



**ÇOK SEVİYELİ EŞİKLEME VE ARI ALGORİTMASI YARDIMIYLA
RENKLİ GÖRÜNTÜLERİN SINIFLANDIRILMASI**

Ahmet Selim KAHRAMAN

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

NİSAN 2019

Ahmet Selim KAHRAMAN tarafından hazırlanan “ÇOK SEVİYELİ EŞİKLEME VE ARI ALGORİTMASI YARDIMIYLA RENKLİ GÖRÜNTÜLERİN SINIFLANDIRILMASI” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Recep DEMİRCİ

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Başkan: Dr. Öğr. Üyesi Hilal KAYA

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Nursal ARICI

Bilgisayar Mühendisliği Ana Bilim Dalı, Yıldırım Beyazıt Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Tez Savunma Tarihi: 30/04/2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Sena YAŞYERLİ
Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ahmet Selim KAHRAMAN

30/04/2019

ÇOK SEVİYELİ EŞİKLEME VE ARI ALGORİTMASI YARDIMIYLA RENKLİ GÖRÜNTÜLERİN SINIFLANDIRILMASI

(Yüksek Lisans Tezi)

Ahmet Selim KAHRAMAN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Nisan 2019

ÖZET

Görüntü sınıflandırma veya ayrıştırma bir görüntünün kendisini oluşturan alt küme veya nesne bileşenlerine ayrılması olup, görüntü analizi ve görüntü tanıma işlemlerindeki en önemli adımlardan birisini teşkil eder. Ayrıştırma işleminin temel amacı görüntü üzerinde farklı konumlardaki pikselleri gruplandırarak anlamlı ve yararlı bilgi elde etmektir. Günümüzde görüntü ayrıştırma alanında birçok çalışma bulunmakla birlikte, bu alanda yer etmiş standart bir algoritma veya çözüm bulunmamaktadır. Görüntü sınıflandırmaya yönelik çalışmalarda hedefe ulaşmak adına günümüze kadar geliştirilen yöntemler kümeleme ve eşikleme esaslı olmuştur. Kümeleme yöntemleri güçlü yaklaşımlar olmasına rağmen hesapsal maliyeti görüntü boyutuna bağlıdır. Dolayısıyla gerçek zamanlı görüntü işleme uygulamalarında yetersiz kalmaktadır. Ayrıca iteratif yapıları ciddi bir dezavantaj oluşturmaktadır. Eşikleme metotları ise görüntü ayrıştırma konusunda tercih edilen yaklaşımlardan bir diğeridir. Otsu ve Kapur yöntemleri bu alanda sıklıkla faydalanılan iki çözümdür. Söz konusu yöntemler gri ölçekli görüntülerde sonuç verirken, renkli görüntülerde yetersiz kalmaktadır. İlave olarak eşik sayısı arttıkça hesaplama maliyeti de artmaktadır. Bu çalışmada renkli görüntülerin çok seviyeli eşiklemesine dayalı yeni bir yöntem önerilmiştir. İlk olarak renkli görüntülerin her bir kanalının histogramının hesaplanmasının ardından yapay arı koloni algoritması ile eşik değerleri bulunmuştur. Akabinde elde edilen eşik değerleri ile renk uzayı bölümlenmiştir. Bu sayede elde edilen küpler içerisinde kalan pikseller aynı sınıfa atanmış ve sınıflandırma yapılmıştır.

Bilim Kodu : 92418

Anahtar Kelimeler : Görüntü İşleme, Sınıflandırma, Çok Seviyeli Eşikleme, Renkli Görüntü, Bölütleme

Sayfa Adedi : 65

Danışman : Prof. Dr. Recep DEMİRCİ

COLOR IMAGE CLASSIFICATION WITH MULTILEVEL THRESHOLDING AND ARTIFICIAL BEE COLONY ALGORITHM

(M. Sc. Thesis)

Ahmet Selim KAHRAMAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

April 2019

ABSTRACT

Image classification or segmentation is the separation of image into its subsets or components. It is one of the most essential steps in image analysis and pattern recognition. The main purpose of image segmentation is to group pixels of the image in different positions to obtain meaningful and useful information. Although there are many studies in this field, a standard algorithm or solution does not exist. Methods that have been developed for classifying images are mainly based on clustering and thresholding. However, both of these methods may be insufficient depending on the image that is being processed. Clustering methods which are mainly used in the real-time image processing applications are inadequate due to the computational cost that increases with the size of the image and also because of the iterative structures. Thresholding methods are another preferred approaches in image segmentation. Otsu and Kapur methods are the two solutions frequently used. On the other hand, the thresholding methods are applicable in gray-scale images but are inadequate in processing colored images. With these type of approaches as the number of thresholds increases, the cost of calculation also increases. Several studies have been carried out to find an alternative solution with a lower computation cost. In this study, a new method based on multilevel thresholds of color images has been proposed. Initially, the histogram of each channel in color images has been calculated, then the threshold values are determined by using artificial bee colony algorithm. Accordingly, the color space is partitioned by the threshold values obtained. The pixels remaining in each sub cubes have been assigned to the same class. Consequently, classification image has been realized as a result.

Science Code : 92418

Key Words : Image Processing, Classification, Multilevel Thresholding, Color
Image, Segmentation

Page Number : 65

Supervisor : Prof. Dr. Recep DEMİRCİ

TEŞEKKÜR

Tez çalışmamın geliştirilmesinde yardımları ve desteği ile rehberlik eden değerli hocam ve tez danışmanım Prof. Dr. Recep DEMİRCİ'ye, tecrübesi ve birikimini aktaran Taymaz Rahkar FARSHI'ye, manevi desteklerini eksik etmeyen değerli aileme ve Borsa İstanbul'daki çalışma arkadaşlarıma teşekkürü bir borç bilirim.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ	1
2. SAYISAL GÖRÜNTÜ İŞLEME	9
2.1. Sayısal Görüntü	9
2.2. Gri Ölçekli Görüntü	11
2.3. Renkli Görüntü.....	11
2.4. Histogram	12
2.5. Görüntü Kümeleme ve Sınıflandırma	14
2.6. Görüntü Eşikleme.....	18
2.7. Yapay Arı Koloni Algoritması.....	22
3. RENKLİ GÖRÜNTÜLERİN ÇOK SEVİYELİ EŞİKLENMESİ	31
3.1. Renk Uzayının Tek Eşikli Bölütlenmesi	31
3.2. Renk Uzayının Çok Eşikli Bölütlenmesi	34
4. DENEYSEL SONUÇLAR VE YORUMLAR	39
4.1. Geliştirilen Arayüz	40
4.2. Tek Eşikli Sonuçlar	41
4.3. Çok Eşikli Sonuçlar.....	46

Sayfa

5. SONUÇ	59
KAYNAKLAR	61
ÖZGEÇMİŞ	65



ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Renk uzayının bölümlenmesi	32
Çizelge 4.1. Çift için bulunan eşikler ($m=1$)	42
Çizelge 4.2. Lena için bulunan eşikler ($m=1$)	43
Çizelge 4.3. Biber için bulunan eşikler ($m=1$).....	44
Çizelge 4.4. Papağanlar için bulunan eşikler ($m=1$).....	45
Çizelge 4.5. Babun için bulunan eşikler ($m=1$).....	46
Çizelge 4.6. Çift için bulunan eşikler ($m=2$)	47
Çizelge 4.7. Çift için bulunan eşikler ($m=3$)	48
Çizelge 4.8. Lena için bulunan eşikler ($m=2$)	49
Çizelge 4.9. Lena için bulunan eşikler ($m=3$)	50
Çizelge 4.10. Biber için bulunan eşikler ($m=2$).....	51
Çizelge 4.11. Biber için bulunan eşikler ($m=3$).....	52
Çizelge 4.12. Papağanlar için bulunan eşikler ($m=2$).....	53
Çizelge 4.13. Papağanlar için bulunan eşikler ($m=3$).....	54
Çizelge 4.14. Babun için bulunan eşikler ($m=2$).....	55
Çizelge 4.15. Babun için bulunan eşikler ($m=3$).....	56

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Görüntü işlemenin temel adımları	9
Şekil 2.2. Sayısal görüntünün gösterimi	11
Şekil 2.3. Gri ölçekli Lena görüntüsü ve piksel değerleri	11
Şekil 2.4. RGB Renk Uzayı	12
Şekil 2.5. Renkli Lena.....	13
Şekil 2.6. Gri ölçekli Lena histogramı	13
Şekil 2.7. Renkli Lena histogramı	14
Şekil 2.8. a) Sentetik görüntü b) Sınıflandırma c) Ayırıştırma	14
Şekil 2.9. a) K-ortalamlar örneği a) Kümelenmesi istenen veri seti b) Merkezlerin rastgele ya da bir yöntemle bağlı olarak seçilmesi c) 1. İterasyon d) 2. İterasyon e) 3. İterasyon f) Kümeleme	16
Şekil 2.10. Eşikleme sonuçları a) Kapur, $t=123$ b) Otsu, $t=117$	21
Şekil 2.11. Eşikleme sonuçları a) Kapur, $t_1=98$, $t_2=164$ b) Otsu, $t_1=93$, $t_2=151$	22
Şekil 2.12. Yiyecek arama çevrimi	24
Şekil 2.13. Yapay arı koloni algoritması akış diyagramı	29
Şekil 3.1. Üç boyutlu renk uzayı ve atanan sınıflar $m=1$	32
Şekil 3.2. Renkli Lena: RGB uzayındaki renk dağılımı	33
Şekil 3.3. a) Biber b) Biber: RGB uzayındaki renk dağılımı	33
Şekil 3.4. a) Papağanlar b) Papağanlar: RGB uzayındaki renk dağılımı	33
Şekil 3.5. a) Babun b) Babun: RGB uzayındaki renk dağılımı	34
Şekil 3.6. Lena RGB renk uzayı ve atanan kümeler, $m=3$	37
Şekil 4.1. Geliştirilen arayüz a) Giriş b) Tek eşikli sonuç örneği c) İki eşikli sonuç örneği	40
Şekil 4.2. Çift	41

Şekil	Sayfa
Şekil 4.3. Çift, $m = 1$ a) Otsu, $c = 2$ b) Kapur, $c = 2$ c) FCM, $c = 2$, süre(s): 1,480	42
Şekil 4.4. Lena, $m = 1$ a) Otsu, $c = 8$ b) Kapur, $c = 7$, c) FCM, $c = 8$, süre(s):1,158..	43
Şekil 4.5. Biber, $m = 1$ a) Otsu, $c = 8$ b) Kapur, $c = 8$ c) FCM, $c = 8$, süre(s):1,138..	44
Şekil 4.6. Papağanlar, $m = 1$ a) Otsu, $c = 8$ b) Kapur, $c = 8$ c) FCM, $c = 8$, süre(s): 1,752.....	45
Şekil 4.7. Babun, $m = 1$ a) Otsu, $c = 8$ b) Kapur, $c = 8$ c) FCM, $c = 8$, süre(s):1,161.	46
Şekil 4.8. Çift, $m = 2$ a) Otsu, $c = 3$ b) Kapur, $c = 3$ c) FCM, $c = 3$, süre(s): 4,884	47
Şekil 4.9. Çift, $m = 3$ a) Otsu, $c = 4$ b) Kapur, $c = 4$, c) FCM, $c = 4$, süre(s): 14,819 ...	48
Şekil 4.10. Lena, $m = 2$ a) Otsu, $c = 21$ b) Kapur, $c = 20$ c) FCM, $c = 21$, süre(s):9,992	49
Şekil 4.11. Lena, $m = 3$ a) Otsu, $c = 39$ b) Kapur, $c = 33$ c) FCM, $c = 39$, süre(s): 46,702.....	50
Şekil 4.12. Biber, $m = 2$ a) Otsu, $c = 23$ b) Kapur, $c = 21$ c) FCM, $c = 23$, süre(s): 13,135.....	51
Şekil 4.13. Biber, $m = 3$ a) Otsu, $c = 48$ b) Kapur, $c = 40$ c) FCM, $c = 48$, süre(s): 73,439.....	53
Şekil 4.14. Papağanlar, $m = 2$ a) Otsu, $c = 23$ b) Kapur, $c = 23$ c) FCM, $c = 23$, süre(s): 18,285.....	54
Şekil 4.15. Papağanlar , $m = 3$ a) Otsu, $c = 45$ b) Kapur, $c = 45$ c) FCM, $c = 45$, süre(s): 93,485.....	55
Şekil 4.16. Babun , $m = 2$ a) Otsu, $c = 25$ b) Kapur, $c = 23$ c) FCM, $c = 25$, süre(s): 13,769.....	56
Şekil 4.17. Babun , $m = 3$ a) Otsu, $c = 52$ b) Kapur, $c = 49$ c) FCM, $c = 52$, süre(s): 89,748.....	57

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ABC	Artificial Bee Colony
ABF	Amended Bacterial Foraging
BA	Bat Algorithm
BFO	Bacterial Foraging Optimization
CCA	Cricket Chirping Algorithm
CMY	Cyan Magenta Yellow
CS	Cuckoo Search
EHO	Elephant Herding Optimization
GWO	Grey Wolf Optimizer
FCM	Fuzzy C-means
FWA	Standard Fireworks Algorithm
HSI	Hue Saturation Intensity
MBFO	Modified Bacterial Foraging Optimization
MÇS	Maksimum Çevrim Sayısı
MR	Magnetic Resonance
MSA	Moth Swarm Algorithm
NASA	National Aeronautics and Space Administration
PSO	Partical Swarm Optimization
RGB	Red Green Blue
WDO	Wind Driven Optimization
WOA	Whale Optimization Algorithm
YAK	Yapay Arı Kolonisi

1. GİRİŞ

Görüntü x ve y uzamsal değişkenler olmak üzere nesnelerin $f(x,y)$ fonksiyonu olarak iki boyutlu düzlemdeki iz düşümüdür. Sayısal görüntü ise f fonksiyonun (x,y) koordinatındaki ışık şiddetinin sayısal bir değer olarak ifade edilmesidir. Sayısal görüntü işleme görüntü algılayıcılar yardımıyla elde edilen sayısal görüntünün bilgisayarlar tarafından işlenerek anlamlı bir hale getirilme süreçleridir.

Sayısal görüntü işlemenin ilk adımları 1920'li yıllara uzanmaktadır. Newyork ve Londra arasında normalde 1 haftadan daha fazla süren ilk görüntülü haberleşme tahmin edilen sürelerin çok altında, 3 saat gibi kısa bir sürede tamamlanmıştır. 1921'lerin sonlarına doğru gerek gönderilen fotoğrafların ton ve kalite anlamında iyileştirilmesi, gerekse görüntü çoğaltmaya dayalı tekniklerin geliştirilmesiyle bu alanda ciddi atılımlar meydana gelmiştir. İlgili gelişmeler bir süre sonra görüntünün hesaplama ve işleme süreçlerini daha hızlı yapacak teknolojilere olan ihtiyacı beraberinde getirmiştir. Söz konusu ihtiyaçlara cevap veren ilk bilgisayar 1960'lı yıllarda ortaya çıkmıştır. 1964 yılında NASA'nın üretmiş olduğu uzay aracı Ranger7 ile Ay'ın ilk fotoğrafları çekilmiş ve bu görüntüler bilgisayarlar aracılığıyla işlenmiştir. Görüntü iyileştirme ve onarma adımlarının temeli bu yıllarda ortaya çıkmıştır. 1970'li yıllara gelindiğinde ise görüntü işleme teknikleri bilgisayarlı tomografinin keşfi ile tıp alanına yayılmıştır. 1960'lı yıllardan günümüze kadar görüntü işleme endüstri, güvenlik ve coğrafya gibi birçok alanda kullanılmıştır ve halen kullanılmaktadır [1].

Sayısal bir görüntü üzerindeki piksellerin fiziksel koordinatları dikkate alınmadan aynı grup altında toplanmasına kümeleme veya sınıflandırma, konumları ve komşulukları göz önüne alınarak yapılan sınıflandırmaya ise ayrıştırma ya da bölütleme denir. Görüntü ayrıştırmadaki temel amaç görüntü üzerindeki farklı nesnelere ait piksellerin aynı küme altında anlamlı bölgeler halinde tasnif edilmesidir. Böylece görüntü analizi işlemlerinin ilk adımı olan nesne çıkarımı mümkün hale gelecektir [2]. Görüntü ayrıştırma problemi için farklı çözüm önerileri yapılmıştır. Bölge büyütme ve genişletme, kenar, ağaç, olasılık, kümeleme, maske tabanlı, yapay sinir ağları, benzerlik ve eşikleme tabanlı yaklaşımlar tipik örneklerdir [3,4]. Görüntü sınıflandırma için en çok kullanılan metotlar K-ortalamlar (K-means) [5] ve bulanık C-ortalamlar (Fuzzy C means: FCM) [6] yaklaşımlarıdır. En eski kümeleme algoritmalarından olan K-ortalamlar algoritmasında bir pikselin sadece bir kümeye atanabilmesi en belirgin dezavantajdır. Bu nedenle K-ortalamlar algoritmasına

bulanık mantık yaklaşımı eklenerek iyileştirme ve geliştirilme yapılmış, sonucunda FCM algoritması ortaya çıkmıştır. FCM metodunda piksellerin iki veya daha fazla kümeye ait olmasına imkân tanınır ve piksellerin ilgili kümelere üyelikleri için 0-1 aralığında değerler atanır. Gerek K-ortalamlar gerekse FCM algoritmaları küme sayısının belli ve az olduğu durumda başarılıdır. Fakat küme sayısı bilinmediğinde görüntü ayrıştırma süreci çok daha zor bir hâl almaktadır. Ayrıca iki algoritmanın da iteratif ve kullanıcı bağımlı bir yapıya sahip olmaları ve küme sayısının arttığı durumda işlem hesap maliyetinin yüksek olması en temel ve ciddi problemlerdir.

Görüntü sınıflandırma yöntemlerinden biri olan eşikleme histogram tabanlı bir çözüm yöntemidir. Histogram yaklaşımında işlenecek veri miktarı görüntünün boyutundan bağımsızdır ve resimdeki bilgiler histogram yardımıyla temsil edilmektedir. Bu nedenle hesaplama maliyeti kümeleme algoritmalarına göre düşüktür [1]. Diğer taraftan eşikleme tekniklerinin de kendine has problemleri mevcuttur. Görüntüdeki sınıf sayısı eşik sayısı ile orantılı olduğundan eşik sayısının seçimi kritiktir. Ayrıca uygun eşikleri tespit etmek için eşik değerlerinin kullanıldığı çok değişkenli amaç fonksiyonunun doğru seçimi önemlidir. Seçilen amaç fonksiyonunu maksimum yapan noktaların tespiti ise çok değişkenli ve kısıtlamalı bir optimizasyon problemidir. Görüntülerin eşikleme tabanlı sınıflandırılması için temelde önerilen iki farklı amaç fonksiyonu mevcuttur. Bunlar histogramın entropi değerini referans alan Kapur [7] ve sınıflararası varyansı temel alan Otsu [8] yöntemleridir. Otsu ve Kapur metotları gri ölçekli resimlerin iki veya çok seviyeli eşiklenmesinde en çok kabul gören sonuçları üretmektedirler. Mevcut yöntemler tek eşik seçiminin yapıldığı durumlarda hızlı sonuç vermektedir. Fakat eşik sayısı arttığında hassasiyet artmasına rağmen hesaplama süresi de artmakta olup bu durum problem teşkil etmektedir. Ayrıca tek bir kanalın histogram bilgisi kullandığından söz konusu yöntemler renkli görüntülerin sınıflandırılmasında verimli olmamaktadır.

Literatüre bakıldığında geçmişten günümüze, tek ve çok seviyeli eşikleme çalışmalarının gri seviyeli görüntüler üzerinde uygulandığı gözlenmiştir. Wang ve Haralick [9] geliştirdiği yöntemde gri ölçekli bir görüntüdeki pikselleri kenar pikseller ve kenar olmayan pikseller olarak iki sınıfa tahsis etmiştir. Kenar pikseller yerel komşuluklar referans alınarak ışıık şiddetine göre karanlık ve karanlık olmayan pikseller olarak kümelendirilmiştir. Karanlık olan bölgeler için ayrı histogram, aydınlık olan bölgeler için ayrı histogram hesaplanarak en yüksek tepe değerleri eşik olarak seçilmiştir. Reddi ve arkadaşları [10] tekli ve çoklu eşikleri

hesaplayabilmek adına koyu ve açık bölgeler arasındaki sınıflar arası değişimi maksimize eden ve Otsu amaç fonksiyonu ile bütünleşmiş hızlı bir arama yöntemi ortaya koymuşlardır. İlgili çalışmada her bir arama şeması için bir ölçüt fonksiyonu tanımlanmıştır. Söz konusu yöntem gri seviyeli görüntülerin histogramı için sürekli olasılık değeri çıkarımına dayanmaktadır. Abutaleb [11] gri ölçekli görüntülerde eşik değerini tespit etmek için yeni bir yöntem geliştirmiştir. İlk etapta entropi tabanlı eşikleme iki boyutlu histogram ile genişletilmiştir. Bu yaklaşımda her piksel ve her piksele ait komşu bölgelerin gri seviye değeri incelenmiştir. Böylece eşik değeri iki girişe sahip bir vektör olarak ifade edilmiştir. Sonuç olarak iki boyutlu entropiyi maksimize eden vektör eşik değerini temsil etmektedir. Pal ve Pal [12] görüntünün gri seviye bilgisine bağlı olarak Kapur modelini referans alıp koşullu entropi çalışmasını ortaya koymuşlardır. Daha sonra eşitlik matrisi aracılığıyla nesneyi arka plandan ayırıp bölgelere tasnif etmişlerdir. Papamarkos ve Gatos [13] gri ölçekli görüntülerin çok seviyeli eşiklenmesinde üç aşamadan meydana gelen yeni bir yöntem geliştirmişlerdir. İlk aşamada görüntünün histogramının elde edilmesinin ardından tepe noktalarını takriben belirlemek için tepe kümeleme tekniği uygulanmıştır. Daha sonra hızlı, doğrusal ve rasyonel yaklaşımlar ile histogram bölümlenerek en büyük-en küçük ilkesi referans alınıp tepe noktaları birbirlerine yaklaştırılmıştır. Son olarak tek boyutlu altın oran arama algoritması vasıtasıyla çok seviyeli eşik değerine karşılık gelen rasyonel fonksiyonun küresel minimum değerleri hesaplanmıştır. Elde edilen sonuçlar eşik değerlerine karşılık gelmektedir. Yin and Chen [14] eşikleme yöntemlerini iyileştirmek için hızlı ve tekrarlanan bir algoritma geliştirmişlerdir. Önerilen metot çok seviyeli eşikleme sonuçlarını bulabilmek için iki seviyeli eşik hesabı ile başlar. Elde edilen sonuçlar çok seviyeli eşik değerlerine genişletilmek için hesaplanır. Daha sonra iteratif bir şekilde eş zamanlı olarak amaç fonksiyonu tespit edilir ve her eşik değeri optimize etmek için ayarlanır. Liao ve arkadaşları [15] gri ölçekli görüntülerde Otsu tabanlı ve hesapsal maliyet açısından avantajlı çok seviyeli eşikleme yöntemi geliştirmişlerdir. İlgili çalışmada sınıflararası varyans değerini maksimum yapan değeri daha hızlı hesaplayan, özyinelemeli yeni bir algoritma tasarlanarak hesaplama verimliliği arttırılmıştır. Albuquerque ve arkadaşları [16] Kapur'un yönteminden ilham alarak Tsallis entropiyi gri seviyeli görüntülerin bölütlemesinde uygulamış ve başarılı sonuçlar almışlardır. Zhang ve arkadaşları [17] ise, morfoloji ve dalgacık dönüşümünü Otsu yöntemi ile birleştirerek gri seviyeli görüntüler üzerinde yeni bir sınıflandırma yöntemi önermişlerdir. Söz konusu çalışmada girdi görüntüsünü işlemek için morfoloji, görüntü üzerindeki nesneleri ayırtmak için ise dalgacık dönüşümü ve Otsu yöntemlerinden faydalanılmıştır. Huang ve arkadaşları [18] Otsu yönteminin temel

prensiplerini analiz ederek ve referans alarak gri ölçekli görüntüler üzerinde kendi ayrıştırma metodunu geliştirmişlerdir. Söz konusu çalışmada ilk olarak en yüksek ve en düşük eşik değerleri hesaplanır. Bu değerler aralığında sınıf içi ve sınıflarası varyanslar elde edilir, minimum varyans oranı bulunur ve geliştirilen algoritma ile en optimal eşik değerleri hesaplanarak sınıflandırma işlemi tamamlanmış olur.

