doğru parçacığın ilerlemesine engel olur. Diğer bir problem de tüm parçacıkların maksimum hıza sahip olduğu durumda ortaya çıkar. Eğer böyle bir durum engellenmeyecek olursa, tüm parçacıklar sadece arama uzayının sınır hatlarında çözümü ararlar. Böylece sınır hattı üzerinde olmayan optimum noktayı bulmaları olanaksız hale gelir.

Eylemsizlik Ağırlığı: Eylemsizlik ağırlığı (inertia weight) ilk olarak Shi ve Eberhart (1998) öne sürülmüştür ve şu an birçok PSO çeşidi eylemsizlik ağırlığını kullanmaktadır. Bu ağırlık değişkeni, hız sıkıştırması ihtiyacını ortadan kaldırdığı gibi sürünün keşif ve odaklanma arasındaki dengenin kontrol mekanizmasını

gerçekleştirir. Eylemsizlik ağırlığı önceki hızın yeni hıza ne kadar katkıda bulunacağını belirler ve böylece eski uçuş yönlerinin yeni hıza etkisini kontrol eder. Buna göre parçacığın yeni hız vektörü aşağıdaki formüle eşit olacak şekilde değişmiş olur:

$$\vec{v}_i = w\vec{v}_i + \vec{U}(0, \phi_1)(\vec{p}_i - \vec{x}_i) + \vec{U}(0, \phi_2)(\vec{p}_g - \vec{x}_i) \quad 2.9$$

Eylemsizlik ağırlığının da alacağı değer PSO'nun davranışında değişikliklere neden olur. Bu değer büyük seçildiğinde keşif kabiliyeti artar, küçük seçildiğinde ise yerel odaklanma gücü artmış olur. Çok küçük seçilen değerler ise sürünün keşif özelliğini tamamen ortadan kaldırabilir. Durum böyle olmasına karşın doğru ağırlık değeri probleme göre değişir ve tespiti oldukça zordur. Bu değerın tespit edilmesinin zorluğu dinamik değer atama yoluna gidilmesine neden olmuştur. Literatürde her iterasyon için farklı rastgele değer atayan (Peng et al., 2000), doğrusal (Suganthan, 1999) veya doğrusal olmayan (Peram et al., 2003) değer azaltan veya bulanık uyarlamalı yöntem kullanan (Shi ve Eberhart, 2001) dinamik ağırlık atama yöntemleri mevcuttur.

Büzme Katsayısı: Eylemsizlik ağırlığına oldukça benzeyen bir yaklaşımdır. Yeni hız değerini büzme katsayısı adı verilen bir katsayı vasıtasıyla büzerek keşfetme-odaklanma dengesini kurar. Bu katsayı χ ile ifade edilir ve bu katsayı ile birlikte hız vektörü aşağıdaki şekilde değiştirilir:

$$\vec{v}_i = \chi[\vec{v}_i + \vec{U}(0, \phi_1)(\vec{p}_i - \vec{x}_i) + \vec{U}(0, \phi_2)(\vec{p}_g - \vec{x}_i)] \quad 2.10$$

Burada;

$$\chi = \frac{2\kappa}{\left|2 - \phi - \sqrt{\phi(\phi - 4)}\right|} \quad 2.11$$

ve $\phi = \vec{U}(0, \phi_1) + \vec{U}(0, \phi_2)$ dır. Bu işlemler $\phi \geq 4$ ve $\kappa \in [0, 1]$ kısıtları içerisinde uygulanmalıdır. Büzme etkisi eylemsizlik ağırlığı yaklaşımına denk sonuçlar üretir ve aynı amacı güder. İki yaklaşım arasındaki temel farklar şu şekildedir:

- Hız sıkıştırma işlemi eylemsizlik ağırlığında gerekebilirken büzme katsayısı için gereksizdir.

- Büzme modeli her zaman için verilen kısıtlar içinde yakınsamayı garantiler
- Parçacıkların yön değişimi sadece ϕ_1 ve ϕ_2 sabitlerini üzerinden gerçekleştirilebilir.

3. RENKLİ GÖRÜNTÜ KESİMLEME

Görüntü kesimleme resmin homojen parçalara ayrılması veya resim içindeki nesnelerin birbirinden ayrıştırılması işlemidir ve görüntü kesimleme birçok uygulama için faydalıdır. Kesimleme sonuçlarından, bir resim bölgesindeki istenen bölgeleri ve nesneleri tespit etmek mümkündür. Bu da görüntü analiz ve tanımda önemli bir rol oynamaktadır.

Geçmişte gri-tonajlı görüntü kesimlemede birçok farklı yöntem geliştirilmiştir. Fakat (1) renkli resimlerin gri-tonajlı resimlerden çok daha fazla bilgi içermesinden, (2) bilgisayar teknolojisinin gelişmesiyle, bilgisayarların renkli resimler üzerindeki işlemleri artık hızlıca yapabiliyor güçte olmasından dolayı renkli görüntü kesimleme son yıllarda birçok araştırmacının dikkatini çekmektedir (Chen et al., 2001).

Renkli görüntü kesimleme, genellikle gri-tonajlı görüntü kesimleme tekniklerinin renk uzayları üzerinde işlem yapmaya uygun hale getirilerek genişletilmesi şeklinde gerçekleştirilir. Herhangi bir kesimleme algoritması, seçilen renk uzayının her bir bileşenine ayrı ayrı uygulanır veya tüm bileşenlerin hepsi için bir ölçüt belirlenir. İşlem sonucunda elde edilen sonuçlar birleştirilerek son çözüm elde edilir. Fakat bu işlemler sonucunda elde edilen sonuçlar gri-tonajlı resimlerde elde edilenlerden farklı olabilir. Çünkü renk bilgisinin her bir piksel için nasıl kullanılacağı belirsizdir. Yapılan seçime göre kesimlemenin başarımı da değişir. Diğer bir problem, kesimleme için hangi renk uzayının seçilmesi gerektiğidir. Her renk uzayının avantajları ve dezavantajları vardır. Herhangi bir kesimleme algoritması bütün kesimleme problemleri için en iyi sonucu vermediği gibi herhangi bir renk uzayı da tüm renkli kesimleme yöntemleri için en iyi sonucu vermede aracı olamaz.

Bir sonraki bölümde gri-tonajlı resimlerde kullanılan görüntü kesimleme yaklaşımlarına değinilmiş, renkli görüntülerde kullanım şekilleri hakkında da bazı ipuçlarına yer verilmiştir. Üçüncü bölümde ise renk gösteriminde kullanılan renk uzaylarından bahsedilmiştir.

3.1 Görüntü Kesimleme

Kesimleme, görüntülerde ilgili yapıların birbirlerinden ve arka plandan ayrılması işlemidir. Görüntü kesimleme, nesne tanıma sistemleri, hareket ve

stereo sistemlerde nesnelerin sınırlarının tespitinde, resim sıkıştırma ve resim düzenleme gibi birçok görüntü işleme konusunda kullanılmaktadır ve bu tip sistemlerin verimli çalışmasında rol oynayan en kritik ve önemli işlemlerden bir tanesidir.

Görüntü kesimlemenin temel hedefi, resmi karakter ve özellik olarak homojen olan farklı bölgelere ayırmaktır. Dolayısıyla kesimleme işlemi, resimlerden özellik çıkarmasında, resimlerin analiz ve temsilde büyük bir öneme sahiptir. Kesimlemenin önemi, bu konu üzerinde birçok farklı kesimleme tekniğinin geliştirilmesine sebep olmuştur. Fakat halen tüm resim çeşitleri için mükemmel standart bir kesimleme tekniği bulunamamıştır. Ayrıca zaman zaman kesimlemenin hedefi çalışmanın hedefine ve resim verilerinin tipine göre farklılık da gösterebilir. Dolayısıyla resmin doğası üzerinde yapılan farklı varsayımlar farklı algoritmaların ortaya çıkmasında etkili olmuştur.

Literatürde, kesimleme algoritmaları farklı şekillerde sınıflandırmıştır. Bu sınıflandırma şemalarının bazıları şu şekildedir (Rogowska, 2000);

- Manüel, yarı-otomatik ve tam otomatik kesimleme algoritmaları
- Piksel-tabanlı, bölge-tabanlı ve kenar-tabanlı kesimleme algoritmaları
- Elle çizim, düşük-seviyeli kesimleme (eşikleme, bölge büyütme vb.), ve model-tabanlı kesimleme (dinamik programlama ve kontur izleme vb.)
- Klasik (eşikleme, kenar-tabanlı ve bölge tabanlı), istatistiksel, bulanık ve yapay sinir ağları tekniklerine dayalı kesimleme algoritmaları

En çok kullanılan sınıflandırma biçimi piksel-tabanlı, bölge-tabanlı ve kenar-tabanlı kesimleme teknikleri sınıflandırmasıdır. Piksel-tabanlı yöntemler sadece renk uzayını kullanarak kesimleme yaparken bölge-tabanlı ve kenar-tabanlı yaklaşımlar resim uzayı üzerinde araştırma yaparak kesimleme yaparlar. Bölge-tabanlı kesimlemede resim uzayında homojen bölgelerin araştırılması söz konusu iken, kenar-tabanlı kesimleme yaklaşımlarında farklı karaktere sahip bölgelerin kenarlarının tespitine çalışılır. Şimdi bu yaklaşımlara kısaca bir göz atalım.

3.1.1 Piksel-tabanlı kesimleme

Renk uzayı üzerinde işlem yapan bu teknikler iki grupta toplanabilirler:

- Histogram-tabanlı teknikler: Resmin piksel değerleri histogramında bir veya birden fazla eşik değeri alınarak bu değerler arasında kalan piksellerin sınıflandırılmasına dayanırlar.
- Kümeleme-tabanlı teknikler: Resim uzayında kümeleme analizi yapan ve buna göre kesimleme yapan tekniklerdir.

Aşağıda bu iki farklı teknikten bahsedilmiştir.

3.1.1.1 Histogram-tabanlı teknikler

Resimlerde kesimleme yapabilmenin en kolay yollarından birisi renk değerleri için bir veya birden fazla aralık seçip bu aralıkların içinde kalan pikselleri birbirinden ayıştırmaktır. Bu aralık değerlerini seçme işlemi eşikleme (thresholding) olarak adlandırılır (Sahoo et al, 1988).

Eşikleme, etkileşimli veya otomatik olarak gerçekleştirilebilir. Etkileşimli yöntemde, resmin histogramı, resimdeki renklerin niceliklerini ve yoğunluklarını ifade eden bir çizge üzerinde gösterilir ve kullanıcı bu çizge üzerinde eşik değerlerini fare yardımıyla işaretler. Böylece belirlenen eşik değerlerine göre pikseller birbirinden ayıştırılarak kesimleme gerçekleştirilir. Fakat insanın görüşüne göre yapılan bu eşikleme yöntemi görüntü işleme alanında pek çok sorunlara neden olduğundan otomatik eşikleme yöntemleri genelde tercih edilmektedir (Russ, 2007). Literatürde birçok otomatik kesimleme yaklaşımı bulunmaktadır. Bunların birçoğu resmin genel histogramı üzerinde işlem yaparak ve/veya yerel bilgileri (piksel komşuluklarını) kullanarak eşikleme yapmaya çalışırlar. Birçok farklı yaklaşımın bulunması resmin doğası üzerinde yapılan varsayımlardan, resimde sadece istenilen bir bölgenin kesimlenme durumunun söz konusu olmasından kaynaklanmaktadır. Dolayısıyla her resim ve istenilen her bölgenin kesimlenmesi için genel bir kesimleme yöntemi bulunmamaktadır.

Otomatik eşiklemenin nasıl varsayımlarla kesimleme yapıldığı hakkında verilebilecek en basit örneklerden birisi resmin histogramındaki tepelerin tespiti ve tepelerin ortasının eşik değerleri olarak belirlenmesidir. Bu oldukça basit bir yöntem olmakla birlikte kararlıdır. Çünkü histogramdaki her bir tepe farklı renk kümelerini ifade ederler. Bu örnekte de görüldüğü gibi, histogramın doğası

hakkında bir varsayım yapılarak kesimleme yapılmıştır. Bunun gibi diğer birçok yaklaşım hakkındaki bilgiye (Russ, 2007) kaynağından erişilebilir.

Renkli resimlerde eşikleme yapılırken, yukarıda bahsi geçen gri-tonajlı resimlerdeki yaklaşımlara benzer ve genişletilmiş yollar izlenir. Bunların en basiti, üç boyuttan oluşan renk uzayını bir boyuta indirecek bir dönüşüm uyguladıktan sonra tek bir boyut üzerinde kesimleme yapma işlemidir (Bonsiepen and Coy, 1991). Bunun dışında her bir renk uzayında ayrı ayrı eşikleme yaparak veya 2-boyutlu ya da 3-boyutlu histogramlar üzerinden eşikleme yapılarak kesimleme yapılabilmektedir.

3.1.1.2 Kümeleme-tabanlı teknikler

Kümeleme, birbiri içinde benzerlik gösteren gruplara verileri atama işlemidir. Bu anlamıyla kümeleme işlemi görüntü işlemeden bağımsız bir tanıma sahiptir. Görüntü işlemede ise kümeleme algoritmaları, resimdeki her bir piksel veya bölge ile ilgili özellikler elde edildikten sonra kümeleme algoritmaları bu piksel veya bölgeleri kendi aralarında homojen kümelerle bölmeye çalışır.

Kümeleme algoritmaları temel olarak iki sınıfta toplanır; parçalayan (partitional) yöntemler ve hiyerarşik yöntemler. Parçalayan yöntemler verileri doğrudan önceden belirlenmiş olan kümelerle atama yaparken, hiyerarşik yöntemler verileri birbiri içine girmiş hiyerarşik veri kümelerine atama yaparlar. Hiyerarşik kümelemede kümeleme tekil kümelerden tüm verileri içinde saklayan bir kümeye doğru olabildiği gibi tam tersi de olabilir. Birincisine, toplayıcı (agglomerative) hiyerarşik kümeleme, ikincisine dağıtan hiyerarşik kümeleme adı verilmiştir.

Parçalayan yöntemler, verileri ilgili K tane kümeye atama yaparken bir ölçüt fonksiyonunu optimize etmeye çalışır. Pratikte, verilen ölçüt fonksiyonuna göre, optimum parçalama işlemi bulunabiliyor olmasına rağmen Brute-Force bir yöntem ile çok maliyetli bir iştir. Çok ufak bir örnekte bile, örneğin 30 verinin 3 kümeye ayrılmasında, yaklaşık olarak 2×10^{14} farklı kümeleme alternatifi vardır. Dolayısıyla sezgisel yöntemlerin kullanılarak yaklaşık en iyi değerlerin bulunması sağlanmalıdır. Bilinen ve en çok kullanılan optimizasyon tekniği K-means algoritmasıdır. Bu algoritma Bölüm 5.2.1 de ayrıntılı olarak anlatılmıştır.

3.1.2 Bölge-tabanlı kesimleme

Bölge-tabanlı kesimlemede resim uzayı üzerinde homojen olan bölgeler tespit edilmeye çalışılır. Bu yaklaşımlar, *bölge-büyütme* (*region-growing*) teknikleri ve *parçala-ve-birleştir* (*split-and-merge*) teknikleri olmak üzere iki alt başlıkta toplanabilirler. Bölge-büyütme teknikleri genel olarak bir kaynak (seed) noktasından başlayarak farklı kıstaslara göre komşu pikselleri kaynak piksele eklemek suretiyle homojen bölgelerin tümünü elde edene dek büyüterek ilerlerler. Bunun aksine, parçala-ve-birleştir tekniklerinde, önce resim homojenlik ölçütü sağlanana dek küçük bölgelere parçalanır. Daha sonra homojen bölgeler elde edilene dek ayrıştırılan bu bölgeler birleştirilerek kesimleme sağlanmış olur.

3.1.2.1 Bölge-büyütme teknikleri

Bölge-büyütme tekniklerinde rastgele veya zekice kaynak pikseller seçilir. Daha sonra bu kaynak pikselin etrafındaki pikseller değerlendirilir. Komşu pikseller belli bir eşik değerinden küçük olacak şekilde kaynak piksele / bölgeye yakın olduğunda bu pikseller homojen olan bölgeye dâhil edilir. Böylece homojen bölgeler belirleninceye kadar aynı işlem devam ettirilir (Koschan and Abidi, 2008).

Aşağıda 5x5 lik gri tonajlı bir resim (Şekil 3.1-(solda)) üzerinde bölge-büyütme işleminin nasıl gerçekleştiği (Şekil 3.1-(sağda)) görülmektedir. Altı çizili değerler kaynak piksellerini ifade etmektedirler ve eşik değeri 3 olarak seçilmiştir. Bu koşullarda bölge-büyütme tekniği uygulandığında resmin iki farklı homojen bölgeye ayrıldığı gözlenecektir.

Bölge-büyütme işleminde kullanılan eşik kıstası renkli ve gri tonajlı resimlerde farklı şekilde tanımlanırlar. Gri tonajlı resimlerde, bu ölçüt iki pikselin renk değerleri arasındaki fark ile ifade edilirken renkli resimlerde iki pikselin vektör (ya da Öklid) uzunluk farkını ifade eder. Bu aşamada kullanılan renk uzayı çok önemlidir. Çünkü her renk uzayında renk değerleri arasındaki uzaklık farklı şekilde ifade edilmekte, bazı renk uzaylarında da renkler arası uzaklıkların tespitinde problemler yaşanmaktadır.

1	1	6	7	8
2	2	6	9	8
1	<u>2</u>	7	<u>8</u>	8
3	1	8	7	7
1	2	6	7	6

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

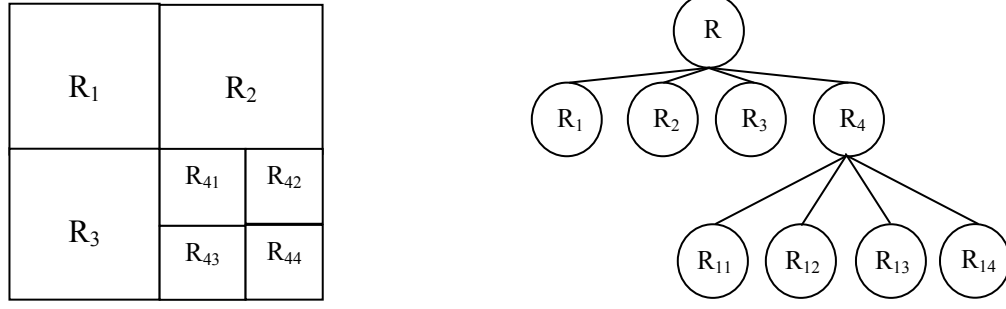
Şekil 3.1. (Solda) kesimleme yapılacak resim, (Sağda) kesimleme sonucunda elde edilen resim (Koschan and Abidi, 2008).

3.1.2.2 Parçala-ve-birleştir teknikleri

Bu yaklaşımda tüm resim bir bütün olarak ele alındıktan sonra, homojenlik ölçütü sağlanana dek resim parçalara, örneğin her defasında dört parçaya, ayrılır. Homojenlik ölçütü, gri-tonajlı resimler için bir bölge içindeki piksel değerlerin değişiminin belli bir eşik değerinin altında olması olabilir. Bu koşul sağlanmadığı müddetçe resmin o parçası yine alt parçalara ayrılacaktır. Tüm parçalama işlemi sonlandığında bazı parçaların birbirlerine yakın değerlerde parçalandığı tespit edilecektir. Böylece bu bölgeler birbiriyle birleştirilerek sonuç resmi elde edilir.

Şekil 3.2'deki örnekte görüldüğü gibi resim önce 4 farklı bölgeye ayrıldıktan sonra bu dört bölgeden sadece birisi homojenlik kistasına uymamış ve bu bölgede tekrar 4 bölgeye ayrıldıktan sonra kıstas sağlanmıştır. Böylece resim homojen bölgelere ayrılmıştır (Koschan and Abidi, 2008).

Parçala-ve-birleştir yöntemi renkli resimlerde kullanıldığında bölge-büyütme yöntemlerinde olduğu gibi farklı kıstasların kullanılmasını gerektirir. Fakat bölge-büyütme yönteminden çok daha fazla güçlüklerle karşılaşılacaktır. Örneğin HSI (ton-doygunluk-yoğunluk) renk uzayı kullanıldığında, eğer yoğunluk değeri çok yüksek ve düşükse ton ve doygunluk değeri önemsizleşmekte, doygunluk zayıf ise ton değeri kararsızlaşmaktadır. Dolayısıyla böyle koşulları içinde barındıran resimlerde bu teknikle kesimleme yapmak oldukça zorlaşmaktadır.



Şekil 3.2. (Solda) kesimlenmiş resim, (Sağda) bölümlerin hiyerarşik yapısı (Koschan and Abidi, 2008).

3.1.3 Kenar-tabanlı kesimleme

Kenar-tabanlı kesimleme yaklaşımları resimlerdeki sınır bölgelerinin yani kenarların belirlenmesi ile ilgilenirler. Bu yüzden bu yaklaşımlara kenar belirleme yöntemleri de denir. Bu yöntemler, gri-tonajlı resimlerde süreksizliğin tespit edilmesine dayanır ve resimde tonaj değişikliklerinin başladığı noktalar belirlenmeye çalışılır (Koschan and Abidi, 2008).

Kenar belirleme teknikleri, sırasal ve paralel teknikler olmak üzere iki kategoriye ayrılır. Sırasal tekniklerde bir noktanın bir kenara ait olup olmadığı daha önceden tespit edilmiş olan noktalara göre belirlenir. Bu yaklaşımlar genel olarak sezgisel tabanlı ya da dinamik programlama tabanlıdır. Fakat performansları ilk seçilen noktanın seçimine göre değişmektedir ve doğru başlangıç noktasını belirlemek oldukça zor bir işlemdir. Paralel kenar belirleme teknikleri ise diğer noktaların bir kenara ait olup olmadığına bakmadan bir noktanın bir kenara ait olup olmadığına karar verirler. Bu yaklaşımlarda, genel olarak bir kenar belirleme operatörü tespit edilir ve tüm resim üzerinde uygulanır. Literatürde birçok farklı kenar belirleme operatörü tanımlanmıştır. Fakat mükemmel bir operatör seçebilmek oldukça zordur. Bu yaklaşımlar sadece iki komşu nokta arasındaki tonaj farklılığını göz önüne aldığından bazı noktalar kenara ait olmadığı halde yanlışlıkla kenar olarak belirlenebilir veya nesne sınırlarına ait bazı noktalar tespit edilemeyebilir. Kesimleme işlemi için, sınır bölgeleri kapalı hatlar olarak belirlenmesi gerektiğinden paralel kenar belirleme teknikleri kesimleme işleminde tek başlarına yeterli olamayabilirler. Dolayısıyla, bu tekniklerden sonra gürültü yok etme ve kenar birleştirme teknikleri beraber kullanılır (Cheng et al, 2001).

Gri-tonajlı resimlerde, kenarlar resimdeki süreksizlikler ve iki bölge arasındaki tonaj farkları ile belirlenirken, renkli resimlerde bilgiler çok daha fazladır. Örneğin, iki nesne arasındaki kenarlarda parlaklık aynı iken, renk tonları farklılık gösterebilir (Macaire et al., 1996). Dolayısıyla, renkli resimlerde kenarlar üç boyutlu renk uzayında süreksizliğin tespit edilmesi ile bulunabilir. Üç boyutlu bir renk uzayında kenar belirleme işlemi üç farklı şekilde gerçekleştirilebileceği tanımlanmıştır (Nevatia, 1977);

- Renk uzayında piksel değerleri için bir fark ölçütü tespit edip bu ölçüte göre süreksizlikleri tespit etmek.
- Her bir boyutu birbirinden bağımsız olarak ele alıp her bir boyutta ayrı ayrı kenar belirleme yaptıktan sonra belirlenen kenarları özel bir işlemle birleştirmek.
- Kenarlar için üç boyutta bazı kısıtlar tanımlayarak bu kısıtlar doğrultusunda her boyutta birbirinden bağımsız kenar belirlemeye olanak tanımak.

3.2 Renk Uzayları

Bir renk uzayı renklerin geometrik bir uzayda temsil edilmelerini sağlar. Genellikle bu uzay üç bileşenden oluşur ve renkler bu bileşenlerin nümerik değerleri ile ifade edilirler.

Birçok uzay renkler arasındaki farkı ölçmek için bir ölçüt tanımlar ve bu ölçüt, renk uzayının renkli resim analizi ve kesimlemesinde önemli rol oynar. Renk uzayları karakteristiklerine göre şu şekilde sınıflandırılabilir (Busin et al., 2008);

- Birincil (Primary) uzaylar: Tristimulus teorisine dayanırlar. Bu teoriye göre bütün renkler üç farklı bileşenin karıştırılmasıyla elde edilebilir. RGB, nRGB uzayları gibi.
- Parlaklık-renklilik (limunance-chrominance) uzayları: Bir bileşenin parlaklığı, diğer iki bileşenin ise renkliliği ifade ettiği uzaylardır. YIQ, YUV, CIE L*a*b*, CIE L*u*v* uzayları gibi.

- Algısal (perceptual) uzaylar: Yoğunluk (intensity), ton (hue) ve renk doygunluğu (saturation) bileşenlerini kullanarak insan algısına yakın renk modellenmesi yapmaya çalışan uzaylardır. HSV ve HSI uzayları gibi.
- Bağımsız eksen (independent axis) uzayları: Birçok farklı istatistiksel yöntem kullanarak elde edilen uzaylardır. İstatistiksel yöntemler, yeni oluşan uzayda bileşenler arasındaki bağımlılığı azaltmaya çalışırlar. (I_1, I_2, I_3) uzayı gibi.

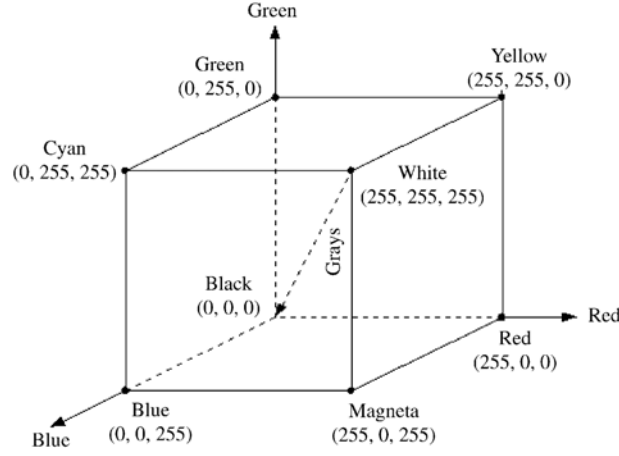
Literatürde birçok renk uzayı ve bunları kullanan yöntemler vardır. Dolayısıyla renkli resim kesimlemede doğru renk uzayının tespiti önemli ve zor bir iştir. Aşağıda çokça kullanılan bazı renk uzayların özellikleri ve renkli görüntü kesimlemede kullanılmalarındaki avantaj ve dezavantajlarına değinilmiştir.

3.2.1 RGB (kırmızı-yeşil-mavi) uzayı

Birincil uzayların başında RGB uzayı gelir. Bu uzayın adı kırmızı, yeşil ve mavinin İngilizcedeki baş harfleriyle tanımlanır. Wyszecki ve Stiles'in (1982) tristimulus teoremine göre renkler, kırmızı (red, R), yeşil (green, G) ve mavi (blue, B) renklerinin eklemeli olarak karıştırılmasından elde edilebilir. Her renk %100 oranında karıştırıldığında beyaz ve %0 oranında karıştırıldığında siyah elde edilir. Bu renk uzayı daha çok televizyon, bilgisayar grafikleri ve fotoğraflarda renklerin temsilinde kullanılır ve RGB uzayı her makinede farklı şekillerde tanımlanabilir.

RGB renk uzayı, Şekil 3.3'te gösterildiği gibi geometrik olarak bir 3 boyutlu küp ile ifade edilir. Küp içindeki herhangi bir nokta rengin R, G ve B değerlerini temsil eder.

RGB, renk gösterimi için uygun olmasına karşın renkli görüntülerin kesimlenmesinde iyi sonuçlar vermemektedir. Bunun nedeni; yoğunluğun (intensity) değişmesi, her bir kırmızı, yeşil ve mavi bileşeninin de değişmesine yol açması ve farklılığı arttırmasıdır. Diğer bir değişle, kırmızı, yeşil ve mavi bileşenlerinin arasında bu şekilde yüksek korelasyonun bulunması kesimlemeyi güçleştirir (Littman and Ritter, 1997). Ayrıca RGB uzayı, renkler arasındaki farkı güzel ifade edemediğinden, RGB uzayında iki renk arasındaki farkı ve benzerliği bulmak güçleşir.



