



**RENKLİ GÖRÜNTÜLERİN ÇOK KATMANLI EŞİKLERLE
AYRIŞTIRILMASI**

Taymaz RAHKAR FARSHİ

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2019

Taymaz RAHKAR FARSHİ tarafından hazırlanan “RENKLİ GÖRÜNTÜLERİN ÇOK KATMANLI EŞİKLERLE AYRIŞTIRILMASI” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Recep DEMİRCİ

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

İkinci Danışman: Doç. Dr. Mohammad-Reza Feizi-Derakhshi

Bilgisayar Mühendisliği, Tebriz Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum

Başkan: Belirsiz

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Belirsiz

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Belirsiz

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Belirsiz

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Belirsiz

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: 28/06/2019

Jüri tarafından kabul edilen bu tezin Doktora 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.

Taymaz RAHKAR FARSHİ

28/06/2019

RENKLİ GÖRÜNTÜLERİN ÇOK KATMANLI EŞİKLERLE AYRIŞTIRILMASI

(Doktora Tezi)

Taymaz RAHKAR FARSHİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2019

ÖZET

Görüntü kümelemede aynı sınıfa atanan piksellerin aynı veya benzer olması beklenir. Başka bir ifade ile oluşan kümenin homojenliği yüksek olmalıdır. Gri seviyeli görüntülerin ayrıştırılmasında belirlenen hedefe eşik sayısı artırılarak ulaşılabilir. Bununla birlikte kullanılacak çoklu eşiklerin tespiti tipik bir problemdir. Ayrıca geleneksel eşikleme algoritmaları renkli görüntülerin ayrıştırılmasında kullanılamamaktadır. Bu çalışmada çok seviyeli eşikleme tekniklerinin renkli görüntülerin kümelenmesi için nasıl kullanılacağı gösterilmiş ve yeni bir sınıflandırma algoritması geliştirilmiştir. Başlangıçta her bir kanal için Otsu ve Kapur yöntemleri ile çok seviyeli eşikler bulunmuştur. Her iki yaklaşımın amaç fonksiyonlarının maksimum değerleri, orman optimizasyon algoritması (OOA) ve parçacık sürü optimizasyon (PSO) algoritması ile ayrı ayrı tespit edilmiştir. Bir sonraki aşamada renk uzayını küçük küplere veya prizmalara bölmek için optimizasyon algoritmaları tarafından belirlenen eşikler kullanılmıştır. Renk uzayında oluşturulan her alt prizma bir küme olarak değerlendirilmiştir. Prizmaların hacimleri oluşturulan kümelerin homojenliğini etkilediğinden, alt küplerin hacimlerini azaltmak için çoklu eşikler kullanılmıştır. Önerilen yöntemin performansı farklı görüntülerle test edilmiştir. Geliştirilen yaklaşımın başarımı bulanık C-ortalamlar (fuzzy C-means: FCM) metodu ile karşılaştırılmıştır. Elde edilen sonuçlar tasarlanan algoritmanın geleneksel yöntemlerden daha verimli olduğunu göstermiştir. Ayrıca FCM tekniğinin hesap maliyeti görüntü boyutu ve küme sayısına bağlı olmasına rağmen, önerilen yöntem bu iki unsurdan bağımsız çalışmaktadır.

Bilim Kodu : 92418

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

Sayfa Adedi : 91

Danışman : Prof. Dr. Recep DEMİRCİ

SEGMENTATION OF COLOR IMAGES WITH MULTILEVEL THRESHOLDS

(Ph. D. Thesis)

Taymaz RAHKAR FARSHI

GAZI UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2019

ABSTRACT

In image clustering, it is desired that pixels assigned in the same class must be the same or similar. In other words, the homogeneity of a cluster must be high. In gray scale image segmentation, the specified goal is achieved by increasing the number of thresholds. However, the determination of multiple thresholds is a typical issue. Moreover, the conventional thresholding algorithms could not be used in color image segmentation. In this study, a new color image clustering algorithm with multilevel thresholding has been presented and, it has been shown how to use the multilevel thresholding techniques for color image clustering. Thus, initially, threshold selection techniques such as the Otsu and Kapur methods were employed for each color channel separately. The objective functions of both approaches have been integrated with the forest optimization algorithm (FOA) and particle swarm optimization (PSO) algorithm. In the next stage, thresholds determined by optimization algorithms were used to divide color space into small cubes or prisms. Each sub-cube or prism created in the color space was evaluated as a cluster. As the volume of prisms affects the homogeneity of the clusters created, multiple thresholds were employed to reduce the sizes of the sub-cubes. The performance of the proposed method was tested with different images. To evaluate the performance of the proposed approach, the algorithm was compared with the most commonly used fuzzy C-means (FCM) clustering method in image clustering. The results were found to be more effective than traditional methods. In addition, although the performance of the FCM method depends on the image size and the number of clusters, the proposed method works independently from these two elements.

Science Code : 92418

Key Words : Image Processing, Clustering; Color space, Multilevel tresholding, Color Image, Segmentation

Page Number : 91

Supervisor : Prof. Dr. Recep DEMİRCİ

TEŞEKKÜR

Bu tezin ortaya çıkmasında çok büyük emeği olan, çalışmalarımın her aşamasında bilgi, öneri ve yardımlarını esirgemeyen danışman hocam Prof. Dr. Recep DEMİRCİ'ye en içten duygularıyla teşekkürü bir borç bilirim. Ayrıca engin bilgi ve tecrübeleriyle çalışmalarına katkı sağlayan değerli hocam Doç. Dr. Mohammad-Reza Feizi-Derakhshi'ye, tavsiyeleri ile yol gösteren Prof. Dr. İlyas ÇANKAYA'ya ve Dr. Öğr. Üyesi Cemal KOÇAK'a, yardımlarını esirgemeyen arkadaşlarım Ali Ufuk YAMAN' a ve Ahmet Selim KAHRAMAN' a teşekkür ederim. Çalışmalarım süresince fedakârlık göstererek beni destekleyen ve zor günlerim dahil her an yanımda olan değerli eşim Sara BEHJAT JAMAL 'e en samimi duygularıyla teşekkür ederim. Ayrıca bugünlere gelmemde emekleri yadsınamaz olan sevgili anneme, babama, ablama ve eşimin ailesine en kalbi duygularıyla teşekkür ederim.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGE LİSTESİ	ix
ŞEKİLLERİN LİSTESİ.....	x
1. GİRİŞ.....	1
2. GÖRÜNTÜ AYRIŞTIRMA VE KÜMELEME	11
2.1. Sayısal Görüntü.....	11
2.1.1. Renk uzayları	11
2.1.2. Gri ölçekli görüntüler.....	12
2.1.3. Renkli görüntüler	13
2.1.4. Görüntü histogramı	14
2.2. Görüntü Kümeleme	17
2.2.1. K-ortalamlar	18
2.2.2. Bulanık C-Ortalamlar	18
2.3. Eşikleme Yöntemleri.....	20
2.3.1. Tek seviyeli eşikleme.....	20
2.3.2. Çok seviyeli eşikleme	22
2.4. Üç Boyutlu Renk Uzayı ile Kümeleme.....	24
3. OPTİMİZASYON YÖNTEMLERİ.....	27
3.1. Sürü Zekâsı	27

3.2. Parçacık Sürü Optimizasyonu	29
3.3. Orman Optimizasyon Algoritması	30
3.3.1. Ağaçları başlatılması.....	31
3.3.2. Ağaçların yerel tohumlaması	32
3.3.3. Popülasyon sınırlayıcı	34
3.3.4. Ağaçların genel tohumlaması.....	34
3.3.5. OOA algoritmasının sözde kodu.....	35
4. RENK UZAYI VE ÇOK SEVİYELİ EŞİKLEME.....	37
4.1. PSO Algoritması ile Çok Seviyeli Eşiklerin Tespiti.....	41
4.2. OOA Algoritması ile Çok Seviyeli Eşiklerin Tespiti.....	42
4.3. Performans Değerlendirme Ölçütleri	43
5. DENEYSEL SONUÇLAR VE YORUMLAR	45
5.1. Geliştirilen Ara Yüz	45
5.2. Deneysel Sonuçlar.....	47
5.2.1. Gri ölçekli görüntülerde alınan sonuçlar.....	47
5.2.2. Renkli görüntülerle alınan sonuçlar	50
6. SONUÇ	83
KAYNAKLAR	85
ÖZGEÇMİŞ	91

ÇİZELGE LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Sınıf etiketi (küp) oluşturma kuralları	25
Çizelge 4.1. İki eşikli sınıf etiketi (küp) oluşturma kuralları	40
Çizelge 5.1. Kameraman için bulunan eşikler	48
Çizelge 5.2. Kameraman performans ölçütleri: F, F' ve Q.....	49
Çizelge 5.3. Lena için bulunan eşikler	51
Çizelge 5.4. Lena performans ölçütleri: F, F' ve Q	51
Çizelge 5.5. Ayakkabı görüntüsü için bulunan eşikler	56
Çizelge 5.6. Ayakkabı performans ölçütleri: F, F' ve Q.....	56
Çizelge 5.7. Mısır görüntüsü için bulunan eşikler	62
Çizelge 5.8. Mısır performans ölçütleri: F, F' ve Q.....	62
Çizelge 5.9. Biber için bulunan eşikler.....	67
Çizelge 5.10. Biber performans ölçütleri: F, F' ve Q	67
Çizelge 5.11. Papağan için bulunan eşikler	71
Çizelge 5.12. Papağan performans ölçütleri: F, F' ve Q.....	71
Çizelge 5.13. Kız Çocuğu için bulunan eşikler.....	75
Çizelge 5.14. Kız Çocuğu performans ölçütleri: F, F' ve Q.....	75
Çizelge 5.15. Test görüntüleri ve bölütleme süreleri (Saniye).	78

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. RGB renk uzayı	12
Şekil 2.2. Gri ölçekli görüntüde renk dağılımı	12
Şekil 2.3. Lena: gri ölçekli.....	13
Şekil 2.4. Renkli görüntüde kanal sıralaması.....	13
Şekil 2.5. Bir pikselin paketlenmiş hali	14
Şekil 2.6. Lena: renkli (200x200)	14
Şekil 2.7. Gri ölçekli Lena görüntüsünün histogramı	15
Şekil 2.8. Renkli Lena ve kanal histogramları a) Kırmızı b) Kırmızı kanalın histogramı c) Yeşil d) Yeşil kanalın histogramı e) Mavi f) Mavi kanalın histogramı	16
Şekil 2.9. Renkli Lena görüntüsünün renk dağılımı	17
Şekil 2.10. Üç boyutlu renk uzayı ve atanan kümeler, $m=1$	25
Şekil 2.11. a) Otsu $t_{r,1}:162, t_{g,1}:102, t_{b,1}:112$ b) Kapur, $t_{r,1}:171, t_{g,1}:140, t_{b,1}:131$	26
Şekil 3.1. Orman Optimizasyon Algoritmasının akış diyagramı	31
Şekil 3.2. m -boyutlu problem uzayı için bir ağaç örneği	32
Şekil 3.3. Bir ağaç üzerinde iki tekrar için yerel tohumlama örneği ($YTD=3$)	33
Şekil 3.4. Dört boyutlu uzayda yerel tohumlama a) $YTD=2$ b) $YTD=1, r'=0,4$ $r' \in [-\Delta x, \Delta x] = [-1, 1]$	33
Şekil 3.5. Dört boyutlu uzayda tek ağaçlı genel tohumlama a) $GTD=2$ b) Sayısal örnek.	35
Şekil 3.6. OOA sözde kodu.....	36
Şekil 4.1. Renkli Lena: üç boyutlu renk uzayı ve kümeleri.....	39
Şekil 4.2. Renkli Lena: renk dağılımı ve atama yapılan sınıfları.....	39
Şekil 5.1. Geliştirilen ara yüz.....	46
Şekil 5.2. Kameraman(300x292) a) Orjinal b) histogramı	49
Şekil 5.3. Kameraman, $m=2$ ve $c=3$ a) Otsu b) Kapur c) FCM, $t=0,891s$	49

Şekil	Sayfa
Şekil 5.4. Kameraman, $m=3$ ve $c=4$ a) Otsu b) Kapur c) FCM, $t=0,917s$	50
Şekil 5.5. Kameraman, $m=4$ ve $c=5$ a) Otsu b) Kapur c) FCM, $t=1,186s$	50
Şekil 5.6. Lena, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=7$ c) FCM, $c=8$, $t=8,150s$	53
Şekil 5.7. Lena, $m=2$ a) Otsu, $c=19$ b) Kapur $c=20$ FCM, $c=20$, $t=11,013s$	53
Şekil 5.8. Lena, $m=3$ a) Otsu, $c=32$ b) Kapur, $c=30$, c) FCM, $c=32$, $t=19,605s$	53
Şekil 5.9. Lena, $m=4$ a) Otsu, $c=51$ b) Kapur, $c=48$ c) FCM, $c=51$, $t=25,863s$	54
Şekil 5.10. Ayakkabı(481x321) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram	55
Şekil 5.11. Ayakkabı, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=6$ c) FCM, $c=8$, $t=19,524s$...	58
Şekil 5.12. Ayakkabı, $m=2$ a) Otsu, $c=19$ b) Kapur, $c=18$ c) FCM, $c=19$, $t=54,051s$...	58
Şekil 5.13. Ayakkabı, $m=3$ a) Otsu, $c=29$ b) Kapur, $c=30$ c) FCM, $c=30$, $t= 101,154s$.	59
Şekil 5.14. Ayakkabı, $m=4$ a) Otsu, $c=52$ b) Kapur, $c=49$ c) FCM, $c=52$, $t= 123,738 s$..	59
Şekil 5.15. Mısır(481x321) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram	61
Şekil 5.16. Mısır, $m=1$ a) Otsu, $c= 8$ b) Kapur, $c= 8$ c) FCM, $c=8$, $t=19,151s$	63
Şekil 5.17. Mısır, $m=2$ a) Otsu, $c= 21$ b) Kapur, $c= 19$ c) FCM, $c=21$, $t= 55,350s$	63
Şekil 5.18. Mısır, $m=3$ a) Otsu, $c= 37$ b) Kapur, $c= 38$ c) FCM, $c=38$, $t= 89,561s$	64
Şekil 5.19. Mısır, $m=4$ a) Otsu, $c= 61$ b) Kapur, $c= 52$ c) FCM, $c=61$, $t=133,562s$	64
Şekil 5.20. Biber (321x481) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram	66
Şekil 5.21. Biber, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=6$ c) FCM, $c=8$, $t=17,175 s$	68
Şekil 5.22. Biber, $m=2$ a) Otsu, $c=18$ b) Kapur, $c=16$ c) FCM, $c=18$, $t=48,723 s$	68
Şekil 5.23. Biber, $m=3$ a) Otsu, $c= 32$ b) Kapur, $c= 27$ c) FCM, $c=32$, $t=89,572s$	68
Şekil 5.24. Biber, $m=4$ a) Otsu, $c= 69$ b) Kapur, $c=54$ c) FCM, $c=69$, $t=157,186s$	69
Şekil 5.25. Papağan (768x512) a) Orijinal b) RGB renk dağılımı c)Alt prizmalar ve kümeler d) RGB Histogram	70
Şekil 5.26. Papağan, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=8$ c) FCM, $c=8$, $t=46,555s$	72

Şekil	Sayfa
Şekil 5.27. Papağan, $m=2$ a) Otsu, $c=24$ b) Kapur, $c=23$ c) FCM, $c=24$, $t=125,738s$	72
Şekil 5.28. Papağan, $m=3$ a) Otsu, $c=47$ b) Kapur, $c=45$ c) FCM, $c=47$, $t=247,708s$	73
Şekil 5.29. Papağan, $m=4$ a) Otsu, $c=84$ b) Kapur, $c=69$ c) FCM, $c=84$, $t=492,019s$	73
Şekil 5.30. Kız çocuğu (765x512) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram	74
Şekil 5.31. Kız Çocuğu, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=8$ c) FCM, $c=8$, $t=31,696s$.	76
Şekil 5.32. Kız Çocuğu, $m=2$ a) Otsu, $c=20$ b) Kapur, $c=20$ c) FCM, $c=20$, $t=152,067s$...	76
Şekil 5.33. Kız Çocuğu, $m=3$ a) Otsu, $c=40$ b) Kapur, $c=37$ c) FCM, $c=37$, $t=215,78s$.	76
Şekil 5.34. Kız Çocuğu, $m=4$ a) Otsu, $c=60$ b) Kapur, $c=64$ c) FCM, $c=60$, $t=362,153s$.	77
Şekil 5.35 Kız Çocuğu, $m=1$, Otsu sınıfların histogramı	80
Şekil 5.36 Kız Çocuğu, $m=2$, Otsu sınıfların histogramı	80
Şekil 5.37 Kız Çocuğu, $m=3$, Otsu sınıfların histogramı	81
Şekil 5.38 Kız Çocuğu, $m=4$, Otsu sınıfların histogramı	81

SİMGELER VE KISALTMALAR

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

Kısaltmalar

Açıklamalar

AB

Amended Bacterial Foraging

ABC

Artificial Bee Colony

FCM

Fuzzy c means

FOA

Forest Optimization Algorithm

GA

Genetic Algorithm

MABC

Modified Artificial Bee Colony

MB

Modified Bacterial Foraging

NFL

No Free Lunch

PSO

Particle Swam Optimization

1. GİRİŞ

Sayısal görüntü üç boyutlu uzaydaki nesnelerin iki boyutlu bir düzlemde sayısal temsidir. Görüntü işleme girişi sayısal görüntü olan bir sinyal işleme sürecidir. Bu sürecin çıktısı yine bir görüntü veya görüntü ile ilgili özellik dizisi veya parametrelerdir [1-3]. Bahsedilen sinyal işleme süreçleri gürültü giderme, ayrıştırma, kenar algılama, sıkıştırma ve şekil tanıma gibi çok çeşitli algoritmalarından oluşur.

Sayısal görüntü işleme alanındaki en zor ve en temel konulardan birisi görüntülerin ayrıştırılmasıdır. Sayısal görüntüdeki nesnelerin diğer nesnelerden ayırt edilmesini sağlayan görüntü ayrıştırma, görüntünün homojen bölgelere ayrılmasını ve aynı bölgede olan piksellerin aynı veya çok benzer olmasını gerektirir [4]. Farklı bölgelerin pikselleri ise farklı olmak zorundadır. Bu konudaki çalışmalar her geçen gün genişleyerek artmakta olup, tıp, askeri, endüstri, uzaktan algılama, güvenlik gibi birçok alanı kapsamaktadır [5,6].

Sayısal bir görüntüyü onu oluşturan alt bölgeler ya da nesneler şeklinde tasnif eden ayrıştırma süreci sonucunda resimdeki nesneler birbirinden izole edilir ve otomatik nesne tanıma sürecinin ilk basamağı gerçekleştirilmiş olur. Diğer taraftan görüntü ayrıştırma ve görüntü kümeleme birbirine çok yakın konular olup, birbiri ile karıştırılmaktadırlar. Ayrıştırmada aynı kümeye atanan piksellerin mekânsal koordinatlar açısından yakın olması gerekirken, kümelemede piksellerin mekânsal koordinatları önemli değildir. Bununla birlikte kümeleme işleminin sonucunda ayrıştırmanın yapılması çok zor bir süreç değildir. Dolayısıyla görüntü sınıflandırma veya kümeleme süreci en temel basamak olarak kabul edilmiştir [7,8]. Günümüze kadar önerilmiş olan sınıflandırma algoritmalarını eşikleme ve kümeleme olmak üzere iki grupta toplamak mümkündür.