Eşikleme teknikleri birçok alanda kullanılmış ve faydalanılmış olup bunların başında manyetik rezonans(Magnetic Resonance: MR) görüntüleri gelmektedir. Wells ve arkadaşları [19] bayes teoremi ve beklenti büyütme (Expectation Maksimization: EM) algoritmasını birleştirerek MR resimlerinin bölütlenebilmesi için uyarlanabilir ayrıştırma adını verdikleri yeni bir yöntem geliştirmişlerdir. Söz konusu yöntem ile MR görüntülerinin homojenlikleri artırılmış olup cerrahi planlama, hastalık araştırması ve ilaç tedavisi gibi yorumların daha etkin bir biçimde yapılabilmesine olanak tanınmıştır. Sathya ve Kayalvizhi [20] ise MR görüntülerinin çok seviyeli eşiklenmesi için iyileştirilmiş bakteri (Amended Bacterial Foraging: ABF) adını verdikleri yeni bir algoritma tasarlamışlardır. Geleneksel eşikleme metodlarının aksine operatör bağımlı olmayan ve hesapsal maliyeti daha başarılı olan bu yaklaşım Otsu ve Kapur yöntemlerinin genişletilmesi ile elde edilmiştir. Vidyarthi ve Mittal [21] beyindeki tümörlerin tespit edilebilmesi adına Otsu metodu ile morfolojik filtreleme yöntemlerini hibrit bir biçime dönüştürerek yeni bir yaklaşım ortaya koymuş ve MR görüntülerinin sınıflandırılmasında başarılı sonuçlar elde etmişlerdir.

Gri seviyeli görüntülerin tek ve çok seviyeli eşiklenmesinde daha etkin ve hızlı sonuçlar almak adına optimizasyon tekniklerinden yararlanılmaktadır. Optimizasyon eniyileme anlamına gelmektedir. Sınırları belli olmak koşuluyla bir amaç fonksiyonunun minimum ya da maksimum değerini elde etme prensibine dayanmaktadır [22]. Günümüze kadar yapılan çalışmalara bakıldığında görüntü ayrıştırma çalışmalarında optimizasyon yöntemlerinin önem teşkil ettiği görülmektedir. Genetik algoritma bu yöntemlerden bir tanesidir. Genetik algoritma evrime dayalı ve gözlemsel deneyler yoluyla elde süreçlerin analiz edilerek, J. Holland tarafından 1975 yılında ortaya koyulan bir optimizasyon yöntemidir [22]. Yin [23] Otsu ve Kapur amaç fonksiyonlarını genetik algoritma ile optimize ederek gri ölçekli görüntülerde çok seviyeli eşikleme yaklaşımını ortaya koymuştur. Tao ve arkadaşları [24] ise geliştirdikleri yöntemde ilk olarak olasılık analizi yardımıyla bulanık entropi tanımlamışlardır. Daha sonra görüntü sırasıyla Z-fonksiyonu, P-fonksiyonu ve S-fonksiyonu ile karanlık, gri ve açık olmak üzere üç ayrı bulanık kümeye bölütlenmiştir. Bulanık bölgenin

nitelikleri entropi prensibi ve genetik algoritma ile hesaplanmıştır. Hammouche ve arkadaşları [25] gri seviyeli görüntülerde çok seviyeli eşikleri tespit etmek amacıyla genetik algoritma tabanlı yeni bir yöntem önermişlerdir. Geliştirilen metotta ilk olarak elde edilen histogram bilgisinin uzunluğu dalgacık dönüşümü kullanılarak azaltılır. Daha sonra histogramın seviyesi referans alınarak genetik algoritma yardımıyla eşik değerleri belirlenir. Elde edilen eşik değerleri ilk etapta hesaplanan orijinal alana yansıtılır ve sonuçlar elde edilmiş olur.

Karınca koloni optimizasyon algoritması görüntü ayrıştırma alanında faydalanan bir diğer optimizasyon tekniğidir. Doğada koloni halinde belirli bir iş paylaşımı içerisinde yaşayan karıncaların davranışlarından esinlenerek modellenen ve ilk olarak 1991 yılında Dorigo ve arkadaşları tarafından ortaya konan bir çalışmadır [22]. Söz konusu yöntem tek ve çok seviyeli eşiklerin tespit edilmesinde kullanılmıştır. Ye ve arkadaşları [26] gri ölçekli görüntülerin eşik değerlerini daha hızlı ve iteratif olmayan bir yapıda hesaplamak adına karınca koloni optimizasyon algoritmasından faydalanmışlardır. İlgili çalışmada karıncaların başlangıç arama çözümleri hesaplanır ve deneme matrisleriyle aday çözümler güncellenir. Bu işlem eşik değerine karşılık gelen maksimum amaç fonksiyonu elde edilene dek tekrarlanır. Tao ve arkadaşları [27] kızılötesi görüntüler üzerinde bulanık entropi ve karınca koloni algoritmasını optimize ederek yeni bir sınıflandırma yöntemi önermişlerdir. Geliştirilen yöntemin geleneksel entropi çalışmalarından daha başarılı sonuç verdiği ifade edilmiştir.

Parçacık sürü optimizasyonu(Partical Swarm Optimization: PSO) ve yapay arı koloni (Artificial Bee Colony: ABC) algoritmaları görüntü ayrıştırma çalışmalarında kullanılan önemli iki optimizasyon tekniğidir. PSO sürü halinde hareket ederek bir dayanışma içerisinde hareket eden canlıları, ABC algoritması ise bir işbirliği içerisinde hareket eden arıların davranışlarından esinlenerek modellenen optimizasyon yaklaşımlarıdır. Yin [28] minimum çapraz entropi eşiği ve PSO algoritmasını optimize ederek gri ölçekli görüntüler üzerinde çok seviyeli eşikleri tespit etmiştir. Maitra ve Chatterjee [29] PSO metodunun varyansını genişleterek melez bir yapı haline getirmişlerdir. Bu sayede PSO yönteminin öğrenme kabiliyeti arttırılmış ve çok seviyeli eşiklemede başarılı sonuçlar elde edilmiştir. Horng ve Jiang [30] ABC algoritmasını maksimum entropi yaklaşımı ile birleştirerek gri seviyeli görüntülerin çok seviyeli eşiklenmesinde yeni bir yöntem ortaya koymuşlardır. Zhang ve Wu [31] gri ölçekli görüntülerde Tsallis entropi ile ABC algoritmasını optimize ederek çoklu

eşikleri tespit etmişlerdir. Er [32] dalgacık dönüşümü, entropi ve ABC algoritmasını kullanarak gri renkli görüntüler üzerinde yeni bir görüntü ayrıştırma yöntemi önermiştir. Geliştirilen metotta gri ölçekli görüntülerin dalgacık dönüşümünün yapılmasının ardından ABC algoritması kullanılarak entropi fonksiyonunu maksimum yapan değer elde edilmiş ve sınıflandırma işlemi tamamlanmıştır. Tural [33] ABC algoritmasının üç farklı varyasyonunu entropi maksimizasyonu ve Otsu yöntemleri ile optimize ederek gri ölçekli görüntüler üzerinde çoklu eşikleri tespit etmiştir.

Son yıllarda gri ölçekli resimlerin çok seviyeli eşiklenmesinde ve ayrıştırılmasında birden fazla optimizasyon tekniği birleştirilerek çözüm aranmıştır. Akay [34] Otsu ve Kapur yöntemlerinin amaç fonksiyonlarını maksimize etmek için PSO ve ABC algoritmalarını kullanmıştır. İstatistiksel sonuçlar ele alındığında eşik sayısının iki olduğu durumda PSO ve ABC'nin aynı oranda performans gösterdiği ve doğrusal Otsu yönteminin nispeten daha başarılı olduğu, ikiden fazla eşik seçimlerinde ise ABC algoritmasının PSO ve lineer arama yöntemlerine göre daha iyi performans gösterdiği ortaya konulmuştur. Bhandari ve arkadaşları [35] çok seviyeli eşikleme problemini sürü zekâsı tabanlı guguk kuşu arama (Cuckoo Search: CS) ve rüzgâr gücü optimizasyonu (Wind Driven Optimization: WDO) algoritmaları ile Kapur'un önerdiği entropiyi maksimize ederek en optimal eşik değerlerini bulan algoritmayı tasarlamışlardır. Söz konusu çalışmada ilk etapta rastgele eşik değerleri üretilir. Elde edilen eşik değerlerinden en iyi çözümler hafızada tutulur. Daha sonra çözümlerin kalitesini değerlendirmek için korelasyon fonksiyonu kullanılmaktadır. Deneysel sonuçlar incelendiğinde önerilen yöntemin örnek uydu görüntüleri üzerinde başarılı sonuçlar verdiği görülmektedir. Singh ve arkadaşları [36] sosyal örümcek ve ateş böceği optimizasyon yöntemleri ile Kapur ve Otsu yöntemlerinin amaç fonksiyonunu optimize ederek gri seviyeli görüntüler üzerinde çok seviyeli eşikleme yapan metodu ortaya koymuşlardır. Deneysel sonuçlar ele alındığında ateşböceği yöntemi ile bulunan eşik değerleri ve hesapsal maliyetin, sosyal örümcek yöntemine göre daha iyi sonuçlar verdiği saptanmıştır. Wang ve arkadaşları [37] tarafından geliştirilen fil sürü optimizasyonu (Elephant Herding Optimization: EHO) yaklaşımını Tuba ve arkadaşları [38] Otsu ve Kapur metotları ile birleştirerek çok seviyeli eşikleme esaslı yeni bir yöntem ortaya koymuşlardır. Tang ve arkadaşları [39] bakteriyel toplama optimizasyonu (Bacterial Foraging Optimization: BFO) yöntemini PSO ve bakteri optimizasyonu (Bacterial Foraging: BF) algoritmaları ile genişleterek değiştirilmiş bakteri optimizasyonu (Modified Bacterial Foraging Optimization: MBFO) metodunu geliştirmiş ve gri ölçekli görüntülerin çok

seviyeli eşiklenmesinde başarılı sonuçlar elde etmişlerdir. El Aziz ve arkadaşları [40] gri ölçekli görüntülerde çok seviyeli eşiklerin tespiti için balina optimizasyonu (Whale Optimization Algorithm: WOA) algoritması ile Kapur ve Otsu yöntemlerini optimize etmişlerdir. Önerilen metodun iyi bir performans gösterdiği ve işlem süresinin makul bir seviyede olduğu ifade edilmiştir. Maurya ve arkadaşları [41] hibrit bir yaklaşım sergileyerek, fişek ve ahenk (Fireworks and Harmony Search) algoritmalarını birleştirmiş ve gri ölçekli görüntüler üzerinde yeni bir eşikleme yöntemi ortaya koymuşlardır.

Önerilen çalışmanın sonuçlarının standart fişek algoritması (Standard Fireworks Algorithm: FWA) ve PSO algoritmaları ile karşılaştırdığında daha başarılı sonuçlar ürettiği ifade edilmiştir. Zhou ve arkadaşları [42] optimizasyon tabanlı meta-sezgisel kebelek sürüsü optimizasyonu (Meta-Heuristic Moth Swarm Algorithm: MSA) algoritmasını Kapur metodu ile birleştirmiş ve gri ölçekli görüntüler üzerinde eşikleri tespit etmişlerdir. Önerilen yöntem sekiz adet test görüntüsü üzerinde uygulanmıştır. Geliştirilen metodun WOA, BA, GWO algoritmalarına göre daha iyi sonuç verdiği ve daha verimli bir şekilde çalıştığı görülmüştür. Sathya ve arkadaşları [43] çekirge ötüş (Cricket Chirping Algorithm: CCA) algoritmasını kullanarak ilk etapta görüntünün histogramı üzerinde rastgele çözüm kümeleri oluşturmuş ve bunları aday çözümler olarak hafızada tutmuşlardır. Daha sonra CCA operatörlerini kullanarak Otsu ve Kapur yöntemleri ile optimize etmiş ve hafızadaki en iyi çözümü eşik değeri olarak kabul etmişlerdir.

Bahsedildiği üzere, geçmişten günümüze dek tek ve çok seviyeli eşikleme tabanlı görüntü sınıflandırma çalışmalarının gri ölçekli görüntülerde etkili olduğu görülmektedir. Hâlbuki günümüzde renkli görüntüler savunma sanayiden tıp alanına, uzay bilimlerinden mühendislik problemlerinin çözümüne kadar birçok alanda yaygın olarak kullanılmaktadır. Dolayısıyla renkli görüntülerin doğru, hızlı ve kullanıcıdan bağımsız bir şekilde gerçek zamanlı bir biçimde işlenerek sınıflandırılmasına yönelik gereksinimlerin varlığından söz etmek mümkündür. Bu sebeple renkli görüntülerin otomatik olarak sınıflandırılmasına yönelik yeni bir metoda ihtiyaç duyulmuştur. Şimdiye kadar ifade edilen yöntemler gri ölçekli görüntülerde sonuç vermektedir ancak renkli görüntülerde yetersiz kalmaktadır. Konu ile alakalı çalışmalar incelendiğinde renkli görüntülerin eşiklenmesine yönelik geliştirilmiş standart bir yöntem bulunmadığı görülmüştür. Gri ölçekli resimlerde tek kanal bilgisi kullanılmaktadır. Renkli resimlerde ise birbirinden bağımsız üç farklı kanal ve her kanalın kendisine ait ayrı histogram bilgisi bulunduğu için bu kanalların birleştirilmesi ve

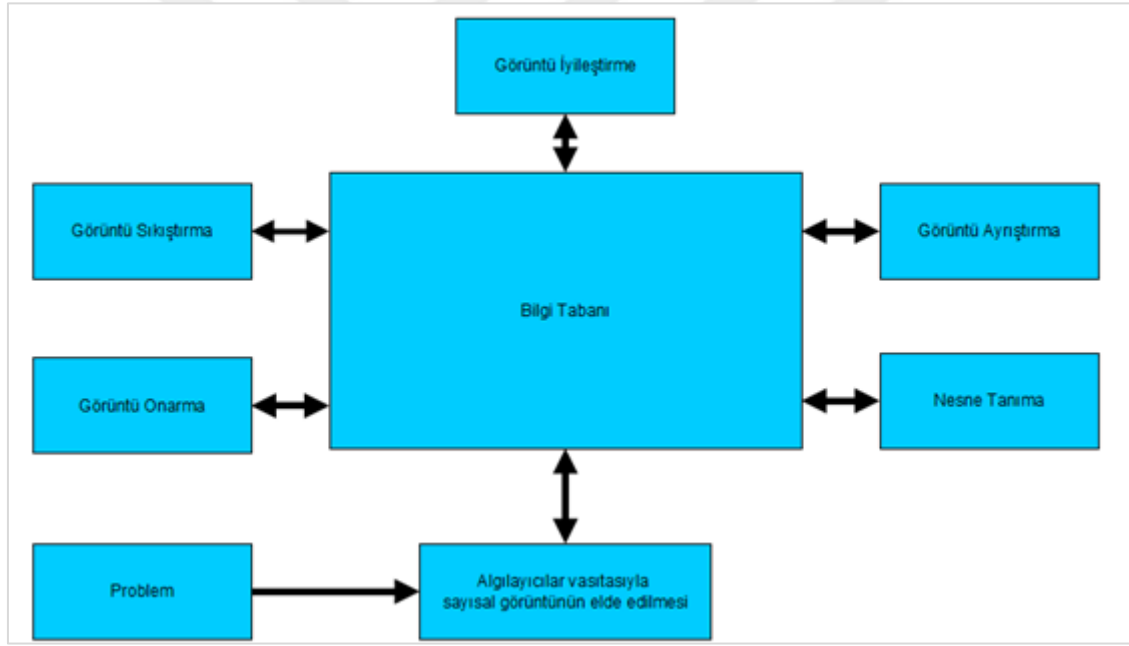
ayrı ayrı eşik değerlerinin hesaplanması ciddi bir sorun teşkil etmiştir. Renkli görüntülerin eşiklenmesine yönelik yeni bir yaklaşım Demirci ve arkadaşları [3] tarafından önerilmiştir. Her bir kanal için Otsu ve Kapur yöntemleri ile eşik değeri hesaplandıktan sonra renk uzayı alt küplere bölünmüş ve aynı küpte kalan pikseller benzer sınıfa dâhil edilmiştir. Geliştirilen metotta tek eşik kullanıldığından sekiz adet alt prizma meydana gelmiştir. Oluşturulan prizma sayısının az ve hacimlerinin büyük olmasından ötürü sınıflandırma için elde edilecek olan bölgelerdeki homojenlik düşük kalmıştır.

Bu çalışmada renkli görüntüler için çok seviyeli eşikleme esaslı yeni bir sınıflandırma yöntemi önerilmiştir. Görüntünün histogramı elde edildikten sonra ABC algoritması aracılığıyla her bir kanal için eşik değerleri ayrı ayrı hesaplanmıştır. Eşik değerlerinin elde edilme süreci Otsu ve Kapur fonksiyonlarının ABC algoritması ile maksimize edilmesi üzerine inşa edilmiştir. Geliştirilen algoritma ile eşik hesabının ardından Demirci ve arkadaşlarının önerdiği yöntem ile entegre edilerek renk uzayı bölümlenmiştir. Bu işlem sonucunda her bir alt küp veya prizmada kalan pikseller aynı sınıfa atanmıştır. Eşik sayısı arttıkça prizmanın hacimlerinin küçülerek homojenliğin arttığı görülmüştür. Böylece renkli görüntüler üzerinde sınıflandırma yapılarak sonuçlar elde edilmiştir.

Önerilen yöntem renkli görüntüler üzerinde çok seviyeli eşikleme yapılan yeni bir sınıflandırma yaklaşımı olup, gri ölçekli görüntülerde de sonuç vermektedir. Bununla birlikte geliştirilen metotta sınıflandırılması istenen görüntünün histogram bilgisi kullanıldığından dolayı görüntü boyutundan bağımsız çalışmaktadır. Bu sayede geleneksel yöntemlerde resim boyutunun artmasıyla yaşanan hesapsal maliyet problemi ortadan kaldırılmıştır. Bununla birlikte doğrusal Otsu ve Kapur yöntemlerinde ikiden fazla eşik seçiminde hesaplama maliyetinde yaşanan sıkıntı önerilen yöntem ile çözülmüş olup çok daha hızlı sonuçlar alınmıştır. Ayrıca kullanıcıdan bağımsız gerçek zamanlı bir yapıya uygun olarak tasarlanması en önemli avantajlarından bir tanesidir.

2. SAYISAL GÖRÜNTÜ İŞLEME

Sayısal görüntü işleme görüntü üzerinde belirli aşama ve işlemlerden meydana gelen süreçleri kapsayan bir bütün olarak tanımlanabilir. Söz konusu süreçler görüntünün algılayıcılar ile elde edilmesi, iyileştirilmesi, onarılması, sıkıştırılması, ayrıştırılması ve nihayetinde nesne tanıma işlemlerinden meydana gelmektedir. Sayısal görüntü işlemenin temel adımları Şekil 2.1’de ifade edilmiştir. İlgili blok diyagramında görüldüğü gibi nihai hedef olan nesne tanıma sürecinin performansı görüntü onarma, sıkıştırma ve ayrıştırma algoritmalarının başarısına bağlıdır. Günümüzde sayısal görüntü işleme tıp, endüstriyel uygulamalar, biyometrik sistemler, güvenlik, savunma sanayi gibi birçok alanda kullanılmaktadır.



Şekil 2.1. Görüntü işlemenin temel adımları [1]

2.1. Sayısal Görüntü

Sayısal görüntü $f(x,y)$ fonksiyonunun iki boyutlu düzlemdeki iz düşümüdür. Burada f nicel, pozitif ve sonlu bir değeri temsil etmekle birlikte o noktadaki yeğinlik değeri olarak ifade edilmektedir. Özetle $f(x,y)$ sıfırdan farklı ve sonlu olup Eş. 2.1’de belirtilmiştir [1].

$$0 < f(x,y) < \infty \quad (2.1)$$

$f(x,y)$ fonksiyonu aydınlatma ve yansıma bileşenleri ile birlikte bir bütündür. Aydınlatma bileşeni $i(x,y)$, yansıma bileşeni ise $r(x,y)$ olarak ifade edilmiştir. Söz konusu iki bileşen $f(x,y)$ 'yi meydana getirmek için Eş. 2.2'de sunulduğu üzere, çarpım işlemi ile birleşmektedir.

$$f(x, y) = i(x, y)r(x, y) \quad (2.2)$$

Burada

$$0 < i(x, y) < \infty \quad (2.3)$$

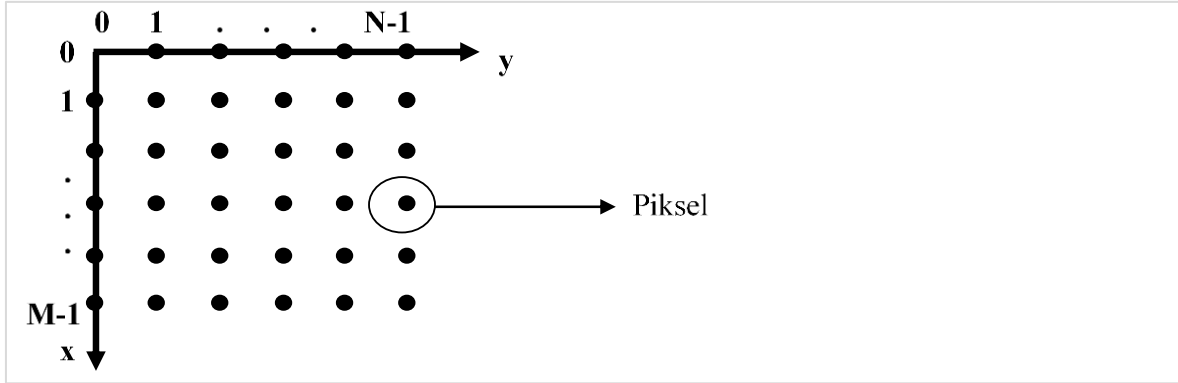
ve

$$0 < r(x, y) < 1 \quad (2.4)$$

olarak sınırlandırılmaktadır [1]. İfade edildiği üzere bir $f(x,y)$ görüntüsü sonlu sayıda bileşenlerden meydana gelmektedir ve her bir bileşenin kendine ait bir değeri vardır. Söz konusu bileşenlere piksel denir. Piksellerin uzamsal düzlem üzerinde kendine ait bir konumu ve değeri bulunmaktadır. (x,y) koordinatındaki bir pikselin koordinatları, $(x+1,y)$, $(x-1,y)$, $(x,y+1)$, $(x,y-1)$ ile verilen yatay ve dikey komşulara sahiptir. İkili bir görüntüde her piksel sadece 0 ve 1 değerlerini alabilir. Gri ölçekli ve renkli bir görüntüde ise her bir piksel görüntünün aydınlık derecesine göre 0 ve 255 arasında bir sayısal değere sahiptir. $f(x, y)$ fonksiyonu ile ifade edilen sayısal bir görüntünün, $M \times N$ 'lik matrisi ise $x = 0, 1, 2, \dots M-1$ ve $y = 0, 1, 2, \dots N-1$ olmak üzere, Eş. 2.5'de verilmiştir.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (2.5)$$

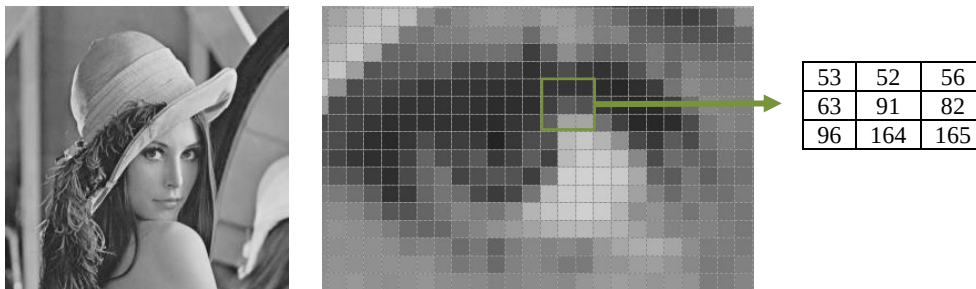
Sayısal bir görüntünün gösterimi ve piksellerden nasıl meydana geldiği Şekil 2.2'de gösterilmiştir.