Şekil 3.3. RGB uzayı (Cheng et al, 2001).

3.2.2 *n*RGB (normalleştirilmiş kırmızı-yeşil-mavi) uzayı

RGB uzayında renk gösterimleri parlaklığa bağımlı olarak değişir. Aynı renkliliğe sahip iki piksel farklı parlaklık değerleri aldığı anda RGB uzayında farklı değerlerle ifade edilirler. Renkli görüntü kesimlemede, parlaklıktaki değişime bağımlılığı düşürmek temel hedeflerden birisidir. Eğer yoğunluk değişimi spektrum boyunca düzgün (uniform) ise, bileşenlerin parlaklıktan bağımsız hale getirilmesi normalleştirme işlemi ile gerçekleştirilir. *n*RGB uzayı şu formülle ifade edilir:

$$\begin{aligned} r &= R / (R + G + B) \\ g &= G / (R + G + B) \\ b &= B / (R + G + B) \end{aligned} \quad 3.1$$

$r+g+b = 1$ olduğundan iki bileşen verildiğinde üçüncü bileşen bulunabilir. Dolayısıyla bu uzayın üç bileşenden sadece ikisi kesimleme işlemi için yeterli olacaktır. İki bileşeni kullanan (*rb*) çalışmalarda aşağıdaki gibi bir üçüncü bileşen kullanılmıştır:

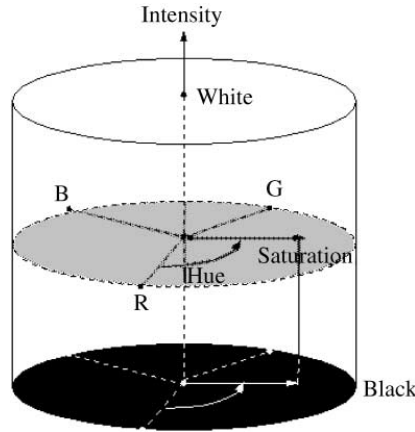
$$Y = c_1 R + c_2 G + c_3 B \quad 3.2$$

c_1 , c_2 ve c_3 birer sabit olup toplamaları bire eşittir ve bu üç bileşen pikselin parlaklık değerini hesaplamak için kullanılırlar. Bu gösterimle renkler ışıktan bağımsız olarak temsil edilebilme özelliğine kavuşurlar.

n RGB parlaklık deęişimine karşı dayanıklı olsa da, düşük yoğunluklarda normalleştirilmiş renkler bu uzayda çok gürültülü görünürler. Bu problem, renk uzayı oluşturulurken RGB den n RGB'e düzgün olmayan (non-uniform) bir dönüşümün yapılmış olmasından kaynaklanmaktadır (Cheng et al, 2001).

3.2.3 HSI (ton-doygunluk-parlaklık) uzayı

Algısal bir uzay olan HSI sistemi, insanın renk algılama mekanizması modellemeye çalışır. Çünkü insanlar farklı ton (Hue) değerlerini kolaylıkla ayırt edebilmektedirler. HSI sisteminin en büyük özellięi, renk bilgisini yoğunluk bilgisinden ayırt edebilmesidir. Bu sistemde "Hue" temel renk tonlarını, "Saturation" rengin doygunluk ölçüsünü, "Intensity" ise ışık etkisinin yoğunluğunu ifade eder. Ayrıca HSI uzayının HSB (B = brightness), HSL (L=lightness) ve HSV (V = value) gibi başka çeşitleri de vardır. HSI geometrik olarak bir silindir şeklinde ifade edilir (Şekil 3.4) (Hoy, 1997).



Şekil 3.4. HSI uzayının geometrik temsili (Cheng et al, 2001).

Bu gösterimde silindirin en alt kısmı siyah rengi, en üstü beyaz rengi ve silindir içinde kalan kısımlar ise diğer renkleri temsil eder. Eksenin taradığı açı ton bilgisini, merkeze olan uzaklık ise yoğunluk bilgisini ifade eder.

HSI uzayının bileşenleri RGB uzayından elde edilir. Her makinedeki RGB tanımı farklı olabileceğinden HSI uzayı da buna baęlı olarak makine baęımlı bir özellik taşımaktadır. RGB uzayından HSI uzayına dönüşüm aşağıdaki gibi gerçekleştirilir;

$$\begin{aligned}
H &= \arctan\left(\frac{\sqrt{3}(G-B)}{2R-G-B}\right), \\
S &= 1 - \frac{\min(R, G, B)}{I}, \\
I &= \frac{R+G+B}{3}
\end{aligned} \tag{3.3}$$

HSI uzayında $F_1 = (H_1, S_1, I_1)^T$ ve $F_2 = (H_2, S_2, I_2)^T$ şeklinde tanımlı İki renk değeri arasındaki renk uzaklığı;

$$\Delta_{HSI}(F_1, F_2) = \sqrt{(\Delta_I)^2 + (\Delta_C)^2} \tag{3.4}$$

şeklinde tanımlanmıştır. Burada $\Delta_I = |I_1 - I_2|$ ve $\Delta_C = \sqrt{S_1^2 + S_2^2 - 2S_1S_2 \cos \theta}$ ile ifade edilir ve θ değeri aşağıdaki gibi hesaplanır;

$$\theta = \begin{cases} |H_1 - H_2| & \text{eğer } |H_1 - H_2| \leq \pi \\ 2\pi - |H_1 - H_2| & \text{eğer } |H_1 - H_2| > \pi \end{cases} \tag{3.5}$$

HSI uzayı renk bilgisini parlaklıktan ayırt ettiğinden renkli görüntü analizi ve kesimlemede çokça kullanılmaktadır. Bu uzay kesimlemede kullanılırken yoğunluk (intensity) bileşeni üzerinde gri-tonajlı kesimleme algoritmaları kullanılabileceği gibi, ton (hue) bileşenini kullanarak da farklı renklerdeki nesnelerin ayrıştırılması sağlanabilir.

HSI'nın en önemli dezavantajlarından birisi R, G ve B nin ufak farklarla ayrıldığı renk silindirin aksisin yanında tekilliğin kaldırılamamasıdır. Bu da dönüştürülen hue değerlerinin renk değerlerine göre çok fazla sıçrama yapmasına neden olur ve renklerde istenmeyen süreksizliğe yok açar (Chapron, 1992). Bu yüzden düşük renk doygunluklarına sahip olan bölgeler kesimlemede her hangi bir bölgeye ayıramayabilirler. Ayrıca yoğunluk değeri beyaza veya siyaha çok yakın olduğu durumlarda, ton ve doygunluk değerleri renklerin ayrıştırılmalarında çok daha az rol oynarlar.

3.2.4 YIQ uzayı

Amerika TV sisteminde kullanılan renk kodlama sistemidir (uzayıdır). RGB den doğrusal (linear) bir dönüşümle şu şekilde elde edilir:

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.253 & -0.312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad 3.6$$

burada $0 \leq R \leq 1, 0 \leq G \leq 1, 0 \leq B \leq 1$

Y bileşeni rengin parlaklık değerinin ölçüsüdür ve resimlerde kenar belirleme için kullanılabilecek güzel bir bileşendir. I ve Q bileşenleri ise ton ve doygunluk bilgilerini taşır. YIQ uzayı, R, G ve B bileşenlerinin birbirine olan korelasyonlarını ortadan kaldırdığı gibi, lineer bir dönüşümle oluşturulduklarından lineer olmayan dönüşümlere göre daha hızlıdır. Dolayısıyla YIQ uzayı, lineer olmayan sistemlere tercih edilebilir.

3.2.5 YUV uzayı

Avrupa TV sisteminde kullanılan renk kodlama sistemidir. YIQ ya benzer bir şekilde RGB uzayından doğrusal bir dönüşümle şu şekilde elde edilir:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & -0.437 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad 3.7$$

burada $0 \leq R \leq 1, 0 \leq G \leq 1, 0 \leq B \leq 1$

Bu uzayda da ilk bileşen parlaklık değerini, diğer iki bileşen ise ton ve doygunluk bileşenlerini taşır. Ayrıca, parlaklık ve ton değeri farklı bit değerleriyle gösterildiği takdirde, YIQ ve YUV gösterimleri görüntü sıkıştırma işleminde oldukça uygundur.

3.2.6 CIE uzayları

CIE uzayı X, Y ve Z ile adlandırılan üç temel bileşene sahiptir ve bir renk bu üç bileşenin bileşiminden elde edilir. X, Y ve Z değerleri RGB den aşağıdaki gibi lineer bir dönüşümle elde edilir.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad 3.8$$

burada $0 \leq R \leq 1, 0 \leq G \leq 1, 0 \leq B \leq 1$

Bu X,Y ve Z bileşenlerinden birçok farklı CIE uzayı elde edilmiştir. Örnek olarak CIE($L^*a^*b^*$) ve CIE($L^*u^*v^*$) verilebilir. Bu uzaylarda L^* bileşeni parlaklığı ifade ederken, u^* , v^* ile a^* , b^* bileşenleri renkliliği ifade eder. Bu iki uzayda, üç boyutlu bir renk uzayda bir nokta gibi algılanabilir ve renkler arasındaki fark Öklid uzaklığı ile hesaplanabilir. Örneğin CIE($L^*a^*b^*$) sistemi için iki renk arasındaki fark şu şekilde hesaplanır;

$$\Delta E_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad 3.9$$

Renk farkını Öklid uzaklığıyla ifade edebilmek kesimlemede önemli bir avantaj sağlamaktadır. Bu uzaylar rengi düzgün bir ölçüye çektiğinden insan algılamasına yakın bilgisayar algılaması gerçekleştirmeye olanak tanır (Tseng and Chang, 1992). Bu uzayın diğer bir özelliği HSI uzayının da bu uzaydan elde edilebiliyor olmasıdır.

Bu uzaylar RGB uzayına göre renk ile yoğunluk bilgilerini daha bağımsız kontrol eder. Öklid uzaklığı dikkate alındığından küçük renk değişimlerini kolaylıkla algılayabilir. Fakat diğer doğrusal olmayan sistemler gibi tekillik problemlerine sahiptirler.

3.2.7 Fizik-tabanlı uzaylar

Klasik renk uzaylarının haricinde bir de ışık etkisini göz önünde bulunduran fizik-tabanlı uzaylar mevcuttur. Bu modeller ışık, gölge ve parlama etkilerini azaltmayı ve renkleri ayrıştırmayı hedef alırlar. Shafer'in (1985) "dichromatic reflection model" i ve Healey'in (1989) "approximate color-reflectance model"'i bu tip yaklaşımların iki temel örneğidir.

Birçok geleneksel resim kesimleme algoritması optik etkileri göz önüne almadıklarından hatalı sonuçlar verebilmektedirler. Birçok resimde gölge, ışık ve parlama etkisi olduğundan fiziksel modeller bu tip bölgelerin doğru sınıflandırılmasına olanak sağlarlar.

Bazı çalışmalar yansıma etkisini malzemenin cinsine göre (homojen ve homojen olmayan malzeme) olarak ayırarak modellemeye çalışmıştır. Bu da algoritma daha çalışmadan maddenin cinsini belirlemede fayda sağlamak ve görüntü işleme ve analizinde büyük fayda sağlamaktadır.

4. ÖNCEKİ ÇALIŞMALAR

KKO ve PSO algoritmaları birçok problemin çözümünde yıllardır kullanılıyor olmasına karşın kesimleme probleminin çözümüne yönelik çalışmalar son yıllarda yapılmaktadır. Bu çalışmalardan hemen hemen tümü gri-tonajlı resimler üzerinde gerçekleştirilmiştir. Kesimleme algoritmalarının sınıflandırmasında olduğu gibi sürü zekâsı tabanlı kesimleme yaklaşımları da bölge büyütme, kenar belirleme ve piksel-tabanlı yaklaşımlar şeklinde sınıflandırılabilir.

KKO tabanlı ilk bölge büyütme yaklaşımı Ramos and Almeida (2000) ve Fernandes et al. (2005) ve tarafından öne sürülmüştür. Önerilen modelde, karıncalar pikseller üzerinde dağıtılmış ve her bir pikseli yalnız bir karıncanın ziyaret etmesine olanak tanınmıştır. Karıncalar, bir pikselden diğerine, muhtemel gidebileceği yönlerle (doğu, batı, kuzey, güney vb.) orantılı bir olasılığa göre hareket edebilmekte, eğer tüm komşu pikseller başka karıncalar tarafından ziyaret edilmişse hareket edememekte, ölmekte ve yeniden başka bir yerde tekrar üretilmektedir. Her bir karınca hareketinden sonra ziyaret ettiği pikseller üzerinde feromon bırakarak ilerler. Feromon izi sabit olarak değil komşuluk bilgisine göre dinamik olarak bırakıldığından resimdeki homojen bölgeler farklı feromon yoğunluğuna göre belirlenebilmektedir.

Huang et al. (2008) bu modeli medikal görüntülerin kesimlenmesinde kullanmıştır. Yaklaşım gerçek beyin MR görüntüleri üzerinde test edilmiş ve altın standarda çok yakın sonuçlar elde edilmiştir.

Bu modele ek olarak Laptik ve Navakaustas (2007) resimlerin farklı bölgelere kesimlenebilmesi için farklı kolonilerin resim üzerinde dolaşması gerektiğini savunmuşlardır. Gerçekleştirdikleri modelde koloniler bir rekabet içerisindedir ve her bir koloni pikselleri sahiplenmeye çalışır. Bunun için ziyaret ettikleri pikseller üzerinde her bir karınca kendi kolonisine ait farklı bir feromon türü bırakmaktadır. Bir piksel üzerinde yalnız bir koloniye ait feromon bulunabilmekte eğer aynı koloniye ait bir karınca tekrar o pikseli ziyaret ederse feromon seviyesi artmakta, farklı bir koloniye ait bir karınca ziyaret ederse feromon seviyesi azalmaktadır. Piksel üzerindeki feromon seviyesi sıfıra yaklaştığında artık başka bir türe ait feromon maddesi piksel üzerinde konulabilmekte ve böylece piksel başka bir koloni tarafından sahiplenilmektedir. Gerçekleştirilen modelden ümit verici sonuçlar elde edilmiştir.

Rezaee (2008), Ramos and Almedia (2007) modelini kenarların belirlenmesinde kullanmıştır. Fakat modelin çalışma zamanı karmaşıklığı yüksek olmasına karşın kötü sonuçlar vermiş ve sadece bu model üzerinde ileride yapılacak çalışmalar için temennilerde bulunulmuştur.

KKO tabanlı kayda değer ilk kenar belirleme yaklaşımı Nezamabadi-pour et al. (2006) tarafından öne sürülmüştür. Bu yaklaşımda karıncalar yine pikseller üzerinde dağıtılmakta ve karıncaların kenarlar üzerinden geçmeleri sağlanmaya çalışılmaktadır. Bunun için karıncalar bir pikselden diğerine KS' deki olasılıklı seçim kuralına göre hareket ederler. Sezgisel fonksiyon bir piksele ait komşu fakat zıt yönlü piksellerin farkının maksimum değeriyle orantılı olarak belirlenmiştir. Bu yaklaşıma göre zıt yöndeki piksellerin farkı ne kadar fazla ise o pikselin kenara ait bir piksel olma olasılığı yüksektir. Gerçekleştirilen bu yöntem test edildiğinde ümit verici sonuçlar verse de klasik kenar belirleme yaklaşımlarından daha kötü ve kalın kenarlı sonuçlar vermekte ayrıca gürültülü resimlerde iyi sonuç verememektedir.

Aydın (2009) bu modelde iki farklı değişiklik yaparak KKO algoritması ile kenarların belirlenme kalitesini klasik kenar belirleme algoritmalarına yaklaştırmıştır hem de üç kat hızlı kesimleme yapabilmeyi başarmıştır. Yapılan iki değişiklikten birisi KS algoritmasında rastlantısallığı arttıran seçim kuralı yerine KKS algoritmasındaki daha agresif olan seçim kuralı tercih edilmesidir. Böylece kalın kenarların oluşmasının önüne geçilmiştir. Diğer bir değişiklik ise zıt piksellerin yoğunluk farkını almak yerine 3x3 komşuluk resminde zıt bölgelerin farkı temel alınarak sezgisel fonksiyonun belirlenmesidir. Böylece bir pikselin bir kenara ait olup olmama olasılığı daha iyi tanımlandığından daha az karınca ve adım sayısı kullanılarak kenarlar belirlenebilmiştir. Aynı model, 5x5 lik komşuluk resimlerini kullanarak sezgisel fonksiyonu belirlendiğinde elde edilen kenar belirleme sonuçlarının gürültü resimlerde bile iyi sonuçlar verdiği hatta birçok klasik kenar belirleme algoritmasından daha sağlam sonuçlar ürettiği tespit edilmiştir (Aydın, 2010).

Kenar belirlemede kullanılan başarılı diğer bir KKO yaklaşımı, kenar belirleme yöntemlerinin en büyük sorunlarından birisi olan bağlantısız kenarların birleştirilmesinde kullanılmıştır (Lu and Chen, 2008). Bu modelde, klasik kenar belirleme algoritmaları sonuç elde edildikten sonra, karıncalar belirlenen kenarların uç noktaları üzerinde bırakılmış ve maksimum komşuluk farkına ve

birleştirme benzerliğine dayalı bir sezgisel kullanılarak kenar birbirlerine birleştirilmeye çalışılmıştır.

Renk uzayı üzerinde kesimleme yapan KKO yöntemleri bu işlemleri eşikleme veya kümeleme yoluyla gerçekleştirmektedirler. Eşikleme yoluyla kesimleme yapan yöntemlerin birçoğu optimum eşik değerlerini bularak kesimleme yapmaktadırlar (Liang et al., 2006; Zhao et al., 2008, Ye et al. 2005). Bazı metotlar da diğer eşikleme algoritmaları ile beraber kullanılarak kesimleme yapmaktadırlar. Örneğin, Tao et al. (2007) bulanık entropi yoluyla kesimleme yaparken parametrelerin seçimini Karınca Kolonisi Optimizasyonu ile gerçekleştirmiştir.

Kümeleme yoluyla kesimleme yapan KKO algoritmalarının bir kısmı kümeleme algoritmalarının dezavantajlarını ortadan kaldırmak için kullanılırken, diğerleri tamamen KKO tabanlı kesimleme gerçekleştirmeye çalışmaktadır. Hung and Sun (2010) K-means algoritmasındaki problemlerden birisi olan küme merkezlerinin doğru ilklenmesi sorununun KKO algoritmasıyla K-means algoritmasını melez bir şekilde kullanarak üstesinden gelmiştir. Bunun için her bir karınca rastgele seçilen küme merkezlerine karıncalar pikselleri atayarak kümeleme yapılmıştır. Benzer bir yaklaşım bulanık c-means algoritması için de kullanılmış ve başarılı sonuçlar elde edilmiştir (Cheng and Gong, 2008). Yu et al. (2009) KKS algoritması ile tüm pikselleri dolaşarak pikseller üzerinde feromon bırakılmasını sağlamış, ikinci aşamada bulanık c-means algoritmasında bu bilgiyi de kullanarak kümeleme yapılmıştır. Böylece gürültülü resimlerde bile rahat kesimleme yapılabilmiştir. Tez yazım sürecinde renkli resimler için gerçekleştirilen tek KKO tabanlı kümeleme ve kesimleme algoritması da bulanık c-means algoritması ile beraber kullanılmıştır (Yu et al., 2010). Çalışmada, KKO algoritması c-means algoritması için küme sayısının ve merkezlerinin ilklenmesinde kullanılmıştır.

Sadece KKO algoritmasını kullanarak kümeleme yapmaya çalışan metotlardan bir tanesi Han ve Shi (2007) tarafından öne sürülmüştür. Buna göre öncelikle resme ait özellikler (ton, renk farklılığı ve komşuluk bilgileri) elde edildikten sonra bu bilgiler 3 boyutlu uzayda bir karıncayı temsil edecek şekilde düşünülmüştür. Daha sonra her karınca olasılıklı seçim kuralına göre bir kümeyi (karıncayı) seçerek kendine birleştirir ve yeni bir küme oluşur. Oluşan her kümenin küme merkezi güncellenir. Bu işlem kümelenmemiş hiç bir karınca kalmayana dek devam eder.

Parçacık Sürü Optimizasyonu algoritmaları bölge büyütme ile kesimlemede genellikle diğer kesimleme tekniklerinin dezavantajlarını ortadan kaldırmak için kullanılmışlardır. Örneğin Mobadi et al. (2006) ve benzer bir çalışma yapan Tseng et al. (2009) yılan (snake) modelindeki konturun ilklenmesi aşamasında kontur kontrol noktalarını PSO algoritması yardımıyla tespit etmişlerdir. Diğer bir çalışmada, Mohsen et al. (2010) bölge büyütme tekniğinde uygun kaynak (seed) pikselleri PSO algoritmasıyla bularak kesimleme yapmışlardır.

PSO algoritmaları ile kenar belirleme algoritmalarının birçoğunun kullandığı farklı filtrele alternatif olarak optimum filtre değerlerini bulmaya çalışılmıştır (Alipoor et al., 2010). PSO ile bulunan filtre resim üzerinde uygulandığında kenarlar daha belirgin bir şekilde ortaya çıkabilmektedir. Öte yandan, Setayesh et al. (2009) kenarları eğrilerin birleşimi olarak düşünmüş ve buna göre bu eğrilere ait kontrol noktalarını PSO algoritması ile bularak kenarları tespit edebilmişlerdir. Yapılan çalışma özellikle gürültülü resimlerde klasik kenar belirleme yaklaşımlarından daha iyi sonuç verdiği tespit edilmiştir. Diğer bir çalışmada (Jianlai et al, 2009), kenarlar belirlenirken hücrel sinir ağı (cellural neural network (CNN)) kullanılmış ve CNN'e ait en uygun şablonlar PSO algoritması ile belirlenmiştir.

Literatürde, eşikleme yolu ile kesimlemede birçok PSO yaklaşımı kullanılmıştır. Fan ve Lin (2007), PSO ile Expectation Maximization algoritmasını, Guo and Li (2007) ile Hongmei et al. (2010) ise PSO ile maksimum entropi algoritmasını melez bir şekilde kullanarak çok seviyeli eşikleme yapmışlardır. Yin (2007) PSO algoritmasını çok seviyeli eşiklemede zaman alıcı ve iyi sonuçlar veremeyen minimum çapraz entropi eşikleme algoritmasının bulduğu eşik değerlerini iyileştirmek için kullanmıştır. Chander et al. (in press) uyarlanabilir ve değiştirilmiş bir PSO algoritmasını iki seviyeli Otsu metoduyla beraber iteratif kullanarak çok seviyeli kesimleme işlemini başarıyla gerçekleştirmiştir. Buna ek olarak, literatürde, ikili PSO ile eşikleme yapan (Djerou et al., 2009) ve diğer eşikleme yöntemlerinin iyileştirmek için PSO kullanan çalışmalar (Chao et al., 2007; Lei and Fan, 2008; Lee et al., 2008; Wei and Kangling, 2008; Gao et al., 2010) mevcuttur. Ayrıca, Hammouche et al. (2010) altı farklı üst-sezgisel algoritmayı eşiklemede kullanarak kıyaslamış, PSO algoritmasının diğer birçok sezgiselden iyi sonuçlar verdiğini göstermiştir.

Kümeleme yoluyla kesimleme yapan PSO algoritmalarından bazıları tamamen PSO tabanlı çalışan yaklaşımlardır. Örneğin, Han and Shi (2007) su

altındaki balık iskeletlerinin kesimlenmesinde PSO algoritmasını kullanmıştır. Bunun için, parçacıklar resim üzerinde bir konumda yerleştirilmiş ve bu konumlar küme merkezleri olarak ilklendirilmiştir. Daha sonra komşuluk bilgilerine göre pikseller kümelenerek parçacıklar hareket ettirilmiş ve iteratif bir şekilde küme merkezlerinin konumu değiştirilerek optimum küme merkezleri tespit edilmeye çalışılmıştır. Omran et al. (2005) tamamiyle PSO tabanlı bir kümeleme algoritması geliştirerek K-means ve bulanık c-means algoritmalarından daha iyi kümeleme ve kesimleme sonuçları elde etmiştir. Diğer PSO yaklaşımları diğer kesimleme yaklaşımlarıyla beraber kullanılmıştır. Örneğin, bir çalışmada (Souza and Costa, 2009), PSO tabanlı bir K-means algoritması K-means ve bulanık c-means algoritmaları ile karşılaştırılmıştır. Diğer bazı çalışmalarda (Zang, 2009; Li and Shen, 2010) da PSO algoritması bulanık c-means algoritması ile kullanılarak kesimleme yapılmıştır.

5. MF/ANT (MATCH FILTER / ANT COLONY) ALGORİTMASI İLE RETİNA GÖRÜNTÜLERİNDE KAN DAMARININ TESPİTİ

Retina görüntülerinde kan damarlarının tespit edilmesi, diyabet, glokoma ve yüksek tansiyon gibi bazı ciddi hastalıkların teşhisinde önemli rol oynamaktadır. Retina görüntülerinin çekilmesi esnasında parlaklık, yansıma, farklı açılardan çekilme gibi birçok etken retina görüntülerinin kalitesini etkilemektedir. Bunun yanında gürültülerin bulunması, damarlı bölge ile arka plan arasında düşük zıtlığın bulunması, damar genişliğinin değişkenliği ve birçok patolojik bölgenin kan damarlarına benzer renklere sahip olmaları kan damarlarının tespiti (kesimlenmesini) güçleştirmektedir.

Literatürde var olan üç temel kan damarı çıkarma yaklaşımı vardır. Bunlar çekirdek-tabanlı, sınıflayıcı-tabanlı ve takip edici-tabanlı yaklaşımlardır (Hoover et al., 2000).

Çekirdek-tabanlı metotlar (Chaudhuri et al., 1989; Nekovei and Sun, 1995), Gauss eğrisi şeklinde modellenen bir filtrenin resim üzerinde döndürülerek damarlı bölgelerin tespit edilmesini esas alır. Bu filtreler önceden tanımlı olup damarlı bölgelerinin özelliklerine göre tanımlanırlar.

Sınıflandırıcı-tabanlı metotlar (Cote et al., 1994; Tamura et al., 1983; Martinez-Perez et al., 1999) iki aşamada damarlı bölgeleri tespit eder. Öncelikle düşük düzeyli bir algoritma ile kesimleme yapılarak resim birbirine bağlı alanlara ayrılır. Daha sonra kan damarları birçok özelliğe göre aday alanlardan sınıflandırılarak elde edilir.

Takip edici-tabanlı metotlar (Chutatape et al., 1998; Tollas and Panas, 1998; Can et al., 1999) ise bilinen bir damar bölgesi üzerinden başlayarak bazı yerel özelliklere göre damar bölgesi üzerinden takip ederek damarları tespit etmeye çalışır.

Bu alanda şu ana dek sürü zekâsı yaklaşımlarını kullanan bir çalışma bulunmamaktadır. Bu tezde, hem damarlı bölgeleri tespit etmede etkili hem de çekirdek-tabanlı yaklaşımların en önemlisi olan match filtre ile birlikte kullanıldığında bu algoritmanın performansını arttıran yeni bir sürü zekâsı algoritması geliştirilmiştir. Bu bölümde, önce kendisiyle birlikte kullanılan Match

filtresine değinilmiş sonra damarların tespitinde geliştirilmiş olan sürü tabanlı kesimleme yöntemine yer verilmiştir. Son olarak gerçekleştirdiğimiz uygulamanın retina resimlerini içeren veri kümeleri üzerindeki test sonuçları gösterilmiştir.

5.1. Match Filtresi (Filter) Algoritması

Retina görüntüleri için tasarlanan match (karşılaştırma) filtreleri (MF) bu görüntülerdeki kan damarlarının özelliklerine göre belirlenir. Bu özellikler;

- Kan damarlarının şekilleri küçük eğrilik gösterir.
- Damarların arka plandan daha koyu renkte olduğu gözlemlenir.
- Damar genişlikleri 2 ile 20 piksel aralığında değişir.