Gri ölçekli sayısal görüntünün histogram bilgisini temel alan eşikleme algoritmalarını tek eşikli ve çok eşikli sınıflandırma şeklinde iki gruba ayırmak mümkündür. İkili sınıflandırma olarak ta adlandırılan tek eşikli sınıflandırmada birçok verinin kaybolması muhtemeldir [9-11]. Ayrıca tek eşikli algoritmaların nesne ayırma performansları yetersiz kalabilmektedir. Bu nedenle araştırmacılar çok seviyeli eşikleme yöntemleri geliştirmişlerdir. Çok seviyeli eşikleme ile elde edilen bölgelerdeki homojenlik ve nesne ayırt edebilme yeteneği artırılabilir. Diğer taraftan eşik sayısı arttığında, eşik bulma algoritmalarının hesap maliyeti de logaritmik olarak artmaktadır [12-14].

Gri ölçekli görüntülerde tek ve çok seviyeli eşiklerin bulunması için önerilmiş olan en yaygın algoritmalar Otsu [9] ve Kapur [10] teknikleridir. Söz konusu algoritmalarda görüntünün histogram bilgisi temelli tek ve çok değişkenli amaç fonksiyonları tanımlanmakta ve ilgili fonksiyonları maksimum yapan parametreler eşik olarak atanmaktadır. Ancak ilgili fonksiyonları maksimum yapan noktaların bulunması hesapsal olarak oldukça maliyetli bir süreçtir ve zaman almaktadır. Dolayısıyla birçok görüntü işleme uygulamaları için yetersiz kalmaktadır. İlgili algoritmaların performansını artırmak için araştırmacılar Otsu ve Kapur amaç fonksiyonlarını çeşitli optimizasyon yöntemleri ile birleştirmişlerdir. İlgili çalışmalarda temel hedef amaç fonksiyonunu maksimum yapan parametrelerin hızlı bir şekilde bulunmasıdır [15]. Ayrıca algoritmanın her çalıştırılmasında aynı sonuçların alınması ideal beklentidir. Diğer taraftan Lunch (No free lunch:NFL) teorisine göre her hangi bir optimizasyon algoritmasının her sorunu çözmekte başarılı olması beklenemez [16]. Bu nedenle çoklu eşikleme problemleri için çeşitli optimizasyon algoritmaları önerilmiştir [17-20]. Söz konusu algoritmaların temel amacı doğrusal yöntemle elde edilen değerlere daha yakın, hızlı ve güvenilir sonuçlar elde etmek olmuştur.

Örneğin Hammouche ve arkadaşları tarafından önerilen yöntemde, Otsu ve Kapur bazlı çok düzeyli bir eşikleme yaklaşımı geliştirilmiştir. Bu amaçla gri ölçekle görüntülerin histogramları üzerinde dalgacık dönüşümü uygulanarak orijinal histogramın uzunluğu azaltılmıştır. Akabinde düşük çözünürlüklü histogram üzerinde Otsu ve Kapur amaç fonksiyonları ve genetik algoritma (GA) kullanılarak eşik sayısı ve değerleri tespit edilmiştir [21].

Öte yandan Horng ve Jiang tarafından yapılan çalışmada maksimum entropi fonksiyonu kullanılarak çok seviyeli eşik tespiti yapılmıştır. Üç farklı optimizasyon algoritması: parçacık sürü optimizasyonu (particle swarm optimization: PSO), bal arısı çiftleşme algoritması ve ateş böceği algoritması kullanılmıştır. Gri ölçekli görüntülerde test edilmiş olan algoritmaların hesaplama hızları karşılaştırılmıştır [22].

Gao ve arkadaşları tarafından yapılan başka bir çalışmada Otsu amaç fonksiyonu kullanılmıştır ve daha hızlı sonuç üretmek için PSO algoritması üzerinde değişiklik yapılmıştır. Gri ölçekli görüntülerle deneyler yapılmıştır [23].

Beyin MR görüntülerinin sınıflandırılması, farklı dokuların tespiti ve gri ölçekli görüntülerin ayrıştırılması için Sathya ve Kayalvizhi tarafından Otsu ve Kapur metotları farklı optimizasyon yöntemleri birlikte ile kullanılmıştır. Bu kapsamda PSO, iyileştirilmiş bakteri (Amended Bacterial Foraging: AB), değiştirilmiş bakteri (Modified Bacterial Foraging: MB), GA ve bakteri optimizasyon (Bacterial Foraging: BF) algoritmaları ile sonuçlar alınarak, hız ve verimlilik açısından karşılaştırmalar yapmışlardır [24,25].

Olivia ve arkadaşları tarafından yapılan çalışmada, itme ve çekme mekanizmasını esas alan elektromanyetik optimizasyon (electromagnetic optimization: EMO) algoritması kullanılarak yeni bir eşik belirleme yöntemi önerilmiştir. Otsu ve Kapur amaç fonksiyonları esas alınmıştır ve referans gri görüntüler test edilmiş, sonuçlar istatistiksel olarak karşılaştırılmıştır. Daha az iterasyon ile neticeye ulaşıldığı gözlenmiştir. Önerilen algoritmanın GA, PSO ve BF optimizasyon yönteminden daha başarılı olduğunu iddia edilmiştir [26].

Uydu görüntülerinin daha hızlı ve güvenilir sınıflandırılması için Bhandari ve arkadaşları yapay arı optimizasyon algoritmasının (Artificial Bee Colony: ABC) üzerinde değişiklik yaparak (Modified ABC) Otsu, Kapur ve Tsalli amaç fonksiyonlarını test etmişlerdir. Elde edilen sonuçlar ABC, PSO ve GA algoritmaları kullanarak karşılaştırılmıştır [27].

Roy ve arkadaşları guguk arama optimizasyon (cuckoo search optimization: CS) algoritması ve minimum karşılıklı entropi (minimum cross entropy: MCE) amaç fonksiyonundan faydalanarak sınıflandırma yapmışlardır. Önerilen algoritma referans gri ölçekli görüntüler üzerinde denenmiştir. Ayrıca minimum karşılıklı entropi amaç fonksiyonu seçilerek PSO ve GA algoritmaları karşılaştırılmıştır. Böylece önerilen yöntemin daha hızlı olduğu ve sonuçların doğrusal arama algoritmasıyla elde edilen çıktılarına yakın olduğu gösterilmiştir [28].

Çoklu eşik tahmini için kullanılan bir diğer yaklaşım ise diferansiyel evrim (differential evolution: DE) optimizasyon algoritmasının kullanılmasıdır. Bahsedilen algoritmanın mutasyon stratejisi değiştirilerek uyarlamalı hale getirilmiş ve Otsu amaç fonksiyonu ile gri ölçekli resimler test edilmiştir. Ayrıca elde edilen sonuçlar PSO ve GA algoritmaları kullanarak karşılaştırılmıştır [29].

Li ve arkadaşları ise arıların birebir ile iletişimde olduğu ve çoklu-iş birliği mekanizmasının kullanıldığı ABC algoritmasını geliştirerek Otsu amaç fonksiyonu ile sonuçlar almışlardır. Aynı şekilde gri ölçekli görüntüler üzerinde testler yapılmış ve sonuçlarının GA, PSO ve ABC optimizasyon yönteminden daha başarılı olduğunu iddia edilmiştir [30].

Son yıllarda ise Pal ve arkadaşları örümcek maymunu optimizasyon (Spider Monkey Optimization: SMO) algoritması ile Otsu ve Kapur amaç fonksiyonlularını birleştirerek yeni bir eşikleme algoritması tasarlamışlardır. Önerilen yaklaşımın başarısını ölçmek için gri ölçekli referans görüntüleri kullanılmış ve PSO algoritmasından daha hızlı sonuca ulaştıklarını iddia etmişlerdir [31].

Çok düzeyli eşikleme gri görüntüler için yaygın olarak kullanılsa da renkli görüntülerin çok düzeyli ayrıştırılmasında sınırlı sayıda çalışmalar yapılmıştır. Renkli görüntüler bir boyutlu histogram yerine üç boyutlu bir histograma sahip oldukları için renkli görüntülerin ayrıştırılması zor bir hale gelmiştir ve birçok araştırmacıda çareyi renkli görüntüleri gri ölçekli görüntülere çevirmekte bulmuşlardır. Ancak söz konusu işlem sonucunda üç kanalın verisi (RGB) tek kanala dönüştüğünden, birçok değerli verinin kaybolması kaçınılmazdır.

Kurugollu ve arkadaşları renkli görüntülerin ayrıştırılması için iki boyutlu histogram tabanlı bir algoritma geliştirmişlerdir. Önerilen yaklaşımda renk kanalları RB, RG ve BG çiftleri şeklinde dikkate alınmıştır. Her bir renk çiftinin iki boyutlu histogramlarının tepe noktaları bulunarak ayrıştırma yapılmıştır. Ayrıca iki boyutlu histogram Gauss filtresi ile yumuşatılmış ve görüntü 5 bitlik hassasiyetle yeniden örneklenmiştir. Böylece görüntüdeki birtakım bilgilerin kaybolması kaçınılmaz olmuştur. İlave olarak ilgili tepe noktaların bulunması ve kanal çiftlerinden gelen bilgilerin birleştirme süreci ayrıca bir hesap maliyetine sahiptir [32].

Navon ve arkadaşları renkli görüntülerin ayrıştırılması için uyarlamalı yerel eşik tabanlı bir yaklaşım önermişlerdir. Önerilen algoritmada renkli görüntüler önce gri ölçekli görüntüye dönüştürülmüş, akabinde Canny kenar algılama algoritması ile kenarlar tespit edilmiştir. Şelale (watershed) algoritması ile ayrıştırma işleminin ardından piksel benzerliği esaslı birleştirme yapılmıştır [33].

Kurban ve arkadaşları renkli görüntülerinin çok seviyeli eşikleme problemini çözmek için bir yenilik önermişlerdir. Sunulan çalışmada Kapur amaç fonksiyonu farklı optimizasyon yöntemleri ile birlikte kullanılmıştır. Ancak gri ölçekli görüntülerden farklı olarak k (eşik sayısı) boyutlu problemin, $3 \times k$ boyutlu uzaya dönüştürülmesinin ardından optimizasyon teknikleri kullanılmıştır. Böylece amaç fonksiyonunun her kanal için ayrı ayrı elde edilmesi yerine bütün kanallar aynı fonksiyonda $3 \times k$ boyutlu çözüm modeli ile hesaplanmaktadır. Diğer taraftan optimizasyon algoritmalarında boyut arttıkça doğrusal yöntemlerden elde edilen sonuçlara yakın cevap üretilmesi daha da zorlaşmaktadır. Söz konusu problemin çözümü için optimizasyon algoritmaların popülasyon ve iterasyon sayılarının artırılması gereklidir ki bu artış sonucunda hesap maliyeti de artıracaktır [34]. Ayrıca Kurban ve arkadaşları tarafından geliştirilen algoritma, Cuckoo arama algoritması ile birleştirilerek renkli uydu görüntülerinin çok seviyeli eşiklenmesi için kullanılmıştır [35].

Görüldüğü gibi eşikleme tabanlı sınıflandırma algoritmaları sadece gri ölçekli görüntülerde uygulanmış, renkli görüntüler için yetersiz kalmıştır. Bununla birlikte hem gri hem de görüntülerin sınıflandırılması için, K-ortalamlar (K-means) [36] ve bulanık C-ortalamlar (Fuzzy C-means: FCM) [37,38] en yaygın kullanılan yaklaşımlardır. Kümeleme yöntemleri verileri benzerliklerine göre gruplandırmaktadır. Kümeleme yaklaşımlarında benzer pikseller konumları dikkate alınmaksızın aynı gruba atanırlar. Piksellerin benzerliği Öklid uzaklığı ile değerlendirilir ve aynı gruba atanan piksellerin birbirine çok yakın olması istenilen durumdur. Her iki algoritmada iteratif yapıda olup küme sayısının kullanıcı tarafından belirlenmesi gereklidir. Ayrıca küme merkezleri ile her piksel arasındaki Öklid mesafenin hesaplanması gerektiğinden hesap maliyeti oldukça yüksektir[39]. Başka bir ifade ile algoritmanın hesap süresi görüntülerin boyutları ile direkt orantılıdır. Bu nedenle söz konusu olan algoritmalar gerçek zamanlı uygulamalar için oldukça yetersiz kalmıştır. Diğer taraftan K-ortalamlar ve bulanık FCM algoritmalarının performansını artırmak için bir takım yaklaşımlar önerilmiştir [39,40]. Örneğin, Patel ve Sinha çalışmalarında gri ölçekli görüntü histogramlarını işleyerek uyarlamalı K-ortalamlar algoritması elde etmeye çalışmışlardır [41].

Yao ve arkadaşları K-ortalama algoritmasının en belirgin iki dezavantajını ortadan kaldırmak için histogramın tepe noktaları ve Otsu algoritması yardımıyla kümelerin başlangıç değerlerini tespit etmişlerdir. Önerilen algoritma balık görüntülerinde test edilmiştir [42].

Moftah ve arkadaşları medikal görüntülerdeki meme kanserinin tespiti için uyarlamalı K-ortalamalar yöntemi geliştirilmişlerdir. İlk aşamada K-ortalamalar algoritması düşük iterasyonda çalıştırılır ve ardından iterasyonun devamında uyarlamalı K-ortalamalar algoritması etkin yapılır. Önerilen algoritma ile oluşan nesnelerin dairesellik testi yapılmakta olup, bulunan nesnelerin şekli daireye yakın oluncaya kadar iterasyona devam edilmektedir [43].

Sikka ve arkadaşları FCM algoritmasında küme merkezlerinin başlangıç noktalarını tahmin için MR görüntü histogramlarının tepe noktalarını kullanmışlardır. Ayrıca piksellerin herhangi bir kümeye üyelik fonksiyonunu geliştirerek, küçük alanlı bölgecikleri ortadan kaldırmayı hedeflemişlerdir [44]. Diğer taraftan küme sayısının ve küme merkezlerinin başlangıç değerlerinin tespiti her iki algoritmanın da en temel problemidir. Söz konusu olan problemlerin çözümü için histogram esaslı yaklaşımlar geçen yıllarda önerilmiştir [40,45]. Piksellerin uzaysal konum bilgilerinin FCM algoritmasına entegre edilmesi yaklaşımı ise son yıllarda önerilen çözümler olup, MR görüntülerinde test edilmiştir [46-49].

Ma ve diğerleri tarafından önerilen yöntem ile FCM algoritmasına iki özellik eklenerek daha güçlü bir hale getirilmiştir. Birinci yaklaşımda merkez piksel ve komşu pikseller arasındaki yerel benzerlik ölçüsü, diğerinde ise global benzerlik ölçüsü kullanmıştır. Böylece algoritmanın performansı artırılmıştır [50].

Eşikleme ve kümeleme yaklaşımları gibi görüntülerin ayrıştırılmasında sıkça kullanılan yöntemlerden biri de bölge genişletme algoritmasıdır. Söz konusu algoritma başlangıçta seçilen ve çekirdek olarak adlandırılan piksellerin önceden belirlenmiş benzerlik kriterlerine göre genişletilmesi esasına dayanmaktadır. İlgili çekirdek pikseller otomatik olarak veya kullanıcı tarafından seçilmektedir. Dolayısıyla seçilecek çekirdek piksellerin sayısının ve uzaysal koordinatlarının belirlenmesi kritik önem taşımaktadır. Ayrıca aday piksellerin ilgili çekirdeklere benzerliğinin hesaplanması ve karar verilmesi oluşacak bölgelerin homojenliği ve büyüklüğünü etkilemektedir. Diğer taraftan algoritmanın hesap maliyeti resimdeki piksel sayısı ile orantılıdır. Doğru çekirdek seçilemediğinde homojenlik kriteri zarar görebilmektedir [51].

Fan ve diğerleri bölge genişletme algoritmasını otomatikleştirmek için kenar algılama odaklı bir çekirdek bulma süreci tanımlamışlardır. Bu kapsamda ayrıştırma işlemi uygulanacak

renkli görüntülerde öncelikle otomatik kenar algılama yapılmıştır. Böylece resimdeki temel geometrik şekiller ortaya çıkmıştır. Bununla birlikte kenarlardaki kopmalar veya gereksiz kenarlar anlamsız geometrik yapıların ortaya çıkmasına neden olmaktadır ki ilave algoritma kullanılması gereklidir. Kenarlardan oluşan geometrik yapıların ağırlık merkezleri başlangıç çekirdek noktaları olarak atanmıştır. Geliştirilen algoritma otomatik yüz tanımda kullanılmıştır [52].

Renkli görüntülerde bölge genişletme algoritmasının çekirdek seçimi ve anlamsız düzeyde küçük bölgelerin oluşumunu engellemek için üyelik fonksiyonu esaslı yöntem de önerilmiştir. Önerilen söz konusu algoritmada kenar algılama ve 3×3 lük maske ile bölge genişletmesi yapılmıştır. Ayrıca oluşan bölgelerin homojenlik bilgisi ve piksellerin fiziksel mesafeleri kullanılmıştır. Böylece gereksiz bölge oluşumu engellenmiş ve otomatik çekirdek seçimi yapılmıştır. Geliştirilen algoritma referans görüntülerde test edilmiştir [53].

Shih ve arkadaşları bölge genişletme algoritmasında çekirdek seçiminin otomatik yapılması için renkli görüntüleri YC_bC_r renk uzayına dönüştürerek 3×3 maske tabanlı bir çekirdek seçim süreci önermişlerdir. İlgili maske içerisindeki kalan piksellerin her bir kanal için öncelikle standart sapması hesaplanmış ve akabinde üç kanalın ortalaması alınmıştır. Daha sonra resimdeki en büyük standart sapma $[0-1]$ aralığına normalize edilerek benzerlik hesaplanmıştır. Resme ait benzerlik bilgileri önceden belirlenen değerlerle eşiklenerek ve Öklit mesafesi yardımı ile çekirdek pikseller tespit edilmiştir [54].

Bölge genişletme algoritmasının çekirdek seçimi problemini çözmek için şelale algoritmasının kullanımı bir başka seçenek olmuştur. Şelale algoritması ile elde edilen bölgeler çekirdek olarak atanmış ve akabinde genişletilmiştir [55].

Kullanıcı bağımlı olan bölge genişletme ve şelale algoritmalarının performansını artırmak için graf tabanlı görüntü ayrıştırma algoritması önerilmiştir. Söz konusu yaklaşımda bölge genişletme ve şelale algoritması tarafından elde edilen bölgeler birer düğüm olarak değerlendirilmiştir. Bölgelerin ortalamaları ve varyansları dikkate alınarak ağırlıklı komşuluk matrisi hesaplanmıştır. Herhangi bir aday pikselin en yakın bölgeye atanması daha önceden belirlenmiş olan eşik değeri ve homojenlik bilgisi kullanılarak yapılmıştır [56]. Diğer yandan ön ayrıştırma sonucunda oluşan bölgelerin kenar bilgileri graf tabanlı ayrıştırma yaklaşımına entegre edilmiş renkli ve gri resimlerde sonuçları alınmıştır [57]. Ön

ayırıştırma sonuncunda elde edilen graf, alt graflar şeklinde kesilerek görüntü ayırıştırması yapılmış ve beyin MR görüntülerin sınıflandırılması için kullanılmıştır [58]. Graf tabanlı görüntü ayırıştırmanın farklı alanlarda uygulanması Camilus ve Govindan tarafından geniş çaplı irdelenmiştir [59].