Şekil 2.2. Sayısal görüntünün gösterimi [1]

2.2. Gri Ölçekli Görüntü

İkili bir görüntüyü oluşturan pikseller sadece siyah ve beyaz renklerden meydana geldiği için bir piksel ya 0 ya da 255 değerini alır. Siyah pikseller 0, beyaz pikseller ise 255 değeri ile ifade edilmektedir. Gri ölçekli bir görüntü ise resmin aydınlık değeri ile doğru orantılı olup 0 ve 255 arasında değer almaktadır. Bu bağlamda en koyu bölge olan siyah 0, en parlak bölge olan beyaz ise 255 ile gösterilir. Gri seviyeli bir resim bellek alanında 1 bayt yani 8 bitlik alan kaplamakta, dolayısıyla ($2^8 = 256$) farklı renk bilgisine sahip olabilmektedir. Şekil 2.3’de görüntü işleme alanında referans resimlerden biri olarak kabul edilen Lena görüntüsü ve görüntü üzerinde seçilen bir bölgedeki piksel, değerleri ile birlikte verilmiştir.



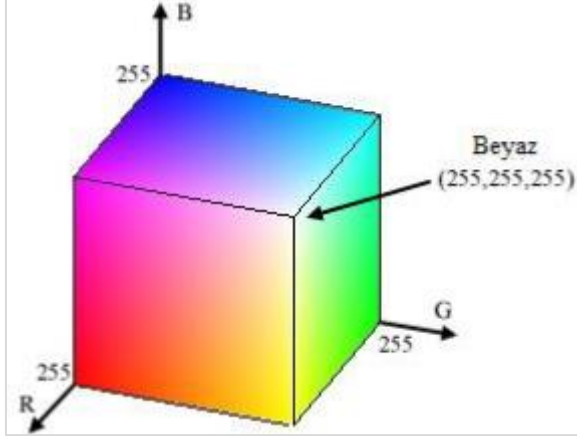
Şekil 2.3. Gri ölçekli Lena görüntüsü ve piksel değerleri

2.3. Renkli Görüntü

İnsanların ve hayvanların algıladığı renkler nesneden yansıyan ışık tarafından belirlenmektedir. Isaac Newton 1666 yılında güneş ışınlarının bir cam prizma içerisinden geçtiğini ve ortaya çıkan ışığın mor renkten kırmızı renge doğru giden bir aralıkta olduğunu keşfetti. Işık elektromanyetik dalgalardan meydana geliyor ve kırmızı en düşük frekansa

sahip renk, mavi ise en yüksek frekansa sahip renk olarak ifade ediliyordu. Bu ışık demetine spektrum adı verilmiştir. Renk spektrumu mor, mavi, yeşil, sarı, turuncu ve kırmızı renklerden oluşan altı ana bölge olarak ifade edilmiştir. Renkli görüntüleri standartlaştırmak adına renk uzayı veya renk sistemi tanımlamaları kullanılmıştır. Renk uzayı koordinat sisteminde her bir noktaya karşılık gelen renk değerinin gösterilmesidir [1].

Sayısal görüntü işleme alanında en sık kullanılan renk modelleri RGB (Red, Green, Blue), HSI (Hue, Saturation, Intensity) ve CMY (Cyan, Magenta, Yellow) dir. Renkli görüntüler üzerinde yapılan işlemlerde ise en sık kullanılan renk uzayı modeli RGB'dir. RGB kırmızı, yeşil ve mavi renk bileşenlerinden meydana gelmektedir. RGB renk uzayında her bir pikselin 8-bit renk derinliğine sahip olduğu düşünüldüğünde RGB renk uzayındaki toplam renk sayısı $(2^8)^3 = 16\,777\,216$ olacaktır. Şekil 2.4'de RGB renk uzayı gösterilmiştir. Renk uzayındaki her bir bileşen 0-255 arasında değer almaktadır. Üç rengin tamamının %100 oranında kullanılması durumunda beyaz, hiçbirinin kullanılmaması durumunda ise %0 yani siyah rengi elde edilmektedir [1].



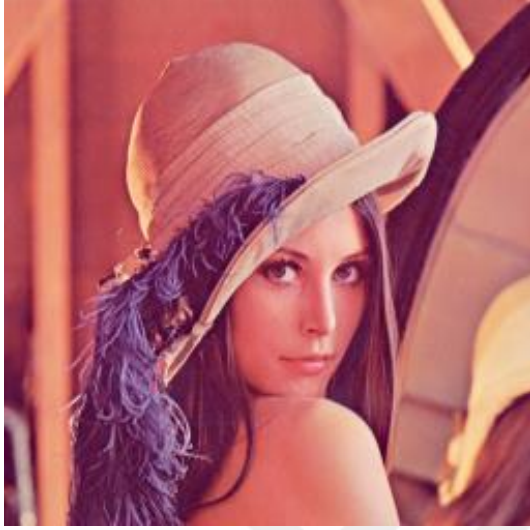
Şekil 2.4. RGB renk uzayı

2.4. Histogram

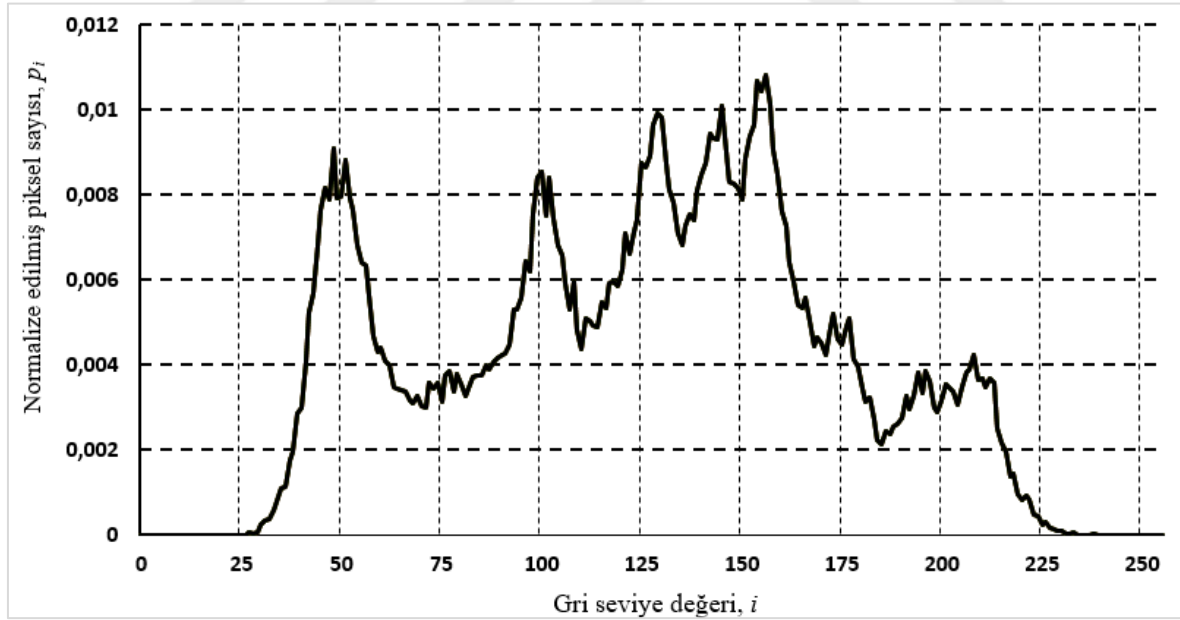
Histogram, $\{0, 1, 2, \dots, (L-1)\}$ yeğinlik değerine sahip bir sayısal görüntünün ayrık olasılık dağılım fonksiyonu olup, $M \times N$ boyutlu gri ölçekli bir resmin, i . gri seviyenin olasılığı,

$$p_i = h_i / (M \times N) \quad (2.6)$$

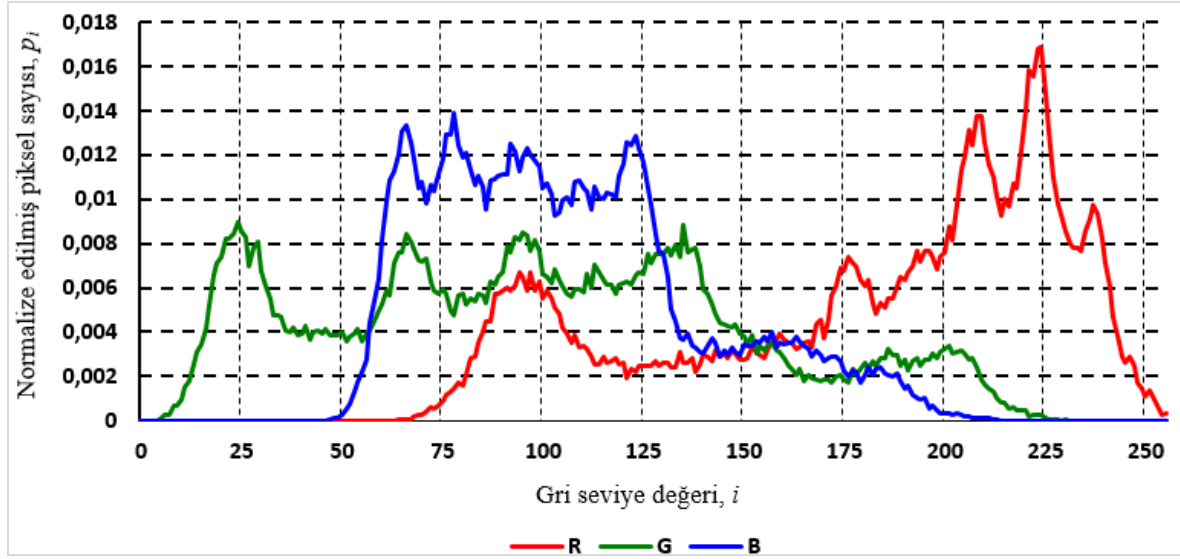
şeklindedir. Burada h_i , i . gri seviyesine sahip piksel sayısını ifade etmektedir. Şekil 2.5’de renkli Lena görüntüsü, Şekil 2.6’da gri ölçekli Lena görüntüsüne ait histogram grafiği, Şekil 2.7’de ise renkli Lena görüntüsüne ait histogram grafiği verilmiştir.



Şekil 2.5. Renkli Lena



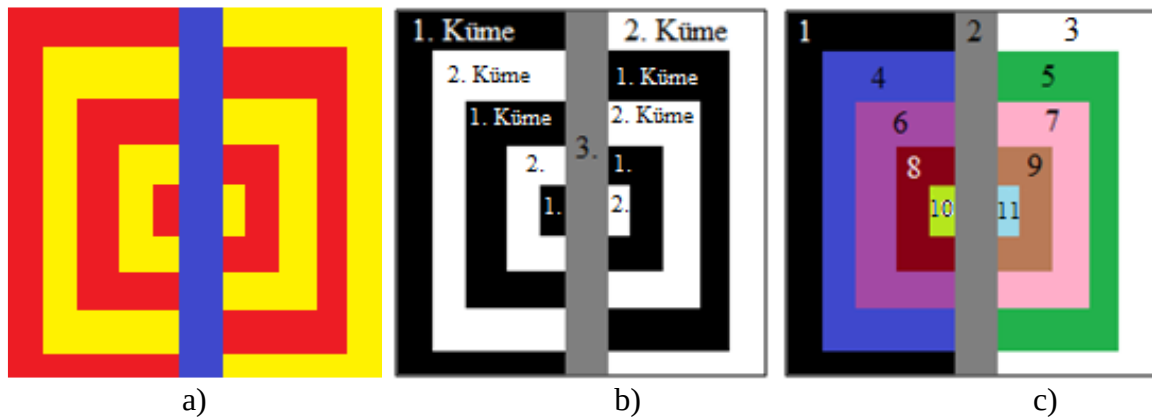
Şekil 2.6. Gri ölçekli Lena histogramı



Şekil 2.7. Renkli Lena histogramı

2.5. Görüntü Kümeleme ve Sınıflandırma

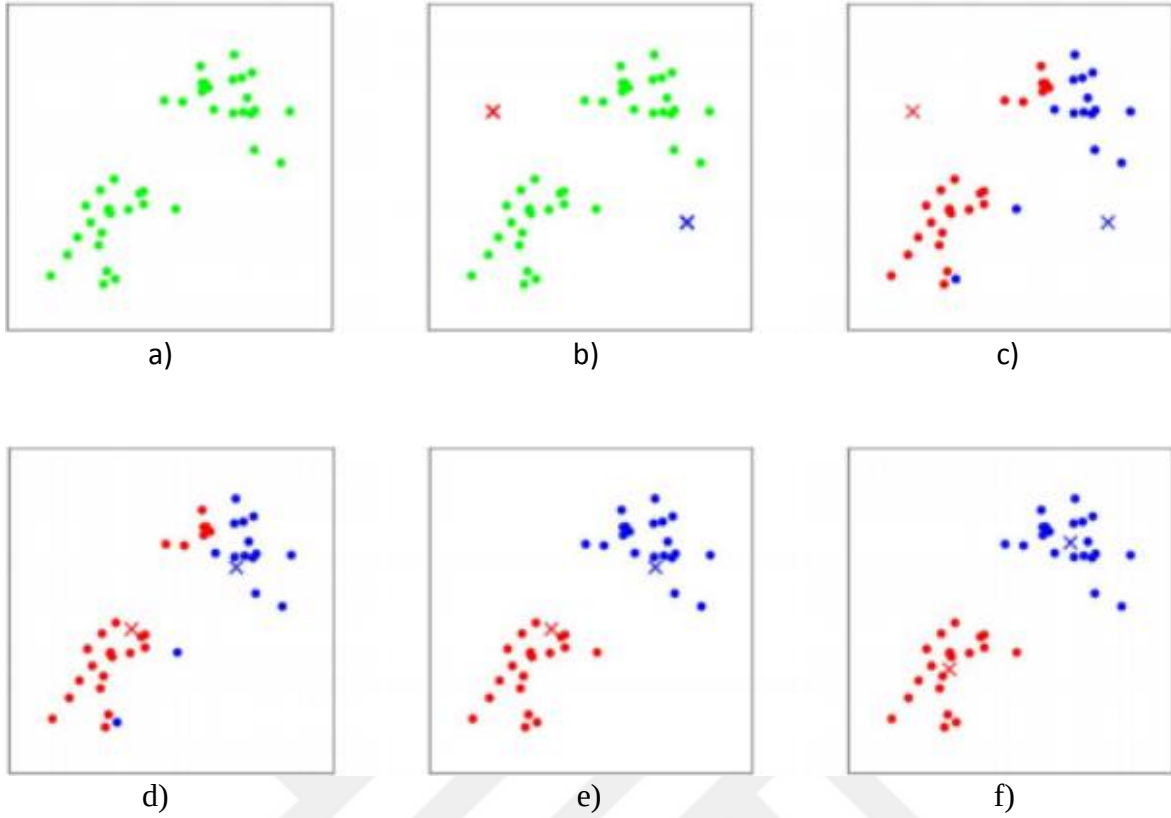
Görüntü ayrıştırma renk, ton, doku gibi belirleyici özelliklere göre görüntü üzerindeki farklı nesnelere ait piksellerin aynı küme altında anlamlı bölgeler halinde tasnif edilmesidir [44]. Görüntü ayrıştırma, nesneyi tanıma ve analiz etme basamaklarındaki en önemli adımlardan bir tanesidir. Görüntü kümeleme ile görüntü ayrıştırma birbirlerine çok yakın kavramlar olsa da aralarında temel farklar bulunmaktadır. Bir resimdeki nesnelerin uzamsal düzlem üzerindeki koordinatları gözardı edilip sadece renk bilgisi referans alınarak gruplandırıldığı takdirde bu işleme görüntü kümeleme veya sınıflandırma, konumsal koordinatlar ve komşuluklar dikkate alınarak gruplandırılır ise bu işleme görüntü ayrıştırma veya bölütleme denir. Sınıflandırma ve ayrıştırma işlemleri arasındaki fark Şekil 2.8’de gösterilmiştir.



Şekil 2.8. a) Sentetik görüntü b) Sınıflandırma c) Ayrıştırma [44]

Şekil 2.8(a)'da 3 farklı renkten meydana gelen orijinal görüntü bulunmaktadır. Şekil 2.8(b)'de ise sınıflandırma işlemi uygulanmış ve 3 farklı kümenin oluştuğu görüntü elde edilmiştir. Anlaşıldığı üzere uzamsal koordinatlar ve komşuluklar dikkate alınmadan benzer renk grupları içerisinde bulunan bölgeler aynı küme altında toplanmıştır. Şekil 2.8(c) incelendiğinde 11 farklı bölgenin oluştuğu görülmektedir. Burada uzamsal koordinatlar ve aynı renk grubu kriteri referans alınarak ayrıştırılma işleminin uygulandığı anlaşılmaktadır. Bununla birlikte aynı özelliklere sahip fakat komşuluğa sahip olmayan gruplar farklı bölgelerde yer almışlardır.

Kümeleme analizi bir veri kümesi içerisinde benzer özelliklere sahip veya benzer davranışlar sergileyen değişkenlerin ortak özellikleri referans alınarak aynı grup altında toplanmasıdır. Bu sayede anlamlı alt kümeler elde edilerek veri kümesi üzerinde işlem yapılabilir. Aynı veri kümesi içerisinde yer alan piksellerin benzerliklerinin fazla, farklı veri kümeleri arasındaki benzerliklerin ise az olması hedeflenir. Literatürde kümeleme çalışmaları için geliştirilmiş birçok yöntem bulunmaktadır. Bu yöntemler arasında en fazla kullanılanları K-ortalamlar, FCM, Gauss ve hiyerarşik kümeleme yaklaşımlarıdır. K-ortalamlar [5] en basit ve denetimsiz kümeleme algoritmasıdır. Burada K küme sayısını ifade eder. Metot önceden belirlenmiş bir veri kümesini k adet küme altında gruplandırma prensibine dayanır. Küme merkezi küme sayısı kadar olacak şekilde rastgele oluşturulur. Daha sonra tüm piksellerin küme merkezlerine olan uzaklığı Öklid formülü ile hesaplanır. Küme merkezine yakın olan pikseller o kümeye atanırken, geri kalan pikseller diğer kümeye atanır. Böylece bir piksel ya 0 ya da 1 kümesine dâhil olur. Amaç fonksiyonu minimize olana kadar bu işlem tekrar edilir. K-ortalamlar yöntemi kullanılarak bir kümeleme işleminin nasıl gerçekleştiği Şekil 2.9'da gösterilmiştir.



Şekil 2.9. K-ortalamlar örneği a) Kümelenmesi istenen veri seti b) Merkezlerin rastgele ya da bir yönteme bağlı olarak seçilmesi c) 1. İterasyon d) 2. İterasyon e) 3. İterasyon f) Kümleme [45]

K-ortalamlar algoritması çok keskin bir yöntemdir. Bir verinin ya da pikselin sadece 1 kümeye ait olmasına imkân tanınır. Söz konusu probleme çözüm üretebilmek adına K-ortalamlar metodu iyileştirilerek FCM [6] algoritması geliştirilmiştir. FCM yönteminde bulanık küme mantığı kullanılarak her bir veri setinin elemanına 0 ve 1 arasında değişen üyelik değerleri atanır. Böylece bir verinin birden fazla sınıfa dâhil olmasına olanak tanınmış olur. Nesne hangi küme merkezine daha yakınsa o sınıfa atanır. FCM yönteminin amaç fonksiyonu Eş. 2.7’de verilmiştir [46].

$$J = \sum_{i=1}^n \sum_{k=1}^c u_{ik}^a |d_i - v_k|^2 \quad (2.7)$$

Eş. 2.7’de a , bulanıklığı,

u_{ik}^a ; k . küme için d_i ’nin üyelik derecesini,

d_i ; i . pikseli,

v_k ; k . kümenin merkezini temsil etmektedir.

Ayrıca n ve c sırayla piksel ve küme sayısıdır.

$|d_i - v_k|^2$; i . piksel ile k . küme merkezi arasındaki uzaklıktır.

u_k ve u_{ik} değerleri ise

$$v_k = \frac{\sum_{i=1}^n u_{ik}^a d_i}{\sum_{i=1}^n u_{ik}^a} \quad (2.8)$$

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{|d_i - v_k|}{|d_i - v_j|} \right)^{\frac{2}{a-1}}} \quad (2.9)$$

şeklinde hesaplanır.

Gerek K-ortalamalar, gerekse FCM yöntemleri görüntü ayrıştırma alanında sıklıkla kullanılan iki yaklaşımdır. Söz konusu yöntemler her ne kadar güçlü olsa da hesapsal maliyetleri görüntünün boyutuna bağlı kalmaktadır. Böylece gerçek zamanlı uygulamalarda yetersiz kalmakta ve verimli olmamaktadırlar. Bununla birlikte bu yapılarda küme merkezlerinin rastgele seçilmesi ya da belirlenmesi ve küme sayısının kullanıcı bağımlı parametre olması ciddi bir dezavantaj teşkil etmektedir.