MF algoritması retina resimleri üzerinde uygulanırken öncelikle resmin sadece yeşil kanalı alınarak işlemler gerçekleştirilir. Çünkü bu kanalda arkan ile damarlar arasındaki tonaj farkı daha belirgindir. Daha sonra bir döndürme matrisi kullanılarak, her defasında tanımlanan filtre döndürülerek resme uygulanır. Burada, filtreye uygulanan döndürme matrisi aşağıdaki şekilde tanımlanmıştır:

$$r = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \quad 5.1$$

Burada θ önceden belirlenmiş bir döndürme açısını ifade eder. Bu döndürme matrisi ile bağlantılı olarak elde edilen filtre, çekirdek filtre ile döndürme matrisinin çarpımından elde edilir ve şu şekilde formüle edilir:

$$\bar{p}_i = [u \quad v] = K(x, y) * r^T \quad 5.2$$

$$K_i(x, y) = -\exp\left(\frac{-u^2}{2\sigma^2}\right) \quad \forall \bar{p}_i \in N$$

Burada $N [u, v]$ alanı içerisindeki komşuluk değerini ($|u| \leq 3\sigma$ ve $|v| \leq L/2$), L ise filtrenin büyüklüğünü ifade eder ve bu değer önceden tahmin edilen damar genişliğine göre belirlenir.

Belirlenen açıya göre döndürülerek oluşturulan her filtre resmin yeşil kanalı üzerinde uygulandığında her bir filtre için ayrı bir gri tonajlı resim üretilmiş olur. Daha sonra bu resimler karşılaştırıldığında damarlı bölgelerin daha fazla filtreye

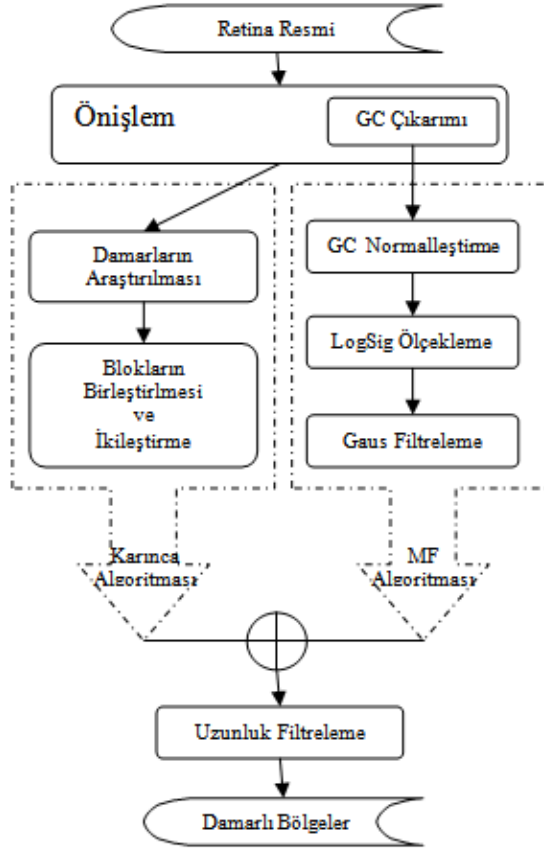
cevap verdiği ve bu nedenle daha açık renkte değerler aldığı gözlemlenir. Son olarak uygun bir eşikleme metodu ile eşikleme yapılarak damar bölgelerini ifade eden siyah-beyaz sonuç resmi elde edilecektir.

5.2. MF/Ant Algoritması

Bu bölümde kan damarlarının tespitinde geliştirilen yöntem ayrıntılı bir şekilde bahsedilmiştir. MF/Ant algoritması ile damarların tespitinde sırasıyla şu adımlar izlenmiş olup bu adımlar Şekil 5.1’de gösterilmiştir:

- **Önişlem adımı:** Bu aşamada, ham retina görüntüleri üzerinde önişleme yapılarak resmin iyileştirilmesiyle damar bölgelerinin daha iyi ortaya çıkması hedeflenmiştir. Bu aşamada kullanılan ön işleme işlemleri: yeşil kanal çıkartımı, görüntü iyileştirme ve blok işaretleme işlemleridir.
- **Karınca Algoritması adımı:** Bu adım Match Filtresi uygulanma adımıyla paralel olarak işletilmektedir. Bir üst aşamada işlenen resim, işaretlenmiş bloklar üzerinde damar araştırması yapma, blokları birleştirme ve ikili resim haline getirme alt basamaklarıyla işletilerek karınca algoritmasının sonucu elde edilir.
- **MF (Match Filtresi) algoritması adımı:** Bu adımda öncelikle yeşil kanal resmi normalleştirilir. Sonrasında LogSig ölçeklendirmesi uygulandıktan sonra, filtre resme uygulanarak ikili resim elde edilir.
- **Birleştirme adımı:** Bu adımda karınca ve MF algoritmalarının sonuçları birleştirilerek sonuç resmi elde edilir.
- **Uzunluk filtreleme adımı:** Bu adım son aşamada yapılır ve bu aşamada küçük parçacıklar ve gürültüler yok edilir.

Aşağıdaki alt başlıklarda bahsedilen adımları teker teker ayrıntılı bir şekilde değinilmiştir.

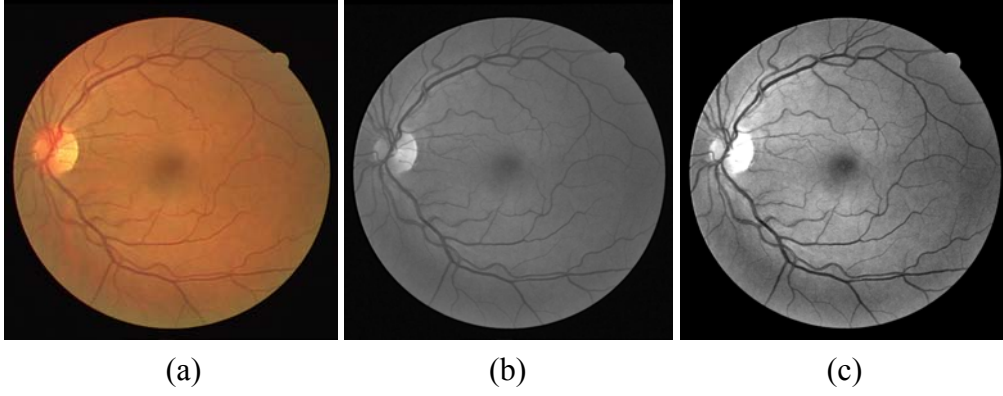


Şekil 5.1 MF/Ant algoritmasının metodolojisi

5.2.1. Önişlemler

Önişlemler basamağında retina resmi üzerinde şu işlemler sırasıyla uygulanmıştır:

- Yeşil kanalda damarlar ile arka plan arasındaki tonaj farkının diğer kanallara ve gri tonaja çevrilmiş halinden daha fazla olduğu bilinmektedir (Perfetti et al., 2007). Dolayısıyla, RGB uzayındaki G uzayı çıkartılarak diğer tüm işlemler bu kanal üzerinde gerçekleştirilmiştir.
- Damarların daha da belirgin olabilmesi için doğrusal olmayan yoğunluk değişimi fonksiyonu uygulanmıştır. Bu işlem sonunda oluşan bir örnek resim şekil 5.2’ deki gibi olacaktır.



Şekil 5.2. (a) Orjinal retina resmi (b) Yeşil kanalın gri tonajda gösterilmiş resmi (c) Yeşil kanalın doğrusal olmayan yoğunluk dönüşümü sonrası görüntüsü

- Damarların 512x512 lik bir resimde genellikle 15 pikselden az bir genişlikte olduğu tespit edilmiştir (Zana and Klein, 1999). Buradan ilham alınarak, resim karıncaların dolaşacakları ufak bloklara bölünmüştür.
- Parçalanmış bu bloklar analiz edilerek içlerinde damar bölgelerinin bulunup bulunamayacağı analiz edilmiştir. Bu işlem entropi kullanılarak kontrol edilir. Eğer entropi değeri blok için çok küçükse, o bloğun bir damar parçası içermediği kabul edilir, aksi durumda damar bölgesi var kabul edilir. Entropisi büyük olan bu bölümler işaretlenir. Sadece işaretli bloklar üzerinde karıncaların dağıtılıp damarlı bölgelerin tespit edilmesine izin verilir.

5.2.2 Karınca algoritması

Bu aşamada her bir işaretli blok üzerinde karınca algoritması uygulanır. Her bir blok, pikselleri noktalar olarak düşünülen iki boyutlu bir çizge olarak kabul edilmiştir. Klasik Karınca Kolonisi Optimizasyonu algoritmalarında, kenarlar çözümü oluşturan çözüm yollarının her bir parçasını oluştururken, burada kenar kavramı sadece piksellerin komşuluk bilgilerini tutmaktadır. Tüm işlemler (sezgisel fonksiyon, feromon güncelleme işlemleri gibi) noktalar (pikseller) üzerinde gerçekleştirilir.

Algoritma ilklenme aşamasında, tüm pikseller üzerindeki feromon değeri sıfıra çok yakın değerlerle ilklenir. Her bir iterasyonda belli sayıdaki karıncalar blok içindeki pikseller üzerinde rastgele dağıtılır. Karıncalar KKS'deki olasılıklı seçim kuralına göre adım adım pikseller üzerinde dolaşarak kendilerine ayrılan maksimum adım sayısı kadar pikseller üzerinde dolaşmaya çalışırlar. Her bir (i,j)

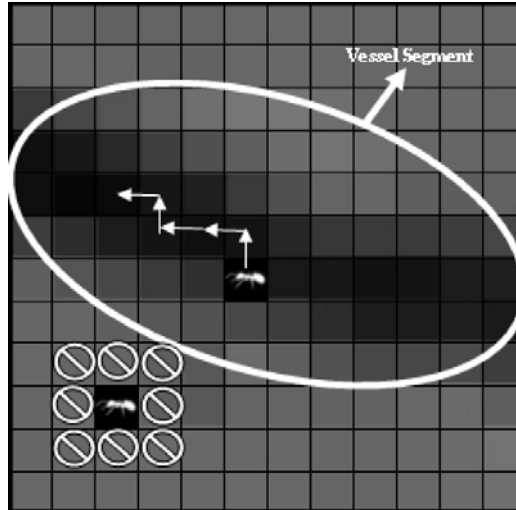
pozisyonundaki piksele ait sezgisel fonksiyon, η_{ij} , aşağıdaki şekilde tanımlanmıştır.

$$\eta_{ij} = \begin{cases} \frac{p_{\max} - p_{ij}}{p_{\max}} & \text{eğer } p_{ij} < \theta \text{ ve } p_{ij} / p_{\max} > \psi \\ 0 & \text{aksi halde} \end{cases} \quad 5.3$$

Burada $p_{\max}$ o blok içindeki en büyük piksel değerini, p_{ij} (i,j) pozisyonundaki pikselin değerini, θ ve ψ iki farklı eşik değerini ifade eder.

Tanımlanan sezgisel fonksiyon karıncaları damarlı bölgeleri içeren piksellere doğru yönelmelerini sağlar. Eğer sezgisel fonksiyon tüm ziyaret edilmemiş komşular için sıfıra eşit çıkarsa, karıncanın etrafında damara ait bir piksel adayının olmadığı anlaşılır ve karınca o piksel üzerinde öldürülür. Diğer bir ölüm olayı karınca maksimum adım sayısını tamamladığı zaman gerçekleştirilir (Bkz Şekil 5.3). Böylece bir sonraki turlar için yeni karıncalar oluşturulmasına olanak tanınır. Yine klasik KKO algoritmalarından farklı olarak, her karınca her adımı sonunda uğradığı pikselin sezgisel fonksiyonu ile doğru orantılı olacak şekilde feromon depolar.

Tüm bu işlemler her bir iterasyon için uygulanır. Tüm iterasyonlar sonucunda damarlı bölgelerde daha çok feromon biriktiği diğer alanlarda daha az feromon bulunduğu görülür.



Şekil 5.3. Karıncanın dolaşımı (ortada), karıncanın ölmesi (sol altta)

Elde edilen tüm feromon depolanmış bloklar, diğer tüm bloklarla (işaretlenmemiş olanlarla) birleştirilerek tüm resim elde edilir. Bu resimdeki pikseller üzerindeki feromon değerleri üzerinde eşikleme yapıldığında retina resmindeki kan damarlarının gösteren ikili resim elde edilmiş olur.

5.2.3 MF algoritmasının uygulanması

Match filtresi paralel olarak resimde uygulanırken aşağıdaki basamaklar takip edilmiştir:

- **Yeşil kanal normalleştirme:** Retina resimleri farklı parlaklık ortamlarında çekilmişlerdir. Bu farklılık MF algoritmasının performansını etkilemektedir. Bu etkinin yok edilmesi için yeşil kanalda elde edilen resim üzerinde normalleştirme aşağıdaki formüle göre uygulanmıştır:

$$GC'(x, y) = \frac{GC(x, y) - \mu_{gc}}{\sigma_{gc}} \quad 5.4$$

Burada GC yeşil kanal resmini μ_{gc} ve σ_{gc} sırasıyla bu kanaldaki piksel değerlerinin ortalama ve standart sapma değerlerini ifade eder

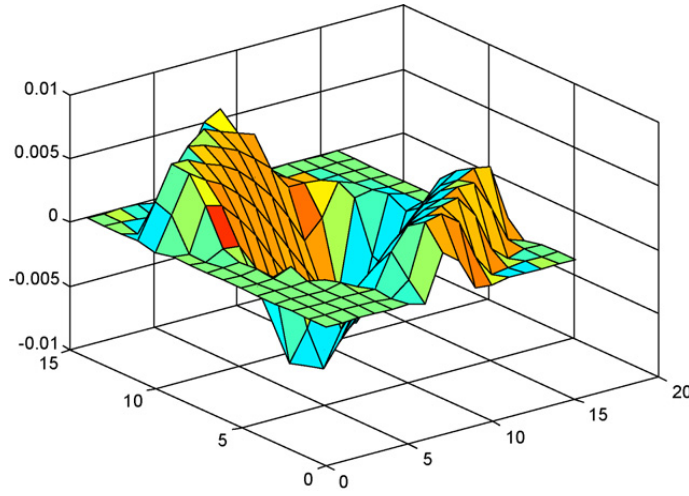
- **LogSig Ölçeklendirme:** Bu ölçeklendirme işlemi resimdeki bazı gürültüleri yok etmek için kullanılmıştır. Yapılan LogSig ölçeklendirme işlemi aşağıdaki gibidir:

$$GC''(x, y) = \frac{1}{1 + \exp^{-2(GC'(x, y) - c)/r}} \quad 5.5$$

Burada c yazılı alanın merkezini, r ise yarıçapını ifade eder ve bu değerler sırasıyla 0 ve 1 olarak kabul edilir.

- **Match (Gauss) Filtreleme:** Match filtrelerini oluşturmanın birçok yolu vardır. Bunlardan en çok bilinenleri Gauss, Chebyshev ve Kaiser tasarımlarıdır. Bu çalışmada da 9 uzunluklu ve standart sapması 2 olan 13 noktalı bir Gauss filtresi kullanılmıştır. Oluşturulan bu filtre 10 derecelik açılarla döndürme matrisine tabii tutulmuş ve resim üzerinde 18 farklı şekilde uygulanmıştır. Üretilen bu filtrelerden birinin grafiksel gösterimi

Şekil 5.4’de gösterilmiştir. Elde edilen resim daha sonra ikili resim haline getirilerek burada da kan damarlarına ait resim elde edilir.



Şekil 5.4. 50 derecelik açı için oluşan ve uygulanan match filtresi

5.2.4 Sonuçları birleştirme ve uzunluk filtreleme

Karınca ve MF algoritmalarından alınan sonuç resimleri OR işleminden geçirilerek birleştirilir. Böylece her ikisinin tanımlamış oldukları damar bölgelerinin birleştiği ve her iki algoritmanın da eksik kalan kısımlarının tamamlanmış olduğu ikili bir resim elde edilmiş olur.

Bu birleştirme sonucunda elde edilen ikili resimde, bazı gürültülerin veya bağlantısız bazı küçük damar bölgelerinin olduğu görülür. Bu bölgelerin yok edilmesi için uzunluk filtreleme işlemi uygulanmıştır. Buna göre, birbirinden bağımsız her bir bölge bir nesne olarak kabul edilerek bir uzunluk tablosuna kapladıkları piksel bilgisiyle beraber eklenmiştir. Daha sonra bu tablo uzunluk sırasına göre sıralanarak bu tabloda belli bir eşik değerinden düşük olan nesneler ve bölgeler resimden temizlenmiştir. Böylece temiz sonuç resmi elde edilmiş olur.

5.3. Deneysel Çalışmalar

5.3.1 Kullanılan veri kümesi ve başarı ölçütleri

Geliştirilen yöntem DRIVE (Staal et al., 2004) veri kümesi üzerinde test edilmiştir. DRIVE veri kümesi 565x584 piksel boyutlu RGB uzayında gösterilen TIFF formatındaki 40 tane farklı retina resminden oluşmaktadır. Bu resimlerden yarısı deneme ve diğer yarısı test için ayrılmıştır. Ayrıca her resim için iki doktor

tarafından belirlenmiş kan damarları resimleri yer doğrulama resimleri olarak sunulmuştur. Bunlardan ilk doktorun ürettiği yer doğrulama resimleri “Altın Standart” olarak kabul edilir ve tüm karşılaştırmalar “Altın Standart” resimler üzerinde gerçekleştirilir (Staal et al., 2004).

Algoritmaların performansının ölçümünde iki farklı ölçüt kullanılır. Bunlardan birincisi ROC (Receiver Operating Characteristics) eğrisi altında kalan alan değeri, diğeri ise maksimum kesimleme başarısıdır.

ROC eğrisi farklı eşik değerlerinde yanlış ile doğru oranlarının değişimi ile ortaya çıkan bir grafikdir. Doğru oranı, altın standart resme göre doğru bulunan piksellerin sayısının altın standart resimdeki damar piksellerinin sayısına bölünmesi ile elde edilir. Yanlış oranı ise yanlış tespit edilen piksellerin sayısının altın standarttaki damar olmayan piksellerin sayısına oranına eşittir. Eğer elde edilen ROC eğrisinin altında kalan alan 1’e eşit olduğunda mükemmel kesimleme yapılmış olduğu anlamına gelir.

Maksimum kesimleme başarısı da elde edilen sonuç resminin ne kadar standart resimle çakıştığını gösteren bir ölçüttür. Doğru sınıflandırılmış (damar veya değil) tüm piksellerin standart resimdeki tüm piksellere bölünmesi ile elde edilen değer başarı ölçüsü olarak kabul edilir.

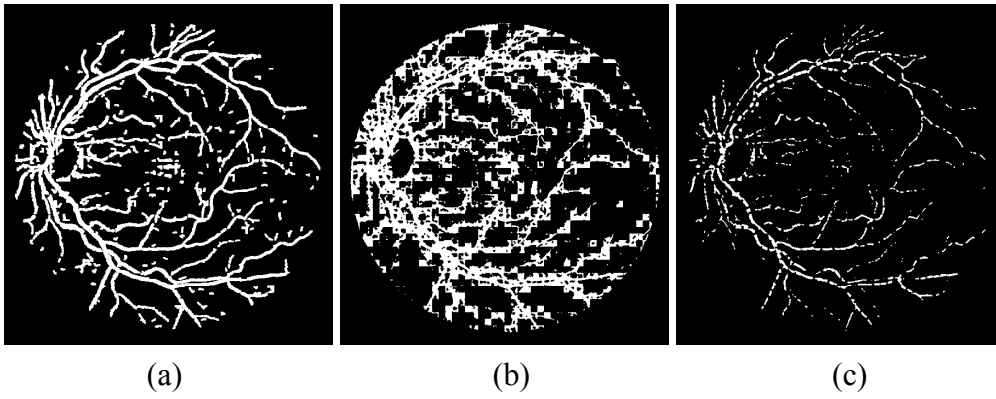
5.3.2 Parametrelerin belirlenmesi

Metodoloji iki farklı algoritma içerdiğinden her bir algoritmaya ait parametre değerleri ayrı ayrı tespit edilmiştir.

Karınca algoritmasının parametrelerinin tespiti oldukça zor olduğundan bu algoritmaya ait temel parametreler DRIVE veri kümesi üzerinde defalarca denenerek elde edilmiştir (Bulunan değerler için Çizelge 5.1’e bakınız). Ayrıca Şekil 5.5’de feromon buharlaşma katsayısının (ρ) ve q_0 parametrelerinin sonuçlar üzerindeki etkisi gösterilmiştir.

Çizelge 5.1. Seçilen parametre değerleri.

Parametre	Değeri
β	5.0
q_0	0.95
ρ	0.7
θ	200
ψ	0.4
İterasyon Sayısı	10



Şekil 5.5. Farklı q_0 ve ρ parametreleri için kesimleme sonuçları. (a) Çizelge 5.1'deki değerleri göre kesimleme sonucu, (b) ve (c) sırasıyla $q_0 = 0.2$, $\rho = 0.9$ ve $q_0 = 0.7$, $\rho = 0.1$ göre kesimleme sonuçları.

Öte yandan, her bir bloktaki karınca sayısı ile bunların adım sayıları blokların boyutları değiştikçe performansı etkilemektedir. Dolayısıyla bu iki değer ile blok boyutu arasında bir bağlantı vardır. Bu bağlantı tek retina resimlerinden elde edilen başarı sonuçlarına göre Çizelge 5.2'de gösterilmiştir. Çizelge 5.2'den de anlaşılacağı gibi karınca sayıları ve maksimum adım sayısı dengeli tutulmalıdır. Testlerde çok karınca ve az adım sayısı ya da az karınca ve çok adım sayısı seçildiğinde algoritma performansının düştüğü saptanmıştır. Çizelge 5.2'den alınan değerlere göre, tüm performans testleri için 12x12 lik bloklara ayrılma seçilmiş ve her blokta 10 karıncanın 50 adım atabilmesine izin verilmiştir. Bu seçiminin makul bir zamanda 0,9208 oranında başarı sağladığı görülebilmektedir.

5x5 lik bloklara parçalandığında daha iyi sonuçlar elde edildiği görülse de bu bloklar tercih edilmemiştir. Bunun sebebi, bu blok seçildiğinde karınca sayısı ve adım sayısı arttıkça algoritmaya ayrılan zamanın gittikçe artması ve 5x5

boyutundaki bazı blokların geniş bir damarın tam ortasında kaldığından blok işaretleme aşamasında işaretlenmeden bırakılabilmesidir. Bu durumda algoritmanın performansı düşeceği gibi ortaya çıkan ikili resimde de kan damarlarının görünümü hatalı olacaktır.

MF algoritmasının parametre seçimi de yine DRIVE veri kümesi üzerindeki bir resim referans alınarak gerçekleştirilmiştir. Filtre uzunluğu 6 dan 25 e birer birer artırılarak sonuç resmi elde edilmiş ve her birisi için maksimum başarı ölçülmüştür. Buna göre en yüksek başarı 13 uzunluğunda olduğu tespit edilmiştir.

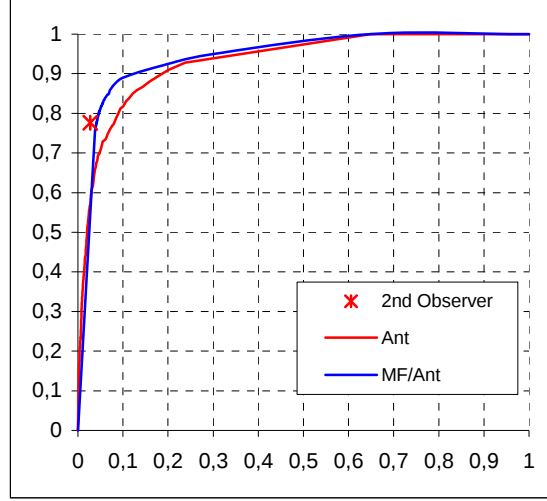
Çizelge 5.2. Blok boyutuna göre karınca sayısı ve adım sayısının etkisi.

Blok Boyutu	Karınca Sayısı	Adım Sayısı			
		10	20	50	100
		Max. Baş. (Zaman)	Max. Baş. (Zaman)	Max. Baş. (Zaman)	Max. Baş. (Zaman)
5 x 5	5	0.924 (25 s)	0.9243 (26 s)	0.9244 (27 s)	0.9249 (27 s)
	10	0.9271 (45 s)	0.9264 (46 s)	0.9276 (47 s)	0.9266 (47 s)
	20	0.9285 (83 s)	0.9284 (88 s)	0.9284 (88 s)	0.9291 (88 s)
	100	0.9288 (425 s)	0.9289 (428 s)	0.9291 (430 s)	0.9290 (430 s)
12 x 12	5	0.9037 (12 s)	0.9092 (17 s)	0.9139 (19 s)	0.9130 (21 s)
	10	0.9042 (19 s)	0.9130 (27 s)	0.9208 (32 s)	0.9181 (37 s)
	20	0.9078 (38 s)	0.9155 (47 s)	0.9214 (66 s)	0.9201 (70 s)
	100	0.9114 (150 s)	0.9174 (232 s)	0.9221 (330 s)	0.9258 (341 s)
24 x 24	5	0.887 (6 s)	0.8892 (8 s)	0.8939 (11 s)	0.8947 (12 s)
	10	0.8918 (9 s)	0.8961 (12 s)	0.9015 (19 s)	0.9053 (24 s)
	20	0.8953 (13 s)	0.9004 (19 s)	0.9072 (32 s)	0.9075 (44 s)
	100	0.9025 (48 s)	0.9062 (82 s)	0.91 (145 s)	0.9214 (217 s)

5.3.3 DRIVE veri kümesi üzerinde test sonuçları

DRIVE veri kümesinin test resimleri üzerinde hem karınca algoritması hem de MF/Ant algoritması test edilmiştir. ROC eğrisinin oluşturulmasında feromon değerleri üzerinde 0'dan 100'e her bir aşamada 0,1 arttırarak elde edilen resimlerden elde edilen sonuçlar kullanılmıştır. Şekil 5.6'da altın standart bir

resim ile karşılaştırıldığında karınca (Ant) ve MF/ant algoritmalarından elde edilen sonuçlardan üretilen ROC eğrileri ve ikinci gözlemcinin (doktorun) (2nd Observer) bu eğriler üzerindeki doğru/yanlış oran değeri gösterilmiştir.



Şekil 5.6. Bir retina resmi için ROC eğrisi ve 2. Gözlemciye göre karşılaştırılması

Tüm resimler için elde edilen ROC eğrileri teker teker oluşturulmuş ve altlarında kalan alanlar hesaplanmıştır. Bu 20 resim için elde edilen ROC eğrisi alan değerleri Çizelge 5.3’de gösterilmiştir.

Çizelge 5.3. 20 resim için elde edilen ROC eğrisi altında kalan alan değerleri

Resim Numarası	ROC Eğrisi Altında Kalan Alan	
	Karınca Algoritması	MF/Ant Algoritması
1	0.9463	0.9546
2	0.9459	0.9528
3	0.9323	0.9410
4	0.9344	0.9387
5	0.9309	0.9385
6	0.9278	0.9357
7	0.9191	0.9275
8	0.9111	0.9299
9	0.9233	0.9286
10	0.9274	0.9361
11	0.9237	0.9376
12	0.9428	0.9488
13	0.9199	0.9277
14	0.9417	0.9490
15	0.9382	0.9468
16	0.9344	0.9357
17	0.9209	0.9367
18	0.9394	0.9492
19	0.9403	0.9509
20	0.9421	0.9493
Ortalama	0.9320	0.9407

Yapılan diğ er bir testte de hem t m resimler i in maksimum ba arı elde edilmi  hem de t m resimler i in se ilmesi gerekli olan feromon e ik değ eri belirlenmi tir. Elde edilen sonu lar  izelge 5.4’de g sterilmi tir.  izelge 5.4’de her bir resme kar ılık gelen e ik değ erleri ve maksimum değ erleri g sterilmi , ayrıca ortalama değ erler sunulmu tur. Her resim i in makul bir ba arı elde etmek i in bu e ik değ erlerinin ortalaması genel e ik değ eri olarak kullanılabilir.

 izelge 5.4. 20 DRIVE veri k mesindeki resimler i in maksimum ba arı değ erleri.