Son yıllarda İncetaş ve arkadaşları kullanıcıdan bağımsız, benzeşim görüntüsü ve graf tabanlı otomatik görüntü ayırıştırma yöntemi geliştirmişlerdir. İlgili çalışma maske tabanlı etiketlemeye dayanmaktadır. Ancak kullanılan maskenin yapısından dolayı fazla ayırıştırma problemi ortaya çıkmaktadır. İlgili problemin çözümü için geçişli kapanış esaslı bir bölge birleştirme yöntemi önerilmiştir. Böylece geleneksel algoritmaların rekürsifliği ortadan kaldırılmıştır. Renkli görüntülerin otomatik olarak ayırıştırılması için önerilen yöntem, Benign Prostatic Hyperplasia (BPH)'li prostat biyopsisi (PB) örneklerinin ayırıştırılmasına olanak sağlayacak şekilde iki aşamalı olarak düzenlenmiştir [60].

Daha önce bahsedilen yaklaşımlara ilave olarak kural tabanlı görüntü ayırıştırma yöntemlerimde yaygın olarak kullanılmıştır. Fisher ve arkadaşları yazı içeren görüntülerde arka plan ve yazıların ayırıştırılması için kural tabanlı bir yaklaşım önermişlerdir. Arka plandan çıkarılan yazı bilgileri çalışma uzunluğu yumuşatma algoritması (run length smoothing algorithm) ile birleştirilmiştir [61].

Moghaddamzadeh ve Bourbakis bulanık mantık ve bölge genişletme algoritmalarını birleştirerek kural tabanlı görüntü ayırıştırma algoritması önerilmişlerdir. Önerilen yöntemde kenar algılama ve bölge birleştirmesi birlikte çalıştırılmıştır [53].

Yannis ve diğerleri tarafından önerilen kural tabanlı görüntü ayırıştırma algoritmasında çok bilinen FCM algoritmasına uzamsal kısıtlamalar eklenmiştir. Aynı sınıfa atanacak piksellerin benzerlik ölçütlerine ilave olarak kural tabanlı konum bilgisi de eklenmiştir. Böylece sınıf içi homojenlik artırılmıştır [62]. Kural tabanlı olarak geliştirilen bir başka FCM algoritmasında ise oluşan bölgelerin ağırlık merkezi ile aday piksel arasındaki konumsal benzerlik hesaplanmış ve kural tabanlı olarak birleştirme yapılmıştır [63].

Demirci bölge büyütme yöntemine benzer kural tabanlı bir ayırıştırma yöntemi geliştirmiştir. Resimdeki piksellerin etiketlerini tutmak için resimle aynı boyutta bir dizi tanımlanmış ve başlangıçta -1 değeri atanmıştır. Bu kapsamda görüntünün sol üst köşesindeki piksel ilk

çekirdek olarak seçilir. Akabinde 3x3 lük maske içindeki benzer komşulara aynı etiket değeri atanır. Piksellerin renk benzerliği bulanık mantığa dayalı kurallarla hesaplanmıştır. Böylece algoritma mevcut bölge sayısı hakkında ön bilgiye ihtiyaç duymamaktadır. Renkli görüntüde üç bağımsız kanal olduğu için benzerlik yüzdesi her kanal için ayrı ayrı hesaplanmıştır. Oluşan bölgelerin homojenliği benzerlik eşik değeri ile kontrol edilmiştir [64].

Son yıllara kadara önerilen algoritmalar ya eşikleme tabanlı olup sadece gri ölçekli resimlerde kullanılmış ya da kümele tabanlı olup hem gri hem de renkli resimlerin sınıflandırılmasında yer almıştır. Kümeleme, graf tabanlı veya bölge genişletme algoritmalarının ortak özelliklerinin hesap maliyetlerinin yüksek olmasıdır. Bununla birlikte 2014 yılında, Demirci ve arkadaşları renk uzayı ve eşikleme yöntemlerini birleştirerek yeni bir sınıflandırma algoritması önermişlerdir. Bu kapsamda renkli bir görüntünün her bir kanalı için ayrı ayrı Otsu, Kapur ve ortalama esaslı eşik seçimi yapılmıştır. Akabinde bulunan eşikler yardımı ile renk uzayı alt prizma ya da kutucuklar şeklinde parçalanmıştır. Her bir kutu içerisinde kalan pikseller aynı sınıfa atanmıştır. Bu yaklaşım ile elde edilecek maksimum sınıf sayısı 8 ile sınırlıdır. Dolayısıyla oluşturulan alt küplerin hacimleri büyük olduğundan ayrıştırma sonucundan elde edilen bölgelerin homojenliği düşük kalmaktadır [65].

Bu çalışmada her bir kanal için çok seviyeli eşikleme yöntemleri kullanılarak elde edilen eşiklerle renk uzayı daha küçük hacimli küpler şeklinde kesilerek, oluşacak sınıfların homojenliğinin artırılması hedeflenmiştir. Çok seviyeli eşiklerin bulunması için Otsu ve Kapur fonksiyonları, orman optimizasyon algoritması (Forest Optimization Algorithm: FOA) [66] ve parçacık sürüsü optimizasyon (Particle Swarm Optimizasyon: PSO) algoritmaları ile birleştirilmiştir. Eşiklerin hesaplanmasında her bir kanalın histogramı kullanıldığından önerilen algoritma resim boyutlarından bağımsızdır. Oluşacak maksimum küme sayısı eşik sayısı ile değişmekle birlikte renk uzayındaki bazı kutucuklar boş kalacağından, oluşacak sınıf sayısı piksellerin renk uzayındaki dağılımına bağlıdır. Önerilen algoritma ile alınan sonuçlar yaygın olarak kullanılan bulanık C-ortalamalar yöntemi ile karşılaştırılmıştır. Ayrıca önerilen yöntemin zaman maliyeti de diğer yöntemlerle karşılaştırılmıştır.



2. GÖRÜNTÜ AYRIŞTIRMA VE KÜMELEME

Sayısal görüntü işleme, bir görüntünün elektronik ortamlara aktarılması ve belirlenen amaç için değiştirilmesidir. Söz konusu olan değişiklikler genel olarak görüntüdeki nesnelerin ayrıştırılması, sınıflandırılması, gürültülerinin arındırılması ve iyileştirilmesi gibi süreçlerden oluşmaktadır. Sayısal görüntü işleme uzaktan algılama ve uzay bilimleri, tıp, savunma sanayi, güvenlik sistemleri ve birçok mühendislik alanlarında kullanılmaktadır. Görüntü analiz çalışmalarının en zorlu konularından birisi de görüntü ayrıştırma işlemleridir. Görüntü ayrıştırma sayısal bir görüntüyü onu oluşturan alt bölgeler ya da nesneler şeklinde tasnif eden bir süreçtir. Bu sürecin sonucunda resimdeki nesneler birbirinden ayırt edilir ve otomatik nesne tanıma sürecinin ilk basamağı gerçekleştirilmiş olur.

2.1. Sayısal Görüntü

Bir sayısal görüntü üç boyutlu uzaydaki nesnelerin iki boyutlu bir düzlemde sayısal temsidir. Görüntü işleme girişi sayısal görüntü olan bir sinyal işleme sürecidir. Bu sürecin çıktısı ise yine bir görüntü veya görüntü ile ilgili özellik dizisi veya parametrelerdir. Elektronik ortamdaki sayısal bir görüntü iki boyutlu bir matris olup, matematiksel açıdan iki değişkenli bir fonksiyondur. Görüntü üzerindeki herhangi bir pikselin konumu iki boyutlu koordinat sistemi ile gösterilir. Böylece bir $f(x_s, y_s)$ şeklindeki görüntü fonksiyonu x_s, y_s uzay koordinatlarındaki aydınlık şiddetini temsil etmektedir. Yatay ve dikey yönde örneklenmiş olan görüntü fonksiyonu $M \times N$ boyutlu bir matris olup,

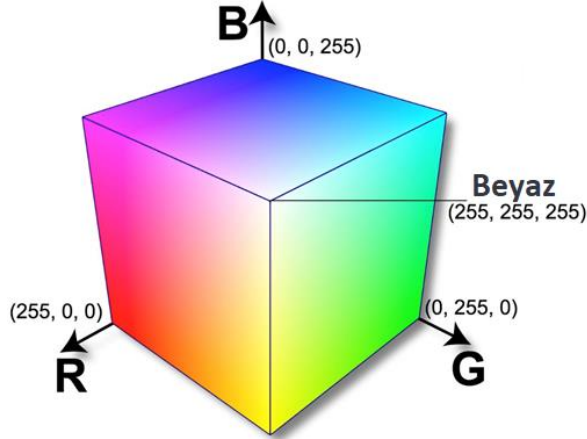
$$f(x_s, y_s) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

şeklinde gösterilir.

2.1.1. Renk uzayları

Görüntüyü oluşturan pikseller gerçekte farklı dalga boyutlarındaki elektromanyetik dalgaların yoğunluklarını temsil etmektedir. Gözün algıladığı dalga boyları 400-700 nanometre aralığında olup, söz konusu aralık kırmızı (Red:R), yeşil(Green:G) ve mavi (Blue:B) kanallar şeklinde tasnif edilmiştir. Böylece herhangi bir piksel RGB olarak

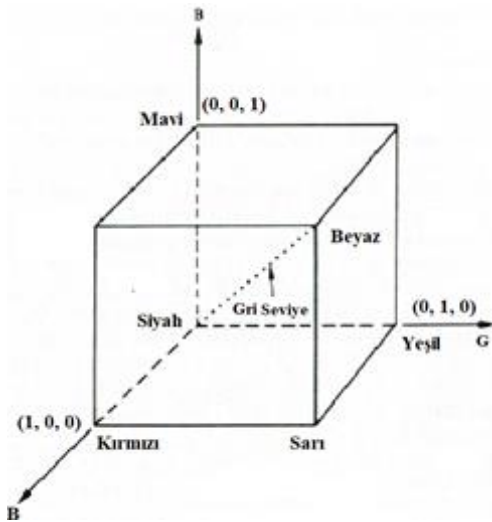
adlandırılan üç boyutlu renk uzayında bir noktaya karşılık gelmektedir. Söz konusu RGB renk uzayında piksellerin veya renklerin dağılımı Şekil 2.1'de gösterilmiştir. Her bir renk boyutu 8 bitlik kuantalamaya tabii tutulduğunda, $255 \times 255 \times 255$ farklı renk tonu elde edilmektedir. Böylece her bir piksel için 24 bitlik hafızaya ihtiyaç duyulur.



Şekil 2.1. RGB renk uzayı

2.1.2. Gri ölçekli görüntüler

Bir görüntünün pikselleri renk uzayındaki (0,0,0) ve (255,255,255) köşegenlerinin doğrultusunda dağıtılsa o görüntüye gri ölçekli görüntü denilir. Gri ölçekli görüntünün RGB renk uzayında dağılımı Şekil 2.2 de gösterilmiştir. Gri ölçekli görüntülerde piksellerin renk değerleri [0-255] aralığında olup, her piksel için 8 bitlik hafıza alanına ihtiyaç vardır. Referans resimlerden olan Lena'nın gri ölçekli hali Şekil 2.3 de gösterilmiştir.



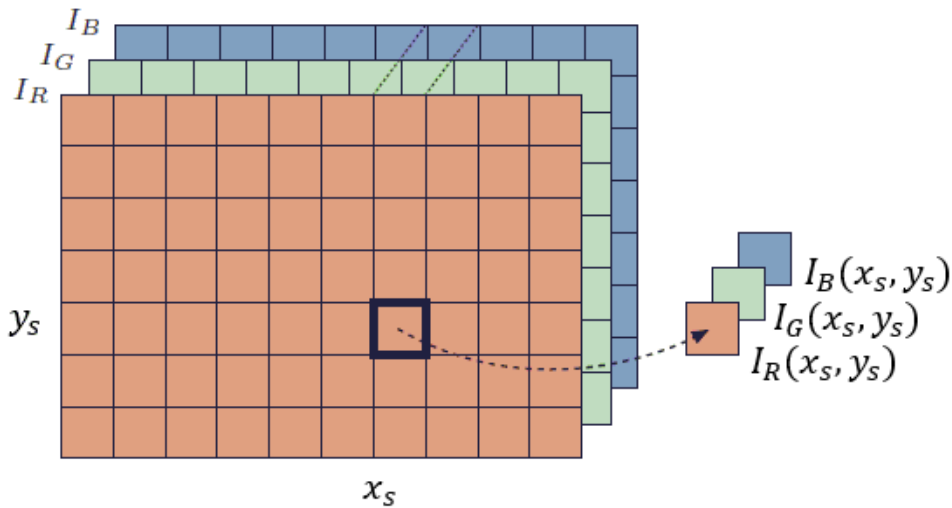
Şekil 2.2. Gri ölçekli görüntüde renk dağılımı



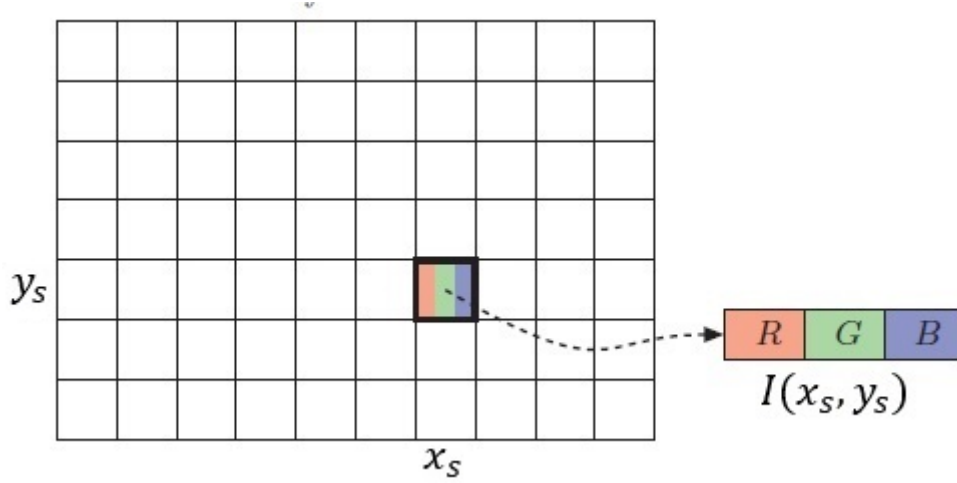
Şekil 2.3. Lena: gri ölçekli

2.1.3. Renkli görüntüler

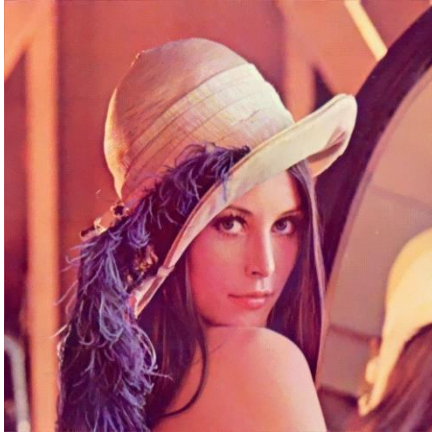
Renkli görüntülerde her piksel Şekil 2.4 ve Şekil 2.5' de gösterildiği gibi kırmızı, yeşil ve mavi olmak üzere, üç kanaldan oluşur. Gri ölçekli görüntülerde olduğu gibi renkli görüntülerde bir pikselin her bir kanalının alacağı değerler [0-255] aralığındadır. Bu nedenle her bir piksel için 24 bitlik hafızaya ihtiyaç duyulur. Şekil 2.6 da orijinal Lena görüntüsü gösterilmiştir. İnsan gözünün algıladığı dalga boyu [400-700] nanometre aralığında olup, retina duvarındaki koni ve çubuk hücreleri yardımıyla ışık bilgisi elektriksel sinyallere dönüştürülmektedir. Çubuk hücreleri gri ölçekli dönüşümde, koni hücreleri ise renkli görüntü oluşturmakta görev alırlar. Diğer taraftan koni hücrelerini de 400 ile 700 nm arasındaki ışık tepkilerine göre kırmızı, yeşil ve mavi koni hücreleri olarak tasnif etmek mümkündür. Bu açıdan bakıldığında RGB renk uzayı insan gözünün yapısına ve algılama sisteme en yakın modeldir [67,68].



Şekil 2.4. Renkli görüntüde kanal sıralaması [69].



Şekil 2.5. Bir pikselin paketlenmiş hali [69].



Şekil 2.6. Lena: renkli (200x200)

2.1.4. Görüntü histogramı

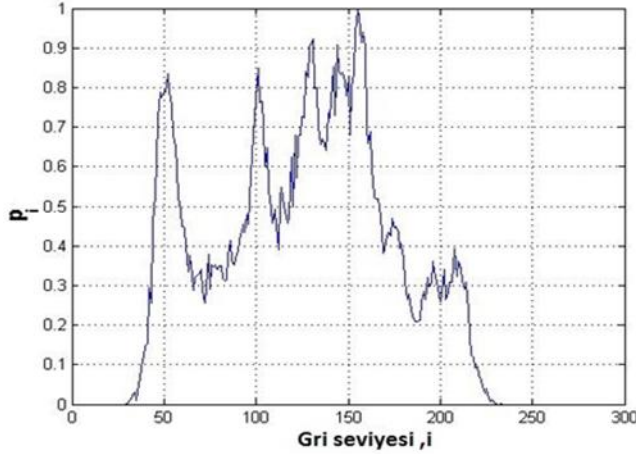
Bir sayısal görüntünün histogramı her parlaklık düzeyindeki piksel sayısını ifade eden bir grafikdir. 8 bitlik gri ölçekli bir görüntüde her piksel $\{0,1,2,\dots,L-1\}$ aralığında bir değer alabilmektedir. $0 \leq i \leq L-1$ olmak üzere, bir pikselin i . gri seviye olasılığı

$$p_i = h_i / (M \times N) \quad (2.2)$$

şeklinde tanımlanabilir. Burada M ve N görüntünün boyutlarını, h_i i . yoğunluk seviye sahip piksellerin sayısını temsil etmektedir. Şekil 2.3' de gösterilen gri ölçekli Lena görüntüsünün histogramı Şekil 2.7' de verilmiştir. Renkli görüntülerde ise üç ayrı kanal bulunduğundan,

her bir kanalın histogramı ayrı ayrı alınmalıdır. Şekil 2.8 de renkli Lena görüntüsün kanal ve histogramları verilmiştir.

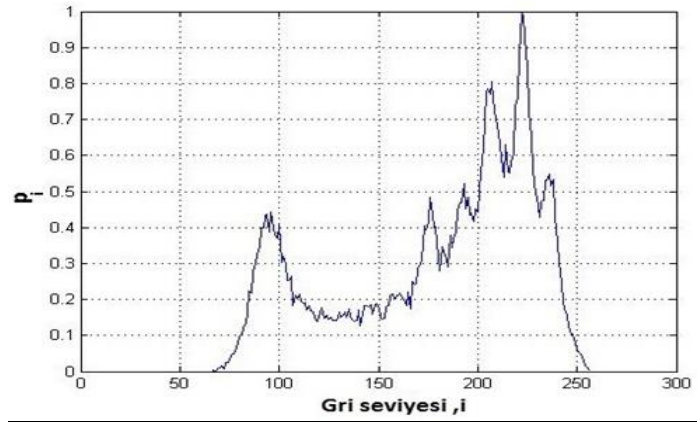
Şekil 2.8 (a)' da kırmızı kanal bileşeni ve Şekil 2.8 (b) histogramı gösterilmiştir. Yeşil kanal bileşeni Şekil 2.8 (c)' de, ilgili kanalın histogramı ise Şekil 2.8 (d) verilmiştir. Şekil 2.8 (e) mavi kanal bilgisini ve Şekil 2.8 (f) ise mavi kanalın histogramını göstermektedir. Görüldüğü gibi her bir kanalın grafikleri farklıdır. Buda görüntüdeki renk dağılımlarının aynı olmadığının göstergesidir. Görüntü üzerindeki fiziksel konumları ne olursa olsun her bir piksel aynı zamanda RGB renk uzayında en az bir noktaya tekabül etmektedir. Ayrıca sayısal görüntüdeki piksellerinin tamamının ya da bir kısmının da RGB renk uzayında bir noktaya karşılık gelmesi mümkündür. Eğer bir sayısal görüntüde 255x255x255 tane farklı piksel mevcut ise Şekil 2.1' de gösterilen RGB renk uzayı tamamen doldurulacaktır. Bu durum her zaman mümkün değildir. Örneğin Şekil 2.6' de gösterilen renkli Lena görüntüsünün RGB renk uzayında dağılımı Şekil 2.9 da gösterilmiştir. Görüldüğü gibi Lena görüntüsü RGB renk uzayında kısmi bir dağılım göstermiştir.