2.6. Görüntü Eşikleme

Eşikleme, resimlerin gri seviye bilgilerini önceden belirlenmiş eşik değerleri baz alınarak aynı sınıfa atayan görüntü sınıflandırma yöntemlerinden bir tanesidir. Eşikleme gri ölçekli görüntüler üzerinde iki seviyeli ve çok seviyeli olarak uygulanabilmektedir. T eşik değeri, $f(x,y)$ iki boyutlu bir resim, Eş. 2.10'da $b(x,y)$ ayrıştırılan görüntü olmak üzere, 1 ile atanan bölgeler nesne, 0 ile atanan bölgeler ise arka plan noktası olacaktır [1].

$$b(x, y) = \begin{cases} 1 & \text{eğer } f(x, y) > T \\ 0 & \text{eğer } f(x, y) \leq T \end{cases} \quad (2.10)$$

Gri ölçekli bir görüntüde histogram hesabının ardından m adet eşik bulunması istenilen durumda $m + 1$ adet sınıf meydana gelecektir. Bu hedefe uygun olarak m değişkenli amaç fonksiyonunun maksimum yapan değer eşikleri temsil etmektedir. Otsu ve Kapur yaklaşımları bu amaçla tercih edilen eşikleme yöntemlerinin başında gelmektedir. Otsu yönteminde sınıflarası varyans, Kapur yönteminde ise entropi tabanlı amaç fonksiyonunu maksimum yapan değer hesaplanır. Kapur yönteminde tek eşik bulmak için Eş. 2.11'deki amaç fonksiyonu kullanılmaktadır.

$$J(t_1) = H_0 + H_1 \quad (2.11)$$

Burada H_0 ve H_1 histogramın kısmi entropilerini ifade etmekte olup

$$H_0 = - \sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{t_1-1} p_i \quad (2.12)$$

$$H_1 = - \sum_{i=t_1}^{L-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \quad \omega_1 = \sum_{i=t_1}^{L-1} p_i \quad (2.13)$$

şeklinde hesaplanır. Eş. 2.11'de t_1 değişkeni amaç fonksiyonunu maksimum yapan değeri temsil etmektedir. Diğer yandan Otsu yönteminde sınıflararası varyans tekniği ile amaç fonksiyonu Eş. 2.14'deki gibi hesaplanır.

$$J(t_1) = \sigma_0 + \sigma_1 \quad (2.14)$$

Eşitlikte her bir sınıfın varyans değeri ise

$$\sigma_0 = \omega_0 (\mu_0 - \mu_T)^2 \quad (2.15)$$

$$\sigma_1 = \omega_1 (\mu_1 - \mu_T)^2 \quad (2.16)$$

şeklindedir. Kümelerin kısmi olasılıkları ve ortalamaları ise

$$\omega_0 = \sum_{i=0}^{t_1-1} p_i, \mu_0 = \sum_{i=0}^{t_1-1} \frac{i p_i}{\omega_0} \quad (2.17)$$

$$\omega_1 = \sum_{i=t_1}^{L-1} p_i, \mu_1 = \sum_{i=t_1}^{L-1} \frac{i p_i}{\omega_1} \quad (2.18)$$

$$\mu_T = \sum_{i=0}^{L-1} i p_i \quad (2.19)$$

şeklindedir. Burada μ_T görüntünün ortalama yoğunluğu ifade etmekte olup

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T \text{ ve } \omega_0 + \omega_1 = 1 \quad (2.20)$$

eşitliği mevcuttur. Kapur ve Otsu yöntemleri çok seviyeli eşikleme için genişletilebilmektedir. Kapur yönteminde çok seviyeli eşikleme formülü

$$J(t_1, t_1, \dots, t_m) = H_0 + H_1 + H_2 + \dots + H_m \quad (2.21)$$

$$H_0 = - \sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \omega_0 = \sum_{i=0}^{t_1-1} p_i \quad (2.22)$$

$$H_1 = - \sum_{i=t_1}^{t_2-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \omega_1 = \sum_{i=t_1}^{t_2-1} p_i \quad (2.23)$$

$$H_2 = - \sum_{i=t_2}^{t_3-1} \frac{p_i}{\omega_2} \ln \frac{p_i}{\omega_2}, \omega_2 = \sum_{i=t_2}^{t_3-1} p_i \quad (2.24)$$

⋮

$$H_m = - \sum_{i=t_m}^{L-1} \frac{p_i}{\omega_m} \ln \frac{p_i}{\omega_m}, \omega_m = \sum_{i=t_m}^{L-1} p_i \quad (2.25)$$

şeklinde olup, Eş. 2.21’i maksimum yapan $t_1, t_2, \dots, t_m$ değerleri optimal eşik değerlerini temsil etmektedir. Böylece optimal çok seviyeli eşikleme, m boyutlu bir optimizasyon problemi olarak ifade edilebilir. Otsu yönteminde ise çok seviyeli eşikler

$$J(t_1, t_2, \dots, t_m) = \sigma_0 + \sigma_1 + \sigma_2 + \dots + \sigma_m \quad (2.26)$$

$$\sigma_0 = \omega_0 (\mu_0 - \mu_T)^2 \quad (2.27)$$

$$\sigma_1 = \omega_1 (\mu_1 - \mu_T)^2 \quad (2.28)$$

$$\sigma_2 = \omega_2 (\mu_2 - \mu_T)^2 \quad (2.29)$$

⋮

$$\sigma_m = \omega_m (\mu_m - \mu_T)^2 \quad (2.30)$$

$$\omega_0 = \sum_{i=0}^{t_1-1} p_i, \mu_0 = \sum_{i=0}^{t_1-1} \frac{i p_i}{\omega_0} \quad (2.31)$$

$$\omega_1 = \sum_{i=t_1}^{t_2-1} p_i, \mu_1 = \sum_{i=t_1}^{t_2-1} \frac{i p_i}{\omega_1} \quad (2.32)$$

$$\vdots$$

$$\omega_m = \sum_{i=t_m}^{L-1} p_i, \mu_m = \sum_{i=t_m}^{L-1} \frac{i p_i}{\omega_m} \quad (2.33)$$

şeklinde genişletilerek hesaplanmaktadır [21]. Şekil 2.10'da Lena görüntüsü üzerinde tek seviyeli eşikleme işlemi Kapur ve Otsu metotları kullanılarak uygulanmış ve sonuçlarıyla birlikte verilmiştir.



Şekil 2.10. Eşikleme sonuçları a) Kapur, $t=123$ b) Otsu, $t=117$

Lena görüntüsü üzerinde Kapur ve Otsu metotları kullanılarak yapılan iki seviyeli eşikleme sonuçları Şekil 2.11'de gösterilmiştir.



Şekil 2.11. Eşikleme sonuçları a) Kapur, $t_1=98$, $t_2=164$ b) Otsu, $t_1=93$, $t_2=151$

2.7. Yapay Arı Koloni Algoritması

Optimizasyon en iyileme anlamına gelmektedir. Belirli sınırlar içerisinde kalmak koşuluyla bir amaç fonksiyonunu minimize veya maksimize eden parametrelerin elde edilmesi optimizasyon problemi olarak tanımlanabilir. Optimizasyon problemlerinde en küçük değer bulunması istenen durumda en düşük maliyete sahip değer, en yüksek değer bulunması istenen durumda ise en büyük değer optimum çözümdür.

Optimizasyon teknikleri son yıllarda birçok problemin çözümünde faydalanan yaklaşımların başında gelmektedir. En fazla tercih edilen optimizasyon yöntemleri ise ısıtım işlem algoritması, genetik algoritmalar, karınca koloni algoritması, parçacık sürü optimizasyonu ve yapay arı koloni algoritmalarıdır. Yapay arı koloni algoritması 2005 yılında Karaboğa [47] tarafından gerçek hayattaki arıların davranışlarını inceleyerek modellediği bir sürü zekâsı optimizasyon yöntemidir.

Karıncalar, kuşlar, balıklar gibi sürü halinde hareket eden canlılar, yeteneklerinin yardımıyla içgüdüsel olarak bir iş planı içerisinde hareket ederler. İş planı ve kendi kendine organize olabilme söz konusu yeteneklerin en önemli özelliği olup sürü zekâsı olarak adlandırılır. Arılar da sürü zekâsı mantığıyla hareket eden canlılardır. Öyle ki yiyecek arama, koloniyi savunma ve keşif gibi eylemler bir iş bölümü içerisinde kurgulanır. Böylece doğal bir arı kolonisinde her arı için tahsis edilmiş bir görev bulunmaktadır. Tereshko [48] 2000 yılında reaksiyon difüzyon denklemlerine dayalı bal arısı kolonisinin yiyecek arama davranışının matematiksel bir modelini geliştirmiştir. Önerilen yaklaşımda üç temel bölüm bulunmaktadır. Bunlar yiyecek kaynakları, görevli işçi arılar ve görevsiz işçi arılardır.

Yiyecek kaynakları

Arıların bal yapmak için gittikleri kaynaklardır. Bir yiyecek kaynağının değeri kovana yakınlığı, miktarı, çeşidi, elde edilmesindeki kolaylık ve zorluk, nektarın kalitesi gibi birçok etmene bağlıdır. Fakat kolaylık açısından nektarın kalitesi tek bir ölçüt olarak ele alınabilir.

Görevli işçi arılar

Görevli işçi arılar ise daha önceden belirlenmiş kaynakların kovana getirilmesinden ve belirli bir gıda kaynağı hakkındaki bilgiyi diğer arılar ile paylaşma sorumluluğundadır. Söz konusu bilgiler nektarın kalitesi, miktarı ve kovana olan uzaklığıdır.

Görevsiz işçi arılar

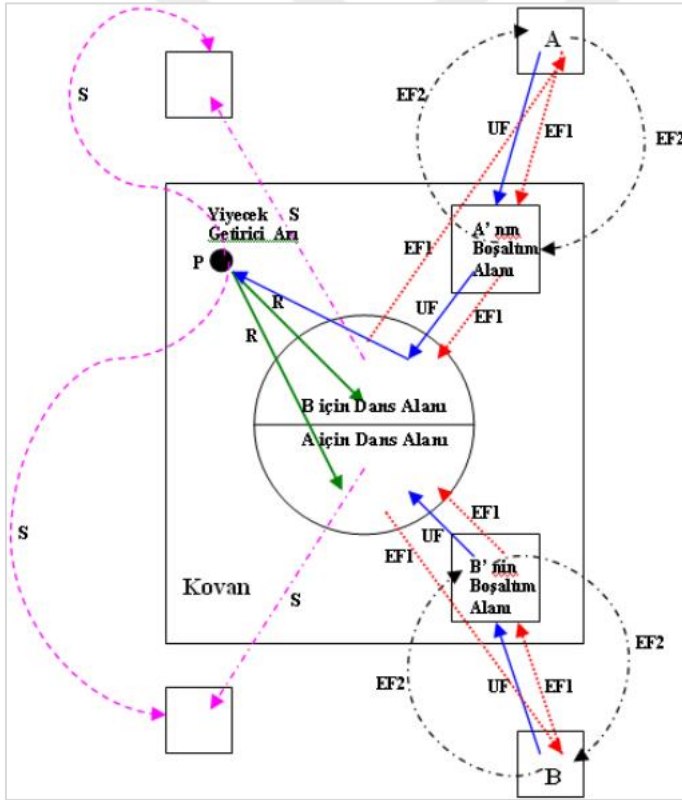
Görevsiz işçi arılar kaynak bulma hedefi içerisinde olan topluluklardır. İki çeşittir: herhangi bir etmene bağlı olmaksızın dürtülerini kullanarak arayış içerisinde olan arılar ve işçi arıların paylaştığı bilgiyi kullanarak kaynağa yönelen arılar. Kâşif arıların tüm koloninin ortalamasına oranı %5 - %10 arasındadır.

Arılar arasında haberleşme

Arılar arasındaki bilgi paylaşımı sürü olarak hareket etmenin en önemli gerekliliklerinden bir tanesidir. Bilgi paylaşımı ne kadar doğru yapılırsa iş bölümü ve nektarın elde edilme süreçleri de bir o kadar doğru ve hızlı olacaktır. Nektarın kalitesi ve bulunduğu konum ile alakalı paylaşımlar ise dans pisti kullanılarak yapılır. Bilgi paylaşımında bulunacak olan arı dans eder, diğer arılar da antenlerini kullanarak nektar ile alakalı bilgileri edinirler. Bu bilgiler tat, koku ve kalite gibi bileşenlerden meydana gelmektedir. Bilgiyi paylaşan arının amacı ise nektara daha fazla arıyı çekmektir. Bu işlemin ardından tekrardan kaynağa yönelir [22].

Nektar konumuna ulaşmak için bilgi paylaşımında bulunan arının yön bilgisi paylaşması gerekmektedir. Yön bilgisinin elde edilmesinin ardından güneşten faydalanılarak konum ve güneş arasındaki açı aracılığıyla yiyecek kaynağına ulaşılır. Yiyecek kaynağının kovana olan uzaklığına göre dairesel dans, kuyruk dansı ve titreme dansı çeşitleri bulunmaktadır.

Dairesel dansa kaynağın kovana olan uzaklığı maksimum 100 metredir. Bu dans türünde uzaklık ve yön bilgisi paylaşılmamaktadır. Titreme dansında ise bilgi paylaşımında bulunan arı, elindeki kaynağın zengin olduğunu fakat bu kaynağın mevcut kovana fazla geleceği bilgisini kovanın hemen hemen her bölgesinde işler. Bu dansın amacı kovan kapasitesi ve kaynak arasındaki ilişkiyi koordine etmektir. 100 metreden 10 kilometreye kadar olan nektarın belirtilmesinden ise 8 rakamını andıran kuyruk dansından faydalanılır. Her 15 saniyede bir tekrarlanan bu dans türünde yiyecek kaynağının konum ve uzaklığı hakkında bilgi verilir. Az tekrar nektarın uzakta olduğunu ifade etmektedir. Gözcü arılar bu dansları analiz ederek hedef nektar kaynağını belirler. Buradaki tercih kaynağın zenginliği ile doğru orantılı olmaktadır [49]. Şekil 2.12’de doğal bir arı kolonisindeki yiyecek arama çevrimi gösterilmiştir.



Şekil 2.12. Yiyecek arama çevrimi [22]

Şekil 2.12’de A ve B iki nektar kaynağı olarak düşünülürse, bu durumda kaynakların konumundan haberdar olmayan bir arı için şu durumlar mevcuttur:

- İçgüdüsel etmenlerle kaynak arayışında olan kâşif arı olabilir. Şekil 2.12’de S ile ifade edilmektedir.

- Dansla elde ettiği rotaya yönelen bir gözcü arı olabilir. Şekil 2.12’de R ile gösterilmiştir.

Yiyecek kaynağının tespit edilmesinin ardından nektar toplanır ve kovana dönülür. Kaynağın kovana aktarılmasının ardından üç seçenek söz konusudur:

- Kaynak terk edilerek bağımsız izleyici olunur. Şekil 2.12’de UF ile gösterilmiştir.
- Kovandaki diğer arılar nektara yönlendirilir. Şekil 2.12’de EF1 ile gösterilmiştir.
- Tek başına kaynağa ilerlenebilir. Şekil 2.12’de EF2 ile gösterilmiştir.

Arıların kendi kendilerine organize olabilmeleri dürtülerinde barındırdığı dört temel özellik ile mümkün olmuştur. Bunlar nektarın miktarı arttıkça gözcü arı sayısının artması, biten kaynağın terk edilmesi, rastgele kaynak arayışı yapmaları ve dans alanındaki bilgi paylaşımlarıdır [22].

Yapay arı koloni algoritması Karaboğa [47] tarafından doğada var olan arıların davranışlarını inceleyerek tasarladığı bir optimizasyon yöntemidir. Karaboğa model üzerinde basitlik amacıyla bazı düzenleme ve kabuller sunmuştur. Her bir yiyecek kaynağı tek bir görevli arı sorumluluğundadır. İşçi arıların sayısının gözcü arıların sayısına eşit olduğu varsayılmaktadır. Bununla birlikte tükenen kaynağın arısı kâşif arı olacaktır. Yapay arı koloni algoritmasında besin kaynağı çözüme tekabül edeceğinden, arama uzayında amaç fonksiyonu en fazla besine sahip olan konum olacaktır. Yapay arı koloni algoritması şu adımlardan meydana gelecektir:

Adım 1. Başlangıç yiyecek kaynağı bölgelerinin üretim süreci,

repeat

Adım 2. İşçi arıların yiyecek kaynağına gönderilmesi ve nektar miktarının hesaplanması,

Adım 3. Olasılık değerlerinin görevli arılardan gelen bilgiye göre hesaplanması,

Adım 4. Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmeleri,

Adım 5. Tükenen kaynakların terk edilişi ve kâşif arı süreci,

until(Çevrim sayısı = Maksimum çevrim sayısı)

Başlangıç besin kaynaklarının üretimi

Başlangıç yiyecek kaynaklarının bulunduğu bölgeler kovan çevresine tekabül eder. Yiyecek kaynaklarının üretimi rassal bir şekilde alt ve üst sınırlar arasında Eş. 2.34'deki şekilde gerçekleştirilir.

$$X_{ij} = X_j^{\min} + rand(0,1)(X_j^{\max} - X_j^{\min}) \quad (2.34)$$

Eşitlikle i nektar kaynağı indisi ($i = 1, \dots, SN$ olmak şartıyla), j optimize edilecek parametre indisi ($j = 1, \dots, m$ olmak şartıyla), X ise başlangıç çözüm değerini ifade etmektedir. $X_j^{\min}$, j . parametrenin alt sınırını, $X_j^{\max}$ ise j . parametrenin üst sınırını temsil eder. Başlangıç yiyecek kaynaklarının belirlenmesinin ardından işçi, gözcü ve kâşif arı süreçleri tetiklenerek daha iyi besin kaynakları tespit edilmeye çalışılır [22].

İşçi arıların besin kaynaklarına gönderilmesi

Başlangıç aşamasında yiyecek kaynaklarının rastgele üretiminin ardından bu kaynakların değerlendirilmesi gerekmektedir. Her bir nektar kalitesinin değerlendirilmesi amacıyla işçi arılar kaynaklara gönderilir. Bununla birlikte besin kaynaklarının komşu konumlarındaki nektarın kalitesi de değerlendirilir. Eğer değerlendirilen kaynak daha kaliteli ise yeni kaynak hafızada tutulur. $\vec{X}_i$ ile gösterilen her bir besin kaynağının j nolu parametresi değiştirilerek, Eş. 2.35'de gösterilen $\vec{X}_i$ komşuluğundaki $\vec{V}_i$ kaynağı bulunur. Burada j rastgele üretilen bir tamsayıyı temsil eder. ($j=1, \dots, m$ aralığında olmak şartıyla) [22].

$$V_{ij} = X_{ij} + \theta_{ij}(X_{ij} - X_{kj}) \quad (2.35)$$

Eşitlikte θ_{ij} , $[-1,1]$ aralığında rassal değer almaktadır. Rastgele seçilen komşu çözüm $\vec{X}_k$ değerinin ($k=1, \dots, SN$ olmak şartıyla) j . parametresi ile mevcut kaynağın j . parametresi θ_{ij} ile ağırlıklandırılır. X_{ij} ve X_{kj} çözümleri birbirine yaklaştıkça değişim miktarı da azalacak

olup, en iyi çözüme yakınlaşılacaktır. V_{ij} değerinin hesaplanmasının ardından alt ve üst parametre sınırlarının dışına çıkılması durumunda, öteleme işlemi j . parametre sınırlarına göre Eş 2.36'da gösterildiği gibi yapılır [22].

$$V_{ij} = \begin{cases} X_j^{\min}, & V_{ij} < X_j^{\min} \\ V_{ij}, & X_j^{\min} \leq V_{ij} \leq X_j^{\max} \\ X_j^{\max}, & V_{ij} > X_j^{\max} \end{cases} \quad (2.36)$$

Sınırlara göre elde edilen $\vec{V}_i$ vektörü yeni kaynağı ifade eder ve kalitesinin uygunluk değeri Eş. 2.37'de gösterildiği şekilde hesaplanır [22].

$$fitness_i = \begin{cases} 1/(1+f_i), & f_i \geq 0 \\ 1+|f_i|, & f_i < 0 \end{cases} \quad (2.37)$$

Eşitlikte, f_i , V_i kaynağının maliyet değerini, $fitness_i$ ise besin uygunluk değerini temsil etmektedir. $\vec{X}_i$ ve $\vec{V}_i$ arasında nektar kaynağına göre seçim işlemi uygulanır. Yeni $\vec{V}_i$ çözümü daha iyi ise hafızaya alınır. Aksi takdirde görevli arılar X_i kaynağında çalışmaya devam ederler. Bu durumda kaynağın geliştirilememe sayacı $failure_i$ değeri bir artar, aksi takdirde bu değer sıfırlanır [22].

Gözcü arıların olasılık değerlerinin hesaplanması

Görevli arılar nektar ile alakalı işlemleri tamamladıktan sonra kovana dönerek besin kaynağının miktarı ile alakalı gözcü arılara bilgi verirler. Gözcü arılar bu bilgilere göre bir seçme işlemi uygular.

Yapay arı koloni algoritmasında bu seçim rulet tekerleği yöntemi kullanılarak Eş. 2.38'deki gibi yapılmaktadır. Tekerlekteki her bir dilim uygunluk değeri ile doğru orantılı olacağından büyük dilimin seçilme olasılığı daha yüksek olacaktır [22].

$$q_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} \quad (2.38)$$

Burada $fitness_i$ i. kaynağın kalitesini ifade ederken, SN görevli arı sayısını temsil etmektedir [22].

Gözcü arıların yiyecek kaynağı

Olasılık değerlerinin hesaplanmasının ardından her kaynak için $[0,1]$ aralığında rassal değişkenler üretilir. Yeni çözümün kalitesi hesaplanır daha iyi olan hafızada tutulur ve kaynağın geliştirilememe sayacı sıfırlanır. Eğer eski çözüm daha iyi ise bu değer arttırılır. Tüm gözcü arılar besin kaynaklarına gidene kadar bu süreç devam ettirilir [22].

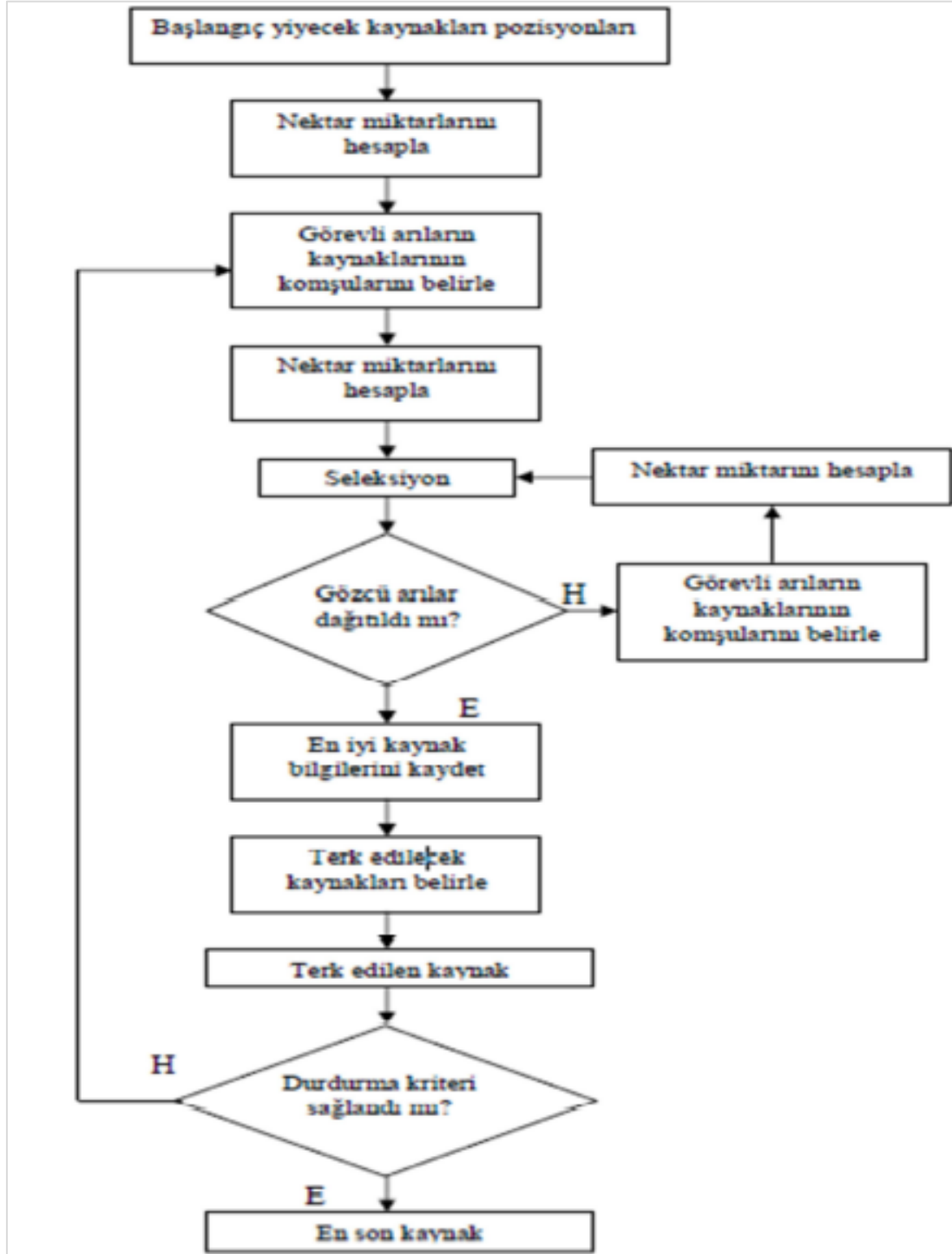
Tükenen kaynağın terk edilişi ve kâşif arı üretimi

Bir çevrimin ardından kaynağın geliştirilememe sayacı kontrol edilerek besin kaynağının tükenip tükenmediği bilgisi elde edilir. Kaynağın tükendiği durumda görevli arı artık kâşif arıya dönüşür. Bu sebeple yiyecek kaynağının tükenip tükenmediğinin belirlenmesi önem teşkil etmektedir. Yapay arı koloni algoritmasında bu bilgi limit olarak ifade edilir [22].