Resim Numarası	E�ik Değ�eri	Maksimum Ba�arı	
		Karınca (Ant) Algoritması	MF/Ant Algoritması
1	10.4	0.9120	0.9323
2	10.9	0.9115	0.9370
3	9.6	0.9118	0.9207
4	9.3	0.9239	0.9368
5	9.4	0.9205	0.9238
6	10.4	0.9208	0.9264
7	7.9	0.9234	0.9269
8	8.6	0.9124	0.9145
9	8.9	0.9309	0.9386
10	8.5	0.9290	0.9341
11	10.5	0.9160	0.9196
12	8.7	0.9233	0.9272
13	9.2	0.9187	0.9267
14	9	0.9213	0.9259
15	12.3	0.9294	0.9219
16	8.1	0.9168	0.9293
17	11.6	0.9169	0.9342
18	11.5	0.9155	0.9354
19	9	0.9208	0.9288
20	9.6	0.9282	0.9365
Ortalama	9.67	0.9202	0.9293

Son olarak elde edilen sonu lar DRIVE veri setini kullanan literat rdeki diğ er Match Filtreleme yakla ımlarıyla kar ıla tırılmı tır. Bu kar ıla tırma

değerleri Çizelge 5.5’de gösterilmiştir. Karşılaştırma sonunda Staal et al. (2004) metodu en iyi sonuçları verse de algoritmanın zaman karmaşıklığı yüksektir ve çok uzun zaman sonucunda damarları belirleyebilmektedir. Öte yandan, literatürdeki MF algoritmalarının parametre değerlerini optimize ederek performanslarını arttırdığı iddia edilen OGMF (optimized MF) (Al-Rawi et al., 2007) ve GAMF (MF + Genetik Algoritma) (Al-Rawi and Karajeh, 2007) adında iki yaklaşım mevcuttur. Bu yaklaşımlar ile elde edilen sonuçlar MF/Ant algoritmasından yüksektir. Fakat her iki algoritma da filtre için gerekli olan uzunluk, L, parametresini tam sayı yerine kayan noktalı bir sayı şeklinde tanımlamaktadır. Bu uzunluk değeri aynı zamanda piksel olarak genişliğe karşılık gelen uzunluk değeri olduğundan bu değer tam sayı olmalıdır. Bu durum algoritmaların geçerliliği ve sonuçların sağlığı konusunda şüphe etmemize neden olduğundan bu iki algoritmaya ait sonuçlara Çizelge 5.5’de yer verilmemiştir.

Çizelge 5.5. Karşılaştırma sonuçları

Damar Tespit Yöntemi	Ortalama ROC Alanı	Ortalama Maksimum Başarı
MF/Ant	0.9407	0.9293
Karınca (Ant)	0.9320	0.9202
Staal et al. (2004)	0.9587	0.9547
GMF (Genel MF) (Chaudhuri et al., 1989)	0.7550	0.8850
Kirsch (Li and Chutatapei, 2000)	0.8687	0.8939
Zana and Klein (1999)	0.9174	0.9439
Sobel operator (Hoover et al., 2000)	0.7571	0.8936
Prewitt operator (Hoover et al., 2000)	0.7486	0.8951

6. IPSOANTK-MEANS ALGORİTMASI İLE ÇİÇEK KESİMİLEME

Çiçek tanımlama ve tanıma, botanikçiler tarafından yürütülen fakat çok zaman alıcı bir süreçtir. Bu yüzden, bilgisayar destekli çiçek tanıma ve alma sistemleri geliştirmek bilgisayar bilimlerinde çözülmesi gerekli olan önemli problemlerden bir tanesidir. Son yıllarda, görüntü işleme tekniklerini destekleyen çeşitli sistemler, çiçek tanıma için geliştirilmiştir (Das et al., 1999; Zou and Nagy, 2004; Evans et al., 2005; Nagy and Zou, 2002; Zhenjiang et al., 2006; An-xiang et al., 2004; Fukuda et al., 2008; Nilsback and Zisserman, 2008; Saitoh and Kaneko, 2000; Saitoh et al., 2004).

Renkli resimlerde çiçekli bölgelerin kesimlenmesi çiçek tanıma ve tanımlama sistemlerinin en temel ve önemli adımıdır. Ayrıca en zor kısmını oluşturmakla birlikte kesimlemenin kalitesi doğrudan tanıma sistemin başarısını da etkilemektedir. Bu amaçla geliştirilen çok az sayıda çiçek kesimleme yaklaşımı bulunmaktadır. Das et al. (1999) bazı çiçekler için veri tabanı oluşturarak bu veri tabanına ait çiçeklerin renk ve bazı özelliklerinden yararlanarak çiçekli bölgeyi arka plandan ayırmaktadır. Diğer bir metot da (Fukuda et al., 2008) yine çiçeklere ait renk bilgilerinden yararlandığı gibi bulanık bir renk kümeleme algoritması kullanmaktadır. Saitoh ve Kaneko (2000) K-means algoritması kullanmışlardır. Fakat çiçeklerin altına siyah bir perde konularak fotoğraf çekildiğinden kesimleme işlemi basitleştirilmiştir. Çizge parçalamaya (graph cut) dayanan başka bir otomatik kesimleme modeli Fukuda et al. (2009) tarafından geliştirilirken, Nilsback and Zisserman' da (2007,2010) iteratif fakat karmaşık bir kesimleme modeli ile otomatik çiçek bölgesi kesimlemesi yapabilmektedirler. Bunun yanında, geliştirilen bir yarı-otomatik kesimleme sistemiyle çoklu, üst üste binmiş veya ince yapraklı çiçeklerin kesimlenmesi sağlanabilmiştir (Zou, 2005).

Bu tezde, renkli resimlerde çiçeklerin kesimlenmesinde kullanılacak sürü zekâsı tabanlı bir kümeleme algoritması önerilmiştir. Bu kesimleme yöntemi çiçek kesimleme konusunda öne sürülen ilk sürü zekâsı tabanlı yöntemdir. Yönteme IPSOAntK-means adı verilmiştir ve üç algoritmanın melez olarak birleştirilmesinden oluşturulmuştur. Bunlar karınca kolonisi algoritması, artırımlı (incremental) parçacık sürü optimizasyonu ve K-means algoritmasıdır. Algoritma temel olarak renkleri en iyi şekilde kümelemeyi amaç edinmektedir. Bunu yaparken de K-means algoritmasında bulunan küme merkezi ilkleme problemini ortadan kaldırmayı hedeflemiştir. Aşağıdaki alt bölümlerde önce K-means

algoritmasına değinilmiş, daha sonra gerekleřtirilen metot ayrıntılı olarak bahsedilmiřtir.

6.1 K-Means Algoritması

K-means algoritması ğreticisiz kmeleme tekniklerinden bir tanesidir. Bu algoritmada elemanlar nceden belirlenmiř sayıda (k adet) kmeye otomatik olarak kmelenir. Bu algoritmada ama tm elemanların kendi kme merkezlerine olan uzaklıklarını en aza indirmektir (Tou ve Gonzales, 1974).

K-means algoritması řu řekilde iřletilir: ncelikle k adet kmeye karar verilir ve bu kmelere ait kme merkezleri rastgele belirlenir. Daha sonra kmelenecek tm elemanlar kendilerine en yakın kme merkezinin olduėu kmeye aktarılır. Her iterasyon sonunda yeni kme merkezleri o merkezdeki tm elemanların ortalamasına gre gncellenir. Bylece her iterasyon sonunda kme merkezleri tm elemanlara uzaklıkları en aza indirecek noktalara getirilmeye alıřılır. Daha biimsel bir ifade ile K-means ařaėıdaki fonksiyonu minimize etmeye alıřır.

$$\arg \min \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad 6.1$$

x_j i kmesine ait bir elemanı, μ_i i kmesinin kme merkezi (ortalamasını) ve S_i i kmesini ifade eder. Algoritma kme merkezlerinde bir hareket gzlemlenmediėinde veya hareket nemsenecek kadar az olduėunda sonlandırılır.

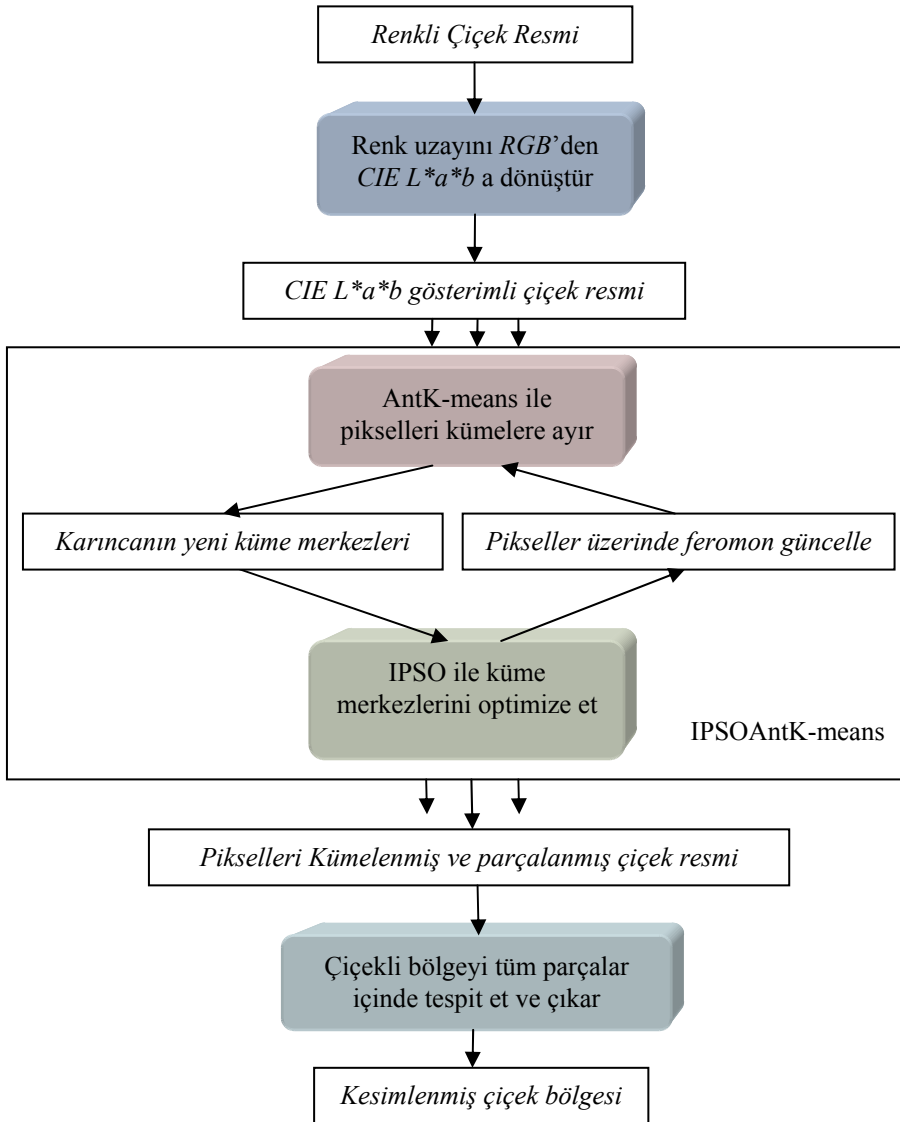
K-means algoritması ğreticisiz bir kmeleme yapmasına raėmen performansı ilk ařamada rastgele atanan kme merkezlerinin deėerinden ve verilerin daėılımından olduka etkilenmektedir. K-means algoritmasının bařarımının veri daėılımından ve kme ilkendirilmesinden baėımsız hale gelebilmesi iin optimizasyon teknikleriyle akıllı kme merkezlerini seebilecek ve bu yolla K-means algoritmasını iyileřtirecek yaklařımlara ihtiya vardır.

Bu tez alıřmasında, renkli resimlerin kesimlenmesi sırasında kme merkezi ilklenmesinden ve piksel daėılımlarından baėımsız olarak zm retebilen sr tabanlı melez bir K-means algoritması nerilmiřtir. Ařaėıdaki blmde nerilen

yaklaşım ile renkli çiçek resimlerinde kesimlemenin nasıl yapıldığı metodolojik olarak anlatılmıştır.

6.2 Renkli Çiçek Resimlerinde Kesimleme Yaklaşımı

Renkli çiçek resimleri için önerdiğimiz yöntem Şekil 6.1'deki gibidir. Şekil 6.1'de görüldüğü gibi öncelikle *RGB* renk uzayındaki resim *CIE L*a*b* renk uzayına dönüştürülür. Daha sonra bu resim kümeleme algoritması kullanılarak homojen bölgelere ayrılmaya çalışılır. Son aşamada kesimlenmek istenen çiçekli bölgeler diğer bölgelerden ayıklanarak çiçekli bölgenin hatları tespit edilmiş olur. Bu yönteme ait ayrıntılar aşağıdaki alt başlıklarda incelenmiştir.



Şekil 6.1. Önerilen yöntem.

6.2.1 Renk uzayının seçilmesi

RGB uzayı resimlerin gösteriminde kullanılmaktadır. Fakat görüntülerin kesimlenmesinde kullanıma pek uygun olmadığı 2. Bölümde bahsedilmişti. Konu piksellerin kümelenmesi olduğunda, akla gelen en önemli sorulardan bir tanesi iki renk arasındaki farkın en basit ve doğru olarak nasıl hesaplanacağıdır. Dolayısıyla, resimler kümeleme algoritmasına gönderilmeden önce renkler arasındaki farkın en rahat ve doğru hesaplandığı uygun bir renk uzayına dönüştürülmelidir.

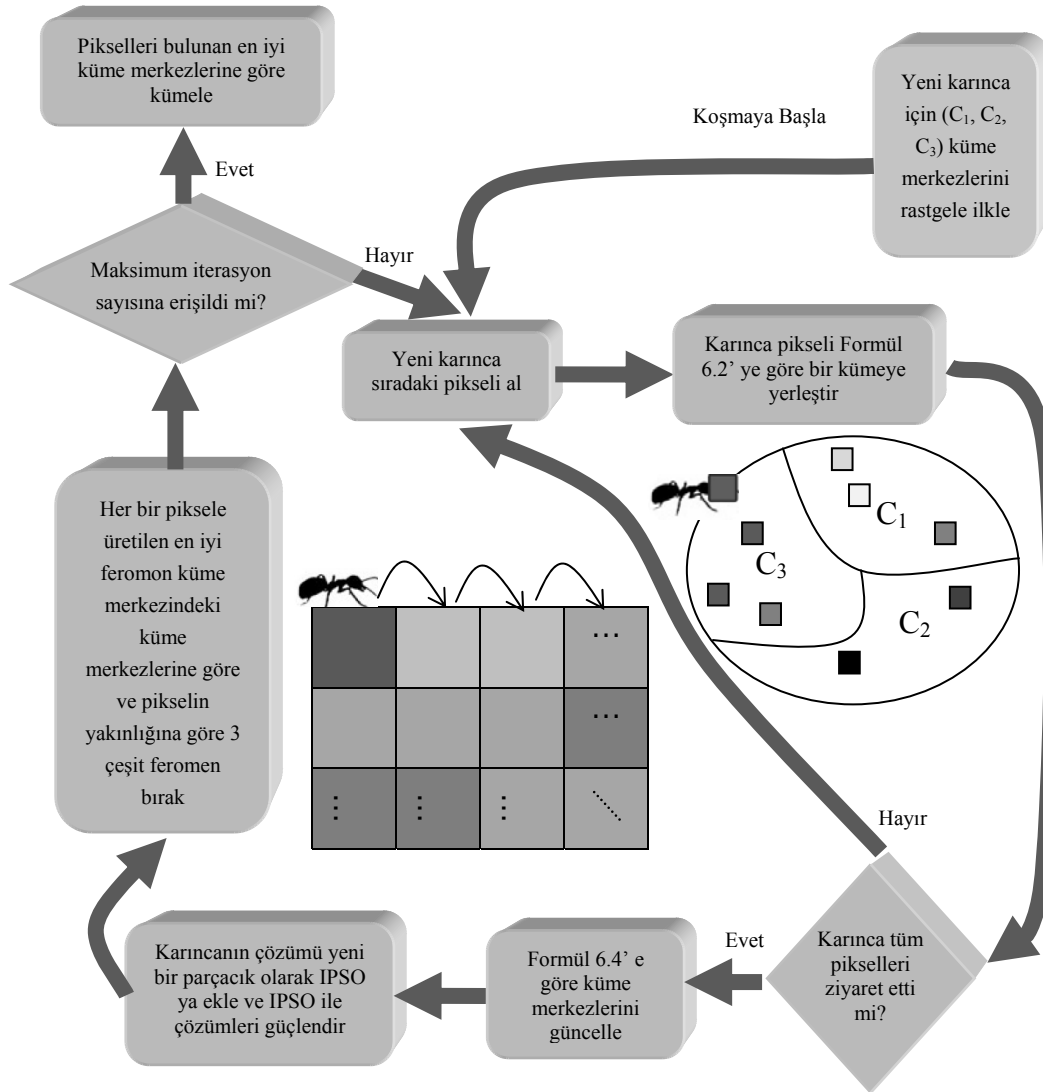
Bu çalışmada, resimler *RGB* uzayından *CIE L*a*b* uzayına dönüştürülerek kümeleme algoritmasına sokulmuştur. Çünkü *CIE L*a*b* uzayında iki renk arasındaki fark bir Öklid uzaklık hesabı ile (Formül 3.9) kolayca hesaplanabilir.

6.2.2 IPSOAntK-means kümeleme algoritması ile kesimleme

K-means algoritmasının en önemli problemlerinden bir tanesinin küme merkezlerinin ilklenme güçlüğü olduğundan bahsetmiştir. Önerilen yeni yaklaşım bu sorunu ortadan kaldırmak ve her durumda iyi kümeleme yapabilecek ve yerel minimumlardan kaçabilecek bir yapıda tasarlanmıştır. Kümeleme algoritmasına IPSOAntK-means algoritması adı verilmiştir ve üç algoritmanın değiştirilerek melez bir şekilde kullanılmasıyla geliştirilmiştir. Bu algoritmalar, artırımlı (incremental) parçacık sürü optimizasyonu (IPSO) (Montes de Oca et al., 2010; Montes de Oca et al., in press), karınca kolonisi algoritması ve K-means algoritmalarıdır.

Gerçekleştirilen algoritma şu şekilde işlemektedir. Her bir iterasyonda yalnız bir karınca hareket eder. Bu karıncaya ait küme merkezleri rastgele seçilir ve karınca tüm pikselleri teker teker ziyaret ederek olasılıklı seçim kuralına göre pikselleri bu belirlenen kümeler atar ve tüm küme merkezleri K-means iterasyonlarında olduğu gibi kümeler atanan piksellerin ortalama değerlerine göre tekrar güncellenir. Bu aşamadan sonra karıncanın ürettiği çözüm yani küme merkezleri, artırımlı parçacık sürü algoritmasındaki sisteme yeni dahil olan bir parçacığın ilk çözüm yolunu ifade eder. Yani karıncanın çözümü yeni parçacığın 3 boyutlu uzaydaki ilk konumudur. IPSO algoritması daha önceki iterasyonlarda algoritmaya dâhil olan muhtemel küme merkezleri ile birlikte (yani diğer parçacıklar ile birlikte) bu yeni parçacığı da dahil ederek küme merkezlerini optimum konumlara doğru hareket ettirir. Eğer yeni iyi konumlar bulunursa en iyi

konum yani küme merkezleri bilgisi günlendir (saklanır). İterasyon sonunda sadece en iyi parçacığın (yani küme merkezlerinin) çözümüne göre pikseller üzerinde feromon depo edilmesine izin verilir. Bu süreç her iterasyon için yapılır ve program bitirme koşuluna gelindiğinde sonlandırılır. Elde edilen en iyi sonuç algoritmanın elde ettiği sonuçtur ve pikseller bu küme merkezlerine göre kümelenecek resim parçalara ayrılır. Bahsedilen algoritmaya ait akış diyagramı Şekil 6.2’de gösterilmiştir.



Şekil 6.2. IPSOAntK-Means algoritmasının akış diyagramı.

Yapılan çalışmada Şekil 6.2’de de görülebileceği gibi toplam üç kümeye pikselleri bölme işlemi gerçekleştirilmiştir. Yani yeni bir karınca pikselleri dolaşırken pikselleri üç kümeden birisine taşır. Yapılan bazı çalışmalar (Biederman, 1985; Mojsilovic’ et al., 2000), insanların genel olarak renkleri 3-4 ana renge bölme ve buna göre gözleriyle kümeleme eğiliminde olduklarını tespit

etmişlerdir. Ayrıca, çok kümeye ayırma resimde çok fazla parçanın elde edilmesine neden olur. Çok parçaya ayrılmış bir resim içinden istenilen bir nesnenin çıkartılması oldukça güçtür. Dolayısıyla, bu çalışmada her bir karınca için pikselleri dolaşmaya başlamadan önce üç farklı küme merkezi rastgele belirlenmiştir ve daha sonra her bir karınca pikselleri teker teker dolaşarak ilgili kümelere atamaktadır. Bir k karıncası (x,y) pozisyonundaki bir $p(x,y)$ pikselini atayacağı j kümesini aşağıdaki formüle göre seçer:

$$j = \begin{cases} \arg \max_{j \in \Omega} \{ \tau_j(x,y)^\alpha \eta_j(x,y)^\beta \} & \text{if } q \leq q_0 + (i_c * ((1 - q_0) / n)) \\ J & \text{otherwise} \end{cases} \quad 6.2$$

Burada n toplam iterasyon sayısını, i_c şu anki iterasyon numarasını, q her defasında rastgele düzgün dağılma göre belirlenen bir sayıyı, q_0 bir parametreyi ($0 \leq q_0 \leq 1$), Ω küme merkezleri kümesini, $\tau_j(x,y)$ $p(x,y)$ pikseli üzerinde depolanan j kümesine ait feromon seviyesini, J ise karınca sisteminde kullanılan olasılıklı seçim kuralına göre seçilen kümeyi ifade eder. Karınca Sistemindeki olasılıklı seçim kuralı buradaki problem için aşağıdaki gibi formülleştirilir:

$$P_j = \begin{cases} \frac{[\tau_j(x,y)]^\alpha [\eta_j(x,y)]^\beta}{\sum_{h \in \Omega} [\tau_h(x,y)]^\alpha [\eta_h(x,y)]^\beta} & \text{if } j \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad 6.3$$

Burada da α ve β iki ayar parametresi olup $\alpha = 1$ olarak belirlenmiştir. Bu problem için sezgisel fonksiyon $\eta_j(x,y) = 1 / \text{dist}(C_j^k, p(x,y))$ şeklinde tanımlanmış olup $\text{dist}(C_j^k, p(x,y))$ pikselin $(p(x,y))$ küme merkezine (C_j^k) olan uzaklığını (renk farkını) ifade eder.

Formül 6.2, KKS'deki olasılıklı seçim kuralına (Formül 2.4) benzemekle birlikte seçme koşulunda bazı farklılıklar gösterir. Bu farklılık algoritmanın rastlantısallığını kontrol etmek için gerçekleştirilmiştir. Bu değişiklik sayesinde, algoritma ilk aşamalarında yani ilk iterasyonlarında rastlantısallığı arttırırken, son süreç ilerledikçe bu rastlantısallığı düşürmekte ve son iterasyonda tamamen ortadan kaldırmaktadır. Böylece algoritmanın ilk aşamalarda keşif gücü arttırılmış ve son aşamalarda odaklanma gücü arttırılmış olur. Bu sayede, algoritma yerel minimumlardan daha kolay kaçtığı gibi daha kararlı bir yapıya da sahip olmuştur. Ayrıca her iterasyon sonunda K-means algoritmasındaki küme merkezi güncelleştirme işlemi uygulanarak küme merkezlerinin daha kararlı bir hale

getirilmesi hedeflenmiştir. Buna göre, her iterasyon sonunda bir k karıncası kendi küme merkezleri C_j^k yi aşağıdaki formüle göre günceller;

$$C_i^k = \frac{1}{N_i^k} \sum_{j=1}^{N_i^k} p_j, i = 1, 2, 3 \quad 6.4$$

burada N_i^k o kümeye atanan toplam piksel sayısını p_j ise bir pikselin renk değerini ifade eder.

Yerel minimumlardan kaçabilmek ve çözümleri güçlendirmek için her iterasyon sonunda ayrıca IPSO algoritması kullanılmıştır. Bu algoritma, sırayla algoritmaya parçacık ekleyerek ilerlediğinden her iterasyon sonunda karıncanın ürettiği çözüm kümesi yeni bir parçacık ve onun ilk konumu olarak IPSO algoritmasına eklenir. İterasyonlar ilerledikçe IPSO algoritmasına yeni parçacıklar eklenecektir. Programın her iterasyonun sonunda IPSO tamamen bağımsız hareket ederek kendine ayrılmış belli bir iterasyon sürecinde parçacık pozisyonlarını en iyi konumlara (en iyi küme merkezlerine) doğru yönlendirmeye çalışır.

IPSO algoritması tamamlandığında, IPSO algoritmasının ürettiği en iyi sonuç programın bir iterasyonu sonunda elde edilen sonucu ifade eder ve bu sonuca göre pikseller üzerinde feromon depolanır. Böylece her iterasyon sonunda sadece o ana kadar üretilen en iyi çözüm için feromon güncelleştirmesine izin verilmiş olur. Feromon güncelleştirme işlemi diğer Karınca Kolonisi algoritmalarından farklı olarak tüm piksellerin üzerinde ve her bir piksel için üç farklı çeşitte bırakılarak yapılır. Her feromon çeşidi o pikselin çözüme ait üç küme merkezinin her birine olan uzaklığıyla orantılı olarak depolanır. Her bir piksel $p(x,y)$ için feromon güncelleştirme işlemi aşağıdaki formüle göre uygulanır:

$$\tau_i(x, y) = (1 - \rho) \cdot \tau_i(x, y) + \eta_i(x, y), \quad i = 1, 2, 3 \quad 6.5$$

Burada $\tau_i(x, y)$ piksel $p(x,y)$ üzerindeki i kümesine ait feromon seviyesini, ρ ise feromon buharlaşma katsayısını ifade eder.

Algoritma için belirlenen maksimum iterasyon sayısına ulaşıldığında elde edilen sonuca göre pikseller kümelenerek birbirinden ayrıştırılır ve resim homojen parçalara ayrılmış olur.

6.3 Deneysel Çalışmalar

IPSOAntK-means kümeleme yaklaşımı iki farklı çiçek veri kümesi üzerinde denenmiştir. Bu veri setleri CAVIAR-flower veri seti (Evans et al., 2005) ve Oxford-17 çiçek veri setidir (Nilsback and Zisserman, 2007). Her iki veri setinin farklı özellikleri olduğundan önerdiğimiz kesimleme yönteminin son basamağı olan istenilen nesnenin resimden çıkartılması basamağı her bir veri seti için farklı şekilde gerçekleştirilmiştir.

IPSOAntK-means algoritması ayrıca iyi ayarlanması gerekli olan birçok parametre içermektedir. Bu parametrelerin otomatik olarak uygun bir şekilde belirlenebilmesi için Iterative (Tekrarlamalı) F-Race algoritması (Balaprakash et al., 2007; Birattari et al., 2009) kullanılmıştır.

Uygulama hem Matlab hem de C++ ve OpenCV kullanılarak gerçekleştirilmiştir. Tüm testler Intel Core 2 Duo 2.53 GHz işlemcili ve 6 GB belleğe sahip bir dizüstü bilgisayar üzerinde gerçekleştirilmiştir.

Aşağıdaki alt bölümlerde önce Iterative F-Race ve parametre seçiminden bahsedilecek, sonra her iki veri setinden elde edilen sonuçlara yer verilecektir.

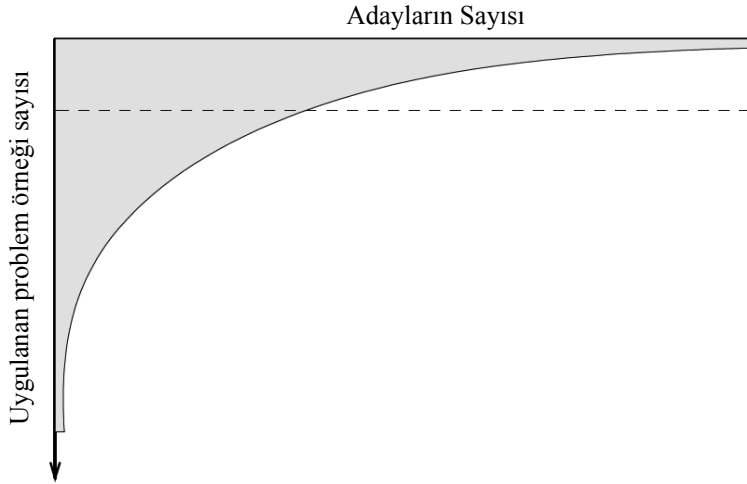
6.3.1 Iterative f-race algoritması

IPSOAntK-means algoritması birkaç tane parametreye sahiptir ve bu parametrelerin değerleri birçok sürü zekâsı yaklaşımında olduğu gibi algoritmanın davranışını büyük oranda etkilemektedir. Fakat bu parametrelerin doğru bir şekilde seçimi oldukça zordur. Bu tür algoritmaların parametre değerleri genellikle ya gelişigüzel bir şekilde seçilir (Hutter et al., 2009) veya birçok analiz yapıldıktan sonra doğru parametre değerlerine karar verilebilir. Dolayısıyla doğru parametre seçme işlemi zaman alıcı ve çaba gerektiren bir süreçtir.