Şekil 2.7. Gri ölçekli Lena görüntüsünün histogramı



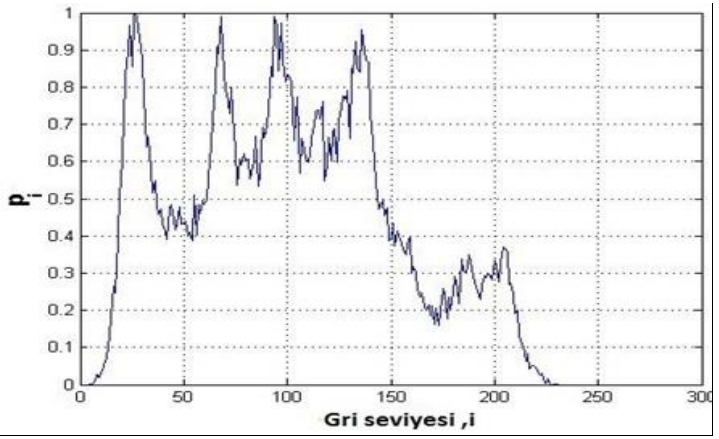
a)



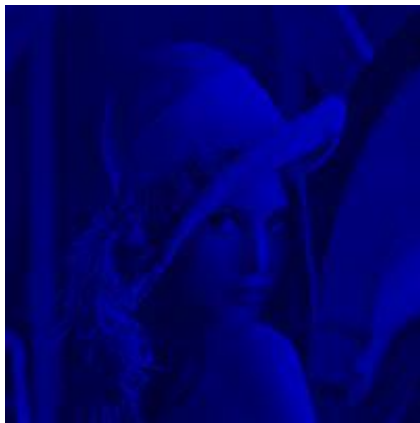
b)



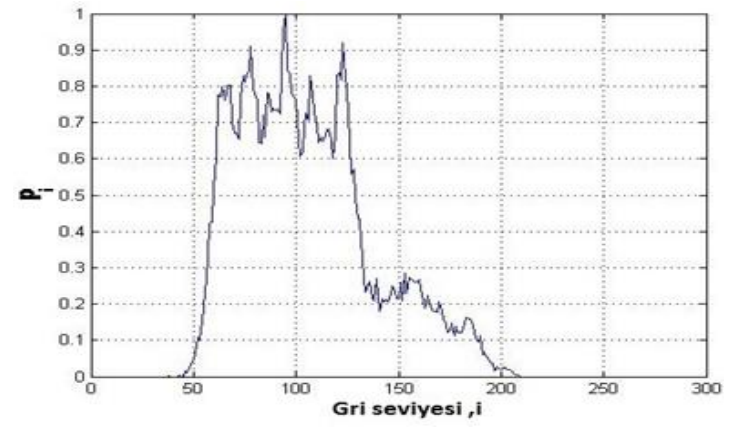
c)



d)

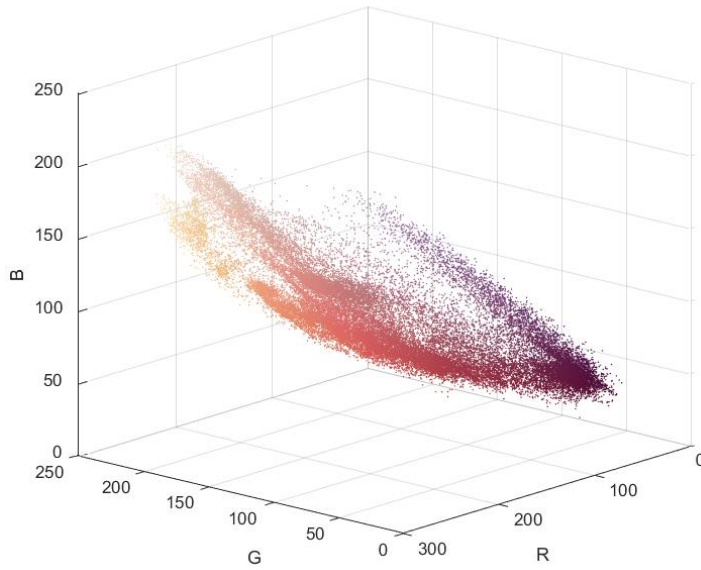


e)



f)

Şekil 2.8. Renkli Lena ve kanal histogramları a) Kırmızı b) Kırmızı kanalın histogramı c) Yeşil d) Yeşil kanalın histogramı e) Mavi f) Mavi kanalın histogramı



Şekil 2.9. Renkli Lena görüntüsünün renk dağılımı

2.2. Görüntü Kümeleme

Sayısal görüntü analizi çalışmalarının en zor ve önemli konularından biri de görüntü ayrıştırma işlemleridir. Bu sebeple görüntü ayrıştırma işlemleri, araştırmacıların dikkatini çekmiş ve bu konu hakkında çok sayıda çalışma yürütülmüştür. Görüntü ayrıştırma, bir görüntünün farklı ihtiyaçlar doğrultusunda görüntünün farklı bölgelere ayrılması işlemidir. Bu işlem sırasında homojen olan, yani benzer değerlerdeki pikseller aynı gruba dâhil edilirler [70]. Bu gruplama aşamasında gri seviyeli görüntüler için parlaklık değeri, renkli görüntüler için 3 kanallı RGB değerleri kullanılır [71]. Görüntü ayrıştırmada ayrıştırılan bölgeler konuma bağlı olarak uzamsal eşleştirme ile de sınıflandırılır. Ayrıştırmanın amacı görüntüyü anlamlı yapılara parçalamak ve kolaylıkla analiz edilebilecek bir görüntü elde etmektir [72].

Görüntü ayrıştırma metotlarını üç ana grupta inceleyebiliriz [69]. Piksel tabanlı ayrıştırma; histogram eşikleme ve renk uzayı kümeleme işlemlerini içerir. Bölge tabanlı ayrıştırma ise ayrıştırma, birleştirme ve bölge geliştirme algoritmalarından oluşmaktadır. Diğer taraftan kenar tabanlı ayrıştırmaları ise yerel teknikler ve genel teknikler olarak gruplamak mümkündür.

Kümelemede pikseller benzerliklerine göre gruplandırılırlar. Bu işlemde konumlar hesaba katılmadan birbirlerine benzer pikseller homojen gruplara ayrılırlar. Renkleri birbirine daha çok benzeyen, yani renk mesafeleri daha az olan pikselleri aynı gruba dâhil edilir. Görüntü kümelemenin görüntü ayırtırmadan tek farkı; ayırtırmada kümeler konuma bağlı olarak sınıflandırılırken, kümelemede konum ve uzamsal bölge eşleştirme söz konusu değildir.

2.2.1. K-ortalamlar

Coleman ve diğerleri [36], k-ortalamlar kümeleme yönteminden yararlanarak bir görüntü ayırtırma metodu geliştirmişlerdir. K- ortalamlar algoritması bir grup nesneyi K adet kümeye bölerek ayırtırma yapar ve her nesne kendine en yakın olan ortalama kümeye dâhil olur. Bu yöntemde küme sayısını gösteren K parametresi, kullanıcı tarafından tanımlanan bir parametredir. Bu yöntemde başlangıçta gerekli sayıda rastgele noktalar seçilir ve ilgili noktaların her biri bir kümenin merkezi olarak tanımlanır. Daha sonra veriler (pikseller) merkezine en yakın olan bir kümeye atanır ve böylece yeni kümeler oluşturulmuş olur. Bu sürecin tekrarlanmasıyla her iterasyonda kümelerin ortalaması hesaplanarak yeni merkez noktaları elde edilmekte ve veriler (pikseller) yeniden yeni bir kümeye atanmaktadır. Bu süreç ya daha önceden belirlenen iterasyon sayısına ulaşana kadar ya da küme merkezlerinde değişiklik gözlemlenmeyene kadar devam eder:

$$J = \sum_{i=1}^Z \sum_{j=1}^C \|q_i - c_j\|^2 \quad (2.3)$$

Burada Z piksel sayısı, C küme sayısı, c_j j . kümenin merkezidir. q_i ise i . veriyi (piksel) ifade etmektedir. K- ortalamlar algoritmasının temel problemi, algoritmanın rasgele merkezlerle başlaması sebebiyle kesin bir sonuca ulaşamaması veya rastgelelik sebebiyle her seferinde farklı sonuçlara ulaşıyor olmasıdır. Başlangıç merkezlerinin hesaplanması için takip edilmesi gereken belirli bir rutin bulunmamaktadır. Daha önceden küme sayısının belirli olması gereklidir.

2.2.2. Bulanık C-Ortalamlar

1973 yılında, Dunn [73] tarafından ortaya koyulan ve 1981 yılında Bezdek [37] tarafından iyileştirilen FCM algoritması, verileri aynı anda birkaç kümeye dâhil etmek için önerilmiştir.

FCM algoritması bir u üyelik matrisini kullanır ve her bir veri için bütün kümelerle bir üyelik değeri atar. Söz konusu değerler 0 ile 1 arasında olmaktadır. Başlangıçta ilgili üyelik değerleri rasgele olarak atanır. FCM yöntemi seçilen amaç fonksiyonunun minimizasyonuna dayanmaktadır. Bu amaçla seçilen fonksiyon

$$J = \sum_{i=1}^Z \sum_{j=1}^C u_{ij}^b \|q_i - c_j\|^2 \quad (2.4)$$

şeklindedir. Burada b değeri j . küme için bulanıklık miktarını ve u_{ij}^b , j . kümenin her q_i için üyelik derecesini göstermektedir. c_j ise küme merkezlerini ifade etmektedir. İlgili kümelerin üyelik değeri ve küme merkezleri

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|q_i - c_j\|}{\|q_i - c_k\|} \right)^{\frac{2}{b-1}}} \quad (2.5)$$

ve

$$c_j = \frac{\sum_{i=1}^Z u_{ij}^b q_i}{\sum_{i=1}^Z u_{ij}^b} \quad (2.6)$$

ifadeleri ile hesaplanmaktadır. Algoritma belli bir iterasyon sayısına gelene kadar devam edebilir veya iki iterasyon arasındaki değişim miktarı önceden belirlenen bir değerden az olursa algoritma durdurulur. τ iterasyon sayısı olmak üzere, eğer $\max_{ij} \left\{ \left| u_{ij}^{(\tau+1)} - u_{ij}^{(\tau)} \right| \right\} < \varepsilon$ şartları sağlandığında algoritma duracaktır. İterasyon sayısı arttıkça işlemin zaman sarfıyatı da büyümektedir [74].

K-ortalamlar ve FCM, veri sınıflandırma yaklaşımına göre iki yaygın görüntü kümeleme algoritmalarıdır. Bu teknikler küme sayısı belirli olan kümeleme işlemlerinde görüntüleri başarılı şekilde sınıflandırabilmektedir. Bununla birlikte, eğer küme sayısı belirli olmaz ise algoritmanın sonuç üretmesi mümkün değildir. Üstelik bu algoritmalar tekrarlamalıdır.

Böylece her piksel üzerinde defalarca işlem yapıldığından hesap maliyeti çok fazla artacaktır. Ayrıca kümelerin merkezleri rastgele atandığından, kümeleme işleminin doğru sonuçlara ulaşabilmesi için algoritmanın birden fazla çalıştırılması gereklidir. Dolayısıyla her seferinde farklı sonuç alınması kaçınılmazdır ve bu durum algoritmanın güvenilirliğini sorgulanır hale getirmektedir. Eş. 2.1 de görüldüğü gibi Z piksel sayısını temsil etmekte olup, sınıflandırılacak görüntü büyüdükçe hesap maliyeti de artacaktır.

2.3. Eşikleme Yöntemleri

Daha önce bahsedilen görüntü sınıflandırma yöntemleri görüntü boyutuna bağlıdır. Diğer taraftan görüntülerdeki bilgilerin tek boyutlu bir diziye ya da histograma dönüştürülmüş halini kullanan eşikleme yöntemleri görüntü boyutundan bağımsız çalışmaktadır. Bu nedenle hem hızlı hem de tercih edilen bir yaklaşım olmuştur. Örneğin Şekil 2.7 de ki histogramın Lena görüntüsünün her bir parlaklık düzeyindeki piksel sayısını göstermektedir. Başka bir ifade ile Lena görüntüsü tek bir grafikte temsil edilmiştir. Sınıflandırmada amaç benzer piksellerin aynı grup altında toplanması olduğundan ilgili grafikler ya da histogramlar çok kullanışlı hale gelmektedir. Dolayısıyla histogram üzerinde seçilecek noktalar yardımıyla sınıflandırma yapmak mümkündür.

Söz konusu eşikleme yöntemlerinde gri ölçekli görüntüler için oluşan sınıf sayısı eşik sayısının bir fazlasıdır. Dolayısıyla tek eşik kullandığında iki sınıf, m tane eşik te ise $m+1$ tane sınıf oluşmaktadır. Eşik seçimi için önerilen yöntemler eşik değerlerini içeren çok değişkenli bir fonksiyonun seçimine ve ilgili fonksiyonun maksimum yapılması prensibine dayanmaktadır.

2.3.1. Tek seviyeli eşikleme

Eşikleme yöntemleri bugüne kadar görüntü ayrıştırma algoritmaları içinde en sık kullanılan yöntemlerden biri olarak tanımlanmıştır. Görüntüdeki nesnenin tek bir eşik değeri kullanılarak arka plandan ayrılması tek seviyeli eşikleme olarak adlandırılır [9,10]. En yaygın şekilde kullanılan eşikleme yöntemleri Kapur'un entropi kriteri yaklaşımı ve Otsu'nun sınıflar arası varyans tekniğidir. Kapur metoduna göre tek seviyeli eşiklemede amaç fonksiyonu

$$J(t_1) = H_0 + H_1 \quad (2.7)$$

gibi seçilir. Burada H_0 ve H_1 histogramın kısmi entropileri olup,

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

ve

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

şeklinde hesaplanır. Eş. 2.7 de verilen amaç fonksiyonunu maksimum yapan t_1 değeri eşik değeri olarak atanır. Diğer taraftan Otsu algoritmasında tanımlanan amaç fonksiyonu

$$J(t_1) = \sigma_0 + \sigma_1 \quad (2.8)$$

şeklinde ifade edilmiş olup, σ_0 ve σ_1 değerleri sınıflar arası varyansı temsil etmektedir. Söz konusu varyanslar

$$\sigma_0 = \omega_0 (\mu_0 - \mu_T)^2 \quad \text{ve} \quad \sigma_1 = \omega_1 (\mu_1 - \mu_T)^2$$

ifadeleri ile hesaplanırlar. Sınıfların ortalamaları ve kısmi olasılıkları ise

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

ve

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

ve

$$\mu_T = \sum_{i=0}^{L-1} i p_i$$

şeklinde bulunmaktadır. Burada μ_T görüntünün genel ortalamasını göstermek üzere,

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T \text{ ve } \omega_0 + \omega_1 = 1$$

şartları sağlanmalıdır.

Sınıflandırmanın amacı aynı veya benzer olan piksellerin aynı gruba atanmasıdır. Görüntü histogramı tek eşik ile iki sınıfa ayrıldığında farklı tonlardaki pikseller aynı sınıfa toplanır. Dolayısıyla oluşan sınıfların homojenliği istenilen düzeyde olmayıp, düşük kalma ihtimali mevcuttur. Başka bir ifade ile görüntüdeki bazı bilgilerin kaybolması muhtemeldir. Bu yüzden çok seviyeli eşikleme kaçınılmaz hale gelmiştir.

2.3.2. Çok seviyeli eşikleme

Tek seviyeli eşik kullanılarak görüntü sınıflandırmanın problemleri daha önceki bölümde tartışılmıştır. Eşikleme kullanılarak homojen sınıflar elde etmenin bir diğer çözümü ise eşik sayısını artırmak olmuştur. Başka bir ifade ile eğer eşik sayısı m tane seçilirse $(t_1, t_2, \dots, t_m)$, gri seviyeli görüntü $m+1$ sınıf şeklinde kümelenecektir. Söz konusu kümeler için eşik değerleri $C_0 [0, \dots, t_1-1]$, $C_1 [t_1, \dots, t_2-1]$ ve $C_m [t_m, \dots, L-1]$ şeklinde olacaktır. Matematiksel açıdan problem histogram bilgisi kullanılarak m tane eşik değerinin nasıl tespit edileceğidir. İlgili problemin çözümü için önerilen yaklaşımlar tek seviyeli eşiklemede kullanılan amaç fonksiyonlarının çok değişkenli fonksiyon şeklinde genişletilmesi şeklinde olmuştur. Böylece Kapur yaklaşımında çok seviyeli eşikleme için önerilen amaç fonksiyonu

$$J(t_1, t_2, \dots, t_m) = H_0 + H_1 + H_2 + \dots + H_m \quad (2.9)$$

şeklinde tanımlanmıştır. Burada $H_0, H_1 \dots H_m$ histogramın kısmi entropileri olup,

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

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

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

$$\vdots$$

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

ifadeleri ile hesaplanırlar. Öte yandan Otsu metoduna göre çok seviyeli eşiklemede önerilen amaç fonksiyonu

$$J(t_1, t_2, \dots, t_m) = \sigma_0 + \sigma_1 + \sigma_2 + \dots + \sigma_m \quad (2.10)$$

şeklinde ifade edilmiş olup, $\sigma_0, \sigma_1 \dots \sigma_m$ değişkenleri sınıflar arası varyansı temsil etmektedir. İlgili varyanslar ise

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

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

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

$$\vdots$$

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

ifadeleri ile elde edilmektedir. Sınıflar arası varyansların hesaplamasında kullanılan sınıf ortalamaları ve kısmi olasılıkları ise

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

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

$$\vdots$$

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

şeklinde bulunmaktadır. Her iki amaç fonksiyonunu $t_1 < t_2 < t_3 < \dots < t_m$ kısıtları ile maksimum yapan parametreler çözüm kümesi olarak atanır. Bu açıdan bakıldığında çok düzeyli eşikleme m boyutlu bir optimizasyon problemidir [14]. Ayrıca yukarıda bahsedilen amaç fonksiyonları ile bulunan eşik değerleri sadece gri seviyeli görüntülerin sınıflandırılması için geçerli olmuştur. Renkli görüntülerde ise üç ayrı kanal olduğundan bu yöntemlerin direkt uygulanması mümkün olmamıştır.