Yapay arı koloni algoritması sürü zekâsı tabanlı, basit ve esnek bir yapıda olup, matematiksel karmaşıklık içermez. Bununla birlikte yaygın olarak kullanılan optimizasyon algoritmaları içerisinde en az kontrol parametresine sahip yaklaşımlardan bir tanesidir. Ayrıca kullanılan seleksiyon teknikleri ve komşu üretme mantığı sayesinde bölgesel araştırmalar için tatminkâr sonuçlar üretmektedir. Yapay arı koloni algoritması performans açısından ciddi bir avantaja sahip olup, başlangıç çözümlerinin değişkenliğinden çok fazla etkilenmemektedir. Söz konusu yöntem nümerik problemler için tasarlanmış olmasına rağmen ayrık problemlere de uygulanabilmektedir [22]. ABC algoritması ile diğer optimizasyon algoritmaları karşılaştırıldığında gerek hesapsal açıdan gerekse üretilen sonuçların doğruluğu açısından başarılı olduğu çalışmalarla kanıtlanmıştır.

Yapay arı koloni algoritmasının temel adımları

Yapay arı koloni algoritmasının temel adımları Şekil 2.13’de verilmiştir.



Şekil 2.3. Yapay arı koloni algoritması akış diyagramı [49]



3. RENKLİ GÖRÜNTÜLERİN ÇOK SEVİYELİ EŞİKLENMESİ

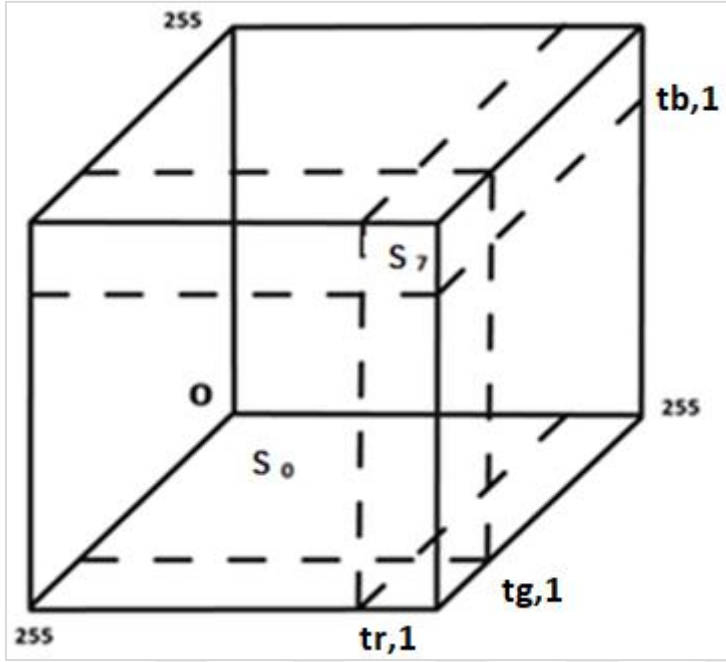
Daha önce bahsedildiği gibi gri seviyeli görüntülerin çok seviyeli eşiklenmesi Otsu ve Kapur yöntemleri kullanılarak yapılabilmektedir. Fakat mevcut yöntemler renkli görüntüler üzerinde verimsiz ve yetersiz kalmakta olup beklentileri karşılayamamaktadır. Bu sebeple renkli görüntülerin tek ve çok seviyeli eşiklenmesine yönelik yeni bir metod geliştirilmiştir. Geliştirilen yöntemde renk uzayı modeli ile ABC algoritması birleştirilmiştir.

3.1. Renk Uzayının Tek Eşikli Bölütlenmesi

Detayları bölüm 2.6’da verilen Otsu ve Kapur metotlarıyla gri ölçekli görüntülerin ikili veya çoklu kümeleme işlemi yapılabilmektedir. Bununla birlikte renkli görüntüler üç kanaldan meydana gelmekte ve her bir kanal kendi eşik değerine sahiptir. Her bir kanal için eşikler hesaplanmış olsa bile, söz konusu bilgilerin anlamlı sınıflar oluşturmak için kullanımı önemli bir problem teşkil etmektedir. Diğer yandan Demirci ve arkadaşları [3] tarafından geliştirilen yöntemde her kanaldan elde edilen tekli eşik bilgileri Şekil 3.1’de gösterildiği gibi birleştirilmiştir. Başka bir ifadeyle renk küpü küçük küplere veya prizmalara bölünmüştür. Bu işlemin ardından alt küp veya prizmanın içerisinde yer alan her piksel aynı sınıfa dâhil edilmiştir. Şekil 3.1 incelediğinde alt küp veya prizmaların hacimlerinin her kanalın eşikğine bağlı olduğu ve aynı olmadığı açıktır. Dolayısıyla alt küplerin veya prizmaların hacimleri resimdeki piksel yoğunluklarının dağılımıyla ilişkili olacaktır. Bununla birlikte büyük hacimlere sahip kümelerin homojenliği küçük hacimli kümelere göre daha düşük olacaktır. Eşik sayısı artırılarak her bir küpün hacmi azaltılır ise oluşacak sınıfların homojenliği de artacaktır. Bu sayede çok seviyeli eşikleme önemli bir çözüm haline gelecektir. Diğer yandan, renkli görüntü için elde edilecek maksimum sınıf sayısı

$$S_{\max} = (m+1)^3 \quad (3.1)$$

şeklinde olacaktır. Renk uzayının tek eşikli bölünmesi Şekil. 3.1’de gösterilmiştir. Aynı yaklaşım çok seviyeli eşikler için genişletilebilir.

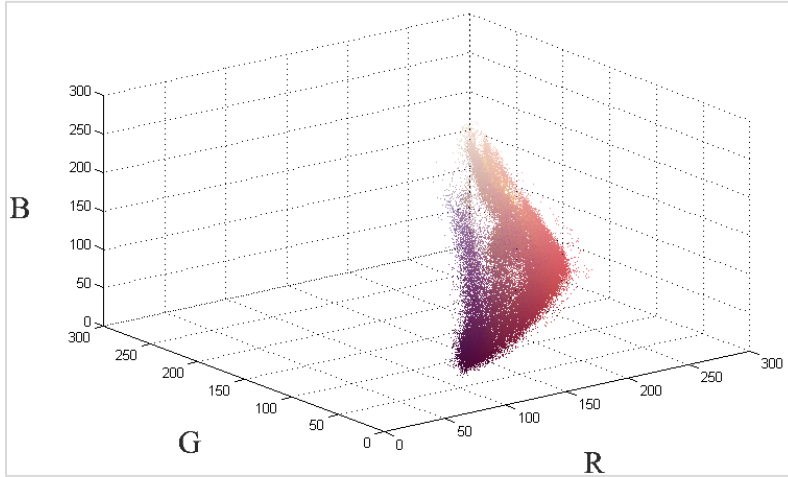


Şekil 3.1. Üç boyutlu renk uzayı ve atanan sınıflar $m = 1$ [3]

Tek eşikle renk uzayının bölünme kuralları ve oluşacak sınıf etiketleri Çizelge 3.1’de verilmiştir. Eş 3.1’e göre tek eşik kullanıldığından ($m=1$) sekiz alt küme oluşacaktır. Eşik sayısı, $m = 2$ ise $S_m = 27$ ve $m = 3$ ise $S_m = 64$ adet alt küme meydana gelecektir. Şekil 3.2’de renkli Lena’nın renk uzayındaki dağılımı verilmiştir. Şekil 3.3(a)’da renkli Biber görüntüsü ve Şekil 3.3(b)’de renk uzayındaki dağılımı, Şekil 3.4(a)’da renkli Papağanlar görüntüsü ve Şekil 3.4(b)’de renk uzayındaki dağılımı, Şekil 3.5(a)’da renkli Babun görüntüsü ve Şekil 3.5(b)’de renk uzayındaki dağılımı gösterilmiştir.

Çizelge 3.1. Renk uzayının bölünmesi [3]

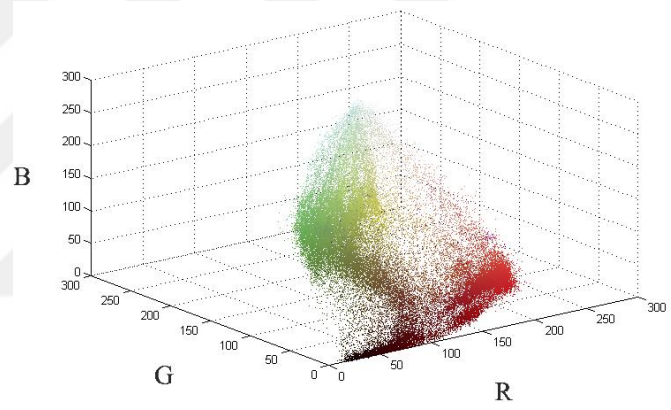
Sınıf Etiketleri	Kurallar	Atanan Renk
S_0	$\text{eğer}(R \leq t_{r,1} \ \& \ G \leq t_{g,1} \ \& \ B \leq t_{b,1})$	0x000000
S_1	$\text{eğer}(R \leq t_{r,1} \ \& \ G \leq t_{g,1} \ \& \ B > t_{b,1})$	0x0000FF
S_2	$\text{eğer}(R \leq t_{r,1} \ \& \ G > t_{g,1} \ \& \ B \leq t_{b,1})$	0x00FF00
S_3	$\text{eğer}(R \leq t_{r,1} \ \& \ G > t_{g,1} \ \& \ B > t_{b,1})$	0x00FFFF
S_4	$\text{eğer}(R > t_{r,1} \ \& \ G \leq t_{g,1} \ \& \ B \leq t_{b,1})$	0xFF0000
S_5	$\text{eğer}(R > t_{r,1} \ \& \ G \leq t_{g,1} \ \& \ B > t_{b,1})$	0xFF00FF
S_6	$\text{eğer}(R > t_{r,1} \ \& \ G > t_{g,1} \ \& \ B \leq t_{b,1})$	0xFFFF00
S_7	$\text{eğer}(R > t_{r,1} \ \& \ G > t_{g,1} \ \& \ B > t_{b,1})$	0xFFFFFF



Şekil 3.2. Renkli Lena: RGB uzayındaki renk dağılımı



a)

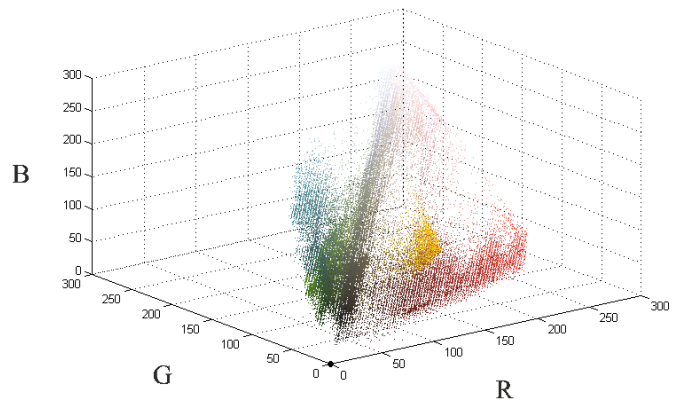


b)

Şekil 3.3. a) Biber b) Biber: RGB uzayındaki renk dağılımı

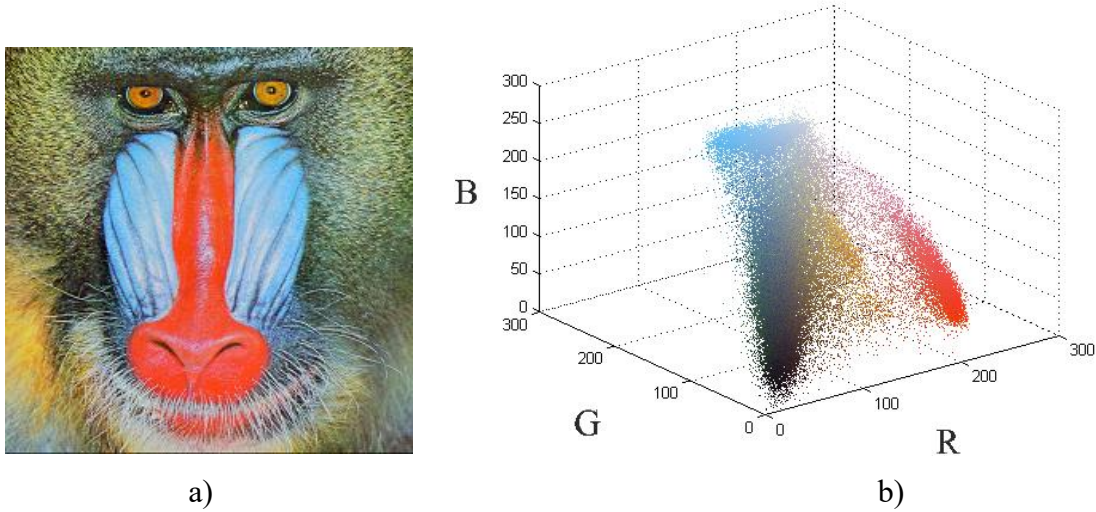


a)



b)

Şekil 3.4. a) Papağanlar b) Papağanlar: RGB uzayındaki renk dağılımı



Şekil 3.5. a) Babun b) Babun: RGB uzayındaki renk dağılımı

Görüntü üzerindeki her bir piksel RGB renk uzayında bir noktaya karşılık gelmektedir. Şekiller ve renk uzayındaki dağılımlar incelendiğinde, tekrar sayısı fazla olan piksel renk değerlerinin küp veya prizma üzerindeki yoğunluğunun daha fazla olduğu görülmektedir. Görüntüyü meydana getiren pikseller renk uzayında farklı noktalarda ve dağınık bir şekilde konumlandığı takdirde boş küp veya prizmaların meydana gelmesi muhtemeldir. Bu durum tek eşikli çözümlerde sıkıntı yaratmayabilir ancak eşik sayısının artışı ile birlikte küp veya prizma sayısı artacak ve boş kümelerin oluşması kaçınılmaz olacaktır. Örneğin Şekil 3.2'deki Lena görüntüsünün RGB uzayındaki dağılımı ile Şekil 3.5(b)'deki Babun görüntüsünün RGB uzayındaki dağılımı incelendiğinde, Babun görüntüsünün dağılımının daha seyrek olduğu ve renk uzayının farklı noktalarında dağınık bir biçimde konumlandığı görülmektedir. Dolayısıyla Babun görüntüsünde eşik sayısının artışı ile birlikte meydana gelecek küp veya prizma sayısının da artacağı ve Lena'da oluşacak küp veya prizma sayısından daha fazla oluşacağı açıktır. Fakat söz konusu küp veya prizmaların bazıları boş kalacak ve görüntü üzerinde karşılığı olmayan anlamsız kümeler oluşacaktır.

3.2. Renk Uzayının Çok Eşikli Bölütlenmesi

Bir önceki başlıkta renk uzayının tek seviyeli eşiklenmesi detaylı olarak ifade edilmiştir. Renkli görüntüler üç kanaldan meydana gelmekte olup her kanal farklı histogram bilgisine sahiptir. Dolayısıyla her renk kanalından elde edilen farklı eşik değerlerinin bir araya getirilmesi temel problemlerden biridir. Bu amaçla ABC algoritması kullanılarak renk uzayının çok seviyeli eşiklenmesine yönelik yeni bir yöntem önerilmiştir.

Önerilen metotta yapay arı koloni algoritması ile eşik değerlerinin tespit edilebilmesi için başlangıç parametrelerine ihtiyaç duyulmaktadır. Bunlar gri veya renkli görüntü bilgisi, maksimum çevrim sayısı, problemin boyutu, limit ve amaç fonksiyonlarıdır. İlk etapta hesaplanmak istenen eşik sayısı, problemin boyutunu temsil edeceğinden her kanal için problemin boyutu kadar değişken tahsis edilir. Böylece R, G ve B kanalı için 3 farklı parametre tanımlanmaktadır. Problem uzayının sınırları ise gri seviyeli görüntünün alt ve üst limitleriyle orantılıdır. Dolayısıyla alt sınır 0, üst sınır ise 255 değerini temsil etmektedir. Optimum eşik değerlerinin tutulduğu çözüm kümesi, amaç fonksiyonu ile uygunluk değeri ve komşu kaynaklar ise eşik değerlerinin tespitinde kullanılan bileşenlerdir.

Yapay arı kolonisinde bir arı yiyecek, çözüm, uygunluk, kaynağın geliştirilememe sayacı bilgilerini beraberinde taşıyan bir yapıdadır. Süreç maksimum çevrim sayısı kadar sırasıyla işçi arı, olasılık hesabı, gözcü arı safhası, en iyi çözümün hafızada tutulması ve kâşif arıların gönderilmesi ile devam eder. Geliştirilen yöntem histogram tabanlı bir çözüm algoritmasıdır. Dolayısıyla hesaplanmak istenen gri ölçekli veya renkli görüntüdeki her bir kanalın histogram bilgisine ihtiyaç duyulmaktadır. Şekil 2.6'da gri ölçekli, Şekil 2.7'de ise renkli Lena görüntüsünün histogramı verilmişti. Söz konusu histogram bilgisinin elde edilmesinin ardından, yapay arı koloni algoritması ile Eş. 2.21'deki Kapur ve Eş. 2.26'daki Otsu amaç fonksiyonlarını maksimum yapan çözümler aranarak eşik değerleri hesaplanır. Elde edilen eşik değerleri kullanılarak RGB renk uzayı alt küp veya prizmalara bölütlenerek süreç tamamlanır.

Yapay arı kolonisinde ilk aşama başlangıç besin kaynaklarının elde edilmesidir. Her bir yiyecek kaynağı sayısı kadar arı tanımlanır. Daha sonra Eş. 2.34'deki X_{ij} yani besin kaynaklarının üretimine geçilir. Burada i seçilen çözüm kümesinin indis numarasına, j optimize edilecek parametrenin indisine tekabül etmekte olup, $X_j^{\min}$ 0, $X_j^{\max}$ değeri 255'e karşılık gelecektir. Dolayısıyla 0 ve 255 arasında rastgele başlangıç çözüm kümeleri üretilir. Bu işlemin ardından görüntü üzerinde üretilen her bir çözüm kümesinin amaç fonksiyonu hesaplanır ve buradan gelen sonucun uygunluk değeri elde edilir. Amaç fonksiyonu Otsu ve Kapur yöntemleri ile hesaplanan sonuçlardır. Uygunluk fonksiyonu ise açgözlü seçimi(greedy selection) mantığı kullanılarak bulunur.

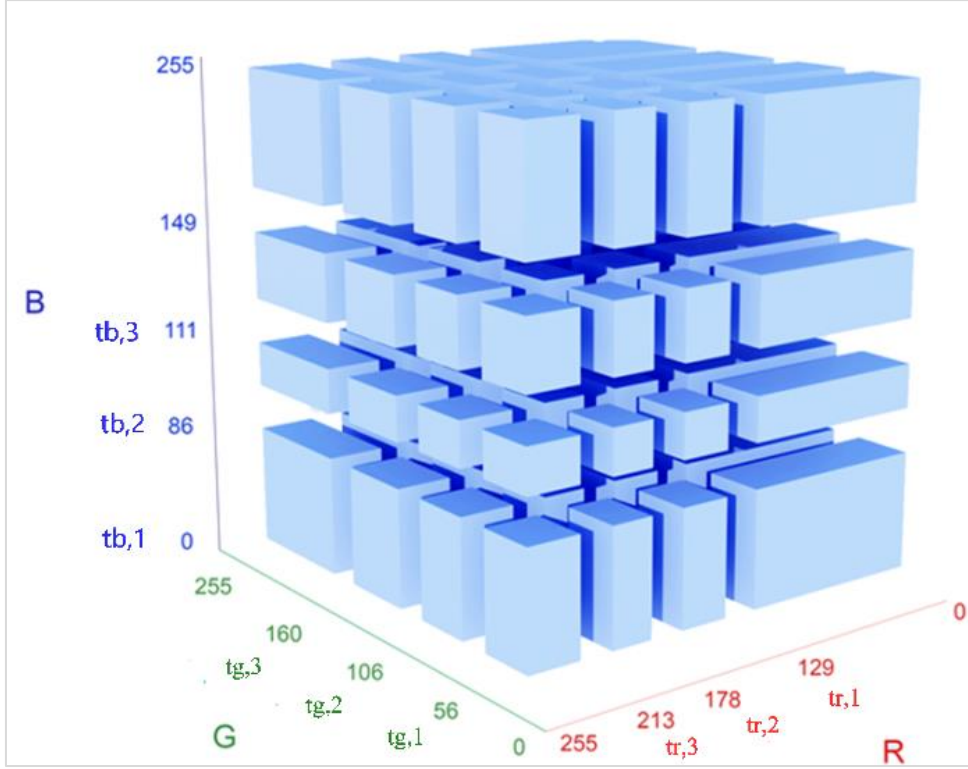
İşçi arılar rastgele elde edilmiş başlangıç çözüm kümelerine giderler. Burada rastgele seçilen komşu kaynaklar -1 ve 1 aralığındaki rassal değişkenler ile optimize edilerek X_{ij} komşuluğundaki V_{ij} kaynağı Eş. 2.35'deki gibi elde edilir. Elde edilen değerin alt ve üst parametre sınırlarının dışına çıkması durumunda öteleme işlemi Eş. 2.36'daki gibi yapılır. Yani 0 'dan küçük değerler 0 'a, 255 'den büyük değerler 255 'e ötelenir. Böylece besin kaynak sınırlarının dışına çıkılması engellenmiş olur. Eğer elde edilen çözümün uygunluk değeri eskisinden daha iyiye yeni çözüm hafızaya alınır ve çözüm geliştirilememe sayacı sıfırlanır. Eğer eski çözüm yeni çözümden daha iyi veya aynı ise ise eski çözümle devam edilir ve çözüm geliştirilememe sayacı bir arttırılır.

Yapay arı kolonisinde ikinci adım olasılık hesabıdır. Söz konusu hesaplama, gözcü arıların seleksiyonda tercih edecekleri ve yönelecekleri kaynakların elde edilmesi için gereklidir. Bu bağlamda uygunluk fonksiyonuna dayalı olasılık hesaplaması rulet tekerleği yöntemine göre Eş. 2.38'deki gibi yapılır. Rulet tekerleği yönteminde her bir dilimin açısı seçilme olasılığıyla doğru orantılıdır. Dolayısıyla bir kaynağın nektar miktarı yani uygunluk değeri ne kadar büyük olursa, o kaynağın gözcü arılar tarafından seçilme ihtimali bir o kadar yüksek olacaktır.