Son zamanlarda, bu sorunun üstesinden gelebilmek için bazı otomatik parametre seçim algoritmaları önerilmiştir. Bu algoritmalarından en çok bilinenleri CALIBRA (Adenso-Diaz and Laguna, 2006), direk arama algoritması yaklaşımı (Audet and Orban, 2006), sıralı parametre optimizasyonu (sequential parameter optimization) algoritması (Bartz-Beielstein, 2006), Iterative F-Race (Balaprakash et al., 2007; Birattari et al., 2009) ve ParamILS (Hutter et al., 2009) algoritmasıdır.

Bu tezde, IPSOAntK-means algoritmasının parametre değerlerinin seçiminde son zamanlarda çokça kullanılan ve başarılı sonuçlar elde edilen Iterative F-Race algoritması kullanılmıştır. Iterative F-Race algoritması F-Race (Birattari, 2009; Birattari et al., 2002) algoritmasının geliştirilmiş bir şeklidir. F-Race algoritması özellikle nümerik (örn. popülasyon sayısı, buharlaştırma oranı vb.) ve kategorik (sürü ağ yapısı gibi) parametrelerin optimizasyonu için önerilmiş bir yaklaşımdır. En iyi parametre değerlerini aday değer setleri kümesinden rastlantısal işlemlerle seçmeye çalışır.

F-Race algoritmasında ilk aşamada rastgele belirlenen aday parametre değer kümeleri sırasıyla problem örnekleri üzerinde denenir. Birkaç örnek üzerinde tüm aday değerler denendikten sonra bu değerler hipotez testine tabi tutulur. İstatistiksel olarak diğer adaylardan farklı olan yani daha kötü çözümler ürettiği tespit edilen aday değerler yok edilir. Bu işlem sadece bir veya birbirine benzer birkaç aday değer kalana dek devam eder ve böylece algoritma için önerilen parametre değerleri tespit edilmiş olur. Problem süresince aday sayılarının nasıl değişebileceği Şekil 6.3’de gösterilmiştir. Görüldüğü gibi F-Race algoritması aday parametre değerlerini problem örnekleri üzerinde uygulayarak onları bir yarışa tabi tutar. Yarışı kazanan aday değerler, parametre değerleri olmayı da hak eder.



Şekil 6.3. Uygulanan problem örneğine göre aday sayısının durumu (Birattari et al., 2009).

F-Race algoritmasında aday değerler birbirinden bağımsız olarak üretilir. Iterative F-Race’de ise F-Race algoritması ile muhtemel iyi değer seçimi yaklaşımı birleştirilir. Her bir iterasyonda F-Race algoritması uygulanır ve adayların üretimi, seçimi ve arındırılması tekrarlamalı bir şekilde elde edilir.

Nümerik parametreler için, her bir iterasyondaki yeni adaylar kümesi bir önceki F-Race'den elde edilen sonucun etrafında ve belli bir standart sapmaya göre normal dağılım gösterecek şekilde üretilir. Böylece her iterasyonda F-Race en iyi adaylar etrafında yeni iyi adayları bulmaya ve arındırmaya çalışır. Kategorik parametrelerin aday değerleri için, en iyi adayların olasılığı yüksek olacak şekilde ayrık dağılıma göre bir örnekleme alınır. Iterative F-Race algoritmasının ayrıntıları (Balaprakash et al., 2007; Birattari et al., 2009) makalelerinde verilmiştir.

Iterative F-Race algoritması doğru parametrelerin seçimi için kullanılmakla birlikte kendisinin de bazı parametreleri vardır. Bu parametrelerin de uygun bir şekilde seçilmesi gereklidir. Neyse ki yapılan çalışmalarda Iterative F-Race parametreleri için uygun değerler saptanmıştır. Iterative F-Race algoritmasına ait parametreler, tanımları ve aldığı değerler Çizelge 6.1'de gösterilmiştir. Parametreler ile ilgili ayrıntılı bilgi yine (Balaprakash et al., 2007; Birattari et al., 2009) makalelerinden elde edilebilir.

Çizelge 6.1. Iterative F-Race algoritmasına ait parametreler, tanımları ve değerleri

Parametre	Formül ve Alacağı Değer	Tanım
L	$L = 2 + round(\log_2(d))$	L : iterasyon sayısı ve d : toplam parametre sayısı
B_l	$B_l = (B - B_{used}) / (L - l + 1)$	B : toplam bütçe ve $B = 10^4$, B_l : bir l iterasyonu için toplam bütçe, B_{used} : o ana kadar kullanılan bütçe
N_l	$N_l = \lfloor B_l / \mu_l \rfloor$	N_l : bir l iterasyonu için belirlenen toplam aday sayısı ve $\mu_l = 5 + l$

6.3.2 Uygun parametre değerlerinin seçilmesi

IPSOAntK-means algoritmasının piksel kümelemeye uygun parametrelerinin seçilmesinde Iterative F-Race algoritması kullanılmıştır. Böylece çok fazla analiz ve çaba harcanmadan iyi sonuçlar üretebilen bir parametre seti elde edilmiştir.

IPSOAntK-means parametre değerlerini Iterative F-Race algoritması ile elde edebilmek için hangi parametrelerin otomatik olarak ayarlanması gerektiğine ve bunların muhtemel değer aralıklarına karar verilmelidir. Çünkü Iterative F-Race algoritması hangi parametrelerin sabit tutulacağını, hangilerinin ayarlanması gerektiğini ve ayarlanmak istenen parametrelerin aralıklarını aday parametre değerlerini üretilmek için kullanıcıdan istemektedir. Buna göre ayarlanmak istenen ve sabit tutulan parametreler ile bunların aralıkları veya değerleri Çizelge 6.2’de gösterilmiştir.

Çizelge 6.2. IPSOAntK-Means parametreleri ve bunların aralık veya değerleri.

Ayarlanmak istenen parametreler	Aralıklar
w	[0,1]
ϕ_1	[0,4]
ϕ_2	[0,4]
β	[0,4]
q_0	[0,1]
ρ	[0,1]
Sabit tutulan parametreler	Değeri
Maksimum iterasyon sayısı	10
IPSO için maksimum iterasyon sayısı	100

Son olarak algoritmanın üzerinde çalıştırılacağı örnek problem kümesi seçilmelidir. Bunun için CAVIAR-flower veri setine ait rastgele seçilen 100 tane örnek resim seçilmiştir. Ayrıca Iterative F-Race'in istatistiksel sonuçları kesimleme algoritmasının başarımlar ölçütü kullanılarak elde edilmiştir.

Bu ayarlamalar gerçekleştirildikten sonra Iterative F-Race algoritması birbirinden bağımsız olarak 10 defa çalıştırılmış ve farklı 10 sonuç elde edilmiştir. Üretilen bu sonuçlardan ortalaması en iyi olan parametre seti IPSOAntK-means için seçilmiştir. Üretilen 10 farklı değer Çizelge 6.3'de gösterilmiştir. Buna göre, seçilen parametre değerleri $w = 0.6435$, $\phi_1 = 1.526$, $\phi_2 = 1.526$, $q_0 = 0.9185$, $\beta = 1.359$ ve $\rho = 0.703$ şeklindedir. Yapılan tüm deneysel testlerde bu parametre seti IPSOAntK-means için kullanılmıştır.

Çizelge 6.3. Iterative F-Race ile IPSOAntK-Means için seçilen parametre değerleri.

w	ϕ_1	ϕ_2	q_0	β	ρ	Ortalama
0.8531	1.279	1.481	0.8463	1.985	0.864	0.78
0.6535	1.526	1.526	0.9185	1.359	0.703	0.89
1.3656	0.986	1.465	0.871	1.095	0.789	0.77
1.1466	0.246	2.578	0.9427	2.125	0.656	0.73
0.9934	1.8426	1.686	0.945	1.408	0.775	0.81
0.419	0.598	0.89	0.805	2.789	0.736	0.71
0.6148	1.422	1.325	0.984	1.756	0.751	0.83
0.6497	2.565	2.186	0.875	2.215	0.816	0.77
0.924	1.667	1.921	0.881	1.649	0.691	0.82
0.6672	1.569	1.487	0.9248	1.883	0.807	0.87

6.3.3 CAVIAR-flower veri kümesi üzerinde test sonuçları

IPSOAntK-means kümeleme algoritması ile çiçekli resimlerdeki çiçek bölgelerinin kesimlenmesinde ilk kullandığımız veri seti CAVIAR-flower veri setidir. Bu veri setine <http://www.ecse.rpi.edu/doclab/flowers> adresinden ulaşılabilir. CAVIAR-flower veri seti içinde 113 farklı çiçek türüne ait 1078 tane farklı renkli çiçek resmi içermektedir. Her bir resim 320x240 piksel boyutunda ve arka planları gerçek görüntülerden oluşmakla birlikte çoğunlukla toprak, yapraklar, dallar ve diğer çiçeklerden oluşmaktadır. Resimler değişken aydınlanma kullanılarak ve kötü koşullar altında çekilmiştir. Dolayısıyla birçok resim keskin gölge parlaklık geçişlerine, yansıma etkilerine, koyuluğa veya bulanıklığa sahiptir. Ayrıca, resimlerde çiçekler genellikle sarı, kırmızı, beyaz ya da mavi renkte olup sabit bir dağılıma sahip değildir. Çiçekler resmin

merkezine gelecek şekilde konumlandırılmakla birlikte çoklu, üst üste binmiş veya oldukça ince bir yapıda olabilmektedir. Tüm bu koşullar, kesimlemenin başarısını oldukça güçleştirmektedir.

CAVIAR-flower veri seti her resim için olmasa da birçok resimde başarı oranını belirleyebilmek için elle üretilmiş doğrulama resimleri barındırmaktadır. Yapılan testler 60 farklı çeşit çiçek türünden alınan toplam 500 adet çiçek üzerinde gerçekleştirilmiştir. Alınan bu çiçek resimlerinin bazı önemli özellikleri şunlardır;

- Her çiçek merkezdedir ve hiç birisinde üst üste çiçek çakışması yoktur.
- Çiçeklerde farklı renkler bulunabilir.
- Her bir resmin başarımının ölçülebilmesi için elle çizilmiş doğrulama resmi bulunur.

Resimlerin birçoğunda aynı çiçekten birkaç tane bulunabilir. Fakat doğrulama resimleri sadece ortadaki ve en büyük boyuttaki çiçeğin sınırlarına göre üretilen resimlerdir. Bu yüzden, kesimleme algoritması tüm çiçekli bölgeleri tespit etse de merkezdeki ve en büyük olanı haricindeki tüm diğer çiçekli kısımları silmelidir. Bunu gerçekleştirebilmek için Fukuda et al. (2008) tarafından önerilen bir ölçüm faktörü kullanılmıştır. Bu ölçüm faktörü E sembolü ile ifade edilir ve şu şekilde hesaplanır;

$$E = \frac{\delta}{\sigma} \quad 6.6$$

Burada δ aday çiçek bölgesindeki toplam piksel sayısını ve σ bu piksellerin resmin merkezine göre dağılımlarını ifade eder ve şu şekilde tanımlanır;

$$\sigma = \frac{1}{\delta} \sum_{s \in S} d(s) \quad 6.7$$

Burada S aday bölgedeki piksellerin kümesini, $d(s)$ ise her bir pikselin merkeze olan Öklid uzaklığını ifade eder. Bu faktöre göre, en büyük E değerine sahip olan bölge merkeze en yakın olan ve en büyük olan bölgedir (Fukuda et al., 2008) ve sonuç olarak bu bölge seçilir ve diğerleri silinir. Buna göre üretilen bazı kesimleme sonuçları Şekil 6.4’de gösterilmiştir.

Başarı performansını ölçebilmek için elde edilen çiçek bölgesinin yer doğrulama resimleriyle ne kadar örtüştüğünü ifade eden bir hesaplama kullanılmıştır. Bu başarı ölçütü basit olarak şu şekilde ifade edilebilir:

$$Başarı = \frac{\text{doğru sınıflandırılmış piksellerin sayısı}}{\text{toplam piksel sayısı}} \quad 6.8$$

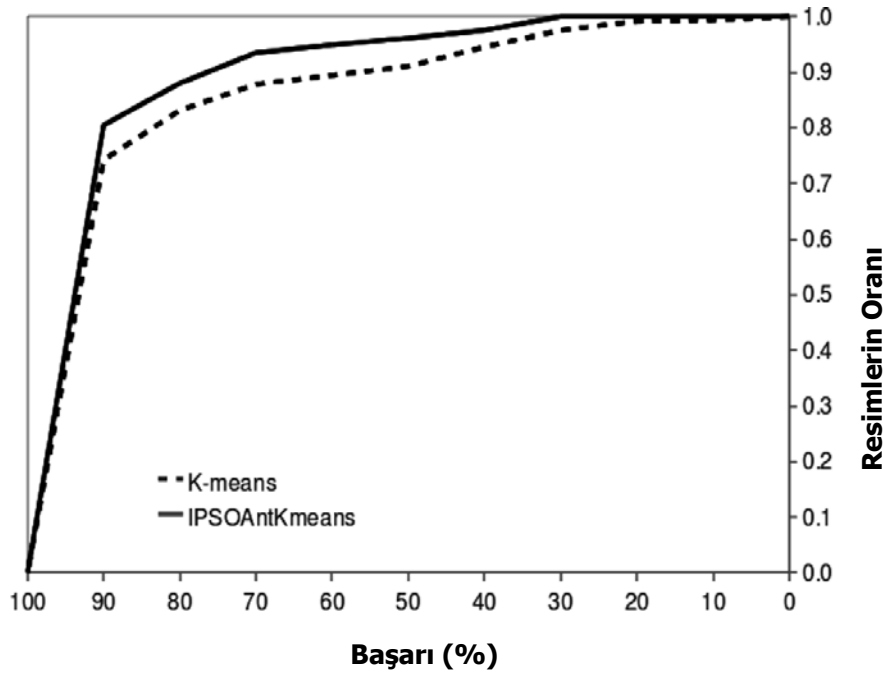


Şekil 6.4. CAVIAR-Flower veri kümesi üzerinde bazı sonuç örnekleri.

Bu başarı ölçütüne göre her resim üzerinde yapılan 10 farklı çalıştırma sonucunda tüm resimlerden elde edilen başarı %96,4 olarak, en başarılı

kesimleme yapılan çiçek türünde ortalama başarı değeri ise %99,3 olarak tespit edilmiştir.

Gerçekleştirilen algoritma küme merkezleri rastgele seçilen K-means algoritmasıyla karşılaştırılmıştır ve aynı koşullar altında K-means algoritmasının tüm resimler üzerindeki ortalama başarısı %92'dir. Daha nicel bir karşılaştırma yapmak için bir ROC eğrisi çizdirilmiştir. Bu eğrinin altında kalan alanın 100 olması durumu mükemmel bir kesilmeyi ifade etmektedir. Eğri üzerindeki her bir nokta, o noktadaki başarı değeri veya üstünde iyi sonuç veren resimlerin oranını ifade etmektedir. Buna göre elde edilen çizge aşağıda gösterilmiştir. Şekil 6.5'de görüldüğü gibi IPSOAntK-means, K-means algoritmasından daha iyi bir performans sergilemektedir.



Şekil 6.5. ROC eğrisi üzerinde IPSOAntK-means algoritması ile K-means algoritmalarının karşılaştırılması

6.3.4 Oxford-17 çiçek veri kümesi üzerinde test sonuçları

Algoritma ayrıca renkli çiçek resimlerini barındıran diğer bir çiçek veri kümesi üzerinde de test edilmiştir. Bu veri seti <http://www.robots.ox.ac.uk/~vgg/data/flowers/> adresinde bulunun Oxford-17 Flower veri kümesidir (Nilsback and Zisserman, 2007). Bu veri seti 17 farklı çiçek türüne ait resimleri içermekle birlikte veri seti sahipleri bunlardan 4 çeşide ait resimleri kesimlemeye uygun bulmamışlardır. Dolayısıyla kesimleme işlemi 14 farklı çiçek türüne ait 753 resim üzerinde gerçekleştirilebilmektedir. Ayrıca her

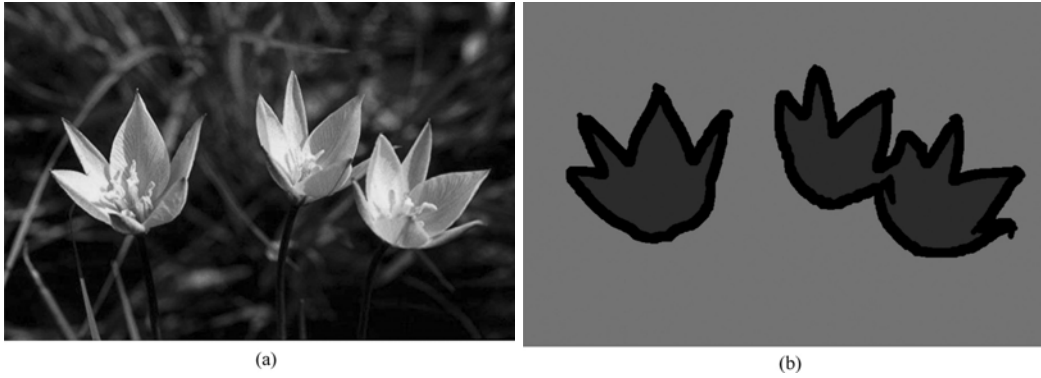
bir çiçek türüne ait yirmişer resim deneme kümesi, geri kalan 493 tane resim ise test kümesi olarak ayrılmıştır (Nilsback and Zisserman, 2010).

753 tane resimden her birine ait elle üretilmiş yer doğrulama resimleri trimap (üçlü harita) şeklinde hazırlanmıştır. Şekil 6.6'da bir resme ait trimap örneği gösterilmiştir. Trimap'lerde kırmızı renkle ifade edilen bölgeler çiçekli kısımları, yeşil renkli olanlar arka planı ve siyah renkteki pikseller ise etiketsiz alanları ifade eder. Etiketsiz alanlar çiçekli bölgeye çok yakın veya yakın renkli bölgeler olup performansa etki etmesi istenilmeyen pikseller kümesidir.

Bu veri setindeki resimler CAVIAR-Flower veri setinde olduğu gibi iyi koşullar altında çekilmeyen ve karmaşık arka plana sahip resimlerden oluşmaktadır. Fakat farklı olarak bu veri setinde çiçekli kısım resmin herhangi bir yerinde olduğu gibi birkaç tane de olabilir. Kesimleme algoritması bu seçenekte sadece bir çiçeğin değil tüm çiçek bölgelerinin kesimlemesini yapabilmelidir.

Bu veri seti için P adı verilen farklı bir başarı ölçütü tanımlanmıştır. P değeri sınıfa eşit olduğunda mükemmel kesimlemeyi ifade eder. Bu P değerinin matematiksel ifadesi aşağıdaki gibidir:

$$P = \frac{\text{doğru önplan} \cap \text{kesimlenen önplan}}{\text{doğru önplan} \cup \text{kesimlenen önplan}} \quad 6.6$$



Şekil 6.6. Oxford-17 veri kümesine ait bir resim ve ilgili trimap resmi

Yine gerekleřtirilen algoritma tm resimler zerinde test edilerek bu veri setini kullanan literatrdeki tm yntemlerle karřılařtırılmıřtır. Bu veri seti zerinde denenen yntemler Saitoh et al. (2004) ve Nilsback and Zisserman (2010) metotlarıdır.

753 resim zerinde yapılan karřılařtırmalar sonucunda, Saitoh et al. (2004) metodunun bařarısı %82, Nilsback and Zisserman (2010) metodunun %94 ve IPSOAntK-means algoritmasının bařarısı %86 olarak hesaplanmıřtır. Yani bařarım olarak IPSOAntK-means algoritması Saitoh et al. (2004) metodundan daha iyi olmasına karřın Nilsback and Zisserman (2010) metodundan daha ktdr. Nilsback and Zisserman (2010) metodu en iyi performansı gstermesine raėmen, ok karmařık kesimleme řeması kullanır ve gerekleřtirimi olduka zordur. Performans dřřnn diėer bir sebebi ise bazı iek trlerinin birine zıt renklerden oluřmasıdır. Byle durumlarda algoritma ieėin sadece bir blmn kesimleme yapabildiėinden bu performansta dřřlere neden olur.

7. KARINCA KOLONİSİ ALGORİTMASI İLE GENEL RENKLİ GÖRÜNTÜ KESİMLEME

Renklerin kümelenmesinde ortaya çıkan problemler vardır. Bunlardan en önemlileri renklerin kaç kümeye ayrılması gerektiği ve bu kümelerin merkezlerin optimum yerlerinin bulunmasıdır (Cheng et al., 2001).

Beşinci bölümde renkli resimlerin kümelenmesinde ortaya çıkan problemlerden birisi olan küme ilkleme problemine IPSOAntK-Means algoritması ile çözüm getirilmeye çalışılmış ve belirlenen üç kümenin merkez konumları optimum noktaları tespit edilmeye çalışılmıştır. Bu bölümde ise hem küme ilkleme problemine çözüm getirilmeye çalışan hem de renklerin kaç kümeye ayrılması gerektiğine karar veren geliştirdiğimiz bir Karınca Kolonisi Optimizasyonu algoritmasına yer verilmiştir.

Gerçekleştirilen bu yaklaşım üç temel aşamadan oluşmaktadır. Bunlar sırasıyla (1) renk uzayının ve aday küme merkezlerinin bulunması, (2) KKO algoritması ile renklerin kümelenmesi ve (3) yakın kümelerin birbirleriyle birleştirilmesidir. Bu yaklaşım ile genel bir kesimleme yaklaşımı geliştirilmek istenmiştir. Bu amaçla, gerçekleştirilen yaklaşım hem doğal resimlerin kesimlenmesinde hem de resim içindeki istenilen bir bölgenin kesimlenmesinde ayrı ayrı test edilerek performansı denenmiştir.

Aşağıdaki bölümlerde her bir aşamadan ayrıntılı olarak bahsedilecektir. Son olarak hem doğa resimlerini içeren veri seti üzerinde, hem de istenilen bölge kesimlemesi için seçtiğimiz çiçekli bölgelerin kesimlenmesinde elde edilen test sonuçlarına yer verilecektir.

7.1 Önerilen Kesimleme Yaklaşımı

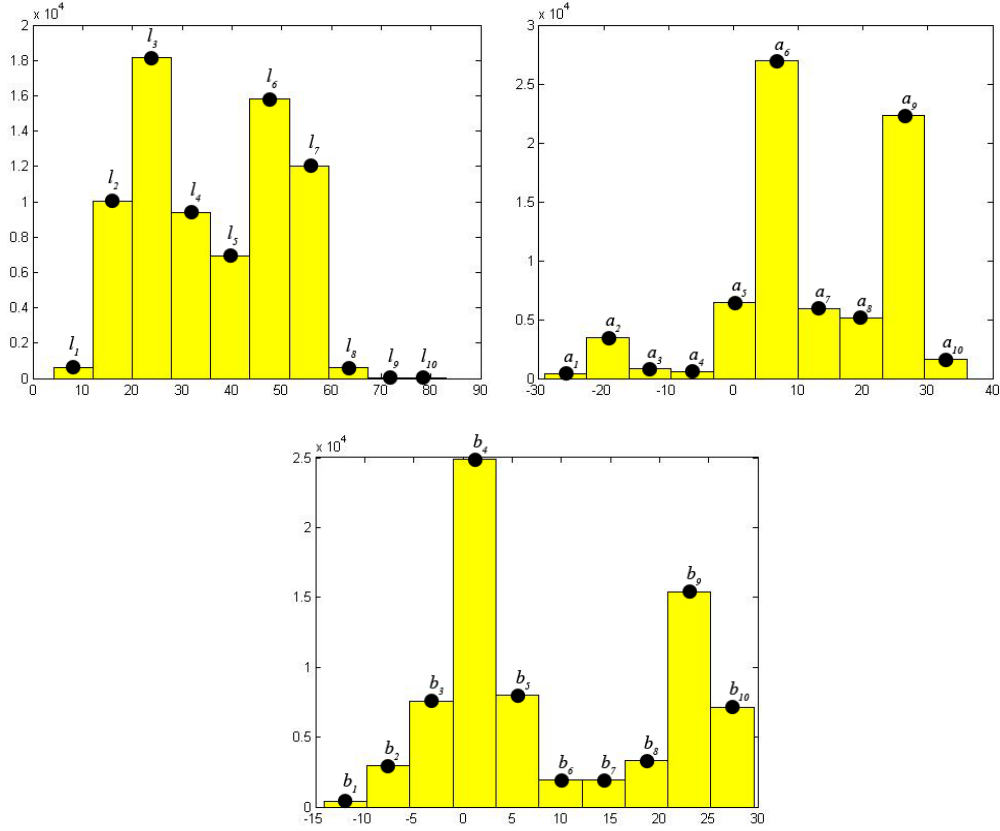
Önerdiğimiz kesimleme yaklaşımında öncelikle karınca kolonisinin üzerinde dolaşacağı çevre, yani arama uzayına karar verilmiştir. Bunun için renk uzayının tespit edilmesi ve aday küme merkezlerinin tespit edilmesi gerekmektedir. Daha sonra her bir aday küme merkezine ait sezgisel bir değerin atanması işlemi gerçekleştirilir. İkinci aşamada, karınca kolonisi algoritması ile karıncalar bu aday küme merkezleri üzerinde dolaşarak en iyi aday küme merkezlerini ve bunların sayısını tespit ederler. Bu seçim sonucunda elde edilen sonuçlar incelendiğinde, seçilen bazı küme merkezleri birbirlerine yakın konumda

olduklarından gerçekleştirilen bir birleştirme mekanizması ile bu küme merkezleri birleştirilerek sonuca ulaşılır. Son olarak elde edilen bu sonuç doğrultusunda renkler kümelenerek kesimleme gerçekleştirilmiş olur. Aşağıda önerilen bu yaklaşımın basamakları sırasıyla ayrıntılı olarak bahsedilmiştir.

7.1.1 Aday küme merkezlerinin ve sezgisel değerlerinin belirlenmesi

KKO algoritması ile kesimleme yapmadan önce karınca kolonisinin yaşayacağı, üzerinde çözümler üreteceği arama uzayına karar verilmelidir. Bu amaçla, öncelikle *RGB* renk uzayındaki resim *CIE L*a*b* uzayına çevrilmiştir. Bunun nedeni daha önce de bahsedildiği gibi, *CIE L*a*b* uzayında iki renk arasındaki ton farkının Öklid uzaklık formülüyle kolaylıkla bulunabilmesidir. Daha sonra, bu renk uzayındaki her bir bileşenin histogramları çıkartılmış ve bu histogramlardaki her bir çubuğun orta noktası tespit edilmiştir. l_i , a_i ve b_i her bir histogramın çubuğunun orta noktası olduğunda ortaya çıkan kümeler $L = \{l_1, l_2, l_3, \dots, l_n\}$, $a = \{a_1, a_2, a_3, \dots, a_n\}$ ve $b = \{b_1, b_2, b_3, \dots, b_n\}$ olacaktır. Aday küme merkezleri de bu üç kümenin bileşimi ile oluşturulan kümeden oluşmaktadır. Bu küme de $C = \{(l_1, a_1, b_1), (l_1, a_1, b_2), \dots, (l_1, a_2, b_1), (l_1, a_2, b_2), \dots, (l_n, a_n, b_{n-1}), (l_n, a_n, b_n)\}$ şeklinde tanımlanmıştır. Buna göre C , n^3 adet aday küme merkezinden oluştuğu görülmektedir.

Her bir i aday küme merkezine ait sezgisel fonksiyon, η_i , $\eta_i = \gamma_i / N$ olarak tanımlanmıştır. Burada γ_i bu küme merkezine yakın olan piksel sayısını, N ise kümedeki toplam piksel sayısını ifade eder. Böylece resimdeki piksellere yakın küme merkezlerinin değeri daha büyük olacaktır. Dolayısıyla sezgisel değeri daha büyük olan küme merkezinin seçilme olasılığı arttırılmış olur.