2.4. Üç Boyutlu Renk Uzayı ile Kümeleme

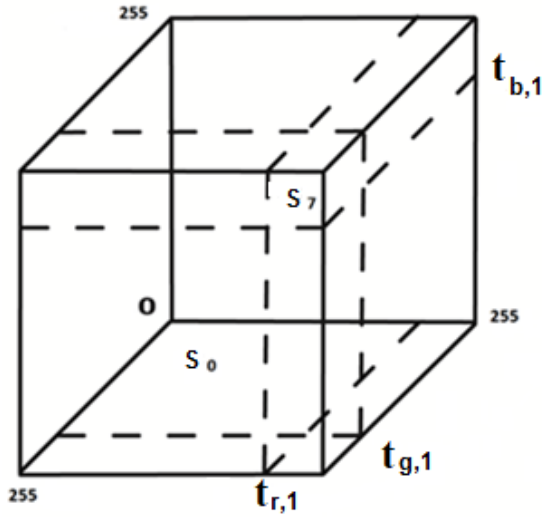
Gri seviyeli görüntülerde tek seviyeli ve çok seviyeli eşikleme yapılarak görüntü kümeleme mümkündür. Diğer taraftan renkli görüntülerde üç ayrı kanal olduğu için her bir kanal için eşikler hesaplanırsa bile birleştirmesi açık problem olarak kalmıştır. Renkli görüntülerin tek seviyeli eşik ve renk uzayı kullanılarak eşiklemesine yönelik ilk çalışma Demirci ve arkadaşları tarafından önerilmiştir [65]. Önerilen yöntemde Otsu ve Kapur amaç fonksiyonları esas alınmıştır. Daha sonrasında tek seviyeli eşikleme işlemi hem gri hem de renkli görüntüler üzerinde gerçekleştirilmiştir. Bu kapsamda eşikleme sonuçlarından elde edilen eşik değerlerinin birleştirilmesi için renk uzayı kullanılarak çözüm üretilmiştir. Şekil 2.10'da gösterildiği gibi eşikler renk uzayının alt kümelerini oluşturmak için kullanılır. Başka bir ifade ile RGB renk uzayı, alt küpler veya prizmalara bölünür. Sonrasında herhangi bir alt küp veya prizma içinde kalan pikseller aynı kümeye atanırlar. Başlangıçta bu yaklaşımın ilgisiz kümeler oluşturulabileceği düşünülebilir ancak yapılan deneylerde elde edilen sonuçlar oldukça mantıklıdır. Şekil 2.10'a dikkatle bakıldığında her bir alt küpün veya prizmanın hacminin her bir kanalın eşik değerlerine bağlı olduğu ve aynı olmadığı kolayca görülebilir. Bu nedenle alt küp veya prizma şekilleri, görüntüdeki piksel yoğunluğunun dağılımları ile ilgilidir. Sonuç olarak, farklı görüntüler için benzersiz bir eşik kümesi ile farklı kesitler elde edilecektir.

Göründeki piksellerin ilgili kümelerle atanması renk uzayını bölümlenmede kullanılan kurallarına dayalı olarak yapılmaktadır. Yani renk uzayında oluşan her bir küpe bir indis atanmakta ve aynı indis uzamsal koordinatlarda dağılmış olan pikselleri etiketlemede kullanılmaktadır. Önerilen algoritmada kullanılan sınıf numaraları, sınıf (küpe) oluşturma kuralları ve sınıf etiketleri Çizelge 2.1'de verilmiştir. Çizelge 2.1'de $t_{r,0}$, $t_{g,0}$ ve $t_{b,0}$ eşikleri her zaman sıfır olarak atanan boş eşiklerdir. Ancak renk uzayında oluşan küplerin

etiketlemesinde eşik indisleri kullanıldığından, etiketleme algoritmasının anlaşması için kullanılmıştır. C_n n . kümeyi veya renk uzayındaki n inci küpü göstermek üzere

$$n_{10} = d_r (m+1)^2 + d_g (m+1)^1 + d_b (m+1)^0 \quad (2.11)$$

ifadesi tanımlanmıştır. Burada n küme etiketini, d_r, d_g ve d_b sırasıyla $(m+1)$ tabanlı sayı sisteminde oluşan rakamları temsil etmektedir. Oluşan üç basamaklı sayının birinci hanesini mavi kanal, ikinci hanesini yeşil kanal ve üçüncü hanesini kırmızı kanal eşik indisleri oluşturmaktadır.



Şekil 2.10. Üç boyutlu renk uzayı ve atanan kümeler, $m=1$ [65].

Çizelge 2.1. Sınıf etiketi (küpe) oluşturma kuralları

Sınıf etiketi C_n	Sınıf (Küp) oluşturma kuralları	Sınıf etiket kodu ($d_r d_g d_b$) $_{m+1}$
C_0	eğer ($R > t_{r,0}$ ve $R \leq t_{r,1}$) ve ($G > t_{g,0}$ ve $G \leq t_{g,1}$) ve ($B > t_{b,0}$ ve $B \leq t_{b,1}$)	000
C_1	eğer ($R > t_{r,0}$ ve $R \leq t_{r,1}$) ve ($G > t_{g,0}$ ve $G \leq t_{g,1}$) ve ($B > t_{b,1}$ $B \leq 255$)	001
C_2	eğer ($R > t_{r,0}$ ve $R \leq t_{r,1}$) ve ($G > t_{g,1}$ ve $G \leq 255$) ve ($B > t_{b,0}$ ve $B \leq t_{b,1}$)	010
C_3	eğer ($R > t_{r,0}$ ve $R \leq t_{r,1}$) ve ($G > t_{g,1}$ ve $G \leq 255$) ve ($B > t_{b,1}$ $B \leq 255$)	011
C_4	eğer ($R > t_{r,1}$ ve $R \leq 255$) ve ($G < t_{g,0}$ ve $G \leq t_{g,1}$) ve ($B > t_{b,0}$ ve $B \leq t_{b,1}$)	100
C_5	eğer ($R > t_{r,1}$ ve $R \leq 255$) ve ($G > t_{g,0}$ ve $G \leq t_{g,1}$) ve ($B > t_{b,1}$ $B \leq 255$)	101
C_6	eğer ($R > t_{r,1}$ ve $R \leq 255$) ve ($G > t_{g,1}$ ve $G \leq 255$) ve ($B > t_{b,0}$ ve $B \leq t_{b,1}$)	110
C_7	eğer ($R > t_{r,1}$ ve $R \leq 255$) ve ($G > t_{g,1}$ ve $G \leq 255$) ve ($B > t_{b,1}$ ve $B \leq 255$)	111



a)



b)

Şekil 2.11. a) Otsu $t_{r,1}:162$, $t_{g,1}:102$, $t_{b,1}:112$

b) Kapur, $t_{r,1}:171$, $t_{g,1}:140$, $t_{b,1}:131$

Otsu ve Kapur amaç fonksiyonları esas alınarak elde edilen eşik değerleri Şekil 2.6'da gösterilen renkli Lena görüntüsünün ayrıştırılması için kullanılmıştır. Şekil 2.11 (a) Otsu yöntemiyle elde edilen ayrıştırma sonucunu gösterirken, Şekil 2.11 (b) ise Kapur yöntemiyle alınan çıktıyı göstermektedir. Şekil 2.11'de görüldüğü gibi ayrıştırma işi tam olarak yapılamamaktadır ve bazı bilgiler kaybolmaktadır. Söz konusu yaklaşım, eşikleme yöntemleri kullanarak renkli görüntüleri farklı bölgelere ayırmakta ancak yetersiz ayrıştırma yapmaktadır.

3. OPTİMİZASYON YÖNTEMLERİ

Mühendisler ve bilim insanları birçok karmaşık problemler ile karşı karşıya kalmaktadırlar. Söz konusu problemler; yöneylem araştırmaları, mekanik sistemlerin tasarımı, görüntü işleme uygulamaları ve endüstriyel verimlilik gibi farklı alanlarda ortaya çıkmaktadır. Bu alanların temel problemi ise en iyi parametrelerin tespit edilmesindeki zorluklardır. Dolayısıyla bilinmeyen parametrelerin bulunmasını içeren tüm problemler bir optimizasyon problemi olarak tanımlanabilir. Ayrıca optimizasyon işlemi değişken türüne bağlı olarak ayrık ve sürekli optimizasyon olmak üzere ikiye ayrılır. Optimizasyon problemlerinde bir ya da birden fazla amaç fonksiyonu tanımlanabilir. Aynı zamanda bütün parametreler göz önüne alınarak, seçilen amaç fonksiyonu minimize ya da maksimize edilmeye çalışılır. Bu açıdan bakıldığında optimizasyon en iyi sonucun bulunması için yürütülen bir süreçtir. Böylece bir bilim insanının veya bir mühendisin önerdiği fikir, düşünce ya da plan, optimizasyon süreci sonucunda daha düzenli bir hale gelmiş olur. Optimizasyon sürecinde başlangıç koşulları farklı yöntemlerle değerlendirilir. Elde edilen veriler, söz konusu fikir, plan ya da düşüncenin iyileştirilmesi için kullanılır. Optimizasyon, farklı problemlerin nasıl çözüme kavuşturulacağını bulması için kullanılan matematiksel bir araç olarak karşımıza çıkmaktadır [75]. Optimizasyonda, herhangi bir problemin çözümünde birbirinden farklı değerler arasında en iyi değerin bulunması amaçlanır. Problemin değerlendirilmesinde ise en iyi cevabın bulunması, kullanılan yöntem ve hata sınır değerlerine bağlıdır. Bu nedenle problemin formüle edilmesi, en iyi cevabın bulunmasını doğrudan etkilemektedir. Araştırmacılar tarafından şimdiye kadar birçok optimizasyon algoritması önerilmiştir [76,77]. Bu algoritmalar genelde doğadan ilham alınarak çözüme ulaştırırlar.

3.1. Sürü Zekâsı

Sürü zekâsı, seçilen problemlere çözüm getirirken; termitler, arılar, karıncalar, kuşlar ve balık sürüleri gibi davranışlarını içinde yaşadıkları topluluğa göre ayarlayan canlıların tutumlarını örnek alan bir yapay zekâ tekniğidir. Arı kolonilerinin kovan etrafında dolaşarak birbirlerine bilgi aktarımları, karıncaların geçtikleri yollara kimyasal madde bırakarak diğer karıncalara bilgi aktarımları, kuş sürülerinin ve balık sürülerinin konum ve hızlarını ayarlayarak ilerlemeleri bu canlıların topluluk bilincine göre yaptıkları davranışlara verilebilecek en iyi örneklerdendir.

Bir sürünün başlıca iki işlevi vardır. Bunlardan birisi kendi başına organize olabilme (*self-organization*), diğeri ise iş bölümüdür. Kendi başına organize olmada kastedilen; sürüdeki temel birimlerin, diğ birimlerle olan etkileşimleri sonucu aldıkları bilgilerle kendi başlarına işlev görmeleri ve sonucunda sistemin bütünü etkilemeleridir. Sistemin diğ birimleri ile etkileşimlerinde söz konusu olan şey; sistemin bir bütün olarak genel başarımından ziyade yakın komşularla bilgilerinin paylaşımıdır. Bonabeau ve arkadaşları kendi başına organize olabilmeyi dört özellik ile karakterize etmişlerdir. Bunlar; pozitif geri besleme (*positive feedback*), negatif geri besleme (*negative feedback*), salınımlar (*fluctuation*) ve çoklu etkileşimdir. Pozitif geri besleme daha uygun yapıların bulunmasını sağlamaktadır. Karıncaların geçtikleri noktalara kimyasal maddeler bırakması ve arıların daha zengin nektar kaynakları hakkındaki bilgileri diğ arılara iletmek için dans etmeleri pozitif geri beslemeye örnek olarak gösterilebilir. Negatif geri beslemede ise esas olan toplanan bilgilerin daha kararlı bir hale getirilmeye çalışılmasıdır. Salınımlar, yeni kaynakların keşfedilmesi için yapılan rastgele dolaşımalar olarak tanımlanabilir. Çoklu iletişim, bir bireyin farklı bir bireye ait bilgiyi kullanabilmesini ifade etmektedir. İş bölümü ise topluluktaki her bireyin aynı anda farklı işler yapmasını tanımlamaktadır. Bu şekilde çalışması, birbirine tabi olmayan bireylerin gösterdikleri performanstan daha üstün bir performans ortaya konulmasını ve araştırma uzayındaki değişimlere cevap verilmesini sağlar [78].

Literatürde yapay arı kolonisi algoritmasıyla birlikte sürü zekâsı üzerine çalışan ve arıların davranışlarını model alan farklı yöntemler bulunmasına rağmen, ayırık problemlerin çözümünde sadece Yang'ın geliştirdiği Sanal Arı (*Virtual Bee*) algoritması kullanılabilmektedir. Bu algoritma ise yalnızca problemlerin iki boyutlu olduğu durumlarda kullanılmaktadır [79]. Yapay arı kolonisi algoritması, Karaboğa tarafından geliştirilmiş [78] ve literatürde var olan algoritmalarla kıyaslandığında oldukça iyi sonuçlar üretmiştir. Yapay arı kolonisi algoritmasının sinyal işlemede sıkça kullanılan, yapay sinir ağlarının eğitiminde gösterdiği performans analiz edilmiş ve elde edilen performans farksal gelişim (*Differential Evolution, DE*) [80] ve parçacık sürü optimizasyon (*Particle Swarm Optimization, PSO*) algoritmalarıyla kıyaslanmıştır [75].

3.2. Parçacık Sürü Optimizasyonu

Doğada sürü halinde yaşayan canlıların, özellikle de balık ve kuş sürülerinin garip davranışları gözlemlenebilmektedir. Ancak sürünün her bir üyesi çok kısıtlı bilgilere sahiptir ve sadece birkaç komşunun konumundan haberdardır. Böyle bir sürüde, hayvanların her biri sadece birkaç basit kuralı takip eder. Sürüdeki bütün bu karmaşık davranışlar basit kuralların birleşimi ile ortaya çıkmaktadır. Kenedy ve Eberhart [75] doksanlı yılların ortasında bu davranışlardan ilham alarak parçacık sürü optimizasyon algoritmasını önermişlerdir. Parçacık sürü algoritmasında bir noktalar dizisi oluşturulur ve her bir noktaya önceden belirlenmiş başlangıç değerleri atanır. Aynı zamanda her parçacığa bir hız vektörü tanımlanır. Söz konusu hız vektörleri rastgele değerler alan değişkenlere göre yenilenir. Her parçacık kendisi için belirlenmiş olan hız vektörüne göre belirli bir açıyla hareket ettirilir. Her parçacığın hızı ve konumunun hesaplanması için

$$V_i(\tau + 1) = W V_i(\tau) + r_{p,1} C_{p,1}(x_i^{best} - x_i) + r_{p,2} C_{p,2}(x_g^{best} - x_i) \quad (3.1)$$

$$x_i(\tau + 1) = x_i(\tau) + V_i(\tau + 1)$$

ifadesi kullanılır. Burada W eylemsizlik kat sayısını, V_i her parçacığın her boyuttaki hızını göstermektedir. x_i her bir parçacığın bulunduğu konumu, x_i^{best} her bir parçacığın şimdiye kadar ulaştığı en iyi konumu ve x_g^{best} ise tüm parçacıkların şimdiye kadar ulaştığı en iyi konumu ifade etmektedir. Ayrıca $r_{p,1}$ ve $r_{p,2}$ 0 ile 1 arasında rasgele atanan birer sayıdır. $C_{p,1}$, her bir parçacığın ulaştığı en iyi konum bilgisinin bir sonraki adımda kullanma kat sayısı ve $C_{p,2}$, ise tüm parçacıkların ulaştığı en iyi konum bilgisinin bir sonraki adımda kullanma kat sayısıdır. Sürü zekâsı uygulamalarında gözlemlenen en temel sürü kurallarına göre, bireyler hızlarını birbirine göre ayarlar ve birbirleriyle olan mesafelerini kontrol ederek birbirlerine çarpmamaya çalışırlar. Aynı zamanda birbirlerine olan yakınlıklarını da korumaya çalışırlar. Bu davranışları ise bireysel zekâlarını artırarak daha iyi sonuç almalarını sağlar [81].

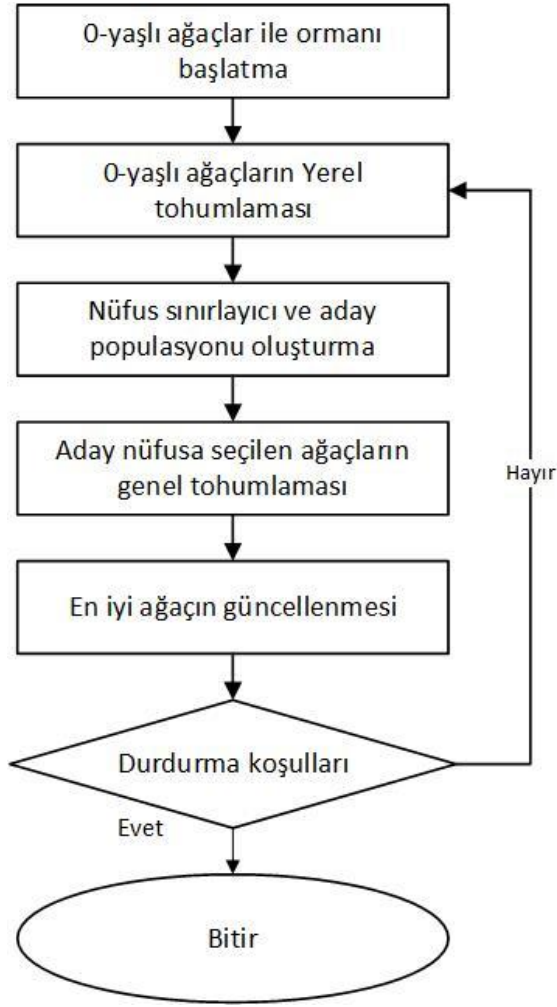
3.3. Orman Optimizasyon Algoritması

Ghaemi ve Feizi [66], tarafından 2014 yılında önerilene orman optimizasyon algoritması (OOA), herhangi bir ormanda belirli bir yıl hayatta kalan ağaçların yaşam mücadelesinden esinlenilerek tasarlanmıştır. Bu algoritma,

1. Ağaçların yerel tohumlaması,
2. Nüfusun sınırlandırılması,
3. Ağaçların genel tohumlaması.

şeklinde üç ana aşamadan oluşmaktadır ve akış diyagramı Şekil 3.1’ de detaylı gösterilmiştir. Diğer evrimsel algoritmalar gibi OOA da ağaçların ilk nüfusu ve konumları rasgele başlatılır. Böylece her ağaç problemin potansiyel bir çözümünü temsil eder. Her bir ağaç değişkenlerin değerlerine ilave olarak ilgili ağacın yaşını temsil eden bir bölüme da sahiptir. Başlangıçta her bir ağacın yaşı ‘0’ olarak ayarlanmaktadır. Ağaçların başlatılmasından sonra, yerel tohumlamada yeni ağaçlar 0 yaş ile ormana eklenir. Aynı zamanda yeni oluşturulmuş ağaçlar hariç ormandaki diğer ağaçların yaşları '1' yaş artırılır.

Daha sonra ormandaki ağaçların uygunluk fonksiyonları ve yaşları dikkate alınarak nüfus kontrolü yapılmaktadır. Belirlenen kriterler dahilinde bazı ağaçlar ormandan kaldırılarak aday nüfus kategorisine alınır. Akabinde aday nüfus kategorisinden bazı ağaçlar kullanılarak genel tohumlama yapılır. Aslında genel tohumlama yerel çözümlerden kurtulmak için ormana bazı yeni potansiyel çözümler eklemektedir. Sonra ormandaki ağaçlar uygunluk değerlerine göre sıralanarak en iyi ağaç seçilir. En iyi ağaç olarak tespit edilen ağacın yaşı 0 atanır.