Gözcü arı safhasında ilk olarak $0-1$ aralığında rastgele sayı üretilir. Üretilen değer olasılık hesabı sonucundan daha küçükse, Eş 2.35'e göre yeni çözüm üretilir ve işçi arı safhasındaki adımlar yeniden uygulanır. Yeni çözüm daha iyiye hafızaya alınır ve eski çözüm hafızadan silinir. Eğer yeni çözüm eski çözümden daha iyi değilse çözüm geliştirilememe sayacı bir arttırılır. İlgili süreç tüm gözcü arılar besin kaynaklarına dağılana kadar gerçekleştirilir.

İşçi ve gözcü arı süreçlerinin tamamlanmasının ardından problemin boyutu yani eşik sayısı kadar olan olan çözümlerin en en iyileri global çözüm kümesine dahil edilir. En iyi besin kaynakları elde edilen eşik sayılarını temsil eder. Kâşif arılar, çözüm geliştirilememe sayacı limiti geçmediği müddetçe Eş. 2.34'deki gibi rastgele besin kaynak arayışına devam ederler. Tıpkı işçi arılarda olduğu gibi çözüm değerinin alt ve üst sınırlar aralığında olup olmadığının kontrolü ve uygunluk fonksiyonun hesaplanması gibi süreçler devam ettirilir. Çözüm geliştirilememe sayacı limit değişkenine eşitlendiği anda yapay arı koloni algoritmasının sonlandırma kriteri sağlanmış olacaktır. Bu işlem sonucunda hafızadaki en iyi çözümler eşik değerlerine karşılık gelecektir.

ABC algoritması ile hesaplanan eşik değerleri elde edildikten sonra renk uzayı alt küp veya prizmalara bölümlenir. Daha sonra ortaya çıkan alt küpler içerisinde kalan pikseller aynı kümeye atanır. Şekil 2.5’de verilen renkli Lena’nın histogram bilgileri kullanılarak arı algoritması ile hesaplanan eşik değerleri sonucunda renk uzayı alt küp veya prizmalara bölümlenmiş olup Şekil 3.6’da gösterilmiştir [50].



Şekil 3.6. Lena RGB renk uzayı ve atanan kümeler, $m = 3$

Grafikte gösterilen her bir alt küp veya prizma bir bölgeye karşılık gelmektedir. Şekil 3.6 dikkatlice incelendiğinde küplerin hacimlerinin eşik değerlerine bağlı olması neticesinde farklı boyutlara sahip olup, aynı olmadıkları görülmektedir. Böylece görüntüdeki piksel dağılımları ile alt küp veya prizma hacimlerinin doğru orantılı olacağı söylenebilir. Bununla birlikte hacmi büyük olan kümelerin homojenliği, hacmi küçük olanlara nazaran düşük kalacaktır. Eşik sayısı artırılarak her bir küpün hacmi azaltılmış olacak ve sınıfların homojenliği de artacaktır. Bu sayede çok seviyeli eşikleme homojenliğin artışında etkili bir unsur olacaktır [50].



4. DENEYSEL SONUÇLAR VE YORUMLAR

Lineer arama yöntemleri ile eşik hesabı kaba kuvvet (brute force) tabanlı olup, yapısı gereği bütün çözüm uzayı tek tek taranmaktadır. Söz konusu yöntemler ile tek ve çok seviyeli eşiklerin tespiti minimum hata ile elde edilebilmektedir. Ancak bu metotlar eşik sayısının artışı ile birlikte hesapsal açıdan yetersiz ve verimsiz kalmaktadırlar. Öyle ki tek eşikte çözüm uzayındaki 255, iki eşikte 255×255 , üç eşğin hesaplanmasında ise $255 \times 255 \times 255$ nokta tek tek taranır. En nihayetinde doğruluk açısından tatminkâr sonuçlar üretilir. Optimizasyon algoritmaları bahsedilen sıkıntıları giderecek ve daha hızlı sonuç verecek biçimde tasarlanan yapılardır. Yapay arı koloni algoritması ile geliştirilen yöntem optimizasyon tabanlı olup, tek ve çok seviyeli eşikleri hesaplamak için bütün çözüm uzayının tek tek taranılması yerine daha hızlı sonuç veren rastgele çözüm kümelerinden faydalanılmaktadır. Söz konusu çözüm kümeleri belirli bir döngü içerisinde iyileştirilerek amaç fonksiyonunu maksimize eden değerler elde edilir. Geliştirilen algoritma lineer arama yöntemlerine en yakın sonuçları verecek şekilde tasarlanmıştır. Lineer yöntemler ile tekli eşiklerin tespitinde sadece 255 nokta taranmakta, böylece hesapsal maliyet açısından en iyi sonucu üretmektedirler. İki eşikli sonuçlarda ise lineer yöntem ve yapay arı kolonisi ile geliştirilen yöntem hesapsal açıdan birbirine yakın sonuçlar üretmektedirler. Dolayısıyla önerilen yöntem üç ve daha fazla eşikte avantajlı hale gelmekte olup, çok daha hızlı sonuç ortaya koymaktadır. Bununla birlikte geliştirilen metot ile elde edilen tekli ve çoklu eşiklerin lineer yöntem ile üretilen eşiklere yakın sonuçlar ürettiği görülmektedir. Eşiklerin tespitinin ardından görüntü üzerinde sınıflandırma işlemi renk uzayı bölütlenerek yapılmaktadır. Önerilen metot yine bu aşamada hesapsal maliyet açısından FCM yöntemi ile karşılaştırdığında daha iyi sonuçlar ortaya koymuştur. Özellikle küme sayısının artışı ile birlikte FCM yöntemindeki hesapsal verimsizlik, geliştirilen yöntem ile ortadan kaldırılmıştır.

4.1. Geliştirilen Arayüz

Önerilen algoritma için C# ortamında bir adet arayüz geliştirilmiştir. Deneysel sonuçlar Intel i7 4600U 2.70 GHz işlemci ve 8 GB RAM donanıma sahip bilgisayarda elde edilmiştir. Geliştirilen arayüz Şekil 4.1(a)'da verilmiştir.

Yapay Arı Koloni Algoritması ile Çok Seviyeli Eşikleme

Eşik Sayısı: ☒ 1, ☐ 2, ☐ 3

Amaç Fonksiyonu: ☐ Otsu, ☒ Kapur

Lineer Yöntem ile Eşik Değerlerini Hesapla

ABC ile Eşik Değerlerini Hesapla

Eşik Değerleri:

R:

G:

B:

Hesaplama Süresi

a)

Yapay Arı Koloni Algoritması ile Çok Seviyeli Eşikleme

Eşik Sayısı: ☒ 1, ☐ 2, ☐ 3

Amaç Fonksiyonu: ☒ Otsu, ☐ Kapur

Lineer Yöntem ile Eşik Değerlerini Hesapla

ABC ile Eşik Değerlerini Hesapla

Eşik Değerleri:

R:

G:

B:

Hesaplama Süresi: 416 saniye

b)

Yapay Arı Koloni Algoritması ile Çok Seviyeli Eşikleme

Eşik Sayısı: ☐ 1, ☒ 2, ☐ 3

Amaç Fonksiyonu: ☒ Otsu, ☐ Kapur

Lineer Yöntem ile Eşik Değerlerini Hesapla

ABC ile Eşik Değerlerini Hesapla

Eşik Değerleri:

R:

G:

B:

Hesaplama Süresi: 549 saniye

c)

Şekil 4.1. Geliştirilen arayüz a) Giriş b) Tek eşikli sonuç örneği c) İki eşikli sonuç örneği

Form üzerinde ilk olarak ayrıştırılması istenen görüntü seçilir. Seçilen orijinal görüntü formun sol kısmında yer almaktadır. Daha sonra hesaplanmak istenen eşik sayısı ve amaç fonksiyonu seçilir. ABC ile eşik değerlerini hesaplama butonu ile süreç tetiklenir. İlk aşamada yapay arı koloni algoritmasının başlangıç parametreleri ayarlanır. Problemin boyutu elde edilmek istenen eşik sayısıdır. Dolayısıyla 2 eşik bulunması istenen durumda problemin boyutu 2, 3 eşik bulunması istenen durumda problem boyutu 3 olacaktır. Başlangıç parametrelerinin ayarlanmasının ardından her bir kanal için ayrı ayrı histogram bilgisi hesaplanır ve hafızada tutulur. Eşik değerleri Otsu veya Kapur amaç fonksiyonlarının yapay arı koloni algoritması ile optimize edilmesine dayanmaktadır. Her bir renk kanalı için Otsu veya Kapur amaç fonksiyonunu ayrı ayrı maksimum yapan eşik değerlerinin hesaplanmasının ardından renk uzayı çok seviyeli olarak bölümlenir ve süreç tamamlanmış olur. Elde edilen sonuçlar Şekil 4.1(b)'de ve Şekil 4.1(c)'de formun sağ kısmında yer alır. İşlemin tamamlanmasının ardından her bir kanal için hesaplanmış olan eşik değerleri ve hesaplama süreleri form üzerinde gösterilir.

4.2. Tek Eşikli Sonuçlar

Önerilen yöntem Şekil 4.2'de verilen 1 adet gri ölçekli Çift ve 4 adet renkli (Lena, Biber, Papağanlar ve Babun) görüntü üzerinde uygulanmıştır.



Şekil 4.2. Çift

Çift görüntüsü için hesaplanan eşik değerleri ($m = 1$) ve hesapsal maliyetler Çizelge 4.1'de verilmiştir. Önerilen yöntemin Otsu amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.3(a)'da, Kapur amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.3(b)'de

gösterilmiştir. Tek eşik ile 2 sınıf oluşacağından, FCM algoritmasında $c=2$ atanarak sınıflandırma yapılmış ve elde edilen sonuç Şekil 4.3(c)'de verilmiştir. FCM algoritması ile süreç 1,480 saniyede tamamlanmıştır. Diğer taraftan lineer arama yaklaşımının ABC yönteminden daha hızlı olduğu görülmüştür. FCM algoritmasında küme merkezleri rastgele atandığından her zaman aynı sonucu alma garantisi olmamakla birlikte, Şekil 4.3 de verilen sınıflandırma sonuçları birbirlerine yakındır.

Çizelge 4.1. Çift için bulunan eşikler ($m=1$)

Amaç Fonksiyonu	Otsu		Kapur	
Arama Yöntemi	Lineer	ABC	Lineer	ABC
Eşikler	t_1	t_1	t_1	t_1
R	105	105	92	93
G	105	106	92	94
B	105	106	92	95
Süre(s)	0,202	0,393	0,207	1,123
Küme sayısı:c	2		2	



a)

b)

c)

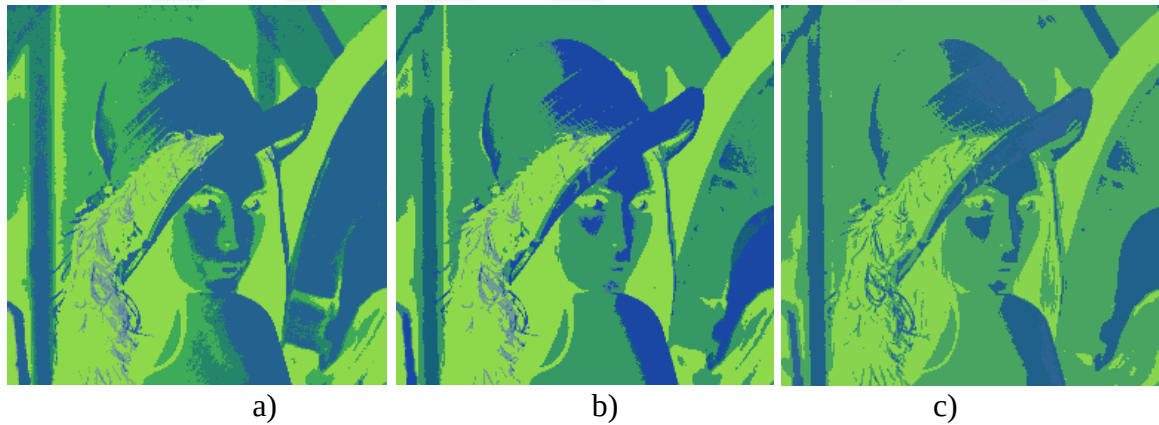
Şekil 4.3. Çift, $m = 1$ a) Otsu, $c=2$ b) Kapur, $c=2$ c) FCM, $c=2$, süre(s): 1,480

Renkli Lena görüntüsü için eşik sayısı ($m = 1$) seçildiğinde tespit edilen eşik değerleri ve hesapsal maliyetler Çizelge 4.2'de verilmiştir. Geliştirilen algoritmanın Otsu amaç fonksiyonu ile elde edilen sonucu Şekil 4.4(a)'da, Kapur amaç fonksiyonu ile elde edilen sonucu Şekil 4.4(b)'de verilmiştir. Otsu ile yapılan deneyde 8 küme, Kapur'da ise 7 küme meydana gelmiştir. FCM algoritmasında $c=8$ atanarak yapılan sınıflandırma sonucu ise Şekil 4.4(c)'de verilmiştir. FCM algoritması ile süreç 1,158 saniyede tamamlanmıştır. Diğer taraftan lineer arama yaklaşımının ABC yönteminden daha hızlı olduğu görülmüştür.

Deneyisel sonuçlara genel olarak bakıldığında Lena'nın yüz hattının, saç ve şapka aksesuarındaki detayların tam olarak ortaya çıkmadığı ve arka plandaki nesnelerin iç içe girdiği anlaşılmaktadır. Lena'nın yüz gösteriminde Otsu yönteminin Kapur'a göre daha iyi sonuç verdiği söylenebilir. Kapur yaklaşımında ise arka planda yer alan detayların daha belirgin olduğu görülmektedir. FCM yönteminde ise küme merkezleri rastgele atandığından dolayı değişken sonuçlar alınmıştır.

Çizelge 4.2. Lena için bulunan eşikler ($m=1$)

Amaç Fonksiyonu	Otsu		Kapur	
Arama Yöntemi	Lineer	ABC	Lineer	ABC
Eşikler	t_1	t_1	t_1	t_1
R	162	165	167	167
G	102	102	141	139
B	112	113	131	129
Süre(s)	0,204	0,393	0,208	0,917
Küme sayısı:c	8		7	



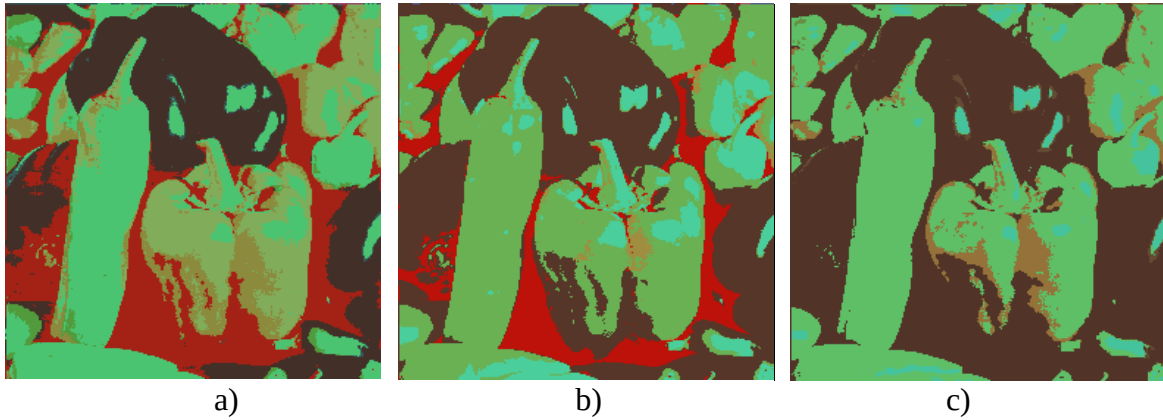
Şekil 4.4. Lena, $m = 1$ a) Otsu, $c=8$ b) Kapur, $c=7$ c) FCM, $c=8$, süre(s):1,158

Biber görüntüsü için bulunan eşik değerleri ($m = 1$) ve hesapsal maliyetler Çizelge 4.3'de verilmiştir. Önerilen metodun Otsu amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.5(a)'da, Kapur amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.5(b)'de gösterilmiştir. İki yöntemde de 8 küme meydana geldiği gözlenmiştir. $c=8$ atanarak FCM algoritması ile yapılan sınıflandırma sonucu Şekil 4.5(c)'de gösterilmiştir. FCM ile süreç 1,138 saniyede tamamlanmıştır. Diğer taraftan lineer arama yaklaşımının ABC yönteminden daha hızlı olduğu görülmüştür.

Deneysel sonuçlar dikkate alındığında, Otsu yönteminde resmin ortasında yer alan yeşil yuvarlak biberin hatlarının arka plan ile iç içe geçtiği ve resmin en alt sol kısmında yer alan biberin kıvrımlarının ortaya çıkmadığı anlaşılmaktadır. Kapur yönteminde ise özellikle arka planda yer alan biberlerin ve uzun biberin sol tarafında yatay şekilde yer alan kırmızı biberin iç içe geçtiği ve kenar hat ayrıntılarının belirgin vaziyette olmadığı görülmektedir. FCM ile ele edilen deneysel sonuçta ise farklı renkteki biberlerin ayırt edilemediği ve arka plan ile iç içe girdiği anlaşılmaktadır. Otsu ve Kapur yöntemleri ile elde edilen sonuçların FCM yöntemine göre daha başarılı olduğu söylenebilir.

Çizelge 4.3. Biber için bulunan eşikler ($m=1$)

Amaç Fonksiyonu	Otsu		Kapur	
Arama Yöntemi	Lineer	ABC	Lineer	ABC
Eşikler	t_1	t_1	t_1	t_1
R	146	149	100	100
G	111	115	121	121
B	71	73	117	119
Süre(s)	0,205	0,402	0,210	1,036
Küme sayısı:c	8		8	



Şekil 4.5. Biber, $m = 1$ a) Otsu, $c = 8$ b) Kapur, $c=8$ c) FCM, $c = 8$, süre(s):1,138

Papağanlar görüntüsü için tahmin edilen eşik değerleri ($m = 1$) ve hesapsal maliyetler Çizelge 4.4’de verilmiştir. Önerilen yöntemin Otsu amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.6(a)’da, Kapur amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.6(b)’de gösterilmiştir.

İki yöntemde de 8 küme meydana gelmiştir. FCM algoritmasında $c=8$ atanarak yapılan sınıflandırma sonucu Şekil 4.6(c)'de verilmiştir. FCM algoritması ile süreç 1,752 saniyede tamamlanmıştır. Diğer taraftan lineer arama yaklaşımının ABC yönteminden daha hızlı olduğu görülmüştür.

Otsu ile yapılan deneyde arka planda yer alan yaprak detayı ve görüntünün solunda yer alan papağanın kanat detaylarının ortaya çıktığı, Kapur'da ise belirgin olmadığı görülmektedir. Diğer yandan Kapur yönteminde, görüntünün sağ tarafında yer alan papağının tüylerinin hatları ile beraber daha anlaşılır olduğu ve göz bölgesindeki detayların daha net bir biçimde ortaya çıktığı anlaşılmaktadır. FCM ile elde edilen sonucun Otsu ve Kapur yöntemleri ile karşılaştırıldığında yetersiz kaldığı söylenebilir.

Çizelge 4.4. Papağanlar için bulunan eşikler ($m=1$)

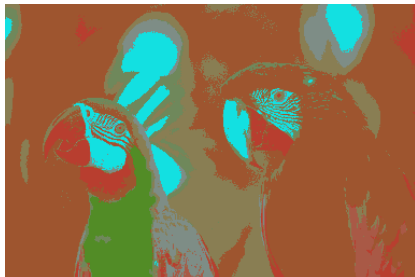
Amaç Fonksiyonu	Otsu		Kapur	
Arama Yöntemi	Lineer	ABC	Lineer	ABC
Eşikler	t_1	t_1	t_1	t_1
R	146	147	134	136
G	119	119	150	151
B	114	115	107	106
Süre(s)	0,305	0,400	0,305	1,096
Küme sayısı:c	8		8	



a)



b)



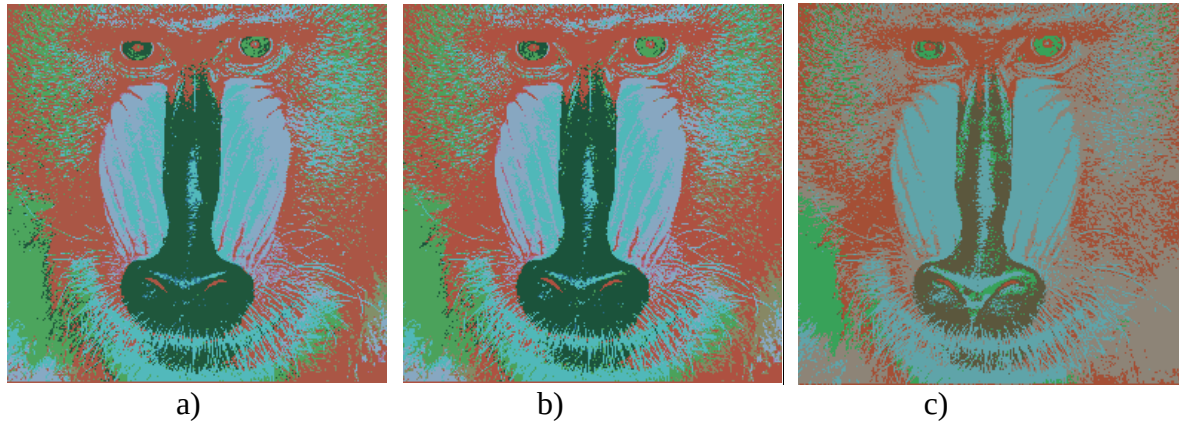
c)

Şekil 4.6. Papağanlar, $m = 1$ a) Otsu, $c = 8$ b) Kapur, $c = 8$ c) FCM, $c = 8$, süre(s): 1,752

Babun görüntüsü için tespit edilen eşik değerleri ($m = 1$) ve hesapsal maliyetler Çizelge 4.5’de verilmiştir. Geliştirilen algoritmanın Otsu amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.7(a)’da, Kapur amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.7(b)’de gösterilmiştir. İki yöntemde de 8 küme meydana gelmiştir. FCM algoritmasında $c=8$ atanarak sınıflandırma yapılmış ve elde edilen sonuç Şekil 4.7(c)’de verilmiş olup süreç 1,161 saniyede tamamlanmıştır. Diğer taraftan lineer arama yaklaşımının ABC yönteminden daha hızlı olduğu görülmüştür. Şekil 4.7’de verilen sınıflandırma sonuçları birbirlerine yakındır.