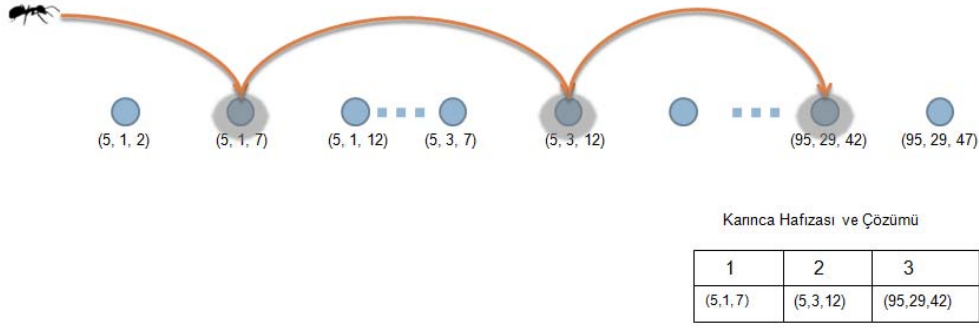


Şekil 7.1. Bir resme ait histogramlar ve her birindeki çubukların orta noktaları

7.1.2 Karınca kolonisi optimizasyonu ile en iyi küme merkezlerinin seçilmesi

Renklerin kümelenmesinde en önemli problemlerden ikisinin küme sayısının ve merkezlerinin tespit edilme zorluğu olduğundan bahsetmiştik. Aday küme merkezleri tanımlandığında KKO ile bu problemin üstesinden gelinebilir. Bu amaçla, KKO algoritması aşağıdaki şekilde tasarlanmıştır;

- Her karınca farklı uzunlukta hafızaya sahiptir. Yani her karınca hafıza uzunluğu kadar küme merkezi seçerek çözümünü oluşturur. Dolayısıyla her karıncanın üreteceği çözüm farklı sayıda kümeden olacaktır.
- Her iterasyon sonucunda en iyi sonucu bulan karıncanın çözümünü oluşturan aday küme merkezleri üzerinde feromon bırakılır. Böylece turlar atıldıkça iyi küme merkezleri üzerindeki feromon miktarı artacağından çözümler bu küme merkezleri etrafında aratılmış olacaktır (Şekil 7.2).



Şekil 7.2. Adım sayısı (hafıza büyüklüğü) üç olan bir karıncanın gerçekleştirdiği bir turun görünümü: tur sonunda karıncanın hafızası (elde ettiği çözüm) ve çözüm üzerinde bıraktığı feromon maddesi.

- Her karınca turunu atarken bir sonraki uğrayacağı küme merkezini Karınca Kolonisi Sistemi algoritmasının olasılıklı seçim kuralına göre (Bkz Formül 2.4) belirler. Fakat Formül 2.4'dekinin aksine her bir aday küme merkezinin olasılığı daha önce ziyaret edilen küme merkezlerinden bağımsız olarak tanımlanır. Yani sadece o aday küme merkezindeki feromon yoğunluğu ve sezgisel fonksiyon değeri olasılığı hesaplamak için yeterlidir.
- Her bir karınca çözümünü oluşturduğunda (turunu tamamlayıp hafızası dolduğunda) o çözüme ilişkin uygunluk fonksiyonu şu şekilde tanımlanmıştır.

$$f_k = \frac{\sum_{i=1}^N \sum_{m=1}^M \min(\text{dist}(p_i, c_m))}{N} + \frac{M \times \max(M) \times \ln(M)}{n} \quad 7.1$$

Burada M , karıncaya ait çözümün uzunluğunu (hafıza uzunluğunu), p_i resime ait her bir pikselin $CIE L^*a^*b$ uzayındaki değerini and c_m karıncanın çözümündeki her bir aday küme merkezin, n histogramdaki çubuk sayısını ve $\max(M)$ karıncaların alabileceği maksimum hafıza uzunluğunu ifade eder. Bu uygunluk fonksiyonu hem küme sayısının en iyi seçilme durumunu hem de seçilecek kümelerin merkezlerinin en iyi konumlarda olup olmadığını değerlendiren bir yapıdadır. Bu uygunluk fonksiyonuna göre en küçük değerleri alan çözümler daha iyi sonuçları ifade etmektedir.

- Her iterasyon sonunda, eğer daha iyi bir çözüm bulunursa, o ana kadarki en iyi çözüm güncellenir. Algoritma tamamlandığında bulunan en iyi karınca çözümü KKO algoritmasının sonucu olarak ifade edilir.

7.1.3 Kümelerin birleştirilmesi ile piksellerin kümelenmesi

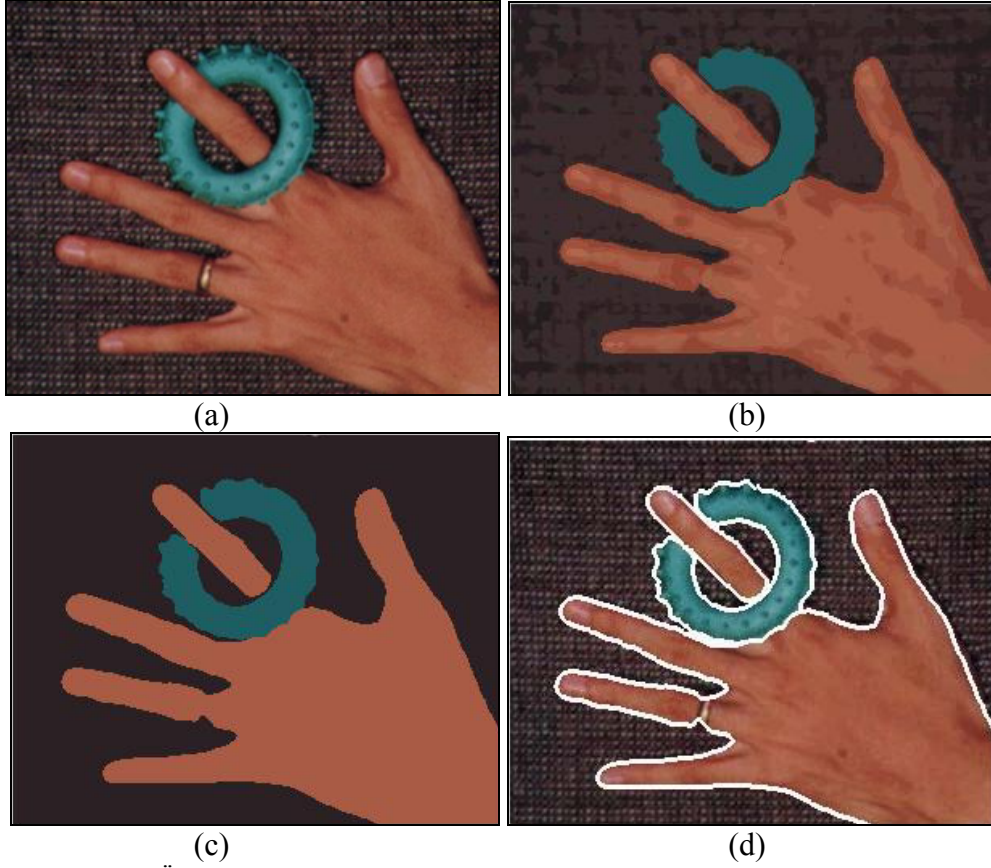
KKO algoritması ile en iyi kümeler belirlendiğinde, bazı küme merkezlerinin birbirine çok yakın değerlere sahip olduğu görülmüştür. Bu durum piksellerin kümelenmesi aşamasında birbirine çok yakın değerdeki piksellerin farklı kümelerde toplanmasına ve resmin istenmeyen küçük parçalara bölünmesine neden olabilir. Bu durumun ortadan kaldırılması için birbirine yakın kümelerin birleştirilmesi gereklidir. Bu amaçla, kümeler için resimdeki piksellerin dağılımının standart sapmasıyla ilişkili bir birleştirme ölçütü önerilmiştir. Eğer iki küme arasındaki fark resimdeki piksellerin standart sapmasından küçük olduğu durumda yeni küme merkezi aşağıdaki gibi belirlenmektedir;

$$C_{merged}(l, a, b) = \frac{\gamma_1 C_1(l, a, b) + \gamma_2 C_2(l, a, b)}{\gamma_1 + \gamma_2} \quad 7.2$$

Bu basit küme birleştirme işleminden sonra pikseller kendilerine en yakın kümelere atanarak kesimleme gerçekleştirilir. Şekil 7.3’de bir resim üzerinde tüm aşamaların sonuçları gösterilmiştir.

7.2 Deneysel Çalışmalar

Gerçekleştirilen KKS algoritması renkleri kümelemek için tasarlandığından genel bir kesimleme algoritması olarak kullanılabilir. Bunun için gerçekleştirilen yaklaşım iki farklı veri seti üzerinde denenerek sonuçlar elde edilmeye çalışılmıştır. Bunlardan bir tanesi yine renkli çiçek resimlerini içeren Oxford-17 veri kümesidir. Diğer doğadan çekilen fotoğraflarla oluşturulmuş Berkeley veri kümesidir. Tüm deneyler Matlab ortamında ve $\alpha = 1$, $\beta = 2$, $\rho = 0.15$, *karınca sayısı* = 5, *iterasyon sayısı* = 20, *n (histogramdaki çubuk sayısı)* = 15, ve $\max(K) = 15$ parametre değerleri için gerçekleştirilmiştir.



Şekil 7.3. Örnek bir resim üzerinde KKO yaklaşımıyla elde edilen kesimleme sonuçları: (a) orijinal resim (b) KKO ile kümeleme sonucunda elde edilen kümeleme, (c) küme birleştirme sonucunda elde edilen kesimleme, (d) kesimleme sonucunda oluşan bölgelerin hatları.

7.2.1 Oxford-17 çiçek veri kümesi üzerinde test sonuçları

Oxford-17 veri kümesi 14 farklı çiçek türünün resimlerini içeren bir veri kümesi olduğu ve özellikleri bölüm 6.3.4’de ayrıntılı bir şekilde anlatılmıştır. Bu veri kümesi için kullanılan performans ölçütü P şeklinde ifade edilir ve bu ölçüte ait hesaplama formülü Formül 6.6’da gösterilmiştir.

Gerçekleştirilen yaklaşım bu veri kümesindeki her resim üzerinde test edilmiş ve P değerleri bulunmuştur. Ayrıca literatürde bu veri kümesini kullanan iki farklı çiçek kesimleme algoritmasıyla karşılaştırılmıştır. Yapılan testler sonucunda tüm resimler için elde edilen ortalama P değeri 0.88 olarak tespit edilmiştir. Karşılaştırma sonucunda ortalama P değeri 0.82 Saitoh et al. (2004) metodundan daha iyi sonuç vermesine karşın ortalama P değeri 0.94 olan Nilsback ve Zisserman (2010) metodundan daha kötü sonuç vermektedir. Fakat tüm resimler için ortalama yerine ortanca P değeri alındığında ortanca P değeri 0.98’e ulaşmaktadır. Bu da en az 350 farklı resim için kesimleme algoritmasının mükemmel sonuç verdiği anlamına gelmektedir. Sonuçlar ayrıntılı bir şekilde

incelendiğinde elliye yakın resimde arka plan rengi çiçeğe çok benzediğinden veya çiçek birbirinden çok farklı renklerden oluştuğundan çok kötü sonuçlar verdiği ($P < 0,5$) tespit edilmiştir. Bu durum, ortalama P değerinin düşmesindeki en önemli nedendir. Şekil 7.4’de gerçekleştirilen kesimlemeye ait bazı elde edilen iyi sonuçlar gösterilmektedir.



Şekil 7.4. Oxford-17 çiçek veri kümesi üzerinde bazı test sonuçları: Birinci kolon: orijinal resim, ikinci kolon: trimap (yer doğrulama) resimleri, üçüncü kolon: küme birleştirme sonucunda elde edilen kesimleme sonuçları, dördüncü kolon: kesimlenmiş çiçek bölgesi resimleri.

7.2.2 Berkeley veri kümesi üzerinde test sonuçları

Berkeley veri kümesi 300 tane farklı doğal ortamda çekilen resimlerden oluşan bir veri kümesidir. Bu resimlerden her birine ait ayrıca elle çizilmiş nesnelerin sınırlarını belirleyen yer doğrulama resimleri de mevcuttur. Diğer yer doğrulama resimlerinden farklı olarak her bir resim yaklaşık 20 farklı gözlemci tarafından elde edilen sonuçların üst üste çalıştırılması sonucu elde edilen gri

tonajlı resimler halinde oluşturulmuştur. Bu yer doğrulama resimleri kesimleme algoritmasının performansını ölçmede aracı olarak kullanılırlar. Veri setine ait 300 resimden 200'ü deneme amaçlı diğer 100 tanesi test amaçlı olmak üzere iki parçaya ayrılmıştır. İlgili veri kümesine ait resimler ve yer doğrulama resimleri <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/> adresinden elde edilebilir.

Veri kümesini oluşturan araştırmacılar kesimleme algoritmasının performansını belirleyebilmek için iki farklı ölçütün kullanılmasını önermişlerdir. Bunlar, Precision-Recall eğrisi ve F-ölçütüdür (Martin et al., 2004). Precision-Recall eğrisi precision ve recall adı verilen iki farklı ölçütten alınan değerlerden elde edilen noktalardan oluşan bir eğridir. Precision, algoritmanın ürettiği sınır pikselinin doğru bir sınır pikseli olma olasılığını, recall ise doğru bir sınır pikselinin tespit edilme olasılığını ifade eder. F-ölçütü ise precision ve recall değerlerinin harmonik ortalamasıdır.

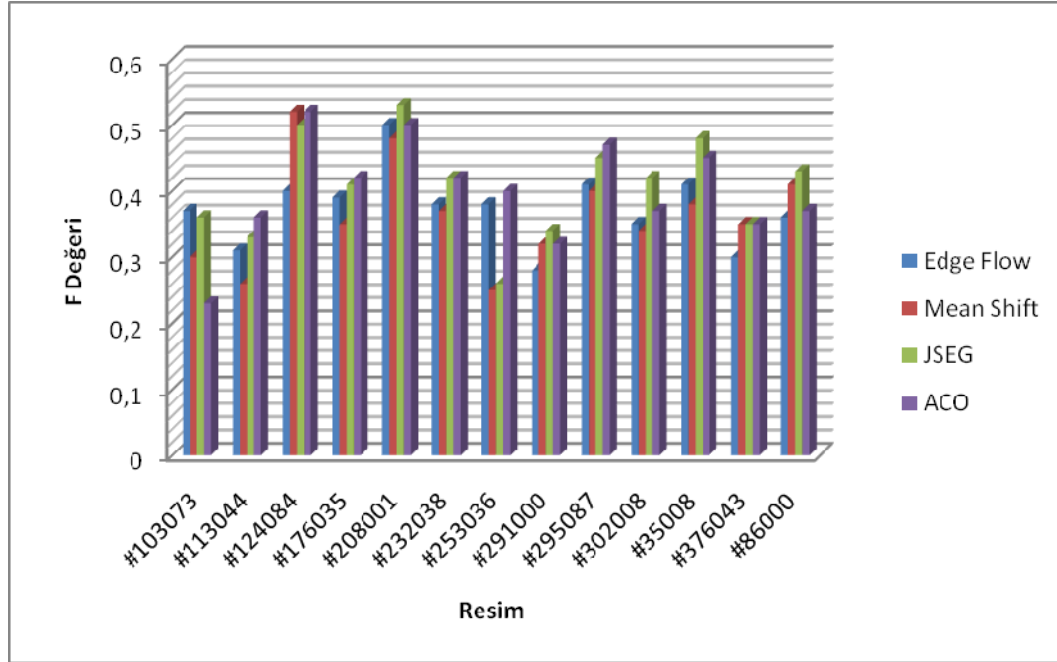
Gerçekleştirilen yöntem Berkeley veri kümesi üzerinde test edilmiş ve üç farklı yöntemle karşılaştırılmıştır. Bunlar, Kenar Takip (Edge Flow) (Ma and Manjunath, 1997), Ortalama Kaydırma (Mean Shift) (Comaniciu and Meer, 2002) ve Deng ve Manjunath (2001) tarafından gerçekleştirilen JSEG yöntemidir. Testler 13 farklı Berkeley veri kümesine ait resim üzerinde denenmiş ve F-ölçütü değerleri ve Precision-Recall eğrileri elde edilmiştir.

Çizelge 7.1 ve Şekil 7.5' de F-ölçütü üzerinden elde edilen karşılaştırma sonuçları gösterilmiştir. Çizelge 7.1' de görüldüğü gibi KKO (ACO) yöntemi ile JSEG yöntemi 13 resmin 7 tanesinde en iyi sonucu vermiştir. Edge Flow ve Mean Shift algoritmaları ise sadece bir resim için diğerlerinden daha iyi sonuç verebilmektedir.

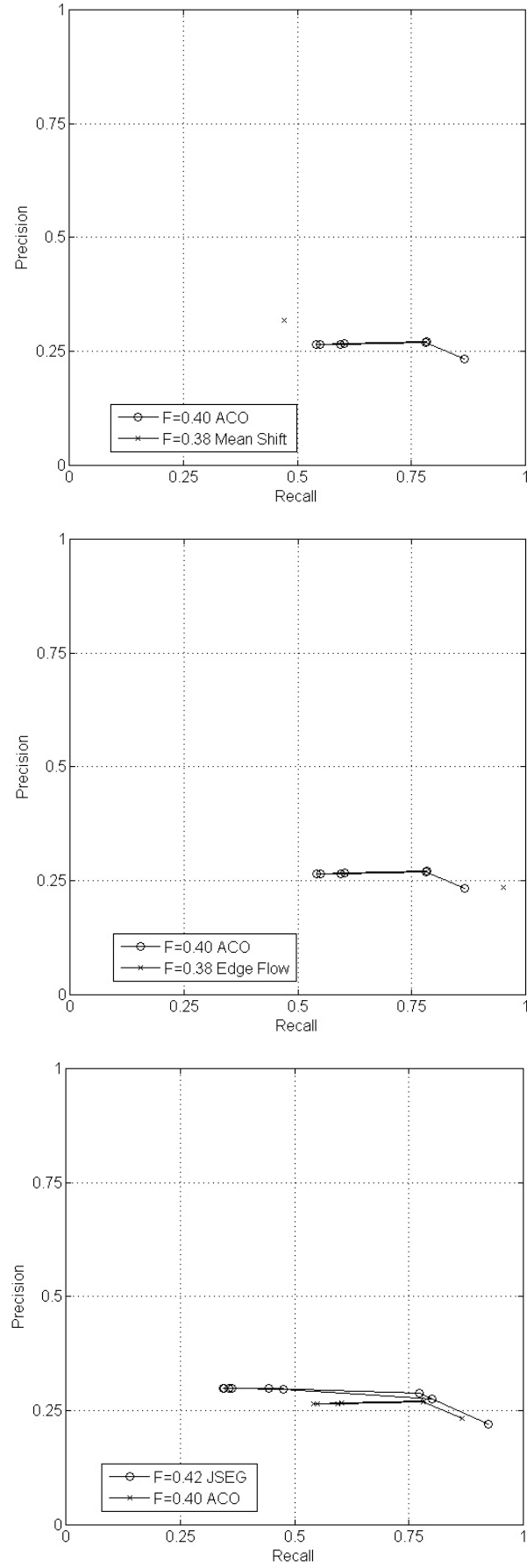
Şekil 7.6' de de bu 13 resmin ortalama değerlerinden elde edilmiş Precision-Recall eğrileri gösterilmiştir. Şekilde görüldüğü gibi KKO yaklaşımı Edge Flow ve Mean Shift algoritmalarından daha iyi sonuç verebilmektedir. JSEG algoritması ile karşılaştırıldığında oldukça yakın sonuçlar vermesine rağmen 0.02 oranında daha kötü sonuçlar vermektedir. Şekil 7.7 ve 7.8'de #124084, #113044, #232038 ve #295087 resimlerine ilişkin algoritmaların belirlediği kenarlar ve kesimleme sonuçları gösterilmiştir. Görüldüğü gibi KKO yaklaşımı nesnelerin sınırlarını daha başarılı bir şekilde belirleyebilmektedir.

Çizelge 7.1. Berkeley veri kümesi üzerinde karşılaştırma sonuçları: F-Ölçütü.

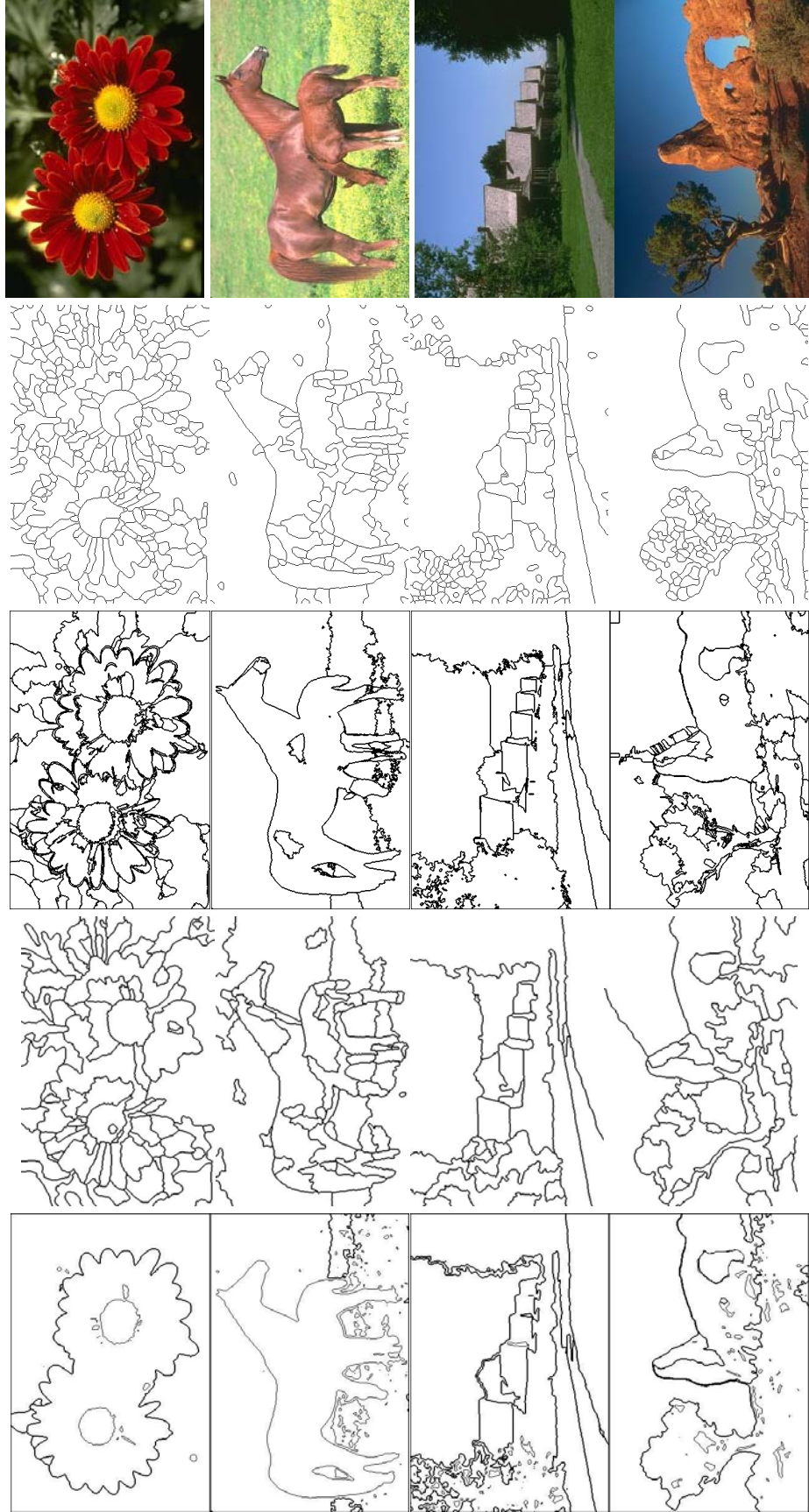
<u>Resim</u>	<u>Edge Flow</u>	<u>Mean Shift</u>	<u>JSEG</u>	<u>ACO</u>
#103073	0.37	0.30	0.36	0.23
#113044	0.31	0.26	0.33	0.36
#124084	0.40	0.52	0.50	0.52
#176035	0.39	0.35	0.41	0.42
#208001	0.50	0.48	0.53	0.50
#232038	0.38	0.37	0.42	0.42
#253036	0.38	0.25	0.26	0.4
#291000	0.28	0.32	0.34	0.32
#295087	0.41	0.40	0.45	0.47
#302008	0.35	0.34	0.42	0.37
#35008	0.41	0.38	0.48	0.45
#376043	0.30	0.35	0.35	0.35
#86000	0.36	0.41	0.43	0.37



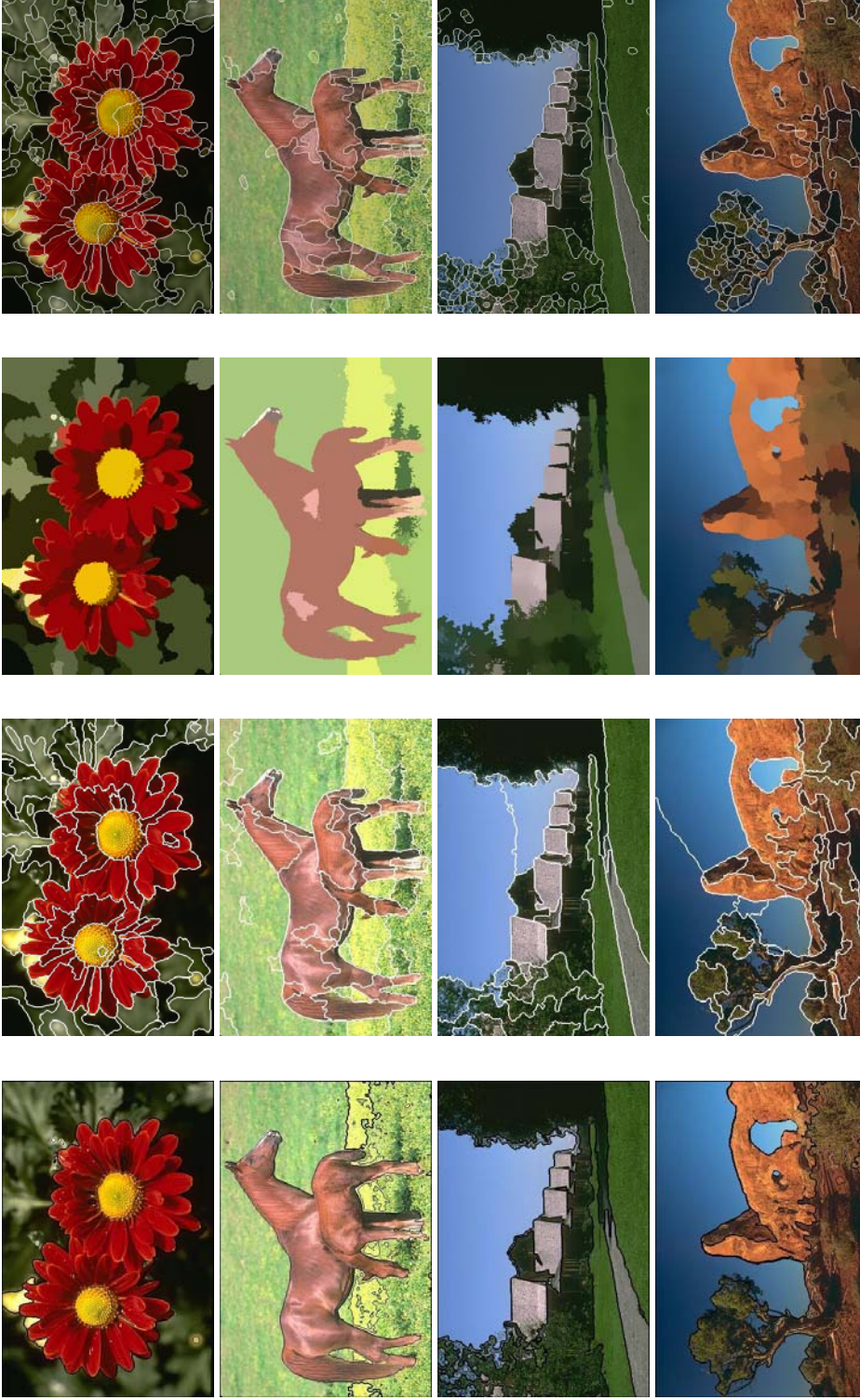
Şekil 7.5. Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları – F-Ölçütü



Şekil 7.6. Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları – Precision-Recall eğrileri



Şekil 7.7. Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları – Bulunan kenarlar:
 (1. satır) orijinal resim, (2. satır) Edge flow, (3. satır) Mean Shift, (4. satır) JSEG,
 (5. satır) gerçekleştirilen ACO algoritması



Şekil 7.8. Berkeley veri kümesi üzerinde bazı karşılaştırma sonuçları - Kesimlenen Bölgeler: (1. satır) orijinal resim, (2. satır) Edge flow, (3. satır) Mean Shift, (4. satır) JSEG, (5. satır) gerçekleştirilen ACO algoritması

8. ÇİÇEK TANIMA SİSTEMİ

Yeryüzünde ismi belli olan 250000'den fazla çiçek türü bulunmaktadır ve bitki türlerinin birçoğu sahip olduğu çiçeklerle ayırt edilirler. Botanikçiler için çiçek türlerinin tespiti oldukça zaman alıcı ve hataya açıktır. Uzman olmayan insanların bir çoğu çiçek türleri ve isimleri hakkında bir bilgiye sahip değildir ve çiçek türleri hakkında bilgileri ancak kitaplardan veya internet yardımıyla elde edinebilirler (Hsu et al., in press). Dolayısıyla çiçek türlerini otomatik olarak tanıyabilecek güçlü sistemlere ihtiyaç vardır.