Şekil 3.1. Orman Optimizasyon Algoritmasının akış diyagramı [66].

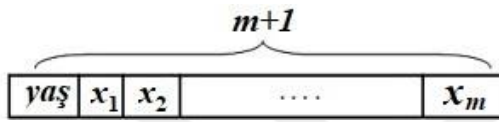
3.3.1. Ağaçları başlatılması

Orman optimizasyon algoritmasında (OOA), seçilen problemin potansiyel çözümünü bir ağaç olarak kabul edilir. Değişken bilgilerine ek olarak ilgili ağacın yaş bilgisi de kayıt altında tutulmaktadır. Başlangıçta 0 olarak atanan ağaç yaş bilgisi, yerel tohumlamadan sonra 1 artırılmaktadır. Ağaçların yaş artışı, ormandaki ağaçların sayısını kontrol etmekte kullanılan bir mekanizmadır. Bir m -boyutlu problemin çözüm kümesi Şekil 3.2’te ki gibi gösterilir. Şekil 3.2’te x_m değişkenlerin değerini ve ‘Yaş’ kısmı ise ilgili ağacın yaşını göstermektedir. Matematiksel olarak her bir ağaca ait vektör;

$$\text{Ağaç} = [\text{yaş}, x_1, x_2, \dots, x_m] \quad (3.2)$$

şeklinde tanımlanır. Eş. 3.2’te görüldüğü gibi her bir ağaç vektör uzunluğu $1 \times (m + 1)$ şeklinde bir dizidir. Bir ağacın izin verilen maksimum yaşı önceden tanımlanmış bir

parametredir ve “*ömür parametresi*” olarak adlandırılır. Ömür parametresi algoritma başlangıcında tespit edilmelidir. Bir ağacın yaşı ömür parametresine ulaştığında, o ağaç ormandan çıkarılır ve aday nüfusa eklenir. Eğer “*ömür parametresi*” çok büyük seçilirse, algoritmanın her tekrarında sadece ağaçların yaşları artacak ve yerel tohumlama sonucunda yeni ağaçlar ormana eklenemeyecektir. Böylece orman eski ağaçlarla dolacaktır. Ters durumda “*ömür parametresi*” çok küçük seçildiğinde, ağaçlar çok çabuk yaşlanacak ve rekabet etme şansı kalmayacaktır. Dolayısıyla optimal çözüme ulaşma ihtimali düşecektir. Bu nedenle ilgili parametrenin doğru seçimi algoritmanın iyi bir şekilde yerel arama yapmasını sağlar.



Şekil 3.2. m -boyutlu problem uzayı için bir ağaç örneği [66].

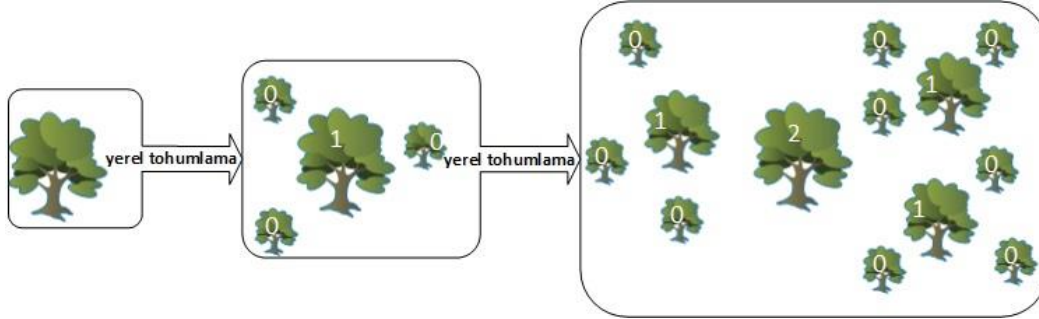
3.3.2. Ağaçların yerel tohumlaması

Doğada ağaçların tohumlama işlemi başladığı zaman, bazı tohumlar sadece ağaçların yakınına düşer ve bir süre sonra genç ağaçlara dönüşür. Sonrasında, yakın ağaçların arasında rekabet başlar ve yeterli güneş ışığı alarak daha iyi bir konumda büyüyen ağaçlar, hayat yarışının kazananları olurlar. Algoritmadaki ağaçların yerel tohumlama aşaması doğanın bu prosedürünü taklit etmektedir. Bu operatör, yaşı 0 olan ağaçlar üzerinde uygulanmakta ve ormanda her ağaca bazı komşular eklemektedir. Şekil 3.3'te bu operatörün bir ağaç üzerinde ikinci tekrara kadar süreci gösterilmiştir. Şekil 3.3'te ağaçların içinde yazılı sayılar, ilgili ağacının “Yaş” değerini göstermektedir ve yeni üretilen her ağacın yaşı sıfır olarak kabul edilmektedir. Yerel tohumlama ile yaşı 0 olan ağaçların yaşı, bu aşamada yeni oluşturulan ağaçlar hariç 1 arttırılır. Ağaçların artan yaşı, ormandaki ağaçların sayısını kontrol eden bir mekanizma gibi davranmaktadır. Yerel tohumlama operatörü ormana birçok ağaç ekleyeceğinden, ağaçların sayısında bir sınırlama olmalıdır. Dolayısıyla ormandaki ağaçların sayısını kontrol etmek için bazı ağaçlar genel tohumlama aşamasında kullanılmak üzere ormandan çıkarılarak aday nüfus kategorisine gönderilirler.

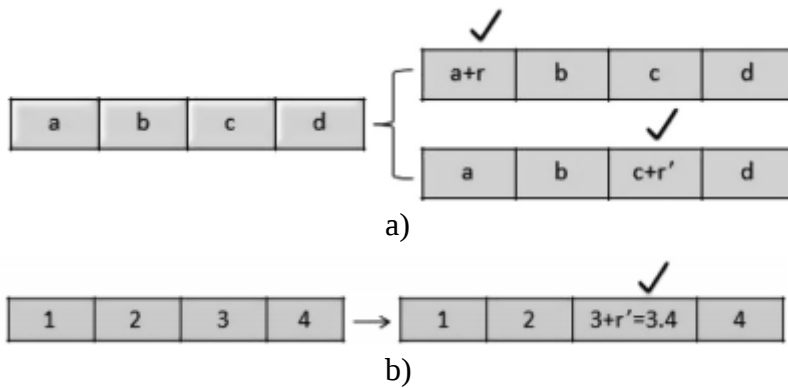
Yerel tohumlama aşamasında bir ağacın performansı umut verici ise, algoritma prosedürü o ağacın yaşını sıfırlar ve tohumlama yapmasına imkân tanınır. Umut verici olmayan ağaçlar ise her tekrarda yaşlanırlar ve birkaç tekrardan sonra ormandan çıkarılırlar. Yerel

tohumlamada ağaçların komşuluğuna birkaç tohum düşebilir ve bu tohumlar daha sonra birer ağaca dönüşebilirler. Her 0 yaşlı ağaç başına üretilen tohum sayısı “Yerel Tohumlama Değişikliği: *YTD*” parametresi ile kontrol edilir. Şekil 3.3’te gösterilen örnekte *YTD* 3 olarak seçilmiş ve iki tekrarda ağaç üretimi yapılmıştır.

Şekil 3.4 (a)’ da 4 boyutlu uzayda tanımlı problemler için yerel tohumlama operatörünün bir örneğini göstermiştir. Şekildeki *a, b, c* ve *d* harfleri her bir boyuttaki değişkeni temsil etmektedir. Görüldüğü gibi burada “*YTD*” 2 olarak atanmıştır. Şekil 3.4 (a)’ da gösterilen *r* ve *r'* değişkenleri $[-\Delta x, \Delta x]$ aralığında rasgele seçilen değerlerdir. Söz konusu Δx , ilgili boyutun üst limitinden küçük olmalıdır. Böylece yerel arama mekanizması da aktif hale gelecektir. Bu operatörü gerçekleştirmek amacıyla rastgele bir boyut seçilir. Sonra seçilen boyuttaki değişkenin değerine $r \in [-\Delta x, \Delta x]$ eklenir. Şekil 3.4 (b)’ de yerel tohumlama operatörüne sayısal bir örnek verilmiştir. Burada *YTD* ve Δx değeri 1 olarak seçilmiştir. Dolayısıyla ilgili boyuttaki değişkenin değerine $[-1, 1]$ aralığında rasgele seçilen $r'=0,4$ değeri eklenmiştir.



Şekil 3.3. Bir ağaç üzerinde iki tekrar için yerel tohumlama örneği (*YTD* =3)[66].



Şekil 3.4. Dört boyutlu uzayda yerel tohumlama a) *YTD* =2 b) *YTD* = 1, $r'=0,4$
 $r' \in [-\Delta x, \Delta x] = [-1, 1]$

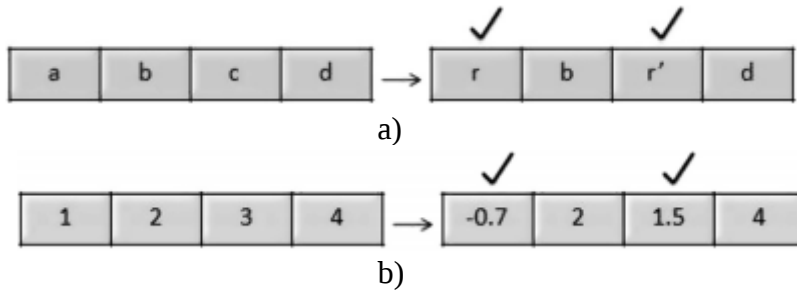
3.3.3. Popülasyon sınırlayıcı

Ormanda ağaç sayısının sonsuz bir şekilde genişlemesinin önlenmesi için bir sınır olmalıdır. Ağaç nüfusunu sınırlamak için “Ömür” ve “Alan sınırı” olmak üzere iki parametre bulunmaktadır. İlk başta yaşları “Ömür” parametresini aşan ağaçlar ormandan çıkarılır ve bunlar aday nüfusa eklenirler. İkinci sınırlama ise “Alan Sınırı” sınırlamasıdır. Söz konusu basamakta ağaçlar uygunluk değerine göre sıralanırlar. Ağaç sayısı orman için tanımlı sınırdan büyükse, ekstra ağaçlar ormandan çıkarılır ve aday nüfus kategorisine atanır. “Alan Sınırı” parametresi başlangıçta seçilen ağaç sayısı ile aynı kabul edilir. Dolayısıyla ormandaki ağaç sayısı başlangıçtaki ağaç sayısına eşittir. Orman nüfusunun kontrolü genel tohumlama aşamasında aday nüfusa atanan ağaçların belirli bir yüzdesi kullanılarak yapılır.

3.3.4. Ağaçların genel tohumlaması

Ormanlarda ağaçların farklı türleri vardır, farklı hayvanlar ve kuşlar bu ağaçların tohum ve meyvelerinden beslenirler. Böylece, ağaç tohumları tüm ormana dağıtılır ve bunun sonucu olarak ağacın yaşam alanı daha da genişlemiş olur. Ayrıca, rüzgâr ve su akışı gibi diğer doğal süreçler tohumların tüm ormana dağıtılmasına yardımcı olur ve ağaçlar imparatorluklarını farklı bölgelerde garanti altına alırlar. Genel tohumlama aşaması ormandaki ağaçların tohumlarının dağıtımını taklit etmek için kullanılır. Genel tohumlama işlemi aday nüfusun önceden belirlenmiş bir yüzdesi kullanılarak yapılır. Söz konusu yüzde “aktarma oranı” olarak tanımlanır ve kullanıcı tarafından seçilmesi gereklidir. Genel tohumlama operatörü aday nüfustan seçilen ağaçlar kullanılarak gerçekleştirilir. Seçilen ağaçların rasgele belirlenen değişkenleri karşılıklı olarak değiştirilir. Yerel tohumlamada olduğu gibi burada da değişken sınırları dikkate alınmalıdır. Seçilen ağaçlardaki değiştirilecek değişken sayısı “Genel Tohumlama Değişikliği: GTD” parametresi olarak tanımlanır ve kullanıcı tarafından başlangıçta seçilmelidir.

Şekil 3.5 (a)’da “GTD” =2 olarak seçildiğinde genel tohumlamanın nasıl yapıldığı, Şekil 3.5 (b) de ise sayısal örneği verilmiştir. r ve r' parametreleri belirlenen sınırlar içinde rasgele seçilirler. Şekil 3.5 (b)’de GTD parametresi 2, r ve r' aralığı ise $[-5, 5]$ şeklinde seçilmiştir. Sayısal örnekte $r=0,7$ ve $r'=1,5$ şeklinde seçim yapılmıştır.



Şekil 3.5. Dört boyutlu uzayda tek ağaçlı genel tohumlama a) GTD = 2 b) Sayısal örnek[66].

3.3.5. OOA algoritmasının sözde kodu

Diğer optimizasyon algoritmaları olduğu gibi OOA algoritmasında her tekrarda bulunan en iyi sonucu korumaktadır. Bu amaçla ağaçlar uygunluk değerlerine göre sıralandıktan sonra, uygunluk değeri en yüksek olan ağaç en iyi ağaç olarak seçilir. Yerel tohumlama aşamasında en iyi ağacın yaşlanmasını önlemek için en iyi ağacın yaşı 0 olarak atanır. Bu şekilde yerel optimumların bulunması da sağlanacaktır. Diğer evrimsel algoritmaları gibi OOA algoritmasında da durdurma koşulları kullanılmaktadır. Söz konusu şartlar; önceden tanımlanmış tekrar, belirli bir tekrarlama sonucunda en iyi ağacın uygunluk değerinde değişiklik olmaması ve istenilen doğruluk seviyesine ulaşılmasıdır. OOA' nın aşamaları Şekil 3.6' da gösterilen sözde *kod* ile tanımlanmıştır.

FOA (Ömür, YTD, GTD, Aktarma Oranı, Alan sınırı)

Girdi: Ömür, YTD, GTD, Aktarma oranı, Alan sınırı

Çıktı: $f(X)$ amaç fonksiyonu için yakın optimal çözüm

1. Rastgele ağaçlar ile Başlangıç Ormanı oluşturma

1.1. Her ağaç m-boyutlu problem için $(m + 1)$ boyutlu X vektörüdür: $X = [yaş, x_1, x_2, \dots, x_m]$

1.2. Her ağacın "Yaşı" nı sıfırla

2. Durdurma koşulu istenilen gibi değilse

2.1. Yaşı 0 olan ağaçlara yerel tohumlama yap

- For $ii=1$: "YTD"
 - Seçilen ağaç için rasgele bir değişken seç
 - Rastgele seçilen değişkene dx ekle ($dx \in [-\Delta x, \Delta x]$)
- Yeni oluşturulan ağaçlar hariç bütün ağaçların yaşını birer arttır

2.2. Nüfus sınırlayıcı

- yaşı "Ömür" parametresinden daha büyük olan ağaçlar kaldır ve aday nüfusa ekle
- Ağaçları uygunluk değerlerine göre sırala
- "Alan Sınırı" nı aşan ağaçları ormandan kaldır ve aday nüfusa ekle

2.3. Genel tohumlama

- Aday nüfustan "Aktarma Oranı" kadar ağaç seç
- Seçilen ağaçlar için
 - "GTD" değişkenlerini seç
 - Değişkenlerin yerlerini değiştir ve ormana 0 yaşında yeni bir ağaç ekle.

2.4. Şimdiye kadarki en iyi ağacın güncellenmesi

- Ağaçları uygunluk değerlerine göre sırala
- En iyi ağacın yaşını 0 yap

3. Sonuç olarak en iyi ağacı gönder

Şekil 3.6. OOA sözde kodu [66].

4. RENK UZAYI VE ÇOK SEVİYELİ EŞİKLEME

Gri ölçekli görüntülerin çok seviyeli eşiklemesine yönelik çeşitli algoritmalar geliştirilmiş olmasına rağmen, renkli görüntülerin çok seviyeli eşiklenmesi çözülmesi gereken problem olarak kalmıştır. Renkli görüntülerin eşiklemesine yönelik ilk çalışma Demirci ve arkadaşları tarafından önerilmiştir. Söz konusu algoritmada RGB renk uzayı ve her bir kanal için tek eşik kullanılmıştır. Eşikler Otsu ve Kapur yöntemleri ile tespit edilmiştir. RGB renk uzayı sekiz alt küpler halinde bölümlenmiş ve her küp içindeki pikseller aynı kümeye atanmıştır. Renkli görüntülerin tek eşik ile sınıflandırılması hızlı olduğu halde oluşturulan sınıfların homojenliği düşük kalmıştır. Bu çalışmada ilgili algoritmanın dezavantajını ortadan kaldırmak için çok düzeyli eşikleme yöntemi önerilmiştir. Her bir kanal için kullanılacak eşik sayısı artırıldığında, oluşacak küplerin hacimlerinin azalacağı açıktır. Böylece her bir sınıfın homojenliği artacaktır. Şekil 4.1 iki eşik kullanıldığında RGB renk uzayında oluşan blokları göstermektedir. Böylece oluşturulabilecek maksimum küme sayısı

$$C_{\max} = (m + 1)^3 \quad (4.1)$$

şeklinde hesaplanabilir. Burada m eşik sayısını temsil etmektedir. Tek eşik kullanıldığında ($m=1$), $C_{\max} = 8$ olacak ve dolayısıyla sekiz alt küp oluşacaktır. Eğer $m=2$ iken, $C_{\max} = 27$ ve $m=3$ iken $C_{\max} = 64$ olacaktır.

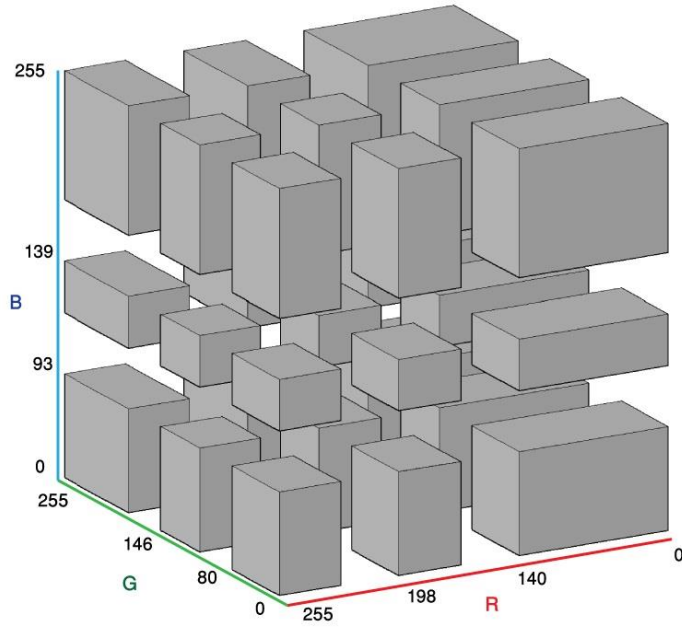
Diğer taraftan eşik sayı artığında histogram bilgileri kullanılarak eşiklerin hesap maliyetleri artmaktadır. Renkli görüntülerde 3 kanal bulunduğundan her bir kanal için m tane eşik hesaplaması gereklidir. Her bir kanal gri ölçekli görüntü gibi kabul edilirse, Eş. 2.9 ve 2.10 da tanımlanan amaç fonksiyonları yardımıyla m tane eşik hesaplanabilir. İlgili fonksiyonlar m değişkenli olup, m boyutlu uzayda Eş. 2.9 ve 2.10'daki ifadeleri maksimum yapan noktalar çözüm kümesidir. Literatürde çeşitli eşikleme yöntemleri olmasına rağmen en yaygın olarak kullanılan eşikleme yöntemleri Eş. 2.9 ve 2.10'da verilen Kapur ve Otsu teknikleridir. Ancak söz konusu bu yöntemlerde hesap maliyeti eşik sayısına bağlıdır. İlgili algoritmaların performansını artırmak için araştırmacılar çeşitli optimizasyon yöntemlerini kullanarak gri görüntüler için eşikleme yapmışlardır.