Çizelge 4.5. Babun için bulunan eşikler ($m=1$)

Amaç Fonksiyonu	Otsu		Kapur	
Arama Yöntemi	Lineer	ABC	Lineer	ABC
Eşikler	t_1	t_1	t_1	t_1
R	145	146	146	146
G	128	128	122	121
B	131	132	129	131
Süre(s)	0,200	0,390	0,205	1,119
Küme sayısı:c	8		8	



Şekil 4.7. Babun, $m = 1$ a) Otsu, $c = 8$ b) Kapur, $c=8$ c) FCM, $c = 8$, süre(s):1,161

4.3. Çok Eşikli Sonuçlar

Çift görüntüsü için hesaplanan eşik değerleri ($m = 2$) ve hesapsal maliyetler Çizelge 4.6’da verilmiştir. Önerilen yöntemin Otsu amaç fonksiyonu kullanılarak elde edilen sonucu Şekil

4.8(a)'da, Kapur amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.8(b)'de gösterilmiştir. İki yöntemde de 3 küme meydana gelmiştir. FCM algoritmasında $c=3$ atanarak sınıflandırma yapılmış ve elde edilen sonuç ise Şekil 4.8(c)'de verilmiştir. FCM algoritması ile süreç 4,884 saniyede tamamlanmıştır. İki eşikli deneyde görüntü üzerindeki nesnelerin nispeten belirgin bir vaziyete kavuştuğu görülmektedir. Özellikle Otsu yönteminde gerek erkek gerekse kadının kıyafet ve görünüm detaylarının ortaya çıktığı anlaşılmaktadır. Bununla birlikte zemin ile eşyalardaki renk ve gölge tonlamaları açısından da başarılı olduğu söylenebilir. Kapur yönteminde ise balkon penceresi ve arka planda yer alan ağaç detaylarında daha iyi sonuçlar elde edilmiştir. Hesapsal maliyet açısından ele alındığında, önerilen yöntemin gerek Otsu gerek ise Kapur metotları ile alınan sonuçlarının FCM yaklaşımına göre daha hızlı sonuç verdiği, Lineer yöntem ile karşılaştırıldığında ise birbirine yakın sonuçlar ürettiği görülmektedir.

Çizelge 4.6. Çift için bulunan eşikler ($m=2$)

Amaç Fonksiyonu	Otsu				Kapur			
Arama Yöntemi	Lineer		ABC		Lineer		ABC	
Eşikler	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂
R	86	145	87	144	91	177	93	175
G	86	145	88	145	91	177	92	176
B	86	145	86	145	91	177	92	174
Süre(s)	0,550		0,439		1,018		1,176	
Küme sayısı:c	3				3			



Şekil 4.8. Çift, $m = 2$ a) Otsu, $c = 3$ b) Kapur, $c = 3$ c) FCM, $c = 3$, süre(s): 4,884

Eşik sayısı ($m = 3$) seçildiğinde Çift görüntüsü için tespit edilen eşik değerleri ve zaman maliyetleri Çizelge 4.7’de verilmiştir. Amaç fonksiyonu Otsu seçildiğinde elde edilen sonuç Şekil 4.9(a)’da, Kapur seçildiğinde ise elde edilen sonuç Şekil 4.9(b)’de gösterilmiştir. İki yöntemde de 4 küme meydana gelmiştir. FCM algoritmasında $c=4$ atanarak yapılan sınıflandırma sonucunda elde edilen sonuç ise Şekil 4.9(c)’de verilmiş olup 14,819 saniyede tamamlanmıştır. Eşyalardaki gölgelendirme ve tonlamalar açısından bakıldığında, iki eşikli deneyde görünmeyen detayların üç eşik ile birlikte daha net bir biçimde ortaya çıktığı görülmektedir. Özellikle Otsu yönteminde kapı, kapı üstünde yer alan duvardaki işleme, abajür ve perdede ortaya çıkan kümeler söz konusu yorumu doğrular niteliktedir. Diğer yandan Çizelge 4.7’de görüldüğü üzere, lineer çözüm Otsu yönteminde 36,928 saniyede, Kapur yönteminde ise 72,418 saniyede sonuç vermiştir. Deneysel sonuçlar hesapsal maliyetler açısından dikkate alındığında, önerilen yöntemin 3 eşikli deneyde en verimli çalışma hızına sahip sonuçlar ürettiği görülmüştür.

Çizelge 4.7. Çift için bulunan eşikler ($m=3$)

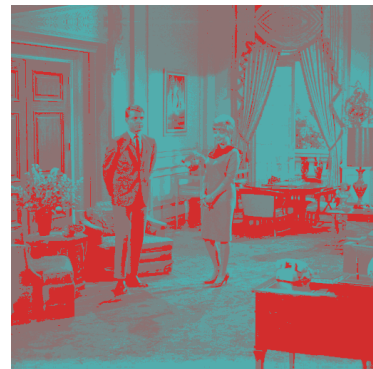
Amaç Fonksiyonu	Otsu						Kapur					
Arama Yöntemi	Lineer			ABC			Lineer			ABC		
Eşikler	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃
R	75	123	163	75	124	164	48	106	178	46	105	175
G	75	123	163	77	122	160	48	106	178	46	102	177
B	75	123	163	76	123	164	48	106	178	48	107	177
Süre(s)	36,928			0,565			72,418			1,225		
Küme sayısı:c	4						4					



a)



b)



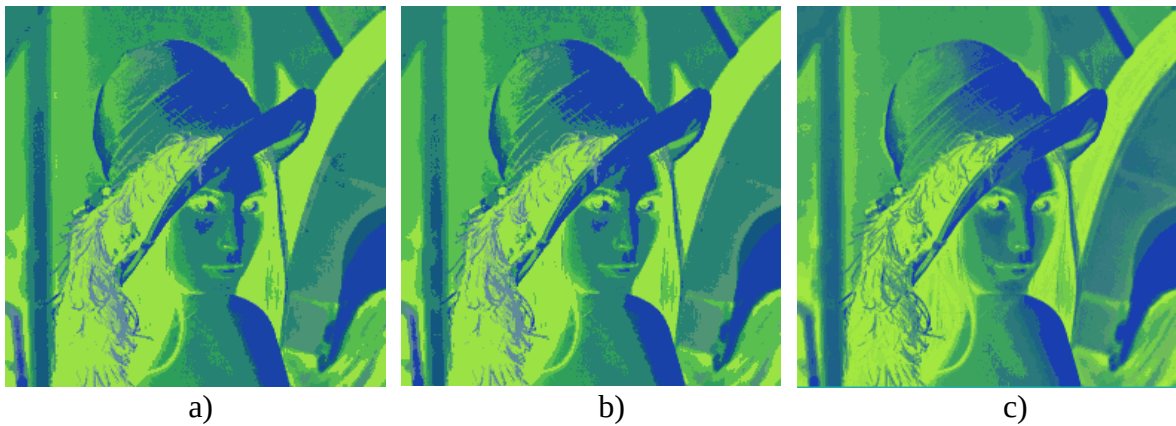
c)

Şekil 4.9. Çift, $m = 3$ a) Otsu, $c=4$ b) Kapur, $c=4$, c) FCM, $c=4$, süre(s): 14,819

Görüntü işleme çalışmalarında referans olarak kabul edilen renkli Lena görüntüsü için tespit edilen eşik değerleri ($m = 2$) ve hesapsal maliyetler Çizelge 4.8’de gösterilmiştir. Geliştirilen algoritmanın Otsu amaç fonksiyonu ile hesaplanan sonucu Şekil 4.10(a)’da, Kapur amaç fonksiyonu ile hesaplanan sonucu Şekil 4.10(b)’de verilmiştir. Söz konusu sonuçlar neticesinde Otsu yönteminde 21, Kapur metodunda ise 20 küme meydana gelmiştir. FCM yaklaşımında $c=21$ atanarak sınıflandırma yapılmış ve Şekil 4.10(c)’de verilmiştir. FCM algoritması ile sonuçlar 9,992 saniyede alınmıştır. İki eşikli deneysel sonuçlar incelendiğinde, Otsu ve Kapur yönteminin birbirine çok yakın sonuçlar verdiği anlaşılmaktadır. Fakat arka plandaki nesnelerin gösteriminde Otsu yönteminin daha başarılı olduğu söylenebilir. Önerilen yöntem, hesapsal açıdan FCM yöntemine göre daha hızlı sonuç verirken, lineer yöntemler ile yakın sonuçlar ortaya koymaktadır.

Çizelge 4.8. Lena için bulunan eşikler ($m=2$)

Amaç Fonksiyonu	Otsu				Kapur			
Arama Yöntemi	Lineer		ABC		Lineer		ABC	
Eşikler	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂
R	139	197	138	196	134	191	134	191
G	79	146	78	146	82	151	82	151
B	96	140	96	141	98	144	98	144
Süre(s)	0,550		0,450		0,930		1,025	
Küme sayısı:c	21				20			



Şekil 4.10. Lena, $m = 2$ a) Otsu, $c=21$ b) Kapur, $c=20$ c) FCM, $c=21$, süre(s):9,992

Renkli Lena görüntüsü için bulunan eşik değerleri ($m = 3$) ve hesapsal maliyetler Çizelge 4.9'da verilmiştir. Şekil 4.11(a)'da Otsu amaç fonksiyonu ile tespit edilen sonuçlar, Şekil 4.11(b)'de Kapur amaç fonksiyonu ile tespit edilen sonuçlar gösterilmiştir. Otsu metodunda 39, Kapur metodunda ise 33 küme meydana gelmiştir. $c=39$ atanarak FCM algoritması ile yapılan sınıflandırma sonucu Şekil 4.11(c)'de verilmiştir. Söz konusu yöntem ile süreç 46,702 saniyede tamamlanmıştır. Deneysel sonuçlar incelendiğinde, küme sayılarının artışı ile beraber Lena üzerindeki ve arka planda yer alan nesnelerin farklı kümelerle ayrışarak görüntü üzerinde yer aldığı anlaşılmaktadır. Fakat üç eşikli deneyde arka planda oluşan bazı piksellerin farklı kümeye dahil olması beklenirken aynı kümeye dahil olduğu görülmektedir. Çizelge 4.9'daki işlem maliyetlerine bakıldığında önerilen yöntemin üç eşikli sonuçlarda büyük bir avantaj barındırdığı açıktır.

Çizelge 4.9. Lena için bulunan eşikler ($m=3$)

Amaç Fonksiyonu	Otsu						Kapur					
Arama Yöntemi	Lineer			ABC			Lineer			ABC		
Eşikler	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃
R	129	178	213	130	180	215	116	159	203	115	155	209
G	58	107	160	55	105	160	61	110	161	59	109	159
B	84	111	147	82	109	146	93	132	170	91	130	170
Süre(s)	36,961			0,551			59,225			1,085		
Küme sayısı:c	39						33					



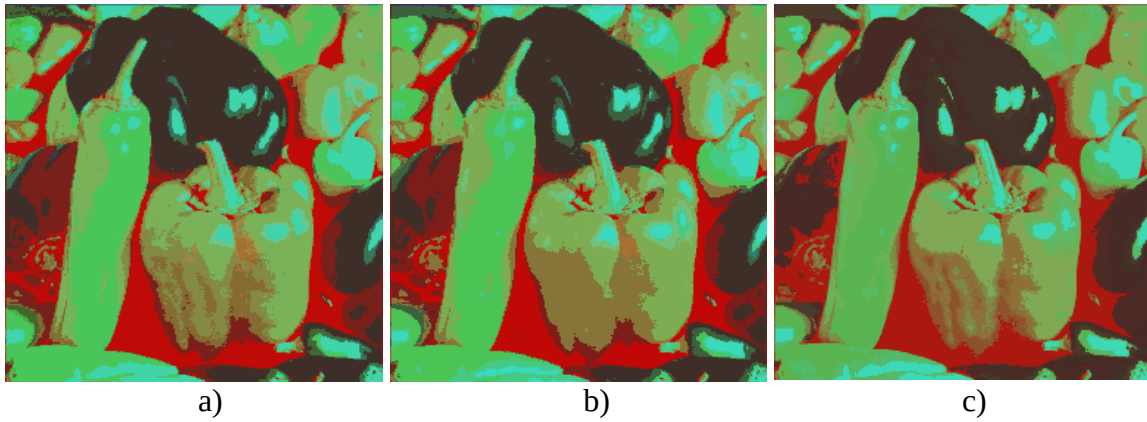
Şekil 4.11. Lena, $m = 3$ a) Otsu, $c=39$ b) Kapur, $c=33$ c) FCM, $c=39$, süre(s): 46,702

Şekil 3.3(b)'de RGB renk uzayındaki renk dağılımı verilen ve Lena görüntüsüne göre uzaydaki dağılımı farklı olan Biber görüntüsü için tahmin edilen eşik değerleri ($m = 2$) ve

hesapsal maliyetler Çizelge 4.10'da gösterilmiştir. Önerilen metodun Otsu amaç fonksiyonu ile elde edilen sonucu Şekil 4.12(a)'da, Kapur amaç fonksiyonu ile elde edilen sonucu Şekil 4.12(b)'de verilmiştir. Deney neticesinde Otsu yöntemi ile 23, Kapur yöntemi ile 21 küme meydana gelmiştir. FCM algoritmasında $c=23$ atanarak yapılan sınıflandırma 13,135 saniyede tamamlanmış olup sonucu Şekil 4.12(c)'de verilmiştir. FCM yaklaşımı ile tek eşikte nesnelerin arka plan ile iç içe girme sıkıntısının iki eşikli deneyde giderildiği, biberlerin daha belirgin vaziyete ulaştığı ve zemin ile nispeten ayrıştığı anlaşılmaktadır. Bununla birlikte özellikle Otsu yönteminde ortada duran biberlerin gölge ve ışıklandırmalarının Kapur yöntemine göre daha başarılı olduğu açıktır. İki eşikli sonuçlarda önerilen yöntem ve lineer yöntemin hesapsal açıdan birbirine yakın sonuçlar verdiği görülmektedir.

Çizelge 4.10. Biber için bulunan eşikler ($m=2$)

Amaç Fonksiyonu	Otsu				Kapur			
Arama Yöntemi	Lineer		ABC		Lineer		ABC	
Eşikler	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂
R	99	160	100	161	96	160	95	159
G	78	156	77	156	68	152	69	152
B	59	126	61	126	108	166	110	167
Süre(s)	0,565		0,452		0,990		1,116	
Küme sayısı:c	23				21			



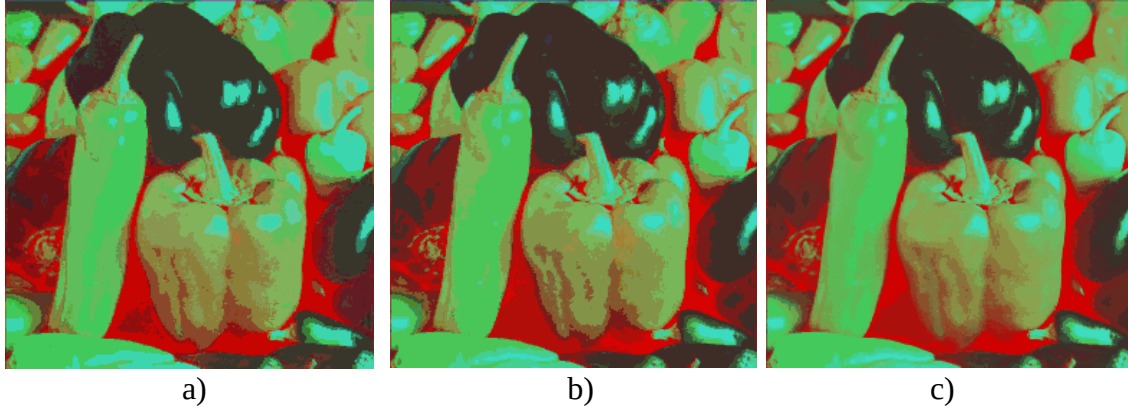
Şekil 4.12. Biber, $m = 2$ a) Otsu, $c=23$ b) Kapur, $c= 21$ c) FCM, $c=23$, süre(s): 13,135

Biber görüntüsü için eşik sayısı ($m = 3$) seçildiğinde tespit edilen eşik değerleri ve hesapsal maliyetler Çizelge 4.11’de verilmiştir. Şekil 4.13(a)’da Otsu amaç fonksiyonu ile bulunan, Şekil 4.13(b)’de Kapur amaç fonksiyonu ile bulunan sonuçlar verilmiştir. Otsu yöntemi ile 48, Kapur yöntemi ile 40 küme meydana gelirken, FCM yönteminde $c=48$ atanarak sınıflandırma yapılmış ve elde edilen sonuç Şekil 4.13(c)’de verilmiştir. FCM metodunun hesaplama süresi 73,439 saniye olmuştur.

Üç eşik ile birlikte hem Otsu hem de Kapur yönteminde görüntü üzerindeki nesneler daha algılanabilir bir şekilde ayırt edilmiş ve sınırlar belirgin hale gelmiştir. Ayrıca görüntü üzerindeki cisimlerin renk tonları daha net bir biçimde ortaya çıkmıştır. Böylece daha fazla detayın yakalanabildiği anlaşılmaktadır. Otsu yönteminde oluşan küme sayısı, Kapur yönteminde oluşan küme sayısından fazladır. Öyle ki Otsu yönteminde oluşan bazı kümelerin Kapur yönteminde meydana gelmediği görülmüştür. Bu açıdan bakıldığında Otsu yönteminin Kapur yöntemine göre daha iyi sonuç verdiği söylenebilir. Çizelge 4.11’den de anlaşılacağı üzere 3 eşik ile birlikte lineer ve FCM yöntemlerinin hesapsal maliyetleri ciddi ölçüde artarken, önerilen yöntemin gerek Otsu gerek ise Kapur ile optimizasyonunun başarılı sonuçlar ürettiği anlaşılmaktadır.

Çizelge 4.11. Biber için bulunan eşikler ($m=3$)

Amaç Fonksiyonu	Otsu						Kapur					
Arama Yöntemi	Lineer			ABC			Lineer			ABC		
Eşikler	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃
R	90	138	177	85	136	176	59	103	162	62	101	164
G	37	98	166	38	101	168	63	119	171	61	114	171
B	29	68	129	30	69	130	58	114	168	56	114	170
Süre(s)	35,963			0,569			67,431			1,200		
Küme sayısı:c	48						40					

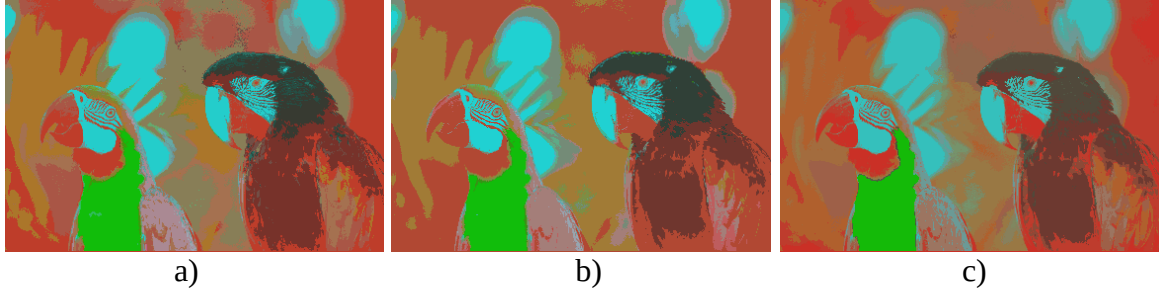


Şekil 4.13. Biber, $m = 3$ a) Otsu, $c=48$ b) Kapur, $c=40$ c) FCM, $c=48$, süre(s): 73,439

Papağanlar görüntüsü için tespit edilen eşik değerleri ($m = 2$) ve hesapsal maliyetler Çizelge 4.12’de verilmiştir. Önerilen yöntemin Otsu amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.14(a)’da, Kapur amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.14(b)’de gösterilmiştir. İki yöntemde de 23 kümenin meydana geldiği gözlenmiştir. $c=23$ atanarak FCM algoritması ile yapılan sınıflandırma sonucu Şekil 4.14(c)’de verilirken, işlem 18,285 saniyede tamamlanmıştır. Eşik sayısı ve küme sayının artışı ile birlikte tek eşikli deneylerde farkedilmeyen arka plandaki nesneler iki eşik ile ortaya çıkarken, papağanlar üzerindeki detayların belirginleştiği görülmektedir. Diğer yandan önerilen yöntemin hem Otsu hem de Kapur ile optimizasyonu hesapsal açıdan FCM yönteminden daha hızlı olup, lineer yöntemlere yakın sonuçlar ortaya koymuştur.

Çizelge 4.12. Papağanlar için bulunan eşikler ($m=2$)

Amaç Fonksiyonu	Otsu				Kapur			
Arama Yöntemi	Lineer		ABC		Lineer		ABC	
Eşikler	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂
R	104	175	104	175	117	183	117	181
G	93	159	93	158	101	161	103	164
B	60	132	61	131	86	167	84	168
Zaman(s)	0,653		0,453		1,126		1,168	
Küme sayısı:c	23				23			



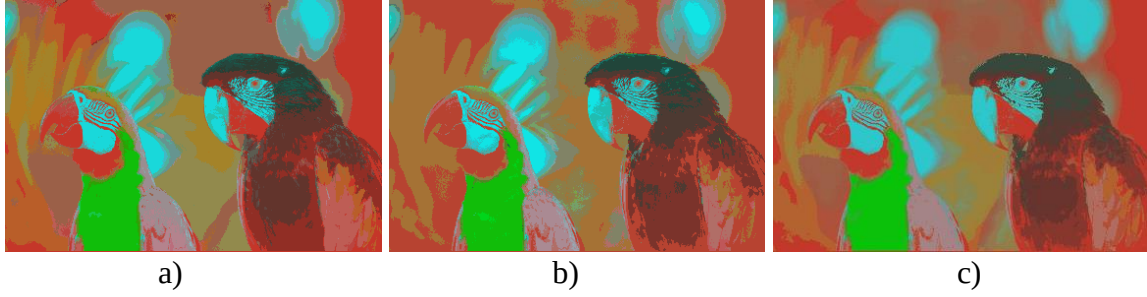
Şekil 4.14. Papağanlar, $m = 2$ a) Otsu, $c=23$ b) Kapur, $c=23$ c) FCM, $c=23$, süre(s): 18,285

Eşik sayısı ($m = 3$) seçildiğinde bulunan eşik değerleri ve hesapsal maliyetler Çizelge 4.13’de gösterilmiştir. Geliştirilen algoritmanın Otsu amaç fonksiyonu ile tespit edilen sonucu Şekil 4.15(a)’da, Kapur amaç fonksiyonu ile tespit edilen sonucu Şekil 4.15(b)’de verilmiştir. FCM algoritmasında $c=45$ atanarak sınıflandırma yapılmış ve Şekil 4.15(c)’de verilmiştir. Söz konusu yöntem ile yapılan sınıflandırma sonucu 93,485 saniyede elde edilmiştir.

Papağanların tüyleri üzerindeki detaylar, farklılıklar ve arka plandaki detaylar üç eşik ile birlikte daha belirgin bir görünüme kavuşmuştur. Bununla birlikte kafa, gaga ve yüz bölgesine bakıldığında başarılı sonuçlar ortaya koyduğu görülmektedir. Yine kanatlardaki renk ve ton değişimlerinin de yansıdığı anlaşılmaktadır. Gerek görüntünün arka planındaki nesnelerin gösteriminde, gerekse papağanların kanat ve gaga bölgesindeki nesnelerin gösteriminde Kapur yönteminin daha başarılı sonuçlar ürettiği söylenebilir.