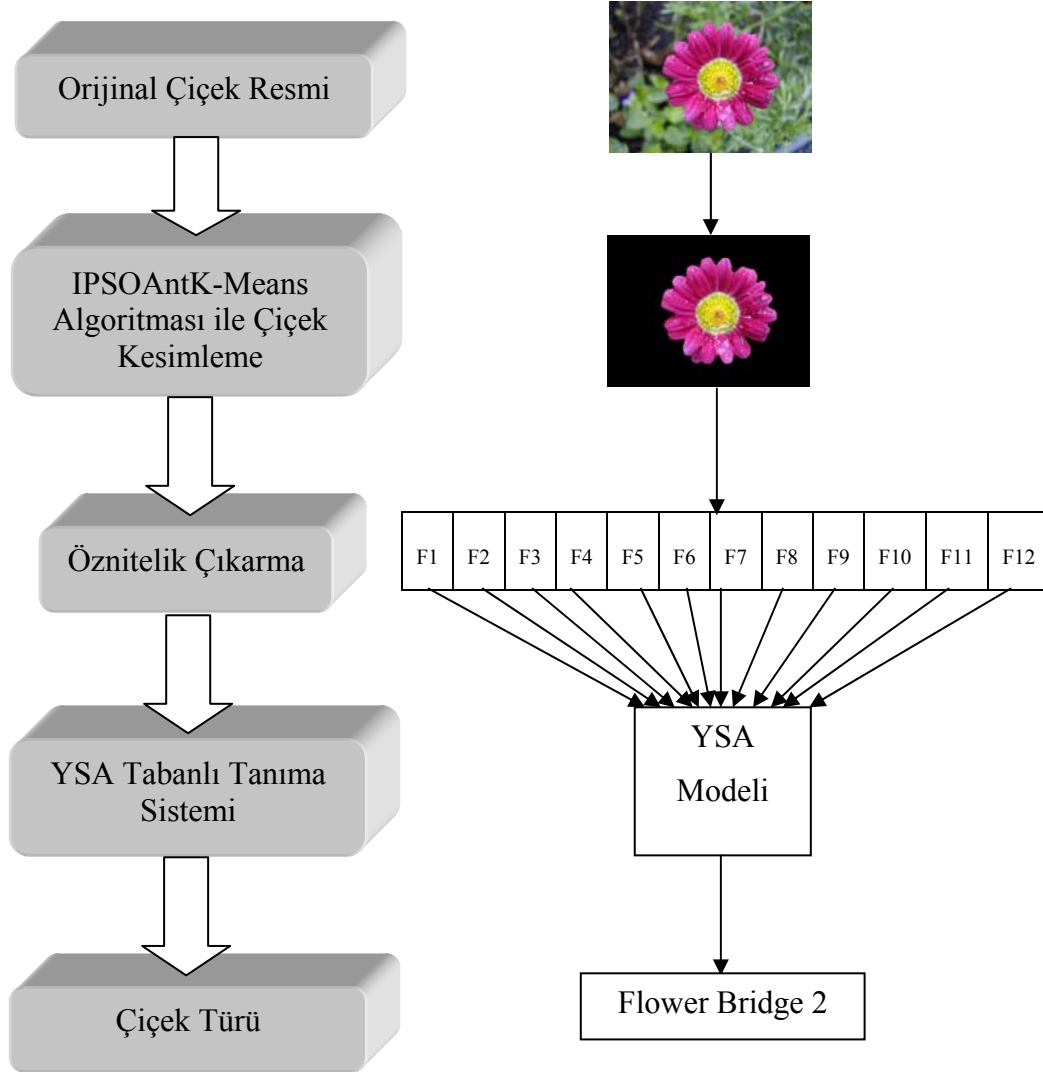
Son yıllarda, görüntü işleme tekniklere dayalı çeşitli çiçek tanıma sistemi geliştirilmiştir. Bu sistemlerden en çok bilineni bilgisayar destekli görsel etkileşimli tanıma sistemidir (CAVIAR; Zou and Nagy, 2004; Evans et al., 2005). Bu sistem model-tabanlı ve etkileşimli bir kesimleme yöntemi ile çiçek bölgesini belirlemekte, şekle ait öznitelikler ile ton ve doygunluğu ifade eden histogram özniteliklerini kullanarak yakın komşu sınıflandırıcısı kullanarak resimleri tanıyabilmektedir. Ayrıca mobil araçlara uyum desteği de sunmaktadır. Das et al. (1999) içerik tabanlı bir çiçek sorgulama sistemini sadece renk dağılımlarından yararlanarak gerçekleştirmektedir. Bu çalışma, çiçeklere ait renkler en uygun bir şekilde tespit edilip renkler indekslendiğinde başarılı sonuçların elde edilebileceğini ortaya koymuştur. Zhenjiang et al. (2006) sadece gül çeşitleri üzerinde ayırım ve sorgulama yapabilen nesneye-dayalı bir sistem geliştirmiştir. Bunların dışında literatürde sunulan az da olsa başka çiçek tanıma sistemleri de mevcuttur. (An-xiang et al., 2004, Fukuda et al., 2008; Nilsback and Zisserman, 2008; Saitoh and Kaneko, 2000; Saitoh et al., 2004; Hsu et al., in press).

Bu bölümde gerçekleştirmiş olduğumuz bir tanıma sisteminden bahsedilecektir. Gerçekleştirilen tanıma sistemi tamamen otomatik olarak işlemektedir. Sistem, sürü tabanlı kesimleme yapan IPSOAntK-mean algoritmasını otomatik çiçek kesimlemede, Yapay Sinir Ağlarını (YSA) ise çiçek tanıma bileşeninde kullanmaktadır.

8.1 Sistemin Yapısı

Gerçekleştirilen çiçek tanıma sistemi temel olarak üç adımdan oluşmaktadır. Birinci adımda sistemin en kritik bileşeni olan kesimleme işlemi gerçekleştirilir ve bu aşamada çiçekli bölgelerin arkapandan ayıklanması sağlanır. Bu kesimleme işlemi için IPSOAntK-means algoritması kullanılmıştır. İkinci aşamada kesimlenen çiçekler için öznitelikler elde edilir. Bu öznitelikler çiçeğin şekline ve

rengine ait bilgileri içermektedir. Üçüncü aşamada gerçekleştirilmiş olan YSA tabanlı tanıma sistemi bu öz nitelikleri kullanarak tanıma işlemini gerçekleştirmektedir.



Şekil 8.1. Çiçek tanıma sisteminin yapısı (Solda), tanıma sisteminin bir örnek resim için işleyişi (Sağda)

Sistemin adımları Şekil 8-1 de gösterilmiştir. IPSOAntK-Means algoritması ile ilgili ayrıntılar Bölüm 6’ da bahsedildiğinden burada sistemin geri kalan iki aşaması aşağıdaki bölümde ayrıntılı olarak bahsedilecek ve gerçekleştirilen tanıma sistemi üzerinde yapılan testlere yer verilecektir.

8.1.1 Çiçeğe ait öz nitelikler (öz nitelik vektörü)

Çiçek tanıma işleminin başarılı bir şekilde yapılabilmesi için doğru kesimleme yapıldıktan sonra çiçeğe ait öz niteliklerin çiçeği en iyi ifade edebilecek

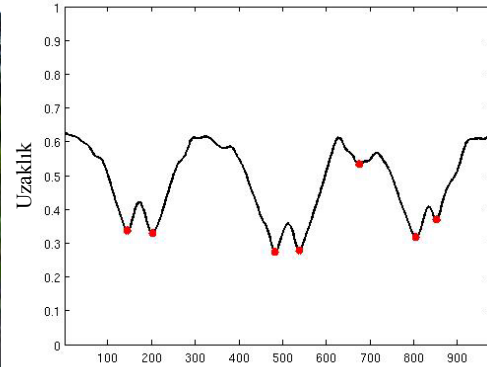
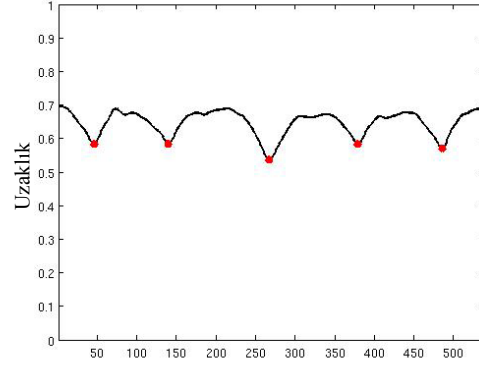
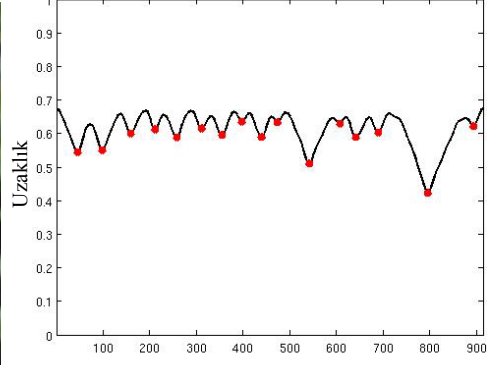
ve diğer çiçeklerden ayırt edebilecek özelliklerin elde edilmesi gerekmektedir. Bu tanıma sistemi için her bir çiçeğe ait şekilsel ve renksel öznitelikler çıkartılmaya çalışılmıştır. Toplam olarak 8 tane öznitelik elde edilmiştir. Bunlardan 6 tanesi şekilsel özellikleri ifade etmekte diğer 2 öznitelik ise çiçeğe ait renk bilgisinden elde edilmiştir. Renkler 3 boyutta ifade edildiğinden renklere ait öznitelikler de 3 değerle ifade edilmektedir. Dolayısıyla 12 tane öznitelik değeri birer değer olarak YSA tanıma algoritmasına girdi olarak verilmektedir. Elde edilen bu öznitelik değerleri aşağıda ifade edilmiştir.

8.1.1.1 Şekilsel Öznitelikler

Kesimleme algoritması ile çiçek arkaplandan ayrıldıktan sonra çiçeğe ait hatlar ve şekil bilgileri elde edilebilmektedir. Çiçeğin morfolojik yapısı (daireliliği, yaprak sayısı, yaprakların büyüklüğü vb.) çiçeği diğer çiçek türlerinden ayıran en önemli özniteliktir. Çiçeğin morfolojik yapısına uygun elde edilen öznitelikler şu şekildedir:

- ***F1 – Dairesellik ölçütü:*** Bu ölçüt $F1=4\pi A/L^2$ formülüyle ifade edilir. Burada A çiçeğin kapladığı alanı L ise çevre uzunluğunu ifade eder (Saitoh et al., 2004). Adından da anlaşıldığı gibi çiçeğin ne kadar dairesel bir yapıya sahip olduğunun bir ölçütüdür ve çiçek dairesel yapıya benzedikçe değer 1'e yaklaşır.
- ***F2-F3 – Yaprakların ortalama genişliği ve standart sapması:*** Bir çiçeği diğer çiçeklerden ayırt eden özniteliklerden birisi de yapraklarının yapısıdır. Bu özneliğin elde edilmesi için öncelikle çiçeğin merkez noktasına en uzak hattan saat yönünün tersine olacak şekilde tüm hattın merkeze olan uzaklıkları hesaplanmış ve bir uzunluk hattı elde edilmiştir. Bu hattaki iniş çıkışlar yaprakların bitiş başlangıç noktalarını ifade etmektedirler. Fakat elde edilen hatlarda çokça iniş çıkışlar bulunduğundan (çiçeğin doğası gereği tırtıklı bir yapıda bulunmasından dolayı) öncelikle elde edilen uzunluk hattı Fourier dönüşümüyle yumuşatılmış ve prüzler ortadan kaldırılmıştır. Daha sonra iniş çıkışların bulunduğu noktalar tespit edilmiştir (Şekil 8.2). Bu noktaların birbirlerine olan Öklid uzaklıkları yaprakların genişliklerini vermektedir. Bu değerler resmin köşegen uzunluğunun yarısına bölünerek normalleştirilmiştir. Daha sonra tüm yaprak genişlikleri bulunarak bir dağılım elde edilmiştir. Bu dağılımın ortalaması (F2) ve standart sapması (F3) birer öznitelik olarak

ıkartılmıřtır. Standart sapma deęeri yaprakların řekillerinin birbirine benzer mi yoksa farklı yapıda mı olduęunu ifade etmektedir (örn. řekil 8.2'deki 3. resim).



řekil 8.2. ieklere ait merkezden uzaklık hatları (normalleřtirilmiř) ve yaprak bařlangı-bitiř noktaları (kırmızı noktalar)

- **F4 – Normalleřtirilmiř yaprak sayısı:** Yapraklara ait bařlangı ve bitiř noktaları belirlendikten sonra ieęin ka yapraęa sahip olduęu da kolaylıkla bilinebilmektedir. Toplam yaprak sayısı elde edildikten sonra

maksimum yaprak sayısı olarak belirlenen 20 yaprak sayısına elde edilen değer bölünerek normalleştirilmiş yaprak sayısı özniteliği elde edilmiştir. Yaprak sayısı 20 den fazla olan çiçeklerde normalleştirilmiş değer 1 olarak alınmıştır.

- ***F5-F6 – Merkezden uzaklıkların ortalama ve standart sapması:*** Çiçeğin şekli hakkında ipucu verebilecek diğer öznitelik bilgileri ise çiçek hattındaki her bir noktanın merkeze uzaklık bilgilerinin oluşturduğu histogramadan elde edilen ortalama ve standart sapma bilgileridir.

8.1.1.2 Renksel Öznitelikler

Çiçeklerin renkleri de çiçekleri ayırd eden diğer bir özniteliktir. Bu çalışmada çiçekli bölgedeki tüm piksellere ait renklerin her bir kanaldaki histogramları elde edilerek renk ortalamaları (kırmızı – F7, yeşil – F8, mavi – F9) ve standart sapmaları (kırmızı – F10, yeşil – F11, mavi – F12) bulunmuştur ve bu değerler öznitelik olarak çıkarılmıştır. Renk ortalaması çiçeğin genel rengi hakkında bilgi verirken standart sapması çiçeğin kaç farklı renkten ve hangi uzaydaki değerlerin değişerek farklı renklerden oluşmuş bir çiçek olduğu bilgisini vermektedir.

8.1.2 Kullanılan YSA modelleri

Yapay Sinir Ağları birçok tanıma sisteminde çokça kullanılan yöntemlerin başında gelmektedir. Bu çalışmada da bazı YSA modelleri üzerinde tanıma sistemi gerçekleştirilerek sistemin performansı ölçülmüştür. Çalışmada kullanılan YSA modelleri; geri yayımlı çok-katmanlı sinir ağları, olasılıklı sinir ağı (PNN) ile radyal tabanlı sinir ağı sınıflayıcısıdır (RBNN) (Wasserman, 1993). Üç farklı geri yayımlı çok-katmanlı sinir ağı modeli olarak ileri-beslemeli (FFNN), kademeli-beslemeli (CFNN), Elman YSA (ENN) modelleri kullanılmıştır. Tüm bu modeller Matlab’ ta YSA aracı içinde hazır olarak bulunmaktadır.

Elde edilen öznitelik bilgileri YSA modelleri için girdi verilerini oluşturduğundan YSA modelleri için girdi katmanı 12 girişlidir. Çıktı katmanında ise tanıma sisteminde belirtilen çiçek türü adedince çıktı nöronu bulunacaktır. Çok katmanlı YSA modelleri için 60 nörondan oluşan bir ara katman koyulmuş, transfer fonksiyonu olarak ara katmanlarda “tansig”, çıkış katmanında “logsig”

kullanılmış ve öğrenme algoritması olarak “learnidx” (Gradient Decent Learning with Adaptive Learning Rate) algoritması tercih edilmiştir.

8.2 DeneySEL SonuÇlar

Gerçekleştirilen tanıma sistemi farklı YSA modelleri kullanılarak gerçekleştirilmiş ve bu modellerin başarıları karşılaştırılmıştır. Deneylerde önce CAVIAR-Flower veri setine ait 35 farklı çiçek türü üzerinde deneyler yapılmış daha sonra internet üzerinden rastgele seçilmiş 4 farklı çiçek türü de eklenerek toplam 39 farklı çiçek türü için testler gerçekleştirilmiştir. Her bir çiçek türü için 5 resim eğitim aşaması için 2 resim de test aşaması için ayrılmıştır. Kullanılan çiçek türlerine ait bazı resimler Şekil 8.3’ de gösterilmiştir. Türler incelendiğinde birçok türün şekil ve renk olarak benzerlik gösterdiği kolayca görülebilmektedir. Bu durum tanıma sistemin başarısını olumsuz şekilde etkilemektedir.



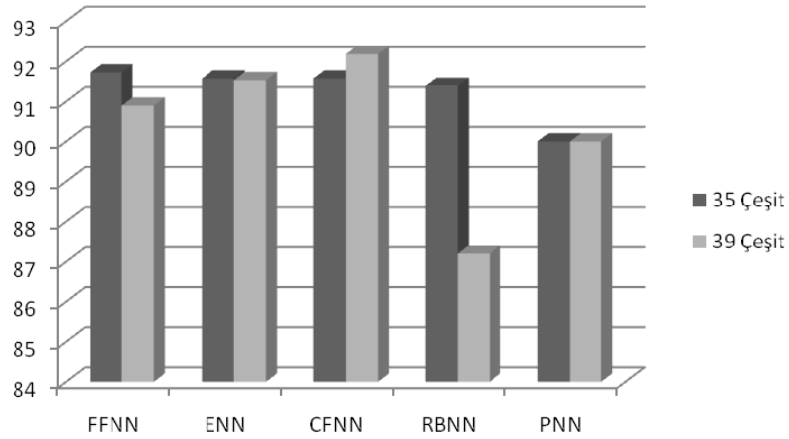
Şekil 8.3. Tanıma sisteminde test edilen bazı çiçek türlerine ait örnek resimler

Tüm sistem ve deneyler Matlab ortamında kodlanmıştır. Her bir yapay sinir ağı modeli için testler onar defa tekrar edilerek sonuçlar elde edilmiştir. Çizelge 8.1’de 35 ve 39 farklı çiçek türü için elde edilmiş olan başarı yüzdeleri görülmektedir.

Çizelge 8.1. Karşılaştırma Sonuçları: Başarı yüzdeleri

Test İndeksi	FFNN		ENN		CFNN		RBNN		PNN	
	35 çeşit	39 çeşit	35 çeşit	39 çeşit	35 çeşit	39 çeşit	35 çeşit	39 çeşit	35 çeşit	39 çeşit
1	97,1	91,03	92,9	91,03	91,4	91,03	91,4	87,2	90	90
2	88,6	91,03	92,9	91,03	94,3	92,31	91,4	87,2	90	90
3	92,9	88,46	92,9	92,31	92,9	94,87	91,4	87,2	90	90
4	92,9	93,59	91,4	88,46	94,3	89,74	91,4	87,2	90	90
5	88,6	88,46	87,1	93,59	90	93,59	91,4	87,2	90	90
6	91,4	91,03	91,4	89,74	88,6	92,31	91,4	87,2	90	90
7	94,3	94,87	94,3	92,31	90	89,74	91,4	87,2	90	90
8	88,6	92,31	92,9	92,31	90	92,31	91,4	87,2	90	90
9	90	89,74	90	93,59	91,4	94,87	91,4	87,2	90	90
10	92,9	88,46	90	91,03	92,9	91,03	91,4	87,2	90	90
Ortalama	91,7	90,898	91,58	91,54	91,58	92,18	91,4	87,2	90	90
Std. Sapma	2,83	2,1	2,09	1,54	1,96	1,76	0	0	0	0

Genel olarak bakıldığında sistemin başarılı bir şekilde tanıma yüzdesi %90 seviyesindedir. RBNN ve PNN her defasında aynı sonucu vermektedirler. PNN yapısı dışarıdan ilave edilen 4 farklı çiçek türüne karşı sağlam bir yapı sergilemekte fakat RBNN yapısı bu 4 farklı çiçek türünde yanlış tahminlerde bulunmaktadır. Benzer bir şekilde FFNN algoritmasının başarısı bu 4 çiçek türü ilave edildiğinde düşmekte, en iyi sonucu CFNN yapısı vermektedir. Bu yapı için başarı yüzdelerinin standart sapmasının düşük olması CFNN yapısının diğer çok katmanlı YSA modellerine göre daha kararlı yapıda olduğunu göstermektedir. Bu durum Şekil 8.4'te daha açık bir şekilde gösterilmiştir.



Şekil 8.4. Karşılaştırma Sonuçları: Başarı yüzdeleri

9. SONUÇ

Bu tez kapsamında, renkli görüntülerin kesimlenme için üç farklı sürü zekâsı algoritması gerçekleştirilmiştir. Bunların bir tanesi karıncaların hareket mekanizmasından ilham alınarak gerçekleştirilmiş ve temel olarak karıncaların resim yüzeyi etrafında dağıtılarak istenen nesneye ait pikseller üzerinde dolaşmalarını sağlayan ve böylece istenilen nesneyi arka plandan ayıran bir yaklaşıma sahiptir. Diğer iki yöntem pikselleri renk bilgisine göre kümeleyerek kesimleme yapmaya çalışan ve bunu bir optimizasyon problemi olarak ele alıp çözmeye çalışan sürü zekâsı teknikleridir. Bu tekniklerden bir tanesi K-means algoritmasının dezavantajlarından bir tanesi olan küme merkezinin doğru ilklenmesi problemini ortadan kaldırmayı hedeflemektedir. Diğer yaklaşım ise hem optimum küme sayısını hem de bu kümeler için olması gerekli olan küme merkezi değerlerini bulmayı hedeflemektedir.

Karıncaların pikseller üzerinde dağıtılarak kesimleme yapan yöntem MF ile beraber kullanılmış ve renkli retina görüntülerindeki kan damarlarının kesimlenmesi probleminin çözümünde kullanılmıştır. Yöntem DRIVE veri kümesi üzerinde test edilmiş ve diğer damar kesimleme yöntemleri ile karşılaştırılmıştır.

Gerçekleştirilen bu damar belirleme yöntemi bu alanda önerilen ilk sürü tabanlı kesimleme yaklaşımıdır. Literatürdeki tüm damar kesimleme yaklaşımlarından daha iyi sonuç veremese de elde edilen sonuçlar kabul edilebilir düzeydedir. Bu alanda yapılan ilk çalışma olmasına rağmen 93%’lere varan bir başarının elde edilmesi geleceğe yönelik yapılabilecek araştırmalar açısından cesaret vermektedir. Gelecek çalışmalarda yöntemin daha da iyileştirilmesi açısından şu konulara dikkat edilmesi gerekmektedir:

- Yöntem sonucunda elde edilen damarlı bölgeler altın standart resimlerinden daha kalındır. Bu durum doğrudan başarıyı düşürmektedir. Damarları olması gerekli olan bir kalınlıkta kesimleme yapabilmeyi sağlayacak bir eklenti algoritma başarımını arttırabilir.
- Karıncalar bazen yanlışlıkla patolojik bölgeler üzerinde dolaşmakta ve bu bölgeleri damar olarak algılayabilmektedir. Bu da başarımın düşmesine neden olmaktadır. Damar özelliklerini daha iyi ifade eden yeni bir sezgisel

tanımlanabilirse karıncalar sadece damarlı bölgelerde dolaşabilir ve böylece daha başarılı sonuçlar elde edilebilir.

- Algoritma diğer damar çıkarma algoritmalarına göre kısmen daha yavaş çalışmaktadır. Bu sorunu ortadan kaldıracak teknikler üzerinde çalışılması gereklidir.

Renk kümelemesi yapan tekniklerden bir tanesi geliştirdiğimiz IPSOAntK-means algoritmasıdır. Bu algoritma artırılmış parçacık sürü optimizasyonu, karınca kolonisi optimizasyonu ve K-means algoritmasının melez bir şekilde birleştirilmesi ile oluşturulmuştur. Gerçekleştirilen algoritma renkli görüntülerde çiçekli bölgelerin kesimlenmesinde kullanılmıştır ve bu alanda önerilen ilk sürü tabanlı kesimleme algoritmasıdır. Bu algoritma CAVIAR-flower ve Oxford-17 veri kümeleri üzerinde ayrı ayrı denenmiş ve başarılı sonuçlar elde edilmiştir.

IPSOAntK-means algoritması küme merkezlerinin zeki bir şekilde ilklenmesine ihtiyaç duymadan rahatlıkla kesimleme yapabilmekte ve bu sayede K-means algoritmasından daha iyi sonuçlar verebilmektedir. Diğer çiçek kesimleme algoritmaları ile karşılaştırıldığında ümit ve cesaret verici sonuçlar elde edilmiştir. Literatürdeki sadece bir tane çiçek kesimleme yaklaşımından daha kötü sonuç vermektedir. Bunun altında yatan temel neden IPSOAntK-means algoritmasının sadece renk özelliklerine göre kesimleme yapmasıdır. Gelecekte, çiçeğe ait diğer özellikler de ele alınarak kümeleme ve kesimleme gerçekleştirilebilirse daha iyi sonuçların alınması olasıdır.

Diğer piksel kümeleme algoritması genel renkli görüntü kesimleme yapabilme amaçlı gerçekleştirilen ilk karınca kolonisi optimizasyonu algoritmasıdır. Bu algortmada önce muhtemel küme adayları belirlendikten sonra KKO algoritması ile en iyi kümelerin seçilmesi sağlanmıştır. Sonuçlar incelendiğinde KKO algoritmasının renkleri başarı ile kümelediği ve resmi renklerine göre kesimlediği görülmüştür.

Bu yaklaşım da Oxford-17 çiçek veri kümesi üzerinde ve doğal resimleri içeren Berkeley veri seti üzerinde denenmiştir. KKO algoritmasının IPSOAntK-means algoritmasından biraz daha başarılı bir şekilde çiçekleri kesimleme yapabildiği görülmüştür. Doğal resimlerin kesimlenmesinde kullanıldığında renkleri iyi kümeleyebilmesine karşın resim içindeki nesnelerin farklı renklerden oluşması, arka plana yakın renge sahip olması ve renk geçişlerinin sürekli olması

gibi durumlardan ötürü nesnelerin sınırlarını iyi bir şekilde belirlemekte zorlandığı görülmüştür.

Bu algoritmanın IPSOAntK-means algoritmasına oranla daha esnek yapıda olduğu ve daha başarılı ve hızlı çalıştığı tespit edilmiştir. Fakat analizler, IPSOAntK-means algoritmasının aksine parametreler zeki bir şekilde seçilmeden rastgele parametre seçilerek yapılmıştır. İlerideki çalışmalarda, daha sağlıklı ve başarılı sonuçların elde edilebilmesi için KKO tabanlı kümeleme algoritmasına ilişkin parametreler parametre ayar algoritmaları ile seçilip analizler yapılmalıdır.

Gerçekleştirilen tanıma sisteminde 39 farklı çiçek türünden elde edilen test sonuçları %90'nın üzerinde bir başarıyla birçok YSA modelinde sonuç vermektedir. Bu 39 farklı çeşitten bir kısmı birbirine çok benzemesine rağmen bu oranda bir başarının gerçekleşmesi tanıma sisteminde kullanılan kesimleme algoritmasının başarısı ve belirlenen özniteliklerin doğru seçilmesiyle orantılıdır. Her ne kadar elde edilen başarı önemsenecek kadar iyiye de ileride daha gelişmiş tanıma modelleri ve geniş çaplı veri kümeleri için daha sağlam bir tanıma sisteminin geliştirilmesine ihtiyaç vardır.