Renkli görüntülerin eşiklenmesi için yapılan ilk çalışmada Otsu ve Kapur amaç fonksiyonları kullanılmıştır. Ayrıca performansları yüksek ve yaygın olduğundan bu

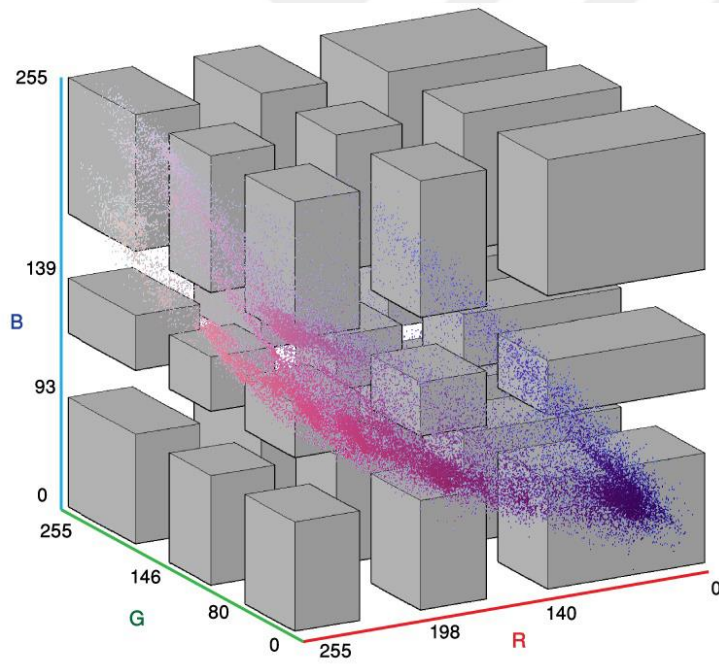
çalışmada Otsu ve Kapur amaç fonksiyonları tercih edilmiştir. Aynı şekilde yaygın ve kodlaması kolay olan PSO ile son yıllarda önerilen OOA optimizasyon algoritmaları kullanılarak çözüm uzayında arama yapılmıştır.

Her bir kanal için m tane eşik bulunduktan sonra renk küplerinin veya sınıfların oluşturulması için maksimum küme sayısı, $C_{\max}$ kadar kural tanımlanması gereklidir. Tek eşikli sınıflandırma için kural tablosu Çizelge 2.1 verilmiştir. İki eşik kullanıldığında ise maksimum 27 tane sınıf oluşacağından, 27 tane kural tanımlanmalıdır. İki eşikli sınıflandırma için gerekli kurallar Çizelge 4.1 de verilmiştir. Eşik sayısı artıkça kural sayısının artması beklenen bir durumdur. Çizelge 2.1 ve Çizelge 4.1 de görüldüğü gibi eşik indisleri sınıf etiket kodlarının üretilmesi için kullanılmıştır. Bu açıdan bakıldığında etiketleme için ayrı bir algoritmanın kullanılmaması algoritmanın tipik bir avantajıdır [82,83].

Renkli Lena görüntüsünün Şekil 4.1 gösterilen histogramları kullanılarak her kanal için ikişer tane eşik hesaplanmış ve elde edilen eşikler yardımıyla oluşturulan bloklar Şekil 4.1, verilmiştir. Şekil 4.2' de ise Şekil 2.9, daki renk dağılımı ile Şekil 4.1 deki blokların birleşilmesi ile elde edilen dağılım gösterilmiştir. Söz konusu şekil dikkatlice incelendiğinde her bir alt renk küpünün veya prizmasının hacimlerinin aynı olmadığı açıktır. Oluşan her alt küp bir bölgeyi temsil ettiğinden, bazı küplerin içerisine herhangi bir pikselin atanmaması veya boş kalması muhtemeldir. Başka bir ifade ile bazı sınıflar için renk uzayında yer tahsis edilmiş olsa bile gelecek görüntüde karşılığı olmayabilir. Bu durumun tek dezavantajı ise fazladan etiket kullanmak olacaktır.



Şekil 4.1. Renkli Lena: üç boyutlu renk uzayı ve kümeleri.



Şekil 4.2. Renkli Lena: renk dağılımı ve atama yapılan sınıfları.

Çizelge 4.1. İki eşikli sınıf etiketi (küp) oluşturma kuralları

Sınıf etiketi C_n	Sınıf (Küp) oluşturma kuralları	Sınıf etiket kodu $(d_r d_g d_b)_{m+1}$
C_0	eğer $(R > t_{r,0} \& R \leq t_{r,1})$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	000
C_1	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	001
C_2	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,2}$ ve $B \leq 255)$	002
C_3	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	010
C_4	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	011
C_5	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,2}$ ve $B \leq 255)$	012
C_6	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	020
C_7	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	021
C_8	eğer $(R > t_{r,0}$ ve $R \leq t_{r,1})$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,2}$ ve $B \leq 255)$	022
C_9	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	100
C_{10}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	101
C_{11}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,2}$ ve $B \leq 255)$	102
C_{12}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	110
C_{13}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	111
C_{14}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,2}$ ve $B \leq 255)$	112
C_{15}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	120
C_{16}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	121
C_{17}	eğer $(R > t_{r,1}$ ve $R \leq t_{r,2})$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,2}$ ve $B \leq 255)$	122
C_{18}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	200
C_{19}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	201
C_{20}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,0}$ ve $G \leq t_{g,1})$ ve $(B > t_{b,2}$ ve $B \leq 255)$	202
C_{21}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	210
C_{22}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	211
C_{23}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,1}$ ve $G \leq t_{g,2})$ ve $(B > t_{b,2}$ ve $B \leq 255)$	212
C_{24}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,0}$ ve $B \leq t_{b,1})$	220
C_{25}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,1}$ ve $B \leq t_{b,2})$	221
C_{26}	eğer $(R > t_{r,2}$ ve $R \leq 255)$ ve $(G > t_{g,2}$ ve $G \leq 255)$ ve $(B > t_{b,2}$ ve $B \leq 255)$	222

4.1. PSO Algoritması ile Çok Seviyeli Eşiklerin Tespiti

Renkli görüntülerde üç tane kanal bulunduğundan, her bir kanal için m tane eşik hesaplanmalıdır. Her bir kanalın histogram bilgileri Eş. 2.9 ve 2.10 da yerine konularak eşik hesaplaması yapılmıştır. Eş. 2.9 ve 2.10 aslında m değişkenli ve kısıtlı bir optimizasyon problemini temsil etmektedir. Her iki amaç fonksiyonunda da $t_1 < t_2 < t_3 < \dots < t_m$ kısıtları mevcuttur. Dolayısıyla çözüm uzayında söz konusu kısıtlar dikkate alınarak arama yapılmalıdır. İlgili problem PSO ortamına Eş.3.1 yardımıyla aktarılmıştır. Eş.3.1'deki x_i i nolu parçacığın bulunduğu konumu temsil etmektedir. Her bir parçacığın bulunduğu konum m boyutlu uzayda bir nokta olduğundan, aslında her bir parçacığın konumu bir çözüm kümesidir. Eş.3.1'deki her bir iterasyonda yeni bir çözüm kümesi bulunacaktır. Bulunan her bir çözümün uygunluğu Eş. 2.9 ve 2.10 aracılığı ile değerlendirilmektir. Değerlendirmede $t_1 < t_2 < t_3 < \dots < t_m$ kısıtlarını sağlamayan parçacıkların uygunluk fonksiyonunun değeri sıfıra çekilerek cezalandırılmaktadır. PSO algoritması ile arama yapılır iken RGB renk uzayının sınırları içerisinde kalınması önemlidir. Bu nedenle en küçük eşik için $0 \leq t_1$, en büyük eşik için $t_m \leq 255$ kısıtları dikkate alınmıştır. Söz konusu süreç durdurma kriterleri sağlanıncaya kadar devam etmektedir. Sonuç olarak PSO tabanlı çok seviyeli arama prosedürü aşağıdaki gibidir:

Aşama 1. Gerekli parametreleri gir, eşik değerlerin alt ve üst sınırları belirle.

Aşama 2. Popülasyondaki her bir parçacığın konumları rastgele oluştur.

Aşama 4. Eş. 2.9 ve 2.10 kullanarak popülasyondaki her bir parçacığın uygunluk değerini hesapla. Uygunluk değeri en yüksek olanı en iyi çözüm ata.

Aşama 5. Bütün parçacıkların konumlarını Eş.3.1 yardımıyla değiştir.

Aşama 6. Bitirme koşullarını sağlanmadı ise Aşama 4'e git.

Aşama 7. Uygunluk değeri en yüksek olan parçacığın konumunu en iyi çözüm olarak al ve algoritmayı durdur.

Eş.3.1 de verilen PSO algoritmasında W parametresi çok küçük seçilirse, parçacıklar çok düşük ivme ile hareket edeceklerdir. Böylece parçacıkların yerel maksimuma takılma ihtimali artacaktır veya parçacık hareketsiz kalarak ölecektir. Diğer taraftan W parametresi çok büyük seçilirse hareketin ivmesini artıracak ve global optimuma odaklanmasını zorlaştıracaktır. Bu nedenle ilgili parametrenin doğru seçilmesi kritik bir husustur. Diğer

tarafından bazı çalışmalarda W parametresi başlangıcında yüksek seçildiği ve algoritma ilerledikçe küçültüldüğü çalışmalarda mevcuttur. Böylece ilk başta geniş bir arama yapıldıktan sonra algoritmanın sonlarına doğru parçacıklar genel optimum çevresinde daha dikkatli bir arama yaparlar. Bu çalışmada $0,5 < w < 1,5$ aralığında seçilmiştir. $C_{p,1}$ ve $C_{p,2}$ parametreleri ise sırasıyla parçacıkların yerel ve genel optimum kat sayıları olup, eşit olmaları daha makuldür. Genel olarak $[0,8-1,5]$ aralığında seçilen söz konusu parametreler, bu çalışmada 0,95 uygulanmıştır. PSO algoritmasının önemli bir parametresi ise parçacık sayısıdır. Parçacık sayısının az olması çözüme ulaşamama riskini taşıırken, çok fazla olması ise gereksiz arama ve hesap maliyetini artırmaktadır. Renkli görüntülerin sınıflandırmak için tasarlanan bu çalışmada parçacık sayısı 250 ve tekrarlama sayısı ise 200 olarak tespit edilmiştir.

4.2. OOA Algoritması ile Çok Seviyeli Eşiklerin Tespiti

Renkli görüntülerin çok seviyeli eşiklenmesi için sürü zekasına dayalı PSO algoritmasıyla çözüm aramaya dayalı yaklaşım önceki bölümde verilmiştir. Diğer taraftan evrimsel algoritmalarla daha çok benzeyen OOA algoritması ile de çözüm kümesi aranmıştır. Her iki yaklaşımda da kullanılan amaç fonksiyonları Eş. 2.9 ve 2.10, da verilmiştir. Her bir renk kanalı için histogramlar kullanılarak m tane eşik bulunması genel hedeftir. OOA algoritmasında her çözüm kümesi bir ağaç ile temsil edildiğinden, Eş. 3.2'deki parametreler yerine eşik değerleri atanarak $Ağaç = [yaş, t_1, t_2, \dots, t_m]$ şeklinde ağaçlar veya orman oluşturulmuştur. Diğer optimizasyon algoritmalarında olduğu gibi OOA algoritmasında da bazı parametrelerin kullanıcı tarafında seçilmesine ihtiyaç duyulur. Örneğin “Ömür” parametresi bir ağacın yaşayabilecek maksimum yaşıdır. İlgili parametrenin küçük seçilirse ağaçların genel optimuma odaklanmasını engelleyebilir iken, diğer taraftan bu parametrenin büyük seçilmesi ağaçları yerel optimuma hapis edebilir. Dolayısıyla doğru seçimin yapılması önemli bir husustur. Bu çalışmada “Ömür” parametresi 10 olarak, maksimum tekrar sayısı ise 200 olarak atanmıştır. Ağaçların sayısı ise 200 olarak tespit edilmiştir. Eş. 2.9 ve 2.10, da tanımlı amaç fonksiyonlarını kullanılarak OOA tabanlı çok seviyeli eşik bulma prosedürü aşağıdaki gibidir:

- Aşama 1. Gerekli parametreleri gir, eşik değerlerin alt ve üst sınırları belirle.
- Aşama 2. 0 yaşlı ağaçlarla rastgele bir orman üret. Her ağaç m boyutlu olup ve m seviyeli eşikleme için bir çözüm adaydır.
- Aşama 4. Eş. 2.9 ve 2.10 kullanarak popülasyondaki her bir ağacın uygunluk değerini hesapla.
- Aşama 5. Ormanda yerel tohumlama yapılarak yeni çözüm adaylar oluştur.
- Aşama 6. Ormandaki ağaç sayısını sınırlamak içine yaşlı ve zayıf ağaçların ortadan kaldır.
- Aşama 7. Genel tohumlama yap.
- Aşama 8. Her ağacın uygunluk değerini hesapla. En iyi ağacı bul ve yaşını sıfırla.
- Aşama 9. Bitirme koşullarını sağlanmadı ise Aşama 4'e git.
- Aşama 10. En iyi ağacın konumunu en iyi çözüm olarak ata ve algoritmayı durdur.

4.3. Performans Değerlendirme Ölçütleri

Görüntü ayrıştırma üzerinde çalışan araştırmacılar sonuçlarını örnek görüntüler üzerinden test ederler ve performans karşılaştırılması genel olarak gözle yapılmaktadır. Tipik olarak bir değerlendirme ölçütü aşağıdaki kriterleri dikkate almalıdır [84]:

- Aşırı ayrıştırma: incelenen görüntünün bir bölgesinin iki veya daha fazla bölge ile temsil edilmesi,
- Eksik ayrıştırma: incelenen görüntünün iki veya daha fazla bölgenin, tek bir bölge ile temsil edilmesi,
- Yanlış sınır yerleştirilmesi: incelenen görüntüdeki nesnelerin sınırlarının olması gereken yerlere yerleştirilememesi,
- Farklı bölge sayısı: Aynı görüntü için farklı sayıda bölge üreten algoritmaları karşılaştırılabilir.

Yukarda bahsedilen ölçütlerin önemli bir kısmını karşılamak için Liu ve Yang 1994 yılında aşağıdaki fonksiyonu önermiştir [85];

$$F = \frac{1}{1000(M \times N)} \sqrt{R} \sum_{i=1}^R \frac{e_i^2}{\sqrt{A_i}} \quad (4.2)$$

Burada $(M \times N)$ görüntüdeki piksel sayısını, R ayrıştırma sonucundan elde edilen bölge sayısını, A_i her bölgedeki piksel sayısını (bölgede alanı), e_i ise i . bölge için bölgeye ait orijinal piksel değerleri ile bölgeye atanan renk değeri arasındaki renk mesafelerinin (Öklid mesafelerinin) toplamını ifade etmektedir. Aslında e_i değişkeni her bölge için ortalama renk hatasını temsil etmektedir. Eş. 4.2' deki $\sqrt{R}$ terimi ile aşırı ayrıştırma cezalandırılmaktadır.

Diğer taraftan $\frac{e_i^2}{\sqrt{A_i}}$ terimi ile de çok küçük bölgeler veya aynı bölgeye atanan piksellerin renk uyumsuzluğu cezalandırılmaktadır. Sonuçta F değerinin küçük olması ayrıştırmanın performansının yüksek olduğunun göstergesidir. Eş. 4.2 de verilen performans ölçütü Borsotti ve arkadaşları tarafından geliştirilerek

$$F'(I) = \frac{1}{10000(N \times M)} \sqrt{\sum_{A=1}^{Max} [R(A)]^{1+\frac{1}{A}}} \times \sum_{i=1}^R \frac{e_i^2}{\sqrt{A_i}} \quad (4.3)$$

şeklinde yeniden tanımlanmıştır [86]. Buradaki amaç çok küçük bölge oluşumunu önlemek için $\sqrt{R}$ terimi değiştirilmiştir. Eş. 4.3' de $R(A)$, A alanına sahip bölgelerin sayısını ve Max ise bölütlenmiş resimdeki en büyük bölgenin alanını temsil etmektedir. Ayrıca $(1+1/A)$ ifadesi, küçük bölgelerin performansa katkısı artırılmıştır. Ayrıştırma sonucunda oluşan küçük alanlı bölgeleri ve homojenliği düşük olan bölgeleri cezalandırmak için Borsotti ve arkadaşları

$$Q(I) = \frac{1}{10000(M \times N)} \sqrt{R} \sum_{i=1}^R \left[\frac{e_i^2}{1+\log A_i} + \left(\frac{R(A_i)}{A_i} \right)^2 \right] \quad (4.4)$$

ölçütünü ayrıca geliştirmişlerdir. Eş. 4.4' de ilave edilen $1+\log A_i$ terimi ile alanı büyük ve homojen olmayan bölgeler cezalandırılmıştır. $\frac{R(A_i)}{A_i}$ terimindeki $R(A_i)$, i . bölge ile aynı alanına sahip bölgelerin sayısını ifade etmektedir. Böylece küçük alana sahip olan bölgeler arttıkça Q büyüyeceğinden, aşırı ayrıştırmaya karşı tedbir alınmış olacaktır. F ve Q ölçütleri ne kadar düşük kalırsa, ayrıştırma işleminin o kadar başarılı olduğu sonucuna varılır.

5. DENEYSEL SONUÇLAR VE YORUMLAR

Geliştirilen algoritmayı test etmek için C# ortamında bir ara yüz tasarlanmıştır. İlgili ara yüz kullanılarak seçilen gri ölçekli ve renkli görüntüler üzerinde deneyler yapılmıştır. Önerilen algoritma görüntülerin histogramlarını kullanarak çok seviyeli eşiklerin tespiti esasına dayanmaktadır. Bu kapsamda Eş. 2.9 ve 2.10' da tanımlanan amaç fonksiyonlarının maksimize edilmesi çözüm kümesine ulaşmak için takip edilen yoldur. Bahsedilen çok değişkenli fonksiyonlar m boyutlu olduğundan, m boyutlu uzayda arama yapılması gereklidir. İlgili arama bahsedilen uzayda belirtilen sınırlar dahilinde arama yaptığından, öncelikle kaba kuvvet (doğrusal arama) algoritması ve optimizasyon algoritmaları kullanılarak deneyler yapılmıştır. Doğrusal arama algoritması yardımı ile çok küçük hata sınırları içerisinde sonuçlar elde edilebilmektedir. Ancak doğrusal aramanın zaman maliyeti çok yüksek olduğundan pratik değildir. Bununla birlikte PSO ve OOA algoritmalarının doğruluğunu test etmek için kullanılmıştır. Her bir deneyden alınan sonuç performans değerlendirme ölçütleri, hesaplama süresi ve görsel çıktılarla değerlendirilmiştir. Ayrıca görüntü sınıflandırmada kullanılan FCM algoritması ile sonuçlar alınmış ve metot bazlı karşılaştırma yapılmıştır.