Çizelge 4.13. Papağanlar için bulunan eşikler ($m=3$)

Amaç Fonksiyonu	Otsu						Kapur					
Arama Yöntemi	Lineer			ABC			Lineer			ABC		
Eşikler	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃
R	85	131	193	82	131	190	111	159	206	110	156	204
G	74	117	170	74	119	173	97	154	206	99	155	209
B	56	108	177	56	106	174	82	137	191	85	139	191
Süre(s)	36,223			0,569			70,045			1,211		
Küme sayısı:c	45						45					

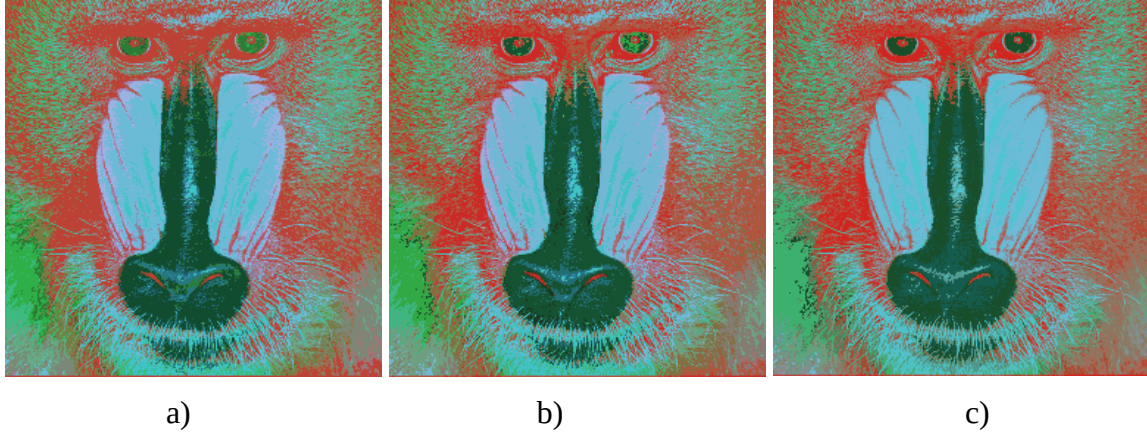


Şekil 4.15. Papağanlar, $m = 3$ a) Otsu, $c=45$ b) Kapur, $c=45$ c) FCM, $c=45$, süre(s): 93,485

Şekil 3.5(b)'de RGB uzayındaki renk dağılımı daha sayrek olan ve farklı noktalarda dağınık bir biçimde konumlanan Babun görüntüsü için elde edilen eşik değerleri ($m = 2$) ve hesapsal maliyetler Çizelge 4.14'de verilmiştir. Geliştirilen metodun, Otsu amaç fonksiyonu ile elde edilen sonucu Şekil 4.16(a)'da, Kapur amaç fonksiyonu ile elde edilen sonucu Şekil 4.16(b)'de gösterilmiştir. Otsu yöntemi ile 25, Kapur yöntemi ile 23 küme meydana gelirken, FCM algoritmasında $c=25$ atanarak yapılan sınıflandırma sonucu Şekil 4.16(c)'de verilmiştir. FCM algoritması ile süreç 13,769 saniyede tamamlanmıştır. Eşik sayısının artışı ile Babunun yüz, burun ve yanak bölgesindeki detayların ortaya çıktığı görülmüştür. Çizelge 4.14 incelendiğinde, Babun görüntüsünde Otsu yöntemi ile meydana gelen küme sayısı Biber ve renkli Lena'da oluşan küme sayısından daha fazladır

Çizelge 4.14. Babun için bulunan eşikler ($m=2$)

Amaç Fonksiyonu	Otsu				Kapur			
Arama Yöntemi	Lineer		ABC		Lineer		ABC	
Eşikler	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂
R	111	178	110	179	98	174	96	174
G	99	155	101	157	61	138	60	136
B	91	167	91	161	91	163	90	162
Süre(s)	0,548		0,444		1,038		1,164	
Küme sayısı:c	25				23			

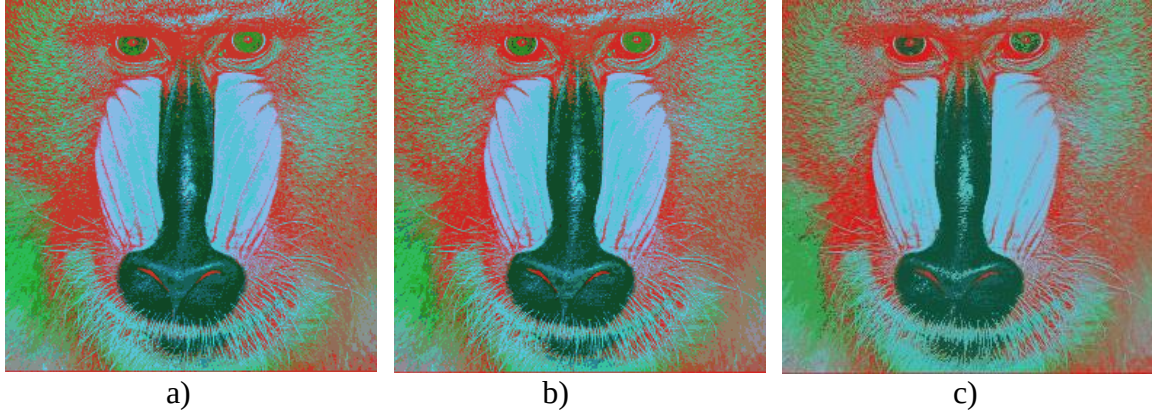


Şekil 4.16. Babun, $m = 2$ a) Otsu, $c=25$ b) Kapur, $c=23$ c) FCM, $c=25$, süre(s): 13,769

Babun görüntüsü için tespit edilen eşik değerleri ($m = 3$) ve hesapsal maliyetler Çizelge 4.15’de verilmiştir. Önerilen yöntemin Otsu amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.17(a)’da, Kapur amaç fonksiyonu kullanılarak elde edilen sonucu Şekil 4.17(b)’de gösterilmiştir. Otsu yöntemi ile 52, Kapur yöntemi ile 49 küme meydana gelmiştir. FCM algoritmasında $c=52$ ile yapılan sınıflandırma sonucu Şekil 4.17(c)’de verilirken süreç 89,748 saniyede tamamlanmıştır. Deneysel sonuçlar dikkate alındığında iki yöntemin de birbirine çok yakın sonuçlar ürettiği görülmektedir. Babunun göz, burun ve tüy detayları üç eşikle birlikte daha da anlaşılır ve belirgin bir vaziyete kavuşmuştur. Çizelge 4.15 incelendiğinde, önerilen yöntemin hesapsal açıdan lineer yöntemle göre başarılı olduğu anlaşılmaktadır. Babun görüntüsünün renk uzayındaki dağılımının daha seyrek olduğu ve farklı noktalarda dağınık bir biçimde konumlandığı daha önceden ifade edilmişti. Çizelge 4.15’de verildiği üzere, oluşan küp veya prizma sayısının renkli Lena, Biber ve Papağanlar görüntülerine göre daha fazla oluşacağı anlaşılmaktadır.

Çizelge 4.15. Babun için bulunan eşikler ($m=3$)

Amaç Fonksiyonu	Otsu						Kapur					
Arama Yöntemi	Lineer			ABC			Lineer			ABC		
Eşikler	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃	t ₁	t ₂	t ₃
R	95	145	199	96	144	198	71	131	191	72	133	189
G	83	125	168	86	127	169	54	109	162	54	106	162
B	70	121	184	71	122	184	67	126	185	66	123	184
Süre(s)	36,045			0,565			73,886			1,223		
Küme sayısı:c	52						49					



Şekil 4.17. Babun, $m = 3$ a) Otsu, $c=52$ b) Kapur, $c=49$ c) FCM, $c=52$, süre(s): 89,748

Deneysel sonuçlar incelendiğinde, önerilen çalışmanın görüntü ayrıştırmanın temel hedefi olan nesneleri ayırt etme, farklı konumdaki pikselleri gruplayarak anlamlı bilgiler elde etme ve benzer pikselleri aynı sınıf altında tasnif etme hedefleriyle büyük oranda örtüştüğü görülmektedir. İlave olarak insan algısına yakın sonuçlar verdiği ve metodun renkli görüntüler için uygulanabilir olduğu anlaşılmaktadır. FCM yönteminde küme merkezlerinin belirlenmesi ve kullanıcı parametrelili olmasının ciddi bir sıkıntı teşkil ettiği daha önce ifade edilmişti. Geliştirilen yöntemin en önemli avantajlarından bir diğeri ise küme sayısının kullanıcı bağımlı bir parametre olmamasıdır.

Tek eşikli sonuçlarda lineer arama yöntemlerinin en hızlı sonuçları verdiği görülmektedir. Önerilen metot ve FCM metodu karşılaştırıldığında ise, önerilen yöntemin her iki deneyinde gerek görüntüdeki detayların algılanması gerekse hesapsal maliyet konularında başarılı olduğu söylenebilir. Hesapsal açıdan bakıldığında, iki eşikli deneylerde lineer ve önerilen yöntemin birbirine yakın sonuçlar ürettiği görülmektedir. Üç eşikli deneylerde, önerilen yöntemin lineer ve FCM yöntemlerine göre ciddi avantaj barındırdığı açıktır. Bununla birlikte görüntü boyutundan bağımsız ve histogram tabanlı bir yaklaşım olduğundan verimli sonuçlar üretmektedir. Böylece önerilen metot gerçek zamanlı uygulamalarda çalışacak bir yapıdadır.

Önerilen yöntemin bahsedilen avantajlarının yanında geliştirilmeye ihtiyaç duyduğu konular mevcuttur. Deneysel sonuçlar incelendiğinde renk uzayında bazı küp veya prizmalarda anlamlı bir bilgiye sahip olmayan pikseller oluşabilir. Örneğin üç eşik seçilmesi halinde en fazla 64 adet küme meydana gelecektir. Fakat görüntü üzerinde 64 farklı nesne olmadığı durumda sınıflandırma işleminde söz konusu sıkıntıların meydana gelmesi kaçınılmazdır.

Bu durum görüntünün renk uzayındaki dağılımı ve seçilecek olan eşik sayısı ile paralel doğrultudadır. Görüntünün renk uzayındaki konumu dağınık bir yapıda olursa eşik sayısının artışı ile birlikte küp veya prizmalar da artacak böylece boş kümelerin meydana gelmesi kaçınılmaz olacaktır. Söz konusu sıkıntılar dikkate alındığında, önerilen metodun bahsedilen sıkıntıları giderecek şekilde iyileştirilmesi gerekmektedir.



5. SONUÇ

Eşikleme yöntemleri görüntü ayrıştırma çalışmalarında sıklıkla tercih edilen yaklaşımların başında gelmektedir. Söz konusu yöntemler gri ölçekli resimlerde sonuç vermesine rağmen, renkli resimlerde verimsiz ve yetersiz kalmaktadır. Renkli görüntüler RGB kanallarından meydana gelmekte olup her bir kanal kendi histogram bilgisine sahiptir. Ayrıca her renk kanalından elde edilen eşik değerlerini bir araya getirmek ciddi bir sıkıntı teşkil etmektedir. Eşikleme yöntemleri söz konusu problemlerden dolayı renkli görüntülerde yetersiz kalmaktadır. Önerilen çalışmada renkli görüntülerin çok seviyeli eşiklenmesi ve sınıflandırılmasına yönelik yeni bir metot geliştirilmiştir. Yapay arı koloni algoritması ile görüntüdeki her bir kanalın eşik değeri Otsu ve Kapur yöntemleri ile hesaplanır. Daha sonra elde edilen eşik değerleri ile RGB renk uzayı çok seviyeli bir biçimde bölütlenir. Alt küp veya prizmanın içerisinde yer alan her piksel aynı sınıfa tahsis edilmiştir. Eşik sayısının artışı ile birlikte renk uzayında oluşan küplerin hacimlerinin küçüldüğü görülmüştür. Bu sayede homojenlik artmıştır. Önerilen yöntem hem gri hem de renkli görüntüler üzerinde sonuç vermektedir. Eşik sayısının artışı ile birlikte karşılaştırılan yöntemlerde hesapsal maliyet problemleri yaşanmaktadır. Özellikle üç eşikten sonra lineer yöntemlerin yetersiz ve verimsiz kaldığı görülmektedir. Fakat yapay arı koloni algoritmasındaki seleksiyon stratejisi ve komşu çözüm tabanlı yapı sayesinde önerilen yöntem, eşik sayısından bağımsız bir biçimde çok daha hızlı sonuçlar üretmektedir. Bu bağlamda geliştirilen metodun ciddi bir avantaja sahip olduğu söylenebilir. Bununla birlikte elde edilen eşik değerlerinin lineer yöntemlere yakın sonuçlar ürettiği görülmektedir.

DeneySEL sonuçlardan anlaşılacağı üzere özellikle tek eşikli sonuçlarda görüntü üzerindeki nesnelerin arka plandaki nesnelerle iç içe geçtiği anlaşılmaktadır. İki ve üç eşik ile birlikte görüntü üzerindeki nesnelerin daha net ve belirgin bir biçimde ortaya çıkarak görüntü ayrıştırma çalışmalarındaki en önemli aşama olan anlamlı sınıflara tahsisin gerçekleştiği görülmüştür. Bu sayede insan algısına yakın sonuçların üretildiği gözlenmiştir. Böylece görüntü ayrıştırma sürecindeki en önemli safhalardan biri olan görüntü analizi ve tanıma işlemi daha mümkün hale gelmiştir.

Geliştirilen algoritmada eşik sayısının artışı ile birlikte sınıflandırma işleminde oluşacak bazı prizma veya küplerin içleri boş kalacak böylece görüntü üzerinde karşılığı olmayan anlamsız kümeler meydana gelecektir. Bununla birlikte görüntü üzerinde gözle görülemeyecek kadar küçük kümeler oluşacaktır. Söz konusu sıkıntıları ortadan kaldırmak adına küme birleştirme algoritmalarından yararlanılabilir. Bu sayede küçük, anlamsız ve piksel yoğunluğu düşük olan kümeler büyük kümelere dağıtılabilir. Önerilen yöntemde bulunması istenen eşik sayısı kullanıcı bağımlı bir yapıda olup, her kanal için ayrı ayrı hesaplanmaktadır. İleriye yönelik çalışmalarda her kanal için en uygun eşikler kullanıcıdan bağımsız bir biçimde hesaplanarak daha etkin sonuçlar ortaya konulabilir.



KAYNAKLAR

1. Gonzalez, R. C. and Woods, R.E. (2014). *Sayısal görüntü işleme*. (Çev, Z. Telatar, H. Tora, F. Arı, ve A. Kalaycıoğlu). Ankara: Palme Yayıncılık. (Eserin orijinali 2008’de yayımlandı). 1-914.
2. Demirci, R. (2006). Rule-based automatic segmentation of color images. *AEU-International Journal of Electronics and Communications*, 60(6), 435-442.
3. Demirci, R., Güvenç, U. ve Kahraman, H. T. (2014). Görüntülerin renk uzayı yardımıyla ayrıştırılması. *İleri Teknoloji Bilimleri Dergisi*, 3(1), 1-8.
4. Incetas, M. O., Demirci, R. and Yavuzcan, H. G. (2014). Automatic segmentation of color images with transitive closure. *AEU-International Journal of Electronics and Communications*, 68(3), 260-269.
5. MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*. Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, 1(14), 281-297.
6. Bezdek, J. C., Ehrlich, R. and Full, W, FCM. (1984). The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3), 191-203.
7. Kapur, J. N., Sahoo, P. K. and Wong, K.A. (1985). New method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3), 273-285.
8. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62-66.
9. Wang, S. and Haralick, R. (1984). Automatic multithreshold selection, *Computer Vision, Graphics, and Image Processing*, 25, 46-67.
10. Reddi, S. S., Rudin, S.F. and Keshavan, H.R. (1984). Optimal multiple threshold scheme for image segmentation, *IEEE Transaction on System Man and Cybernetics*, 14(4), 661–665.
11. Abutaleb, A. S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropy. *Computer Vision, Graphics, and Image Processing*, 47(1), 22-32.
12. Pal, N. R. and Pal, S. K. (1989). Entropic thresholding. *Signal Processing*, 16(2), 97-108.
13. Papamarkos, N. and Gatos, B. (1994). A new approach for multilevel threshold selection. *CVGIP: Graphical Models and Image Processing*, 56(5), 357-370.
14. Yin, P. Y. and Chen, L. H. (1997). A fast iterative scheme for multilevel thresholding methods. *Signal Processing*, 60(3), 305-313.

15. Liao, P. S., Chen, T. S. and Chung, P. C. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, 17(5), 713-727.
16. De Albuquerque, M. P., Esquef, I. A., and Mello, A. G. (2004). Image thresholding using Tsallis entropy. *Pattern Recognition Letters*, 25(9), 1059-1065.
17. Zhang, G. M., Chen, S. P., and Liao, J. N. (2011, March). *Otsu image segmentation algorithm based on morphology and wavelet transformation*. In Computer Research and Development (ICCRD), 2011 3rd International Conference on (Vol. 1, pp. 279-283). Shanghai, China.
18. Huang, M., Yu, W., and Zhu, D. (2012, August). *An improved image segmentation algorithm based on the Otsu method*. At the paper Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on (pp. 135-139). Kyoto, Japan.
19. Wells, W. M., Grimson, W. E. L., Kikinis, R., and Jolesz, F. A. (1996). Adaptive segmentation of MRI data. *The Institute of Electrical and Electronics Engineers Transactions on Medical Imaging*, 15(4), 429-442.
20. Sathya, P. D. and Kayalvizhi, R. (2011). Amended bacterial foraging algorithm for multilevel thresholding of magnetic resonance brain images. *Measurement*, 44(10), 1828-1848.
21. Vidyarthi, A. and Mittal, N. (2013, December). *A hybrid model for extraction of brain tumor in MR images*. At the paper Signal Processing and Communication (ICSC), 2013 International Conference on (pp. 202-206), Noida, India.
22. Karaboğa, D. (2011). *Yapay Zekâ Optimizasyon Algoritmaları, Giriş*, 4. Basım. Ankara: Nobel Yayın Dağıtım, 1 - 232.
23. Yin, P. Y. (1999). A fast scheme for optimal thresholding using genetic algorithms. *Signal Processing*, 72(2), 85-95.
24. Tao, W. B., Tian, J. W. and Liu, J. (2003). Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. *Pattern Recognition Letters*, 24(16), 3069-3078.
25. Hammouche, K., Diaf, M. and Siarry, P. (2008). A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Computer Vision and Image Understanding*, 109(2), 163-175.
26. Ye, Z., Zheng, Z., Yu, X. and Ning, X. (2005, October). *Automatic threshold selection based on ant colony optimization algorithm*. In Neural Networks and Brain. ICNN&B'05. International Conference on (Vol. 2, pp. 728-732). Beijing, China.
27. Tao, W., Jin, H., and Liu, L. (2007). Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recognition Letters*, 28(7), 788-796.
28. Yin, P. Y. (2007). Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Applied Mathematics and Computation*, 184(2), 503-513.

29. Maitra, M. and Chatterjee, A. (2008). A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. *Expert Systems with Applications*, 34(2), 1341-1350.
30. Horng, M. H. and Jiang, T. W. (2010, October). *Multilevel image thresholding selection using the artificial bee colony algorithm*. International Conference on Artificial Intelligence and Computational Intelligence (pp. 318-325). Springer, Berlin, Heidelberg.
31. Zhang, Y. and Wu, L. (2011). Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach. *Entropy*, 13(4), 841-859.
32. Er, F. (2014). *Arı Koloni Optimizasyon Yöntemi İle Gri Seviyeli Görüntü Bölütleme*, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 1-31.
33. Tural, A. (2014). *Çok Seviyeli Eşikleme Probleminin Yapay Arı Kolonisi Algoritmaları ile Çözülmesi*, Yüksek Lisans Tezi, Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü, Kütahya, 1-70.
34. Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 13(6), 3066-3091.
35. Bhandari, A. K., Singh, V. K., Kumar, A., and Singh, G. K. (2014). Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. *Expert Systems with Applications*, 41(7), 3538-3560.
36. Singh, R., Agarwal, P., Kashyap, M., and Bhattacharya, M. (2016, April). *Kapur's and Otsu's based optimal multilevel image thresholding using social spider and firefly algorithm*. Communication and Signal Processing (ICCSP), 2016 International Conference on (pp. 2220-2224). IEEE.
37. Wang, G. G., Deb, S., Gao, X. Z., and Coelho, L. D. S. (2016). A new metaheuristic optimisation algorithm motivated by elephant herding behaviour. *International Journal of Bio-Inspired Computation*, 8(6), 394-409.
38. Tuba, E., Alihodzic, A., ve Tuba, M. (2017, June). *Multilevel image thresholding using elephant herding optimization algorithm*. Engineering of Modern Electric Systems (EMES), 2017 14th International Conference on (pp. 240-243), Oradea, Romania.
39. Tang, K., Xiao, X., Wu, J., Yang, J., and Luo, L. (2017). An improved multilevel thresholding approach based modified bacterial foraging optimization. *Applied Intelligence*, 46(1), 214-226.
40. El Aziz, M. A., Ewees, A. A., Hassanien, A. E., Mudhsh, M., and Xiong, S. (2018). *Multi-objective Whale Optimization Algorithm for Multilevel Thresholding Segmentation*. In Advances in Soft Computing and Machine Learning in Image Processing (pp. 23-39). Springer, Cham.
41. Maurya, L., Sharma, E., Mahapatra, P., and Doegar, A. (2018). *A Hybrid of Fireworks and Harmony Search Algorithm for Multilevel Image Thresholding*. In Advanced Computing and Communication Technologies (pp. 11-21). Springer, Singapore.

42. Zhou, Y., Yang, X., Ling, Y., and Zhang, J. (2018). Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation. *Multimedia Tools and Applications*, 1-29.
43. Sathya, S. S., and Deuri, J. (2018). *Multilevel Thresholding for Image Segmentation Using Cricket Chirping Algorithm*. Bio-Inspired Computing for Image and Video Processing.
44. İncetaş, M. O. (2014). *Tıbbi Görüntülerin Uyarlanabilir Bölge Genişletme Algoritması İle Analizi*, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 1-80.
45. İnternet: Piech Chris. K Means. Stanford. URL:
<http://www.webcitation.org/query?url=http%3A%2F%2Fstanford.edu%2F%7Ecpiech%2Fcs221%2Fhandouts%2Fkmeans.html&date=2019-03-20>, Son Erişim Tarihi: 20.03.2019.
46. Moradi, B. (2015). *Bulanık ve Graf Tabanlı Görüntü Ayrıştırma*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 1-66.
47. İnternet: Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). *Technical report-tr06*, Erciyes university, engineering faculty, computer engineering department.
<http://www.webcitation.org/query?url=https%3A%2F%2Fpdfs.semanticscholar.org%2F015d%2Ff4d97ed1f541752842c49d12e429a785460b.pdf&date=2019-03-22>, Son Erişim Tarihi: 20.03.2019.
48. Tereshko, V. (2000). *Reaction-diffusion model of a honeybee colony's foraging behaviour*. International Conference on Parallel Problem Solving from Nature, 807-816.
49. Akay, B. (2009). *Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial BeeColony) Algoritmasının Performans Analizi*, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri, 1-301.
50. Kahraman, A., Farshi, T. ve Demirci, R. . (2018). Renkli görüntülerin çok seviyeli eşiklenmesi ve sınıflandırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 6(4), 846-859.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KAHRAMAN, Ahmet Selim
 Uyruğu : T.C.
 Doğum tarihi ve yeri : 12.06.1990/ İstanbul
 Medeni hali : Bekâr
 Telefon : 0 (555) 448 19 19
 e-mail : ahmetselimkahraman@gmail.com



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam Ediyor
Lisans	İ.Aydın Üniversitesi / Yazılım Mühendisliği	2014
Önlisans	Trakya Üniversitesi / Bilgisayar Programcılığı	2010
Lise	İstanbul Ticaret Odası ATL / Bilgisayar	2008

İş Deneyimi

Yıl	Yer	Görev
2018-Halen	Borsa İstanbul	Yazılım Geliştirici
2014-2018	TÜBİTAK	Yazılım Geliştirici

Yabancı Dil

İngilizce

Yayınlar

Kahraman, A., Farshi, T. ve Demirci, R. (2018). Renkli görüntülerin çok seviyeli eşiklenmesi ve sınıflandırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 6(4), 846-859.

Hobiler

Spor, Müzik



GAZİ GELECEKTİR..