KAYNAKLAR DİZİNİ

- Adenso-Diaz, B. and Laguna, M.**, 2006, Fine-Tuning of Algorithms using Fractional Experimental Design and Local Search, *Operations Research*, 54(1), 99–114pp.
- Alipoor, M., Imandoost, S., and Haddadnia, J.**, 2010, Designing Edge Detection Filters using Particle Swarm Optimization, In *Proceeding of 18th Iranian Conference on Electrical Engineering*, 548-552pp.
- Al-Rawi, M. and Karajeh, H.**, 2007, Genetic Algorithm Matched Filter Optimization for Automated Detection of Blood Vessels from Digital Retinal Images, *Computer Methods and Programs in Biomedicine*, 87, 248-253pp.
- Al-Rawi, M., Qutaishat, M. and Arrar, M.**, 2007, An Improvement Matched Filter for Blood Vessel Detection of Digital Retinal Images, 37, 262-267pp.
- An-xiang, H., Gang, C., Jun-li, L., Zhe-ru, C., and Dan, Z.**, 2004, A Flower Image Retrieval Method Based on ROI Feature, *Journal of Zhejiang University – Science*, 5(7), 764–772pp.
- Audet, C. and Orban, D.**, 2006, Finding Optimal Algorithmic Parameters using The Mesh Adaptive Direct Search Algorithm, *SIAM Journal on Optimization*, 17(3), 642–664pp.
- Aydın, D.**, 2009, A Modified Ant-Based Approach to Edge Detection, N.T. Nguyen, R. Kowalczyk, and S.-M. Chen (Eds.): *ICCCI 2009, LNAI 5796*, 620–628pp.
- Aydın, D.**, 2010, An Efficient Ant-Based Edge Detector, *LNCS Transactions on Computational Collective Intelligence*, 1, 39-55pp.
- Balaprakash, P., Birattari, M., and Stützle, T.**, 2007, Improvement Strategies for the F-Race Algorithm: Sampling Design and Iterative Reinforcement. In *Proceedings of the International Workshop on Hybrid Metaheuristics (HM 2007)*, T. Bartz-Beielstein et al (Eds.), *LNCS 4771*, Berlin: Springer, 108–122pp.
- Bartz-Beielstein, T.**, 2006, *Experimental Research in Evolutionary Computation: The New Experimentalism*, Natural Computing Series. Springer Verlag, Berlin, Germany.
- Biederman, I.**, 1985, Human Image Understanding: Recent Research and Theory. *Computer Vision, Graphics, and Image Processing*, 32(1), 29-73pp.
- Birattari, M.**, 2009, *Tuning Metaheuristics: A Machine Learning Perspective*, Springer, Berlin, Germany.

KAYNAKLAR DİZİNİ (devam)

- Birattari, M., Stützle, T., Paquete, L. and Varrentrapp, K.**, 2002, A Racing Algorithm for Configuring Metaheuristics. In Proceedings of the Genetic and Evolutionary Computation Conference, Langdon et al. (Eds.), 11-18pp.
- Birattari, M., Yuan, Z., Balaprakash, P., and Stützle, T.**, 2009, F-Race and iterated F-Race: An overview. In Experimental methods for the analysis of optimization algorithms, T. Bartz-Beielstein et al. (Eds.), Berlin, Germany: Springer, 311–336pp.
- Bonabeau, E., Dorigo, M. and Theraulaz, G.**, 1999, Swarm Intelligence From Natural to Artificial Systems, Oxford University Press, New York NY.
- Bonsiepen, L. and Coy, W.**, 1991, Stable Segmentation using Color Information, In proceeding of 4th International Conference on Computer Analysis, Image and Patterns, R. Klette (Eds.), 77-84pp.
- Buhr, J., Sumpter, D.J.T., Cousin, I.J., Hale, J.J., Despland, E., Miller, E.R. and Simpson, S.J.**, 2006, From Disorder to Order in Marching Locusts, Science, 312(5778), 1402-1406pp.
- Bullnheimer, B., Hartl, R.F. and Strauss, C.**, 1999, A new Ranked-Based Version of the Ant System: A Computational Study, Central European Journal for Operational Research, 7, 25-38pp.
- Busin, L., Vandenbroucke, N. and Macaire, L.**, 2008, Color Spaces and Image Segmentation, Advances in Imaging And Electron Physics, 151, 66-169pp.
- Can, A., Shen, H., Turner, H.J., Tanenbaum, H.L., and Roysam, B.**, 1999, Rapid Automated Tracing and Feature Extraction from Retinal Fundus Images using Direct Exploratory Algorithms, IEEE Transactions on Information Technology and Biomedicine, 3(2), 125–138pp.
- Chander, A., Chatterjee, A., and Siarry, P.**, 2011, A New Social and Momentum Component Adaptive PSO Algorithm, Expert Systems with Applications, (in press)
- Chao, N., Qi, L. and Liangzheng, X.**, 2007, An Improved General Particle Swarm Optimization Algorithm for Fast Infrared Image Segmentation, In Proceeding of Chinese Control Conference, 558-562pp.
- Chapron, M.**, 1992, A New Chromatic Edge Detector Used for Color Image Segmentation”, In Proceeding of the IEEE International Conference on Pattern Recognition, A, 311-314pp.

KAYNAKLAR DİZİNİ (devam)

- Chaudhuri, S., Chatterjee, S., and Katz, N.**, 1989, Detection of Blood Vessels in Retinal Images Using Two-Dimensional Matched Filters, IEEE Transactions on Medical Imaging, 1989, 8(3), 263-269pp.
- Cheng, H.D., Jiang, X.H., Sun, Y. and Wang, J.**, 2001, Color Image Segmentation: Advances and Prospects, Pattern Recognition, 34(12), 2259-2281pp.
- Cheng, X. and Gong, X.**, 2008, An Image Segmentation of Fuzzy C-means Clustering Based on the Combination of Improved Ant Colony Algorithm and Genetic Algorithm, In Proceeding of International Workshop on Education Technology and Training, 804-808pp.
- Chutatape, O., Zheng, L. and Krishnan, S.M.**, 1998, Retinal Blood Vessel Detection and Tracking by Matched Gaussian and Kalman Filters, in Proceeding of 20th Annual International Conference of IEEE Engineering in Medicine and Biology, 3144-3149pp.
- Comaniciu, D. and Meer, P.**, 2002, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 603-619pp.
- Cordon, O., Viana, I.F. and Moreno, L.**, 2000, New ACO Model Integrating Evolutionary Computation Concepts: The Best-Worst Ant System, In Proceeding ANTS 2000, Brussels, 22-29pp.
- Cote, B., Hart, W., Goldbaum, M., Kube, P., and Nelson, M.**, 1994, Classification of Blood Vessels in Ocular Fundus Images, Computer Science and Engineering Dept., Univ. of California, San Diego, Technical Report.
- Das, M., Manmatha, R. and Riseman, E. M.**, 1999. Indexing Flower Patent Images using Domain Knowledge, IEEE Intelligent Systems, 14(5), 24-33pp.
- Deneubourg, J.L. and Goss, S.**, 1989, Collective Patterns and Decision Making, Ethology Ecology and Evolution, 1, 295-311pp.
- Deneubourg, J.L., Pasteels, J.M. and Verhaege, J.C.**, 1983, Probabilistic Behaviour in Ants: A Strategy of Errors?, Journal of Theoretical Biology, 105, 259-271pp.
- Deneubourg, J.L., Lioni, A. and Detrain, C.**, 2002, Dynamics of Aggregation and Emergence of Cooperation, Biological Bulletin, 202(3), 262-267pp.

KAYNAKLAR DİZİNİ (devam)

- Deng, Y. and Manjunath, B.S.**, 2001, Unsupervised Segmentation of Color-Texture Region in Images and Video, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8, 800-810.
- Djerou, L., Khelil, N., Dehimi, H.E. and Batouche, M.**, 2009, Automatic Multilevel Thresholding Using Binary Particle Swarm Optimization for Image Segmentation, In Proceeding of International Conference of Soft Computing and Pattern Recognition, 66-71pp.
- Dorigo, M. and Gambardella, L.M.**, 1997, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Transaction on Evolutionary Computation, 1, 53–66pp.
- Dorigo, M., Maniezzo, V. and Colorni, A.**, 1996, The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transaction on Systems ,Man, and Cybernetics Part-B, 26, 29–41pp.
- Engelbrecht, A.P.**, 2007, Computational Intelligence: An Introduction, John Wiley & Sons Ltd., England.
- Evans, A., Sikorski, J., Thomas, P., Cha, S.-H. and Tappert, C.**, 2005, Computer Assisted Visual Interactive Recognition (CAVIAR) Technology. In Proceeding of IEEE International Conference on Electro/Information Technology, 22-25pp.
- Fan, S.-K.S. and Lin, Y.**, 2007, A Multi-level Thresholding Approach Using a Hybrid Optimal Estimation Algorithm, Pattern Recognition Letters, 28, 662-669pp.
- Fernandes, C., Ramos, V., and Rosa, A.C.**, 2005, Self-Regulated Artificial Ant Colonies on Digital Image Habitats, International Journal of Lateral Computing, 1(2), 1-8pp.
- Fukuda, K., Takiguchi, T. and Ariki, Y.**, 2008, Multiple Classifier Based on Fuzzy C-Means for A Flower Image Retrieval, In Proceeding of International Workshop on Nonlinear Circuits and Signal Processing, 76-79pp.
- Gao, H., Xu, W., Sun, J. and Tang, Y.**, 2010, Multilevel Thresholding for Image Segmentation Through an Improved Quantum-Behaved Particle Swarm Algorithm, IEEE Transactions on Instrumentation and Measurement, 59(4), 934-946pp.

KAYNAKLAR DİZİNİ (devam)

- Garnier, S., Gautrais, J. and Theraulaz, G.,** 2007, The Biological Principles of Swarm Intelligence, *Swarm Intelligence*, 1, 3-31pp.
- Gou, C. and Li, H.,** 2007, Multilevel Thresholding Method for Image Segmentation Based on an Adaptive Particle Swarm Optimization Algorithm, M.A. Orgun and J. Thornton (Eds.): *AI 2007, LNAI 4830*, 654–658pp.
- Grassé, P.P.,** 1959, La Reconstruction dun id et les Coordinations Interindividuelles Chez Bellicosimetes Natalensis et Cubitemes sp. la Théorie de la Stigmerie: Essai d'Inrprétation du Comportement des Termites Constructeurs. *Insectes Sociaux*, 6, 41-81pp.
- Han, Y. and Shi, P.,** 2007, An Improved Ant Colony Algorithm for Fuzzy Clustering in Image Segmentation, *Neurocomputing*, 70, 665-671pp.
- Han, Y. and Shi, P.,** 2007, An Efficient Approach for Fish Bone Detection Based on Image Processing and Particle Swarm Clustering, D.-S. Huang, L. Heutte, and M. Loog (Eds.): *ICIC 2007, CCIS 2*, 940–948pp.
- Huang, P., Cao, H. and Luo, S.,** 2008, An Artificial Ant Colonies Approach to Medical Image Segmentation, *Computer Methods and Programs in Biomedicine*, 92, 267-273pp.
- Healey, G.,** 1989, Using Color for Geometry-Insensitive Segmentation, *Optical Society of America*, 22 (1), 920-937pp.
- Hammouche, K., Diaf, M. and Siarry, P.,** 2010, A Comparative Study of Various Meta-heuristic Techniques Applied to the Multilevel Thresholding Problem, *Engineering Applications of Artificial Intelligence*, 23(5), 676-688.
- Hongmei, T., Cuixia, W., Liying, H. and Xia, W.,** 2010, Image Segmentation Based on Improved PSO, In *Proceeding of International Conference on Computer and Communication Technologies in Agriculture Engineering*, 191-194pp.
- Hoover, A., Kouznetsova, V., and Goldbaum, M.,** 2000, Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response, *IEEE Transactions on Medical Imaging*, 19 (3), 203–210pp.
- Hoy, D.E.P.,** 1997, On the Use of Color Imaging in Experimental Applications, *Experimental Techniques*, 21(4), 17-19pp.
- Hsu, T.-H., Lee, C.-H., and Chen, L.-H.,** 2011, An Interactive Flower Image Recognition System, *Multimedia Tools and Applications*, In press.

KAYNAKLAR DİZİNİ (devam)

- Hung, C.-C. and Sun, M.**, 2010, Ant Colony Optimization for the K-means Algorithm in Image Segmentation, in Proceeding of the 48th Annual Southeast Regional Conference, 1-4pp.
- Hutter, F., Hoos, H.H., Leyton-Brown, K., and Stützle, T.**, 2009, ParamILS: An Automatic Algorithm Configuration Framework, Journal of Artificial Intelligence Research, 36, 267-306pp.
- Janson, S., Middendorf, M. and Beekman, M.**, 2005, Honeybee Swarms: How do Scouts Guide a Swarm of Uninformed Bees?, Animal Behaviour, 70(2), 349-358pp.
- Jianlai, W., Chunling, Y., and Chao, S.**, 2009, A Novel Algorithm for Edge Detection of Remote Sensing Image Based on CNN and PSO, In Proceeding of International Congress on Image and Signal Processing, 1-5pp.
- Karsai, I. and Theraulaz, G.**, 1995, Nest Building in a Social Wasp: Postures and Constraints. Sociobiology, 26, 83-114pp.
- Koschan, A. and Abidi, M.**, 2008, Digital Color Image Processing, John Wiley & Sons Inc, New Jersey.
- Laptik, R. and Navakauskas, D.**, 2007, Application of Ant Colony Optimization for Image Segmentation, Electronics and Electrical Engineering, 8(80), 13-18pp.
- Lee, C.-C., Chiang, Y.-C., Shih, C.-Y. and Hu, W.-S.**, 2008, Image Thresholding Using Mean-Shift Based Particle Swarm Optimization, In Proceeding of 8th International Conference on Intelligent Systems Design and Applications, 65-70pp.
- Lei, B. and Fan, J.-L.**, 2008, Parameter Selection of Generalized Fuzzy Entropy-Based Thresholding Segmentation Method with Particle Swarm Optimization, In Proceeding of International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 901-904pp.
- Li, H. and Chutatape, O.**, 2000, Fundus Image Feature Extraction, in Proceeding of the 22nd IEEE Conference on Engineering in Medicine and Biology Society, 1, 3071-3073pp.
- Li, Y.-L., Shen, Y.**, 2010, Fuzzy C-means Cluster Segmentation Algorithm Based on Hybridized Particle Swarm Optimization, In Proceeding of 5th International Conference on Bio-Inspired Computing: Theories and Applications, 811-815pp.

KAYNAKLAR DİZİNİ (devam)

- Liang, Y.-C., Chen, A. H.-L. and Chyu, C.-C.**, 2006, Application of a Hybrid Ant Colony Optimization for the Multilevel Thresholding in Image Processing, King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, 1183–1192pp.
- Littmann, E. and Ritter, H.**, 1997, Adaptive Color Segmentation - A Comparison of Neural and Statistical Methods”, IEEE Transaction on Neural Network, 8(1), 175-185pp.
- Lu, D.-S. and Chen, C.-C.**, 2008, Edge Detection Improvement by Ant Colony Optimization, Pattern Recognition Letters, 29(4), 416-425pp.
- Ma, W.Y. and Manjunath, B.S.**, 1997, Edge Flow: A Framework for Boundary Detection and Image Segmentation, In Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition, 744-749pp.
- Macaire, L., Ultre, V. and Postaire, J.-G.**, 1996, Determination of Compatibility Coefficients for Color Edge Detection by Relaxation, In Proceeding of International Conference on Image Processing, C, 1045–1048pp.
- Martin, D., Fowlkes, C., and Malik, J.**, 2004, Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(5), 530-549pp.
- Martinez-Perez, M.E., Hughes, A.D., Stanton, A.V., Thom, S.A., Bharath, A.A., and Parker, K.H.**, 1999, Segmentation of Retinal Blood Vessels Based on The Second Directional Derivative and Region Growing,” in Proceeding of International Conference on Image Processsing, 173–176pp.
- Mobahi, H., Ahmadabadi, M.N., and Araabi, B.N.**, 2006, Swarm Contours: A Fast Self-Organization Approach for Snake Initialization, Complexity, 12, 41-52pp.
- Mohsen, F.M.A., Hadhoud, M.M., and Amin K.M.**, 2010, Particle Swarm Optimization Method for Very Low Bit Rate Segmented Image Compression, in Proceeding of International Conference on Computer Engineering and Systems, 15-20pp.
- Mojsilovic', A., Kova'cevic', J., Hu, J., Safranek, R.J. and Ganapathy, S.K.**, 2000, Matching and Retrieval Based on Vocabulary and Grammar of Color Patterns, IEEE Transactions on Image Processing, 1(1), 38-54pp.
- Montes de Oca, M.A., Stützle, T., Van den Enden, K. and Dorigo, M.**, 2010, Incremental Social Learning in Particle Swarms, IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 99, 1-17pp.

KAYNAKLAR DİZİNİ (devam)

- Montes de Oca, M.A., Aydın, D. and Stützle, T.,** 2010, An Incremental Particle Swarm for Large-Scale Continuous Optimization Problems: An Example of Tuning-in-the-loop (Re)Design of Optimization Algorithms, *Soft Computing* (in press).
- Nagy, G. and Zou, J.,** 2002 Interactive Visual Pattern Recognition, In *Proceeding of International Conference on Pattern Recognition*, 2, 478–481pp.
- Nekovei, R. and Sun, Y.,** 1995, Back-Propagation Network and Its Configuration for Blood Vessel Detection in Angiograms, *IEEE Transactions on Neural Networks*, 6 (1), 64–72pp.
- Nevatia, A.,** 1977, Color Edge Detector and Its Use in Scene Segmentation, *IEEE Transaction on System Man Cybernetics*, SMC-7 (11), 820–826pp.
- Nezamabadi-pour, H., Saryazdi, S., Rashedi, E.,** 2006, Edge Detection Using Ant Algorithms, *Soft Computing*, 10, 623–628pp.
- Nilsback, M-E. and Zisserman, A.,** 2007, Delving into the Whorl of Flower Segmentation. In *Proceeding of the British Machine Vision Conference*, 570–579pp.
- Nilsback, M.-E. and Zisserman, A.,** 2008, Automated Flower Classification over a Large Number of Classes. In *Proceeding of Indian Conference on Computer Vision, Graphics & Image Processing*, 722–729pp.
- Nilsback, M.-E. and Zisserman, A.,** 2010, Delving deeper into the whorl of flower segmentation, *Image and Visual Computing*, 28(6), 1049–1062pp.
- Omran, M., Engelbrecht, A.P., and Salman, A.,** 2005, Particle Swarm Optimization Method for Image Clustering, *International Journal of Pattern Recognition and Artificial Intelligence*, 19(3), 297–322pp.
- Peer, E.S., Bergh, F. and Engelbrecht, A.P.,** 2003, Using Neighbourhoods with the Guaranteed Convergence PSO. In *Proceeding of the IEEE Swarm Intelligence Symposium*, 235–242pp.
- Peng, J., Chen, Y. and Eberhart, R.C.,** 2000, Battery Pack State of Charge Estimator Design using Computational Intelligence Approaches, In *Proceeding of the Annual Battery Conference on Applications and Advances*, 173–177pp.

KAYNAKLAR DİZİNİ (devam)

- Peram, T., Veeramachaneni and K., Mohan, C.K.**, 2003, Fitness-Distance-Ratio based Particle Swarm Optimization, In Proceeding of the IEEE Swarm Intelligence Symposium, 174-181pp.
- Perfetti R., Ricci E., Casali D. and Costantini G.**, 2007, Cellular Neural Networks with Virtual Template Expansion for Retinal Vessel Segmentation, IEEE Transactions on Circuits and Systems, 54(2), 141-146pp.
- Ramos V. and Almeida, F.**, 2000, Artificial Ant Colonies in Digital Image Habitats- A Mass Behaviour Effect Study on Pattern Recognition, Second International Workshop on Ant Algorithms, 113-116pp.
- Rezaee, A.**, 2008, Extracting Edge of Images with Ant Colony, Journal of Electrical Engineering, 59(1), 57-59pp.
- Rogowska, J.**, 2000, Overview and Fundamentals of Medical Image Segmentation, Handbook of Medical Imaging, Academic Press, 69-85pp.
- Russ, J.C.**, 2007, The Image Processing Handbook, CRC Press, Boca Raton.
- Sahoo, P.K., Soltani, S., Wong, A.K.C. and Chen, Y.C.**, 1988, A Survey of Thresholding Techniques, Computer Vision, Graphics, and Image Processing, 41(2), 233-260pp.
- Saitoh, T., Aoki, K., and Kaneko, T.**, 2004, Automatic Recognition of Blooming Flowers, In Proceeding of International Conference on Pattern Recognition, 27–30pp.
- Saitoh, T. and Kaneko, T.**, 2000, Automatic Recognition of Wild Flowers, In Proceeding of International Conference on Pattern Recognition, 507–510pp.
- Setayesh, M., Zhang, M., and Johnston, M.**, 2009, A New Homogeneity-based Approach to Edge Detection using PSO, in Proceeding of 24th International Conference Image and Vision Computing, 231-236pp.
- Shafer, S.A.**, 1985, Using Color to Separate Reflection Components, Color Research Applications, 10 (4), 210-218.
- Shi, Y. and Eberhart, R.C.**, 1998, A Modified Particle Swarm Optimizer, In Proceeding of the IEEE Congress on Evolutionary Computation, 69-73pp.
- Shi, Y. and Eberhart, R.C.**, 2001, Fuzzy Adaptive Particle Swarm Optimization, In Proceeding of the IEEE Congress on Evolutionary Computation, 1, 101-106pp.

KAYNAKLAR DİZİNİ (devam)

- Souza, J.G. and Costa, A.F.**, 2009, Unsupervised Data Clustering and Image Segmentation Using Natural Computing Techniques, In Proceeding of IEEE International Conference on Systems, Man, and Cybernetics, 5045-5050pp.
- Staal, J.J., Abramoff, M.D., Niemeijer, M., Viergever, M.A. and van Ginneken, B.**, 2004, Ridge Based Vessel Segmentation in Color Images of the Retina", IEEE Transactions on Medical Imaging, 23, 501-509pp.
- Stützle, T. and Hoos, H.H.**, 2000, Max-Min Ant System, Future Generation Computer System, 16, 889-914pp.
- Suganthan, P.N.**, 1999, Particle Swarm Optimiser with Neighborhood Operator, In Proceeding of the IEEE Congress on Evolutionary Computation, 1958-1962pp.
- Tamura, S., Tanaka, K., Ohmori, S., Okazaki, K., Okada, A., and Hoshi, M.**, 1983, Semi-Automatic Leakage Analyzing System for Time Series Fluorescein Ocular Fundus Angiography, Pattern Recognition., 16(2), 149–162pp.
- Tao, W., Jin, H. and Liu, L.**, 2007, Object Segmentation Using Ant Colony Optimization and Fuzzy Entropy, Pattern Recognition Letters, 28, 788-796pp.
- Theraulaz, G., Bonebeau, E. and Deneubourg, J.L.**, 1998, The Origin of Nest Complexity in Social Insects, Complexity, 3(6), 15-25pp.
- Theraulaz, G. and Bonebeau, E.**, 1999, A Brief History of Stigmergy, Artificial Life, 5, 97-116pp.
- Thorpe, W.H.**, 1963, Learning and instinct in animals, Londra, Methuen.
- Tolias, Y. and Panas, S.M.**, 1998, A Fuzzy Vessel Tracking Algorithm for Retinal Images Based on Fuzzy Clustering, IEEE Transactions on Medical Imaging, 17(2), 263–273pp.
- Tou, J.T. and Gonzalez, R.C.**, 1974, Pattern Recognition Principles, Addison-Wesley.
- Tseng, C.-C., Hsieh, J.-G., and Jeng, J.-H.**, 2009, Active Contour Model via Multi-population Particle Swarm Optimization, Expert Systems with Applications, 36, 5348-5352pp.
- Tseng, D.C. and Chang, C.H.**, 1992, Color Segmentation Using Perceptual Attributes, in Proceeding of the IEEE International Conference on Pattern Recognition, A, 228-231pp.

KAYNAKLAR DİZİNİ (devam)

- Uğur, A., Aydın, D.,** 2009, An Interactive Simulation and Analysis Software for Solving TSP using Ant Colony Optimization Algorithms, *Advances in Engineering Software*, 40(5), 341-349pp.
- Wasserman, P.D.,** 1993, *Advanced Methods in Neural Computing*, New York: Van Nostrand Reinhold.
- Wei, C. and Kangling, F.,** 2008, Multilevel Thresholding Algorithm Based on Particle Swarm Optimization for Image Segmentation, In *Proceeding of 27th Chinese Control Conference*, 348-351pp.
- Wyszecki, G. and Stiles, W.S.,** 1982, *Color Science*, Wiley, New York.
- Ye, Z., Zheng, Z., Yu, X., and Ning, X.,** 2005, Automatic Threshold Selection Based on Ant Colony Optimization Algorithm, *International Conference on Neural Networks and Brain*, 728-732pp.
- Yin, P.-Y.,** 2007, Multilevel Minimum Cross Entropy Thresholding Selection based on Particle Swarm Optimization, *Applied Mathematics and Computation*, 184, 503-513pp.
- Yu, Z., Au, O.C., Zou, R., Yu, W., and Tian J.,** 2010, An Adaptive Unsupervised Approach Toward Pixel Clustering and Color Image Segmentation, *Pattern Recognition*, 43, 1889-1905pp.
- Yu, Z., Zou, R., Yu, S. and Mou, H.,** 2009, Noise-robust Binary Segmentation based on Ant Colony System and Modified Fuzzy C-Means Algorithm, in *Proceeding of the 11th Conference on Congress on Evolutionary Computing*, 2488-2493pp.
- Zana, F. and Klein, J.C.,** 1999, A Multimodal Registration Algorithm of Eye Fundus Images Using Vessels Detection and Hough Transform, *IEEE Transactions on Medical Imaging*, 1999, 18(5), 419-428pp.
- Zang, J.,** 2009, Image Segmentation Using Possibilistic C-Means Based on Particle Swarm Optimization, In *Proceeding of WRI Global Congress on Intelligent Systems*, 119-123pp.
- Zhao, X., Lee, M.-E., and Kim, S.-H.,** 2008, Improved Image Thresholding using Ant Colony Optimization Algorithm, *International Conference on Advanced Language Processing and Web Information Technology* 08, 210-215pp.
- Zhenjiang, M., Gandelin, M.-H., and Baozong, Y.,** 2006, An OOPR-based Rose Variety Recognition System, *Engineering Applications of Artificial Intelligence*, 19, 79–101pp.

KAYNAKLAR DİZİNİ (devam)

- Zou, J.**, 2005, A Model-Based Interactive Object Segmentation Procedure. In Proceeding of IEEE Workshops on Applications of Computer Vision, 522–527pp.
- Zou, J. and Nagy, G.**, 2004, Evaluation of Model-Based Interactive Flower Recognition, In Proceeding of International Conference on Pattern Recognition, 311–314pp.

ÖZGEÇMİŞ

1982’de İzmir’de doğdu. İlk ve ortaokulu tamamladıktan sonra liseyi İzmir Atatürk Lisesi’nde tamamladı. Lisans eğitimine 2000 yılında Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü’nde başladı ve 2005 yılında mezun oldu. Aynı yıl Ege Üniversitesi Bilgisayar Mühendisliği Bölümü’nde doktora eğitimine başladı. Şu an, sürü zekâsı ve optimizasyon algoritmaları, hesaplama zekâ ve bunların zor problemlerin çözümünde uygulanması, görüntü işleme ve bilgisayar grafikleri alanlarında çalışmalarını sürdürmektedir. Bu alanlarda, ondan fazla uluslararası ve ulusal bildiri ve makale çalışmaları vardır.

Tez Süresince Yayımlanan Makaleleri

Uğur, A., Aydın, D., 2009, An Interactive Simulation and Analysis Software for Solving TSP using Ant Colony Optimization Algorithms, *Advances in Engineering Software*, 40(5), 341-349pp.

Cinsdikici, M.G., Aydın, D., 2009, Detection of Blood Vessels in Ophthalmoscope Images using MF/Ant (Match Filter / Ant Colony) Algorithm , *Computer Methods and Programs in Biomedicine*, 96(2), 85-95pp.

Aydın, D., 2010, An Efficient Ant-Based Edge Detector, *LNCS Transactions on Computational Collective Intelligence*, 1, 39-55pp.

Aydın, D., Uğur, A., 2010, Automatic Flower Boundary Extraction using IPSOAntK-means algorithm, *Cybernetics and Systems: An International Journal*, 41(6), 416-434pp.

Aydın, D., Uğur, A., 2011, Extraction of Flower Regions in Color Images Using Ant Colony Optimization, *Procedia Computer Science Journal*, WCIT2010, In press.

Tez Süresince Yayımlanan Bildirileri

Aydın, D., Cinsdikici, M.G., 2009, An Ant-Based Algorithm to Detection of Blood Vessels in Retinal Images", *International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2009)* , 512-516pp.

Aydın, D., 2009, A Modified Ant-Based Approach to Edge Detection, N.T. Nguyen, R. Kowalczyk, and S.-M. Chen (Eds.): *ICCCI 2009, LNAI 5796*, 620–628pp.