5.1. Geliştirilen Ara Yüz

Önerilen ayrıştırma algoritmasının test sonuçlarını elde etmek için C# programlama dilinde Şekil 5.1' de gösterilen ara yüz tasarlanmıştır. “Görüntü Seç” butonu ile ayrıştırılacak görüntü sol üst köşedeki “Picture Box” kutusuna alınmaktadır. Akabinde ayrıştırmada kullanılacak eşik sayısı etiketi “Eşik Sayısı” olan bir “Text Box” aracılığı ile kullanıcıdan alınmaktadır. Sonraki aşamada ise “Çoklu Eşik” butonuna basılarak algoritma çalıştırılır. Test edilecek görüntü için elde edilen eşikler kırmızı, yeşil ve mavi etiketli “Text Box” larında gösterilmiştir. Aynı zamanda ayrıştırılmış görüntü sağ üst köşedeki “Picture box” da gösterilmiştir. Ayrıştırma işlemi ile birlikte performans değerlendirmesi de yapılmakta olup, sonuçlar sağ alt köşedeki F , F' ve Q etiketli “Text Box” larında gösterilmiştir.



Şekil 5.1. Geliştirilen ara yüz

5.2. Deneysel Sonuçlar

Deneysel sonuçların da kullanılan bütün algoritmalar Intel Core, i7-3632QM 2.2 GHz işlemcili ve 8 GB RAM'e sahip bir bilgisayar ortamında test edilmiştir. Görüntüleri sınıflandırmak için K-ortalamlar veya FCM algoritması gibi iteratif yöntemler kullandığında, büyük boyutlu görüntüler küçük boyutlu görüntülere kıyasla daha uzun zaman almaktadır. Diğer taraftan geliştirilen algoritma görüntünün sadece histogramını kullandığından hesaplama maliyeti görüntü boyutlarından bağımsızdır. Görüntü histogramı ise tek boyutlu olup, uzunluğu sabittir. Görüntünün histogramın alındığında resim üzerindeki piksel bilgileri bir anlamda histogram verisine dönüştürülmektedir.

Çok seviyeli eşikleme kullanarak görüntü sınıflandırmada ise sadece histogram bilgisi kullanılmaktadır. Başka bir ifade ile histogram verileri üzerinde tanımlanan çok değişkenli amaç fonksiyonları maksimum yapılarak eşikler tespit edilmektedir. Elde edilen eşikler ile de uzamsal düzlemdeki pikseller sınıflandırılmaktadır. Bu çalışmada Kapur ve Otsu amaç fonksiyonları temel alınmış, OOA ve PSO algoritmaları kullanarak eşikler tahmin edilmiştir. Yani sınıflandırma eşikler kullanılarak yapılmıştır. Diğer taraftan en yaygın kullanılan sınıflandırma algoritması FCM olduğundan, eşikleme ile elde edilen sonuçların performansı FCM algoritmasının çıktıları ile karşılaştırılmıştır.

5.2.1. Gri ölçekli görüntülerde alınan sonuçlar

Günümüze kadar çok seviyeli eşikleme sadece gri ölçekli görüntülerde kullanılmıştır. Bu çalışmada geliştirilen teknik ise hem gri hem de renkli görüntülerde çalışmaktadır. Başka bir ifade ile gri ölçekli görüntüler test edildiğinde, sadece Şekil 4.1' de gösterilen diyagonal alt küpler aktif hale gelir. Diğer küplere herhangi bir piksel ataması olmayacaktır. Bu nedenle önerilen algoritmanın gri ölçekli görüntülerde çalıştığını göstermek için Şekil 5.2 (a)' da orijinali, Şekil 5.2 (b)' de histogramı gösterilen Kameraman görüntüsü test edilmiştir. Doğrusal, OOA ve PSO yöntemleri ile elde edilen eşikler ve hesaplama süreleri Çizelge 5.1' de verilmiştir. Görüldüğü gibi eşik sayısı arttıkça hesaplama süreside artmaktadır. Söz konusu artış bir anlamda kaba kuvvet algoritması olan doğrusal aramada neredeyse logaritmik iken, optimizasyon yöntemlerinde çok fazla değişim görülmemiştir. Başka bir ifade ile optimizasyon yöntemleri eşik sayısından çok fazla etkilenmemektedir.

Eşik değerleri ve hesaplama süreleriyle ilaveten, F , F' ve Q performans ölçütleri ise Çizelge 5.2' de verilmiştir. Görüldüğü gibi eşik sayısı arttıkça homojenliğin göstergesi olan F , F' ve Q parametreleri sıfıra doğru yaklaşmaktadır.

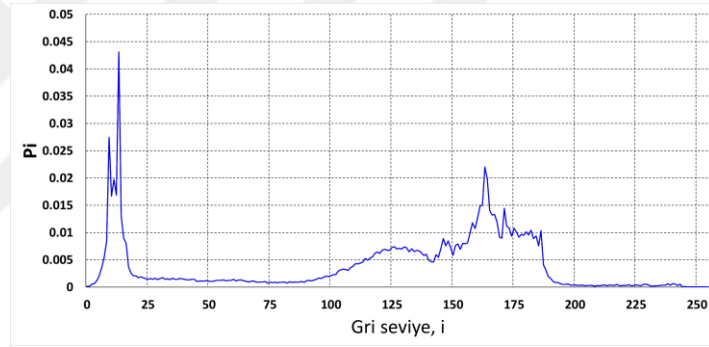
Eşik sayısı $m=2$ olarak seçildiğinde, Otsu yöntemi ile alınan sonuç Şekil 5.3(a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.3 (b) de ve FCM algoritmasının çıktısı ise Şekil 5.3(c)' de gösterilmiştir. Genel olarak bakıldığında bazı nesnelerin birbirine karıştığı açıktır. Aynı şekilde eşik sayısı $m=3$ olarak seçildiğinde, Otsu yöntemi ile alınan sonuç Şekil 5.4(a) da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.4' (b) de ve FCM algoritmasının çıktısı ise Şekil 5.4 (c)' de verilmiştir. Kapur yönteminde sonuç çok fazla değişmemekle birlikte, detaylar ortaya çıkmaya başlamıştır. Eşik sayısının artması sınıf sayısını artıracığından, $m=4$ seçilerek deneyler tekrarlanmıştır. Otsu yöntemi ile alınan sonuç Şekil 5.5 (a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.5(b) de ve FCM algoritmasının çıktısı ise Şekil 5.5 (c)' de gösterilmiştir. Görüldüğü gibi bina ve kameranın ayakları geriye kalan nesnelerden ayırt edilir hale gelmiştir. Ancak FCM algoritmasının hesap süreside artmıştır.

Çizelge 5.1. Kameraman için bulunan eşikler

Otsu					Kapur					
	Süre,t (s)	t ₁	t ₂		Süre,t (s)	t ₁	t ₂			
Doğrusal	0,200	71	145		0,206	124	192			
OOA	0,325	72	146		0,114	125	191			
PSO	0,312	72	146		0,317	124	192			
Küme: c		3/3				3/3				
		t ₁	t ₂	t ₃		t ₁	t ₂	t ₃		
Doğrusal	164,713	46	102	150	243,758	45	103	192		
OOA	0,472	48	104	150	0,123	45	103	192		
PSO	0,427	46	102	150	0,422	45	103	192		
Küme: c		4/4				4/4				
		t ₁	t ₂	t ₃	t ₄		t ₁	t ₂	t ₃	t ₄
Doğrusal										
OOA	0,513	42	95	140	170	0,598	44	97	146	197
PSO	0,508	42	95	140	170	0,515	44	97	146	197
Küme: c		5/5					5/5			

Çizelge 5.2. Kameraman performans ölçütleri: F, F' ve Q

Eşikleme Metodu	Otsu		Kapur		FCM
Algoritma	PSO	OOA	PSO	OOA	-
Eşik Sayısı/ Küme	$m = 2 / C = 3$		$m = 2 / C = 3$		$C = 3$
F	0,000114	0,000114	0,000761	0,000761	0,000382
F'	0,000011	0,000011	0,000076	0,000076	0,000038
Q	0,000329	0,000329	0,002337	0,002337	0,001088
Eşik Sayısı/ Küme	$m = 3 / C = 4$		$m = 3 / C = 4$		$C = 4$
F	0,000086	0,000086	0,000345	0,000345	0,000273
F'	0,000009	0,000009	0,000048	0,000035	0,000027
Q	0,000222	0,000222	0,001268	0,001268	0,000766
Eşik Sayısı/ Küme	$m = 4 / C = 5$		$m = 4 / C = 5$		$C = 5$
F	0,000057	0,000057	0,000093	0,000093	0,000166
F'	0,000006	0,000006	0,000009	0,000009	0,000017
Q	0,000133	0,000133	0,000262	0,000262	0,000386



Şekil 5.2. Kameraman(300x292) a) Orjinal b) histogramı



a)

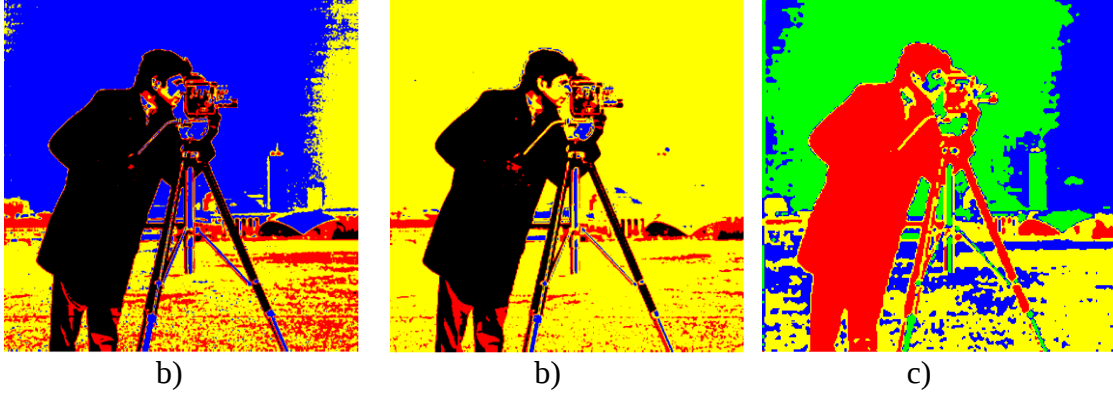


b)

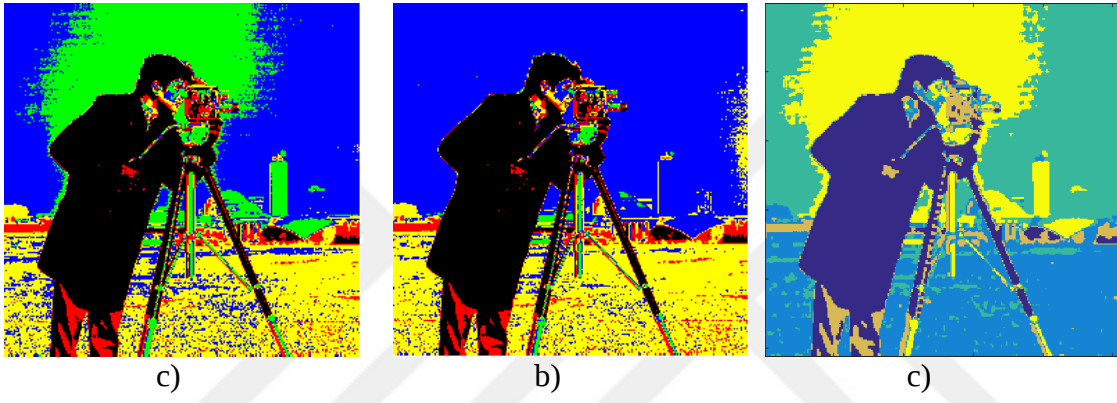


c)

Şekil 5.3. Kameraman, $m=2$ ve $c=3$ a) Otsu b) Kapur c) FCM, $t= 0,891s$



Şekil 5.4. Kameraman, $m=3$ ve $c=4$ a) Otsu b) Kapur c) FCM, $t=0,917s$



Şekil 5.5. Kameraman, $m=4$ ve $c=5$ a) Otsu b) Kapur c) FCM, $t=1,186s$

5.2.2. Renkli görüntülerle alınan sonuçlar

Bu çalışmanın asıl amacı renkli görüntülerin çok seviyeli eşiklemesi olduğundan, önerilen algoritma referans görüntüler kullanılarak ta test edilmiştir. Bu kapsamda Lena, Ayakkabı, Mısır, Biber, Papağan ve Kız çocuğu görüntüleri dikkate alınmıştır.

Orijinali Şekil 2.6' da gösterilen Lena görüntüsünün renk dağılımı Şekil 2.9' da verilmiştir. Şekil 2.8' de gösterilen Lena' nın renk kanallarına ait histogramlar kullanılarak eşikler hesaplanmıştır. Doğrusal, OOA ve PSO yöntemleri ile elde edilen eşikler ve hesaplama süreleri Çizelge 5.3' de gösterilmiştir. Çizelge 5.3'deki eşikler kullanılarak Şekil 2.9' daki renk dağılımı sınıflandırıldığında bazı alt küplerin veya prizmaların boş kalacağı açıktır. Bu nedenle seçilen eşişe bağlı olarak oluşacak maksimum küme sayısı ve doldurulan veya aktif olan küp sayısı tabloda ayrıca belirtilmiştir. Eşik değerleri ve hesaplama sürelerine ilaveten, F , F' ve Q performans ölçütleri ise Çizelge 5.4' de gösterildiği gibidir.

Çizelge 5.3. Lena için bulunan eşikler

Otsu						Kapur							
Doğrusal			OOA		PSO	Doğrusal			OOA		PSO		
			t_1					t_1					
R	162		160		160	167		168		168			
G	102		103		103	141		140		140			
B	112		110		110	131		131		131			
Süre, t (s)	0,008		0,244		0,244	0,029		0,294		0,333			
Küme:c	8/8					7/8							
	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	
R	139	197	141	192	141	198	134	191	135	190	135	190	
G	79	146	80	46	80	146	82	151	84	150	84	150	
B	96	140	96	140	96	140	98	144	100	144	100	144	
Süre, t (s)	5,572		1,141		1,567		6,628		1,022		2,310		
Küme:c	19/27					20/27							
	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	126	172	210	126	172	210	116	159	203	
G	58	107	160	59	108	160	59	108	160	61	110	161	
B	84	111	147	86	115	147	86	112	147	93	132	170	
Süre, t (s)	512,833			1,163			1,818			676,992			
Küme:c	32/64					30/64							
	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4	
R	-	-	-	-	118	156	190	217	118	157	191	218	
G	-	-	-	-	50	86	121	166	50	86	121	166	
B	-	-	-	-	84	108	135	165	84	108	135	165	
Süre, t (s)					5,54				5,50				
Küme:c					51/125				51/125				
									48/125				

Çizelge 5.4. Lena performans ölçütleri: F, F' ve Q

Eşikleme Metodu	Otsu		Kapur		FCM
Algoritma	PSO	OOA	PSO	OOA	-
Eşik Sayısı/ Küme	$m = 1 / C = 8$		$m = 1 / C = 7$		$C = 7$
F	0,008639	0,008639	0,008696	0,008696	0,001388
F'	0,000864	0,000864	0,000870	0,000870	0,000139
Q	0,023248	0,023248	0,024710	0,024710	0,002677
Eşik Sayısı/ Küme	$m = 2 / C = 19$		$m = 2 / C = 20$		$C = 20$
F	0,003746	0,003746	0,003757	0,003757	0,000612
F'	0,000375	0,000375	0,000376	0,000376	0,000061
Q	0,007843	0,007843	0,008098	0,008098	0,000753
Eşik Sayısı/ Küme	$m = 3 / C = 32$		$m = 3 / C = 30$		$C = 32$
F	0,002323	0,002323	0,002416	0,002444	0,000170
F'	0,000249	0,000249	0,000242	0,000259	0,000017
Q	0,003944	0,003944	0,004367	0,004448	0,000164
Eşik Sayısı/ Küme	$m = 4 / C = 51$		$m = 4 / C = 48$		$C = 51$
F	0,001108	0,001108	0,001205	0,001205	0,000242
F'	0,000125	0,000125	0,000128	0,000128	0,000024
Q	0,001348	0,001348	0,001492	0,001492	0,000177

Eşik sayısı $m=1$ olarak seçildiğinde, Otsu yöntemi ile alınan sonuç Şekil 5.6 (a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.6 (b)' de ve FCM algoritmasının çıktısı ise Şekil 5.6 (c)' de gösterilmiştir. Otsu algoritması ile doldurulan küme sayısı, 8 ve geçen süre 0,244 saniyedir. Öte yanda Kapur algoritması ile 7 prizma aktif olmuştur. Karşılaştırma yapmak amacı ile küme sayısı, $c=8$ seçilerek FCM algoritmasının çalıştırıldığında geçen süre 8,150 saniyedir.

Otsu ile elde edilen sonuca bakıldığında, Lena'nın şapka, yüz ve aksesuarında bulunan detayların Kapur yöntemine göre daha iyi sonuç verdiği görülmektedir. Yine Lena'nın yüz ve vücut bölgesi daha net bir biçimde ortaya çıkmış olup, orijinal görüntüye en yakın sonucu vermiştir. Bununla birlikte görüntüdeki arka plan bölgesi, Kapur yönteminde daha detaylı bir biçimde ortaya çıktığı görülmektedir. Her iki yöntemde de tek eşikte yer yer Lena'nın arka planı ile ön planı iç içe geçtiği görülmektedir. FCM yönteminde görüldüğü gibi Lena'nın yüz kısmında aşırı ayrıştırma problemi gözlenmektedir.

Eşik sayısı $m=2$ olarak seçildiğinde, Otsu yöntemi ile alınan sonuç Şekil 5.7 (a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.7 (b)' de ve FCM algoritmasının çıktısı ise Şekil 5.7 (c)' de gösterilmiştir. Görüntü üzerinde detayların daha belirginleştiği görülmektedir. Özellikle şapka ve aksesuardaki detayların hem Otsu hem de Kapur yönteminde daha belirgin bir şekilde ortaya çıktığı açıktır. FCM yönteminde ise hemen hemen her bölgede aşırı ayrıştırma problemi dikkat çekmektedir.

Eşik sayısı $m=3$ olarak alındığında, Otsu yöntemi ile alınan sonuç Şekil 5.8 (a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.8 (b)' de ve FCM algoritmasının çıktısı ise Şekil 5.8 (c) de gösterilmiştir. Otsu amaç fonksiyonu ile doldurulan küme sayısı, 32 iken, Kapur algoritması ile 30 prizma aktif olmuştur. Karşılaştırma yapmak amacı ile küme sayısı, $c=32$ seçilerek FCM algoritmasının çalıştırıldığında geçen süre 19,605 saniyedir. Eşikleme işleminde detayların hemen hemen en ince ayrıntısına kadar ayrıştığı görülmektedir. Ancak arka planda, sağ üstte gözüken çerçeve Otsu ve Kapur ile düzgün şekilde ayrıştırılırken, FCM yöntemi ile bu bölge tek parça halinde çıkmayıp birden fazla bölgeye ayrılmıştır.

Son olarak eşik sayısı $m=4$ olduğunda, Otsu yöntemi ile alınan sonuç Şekil 5.9 (a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.9 (b)' de ve FCM algoritmasının çıktısı ise Şekil 5.9 (c)' de gösterilmiştir. Otsu amaç fonksiyonu ile doldurulan küme sayısı, 51 iken, Kapur

algoritması ile 48 prizma doldurulmuştur. Sınıf sayısı, $c=51$ seçilerek FCM algoritmasının çalıştırıldığında geçen süre 25,863 saniyedir. FCM algoritması ile hem işlem süresinin uzadığı hem de elde edilen çıktının aşırı ayrıştırmaya doğru gittiği açıktır.



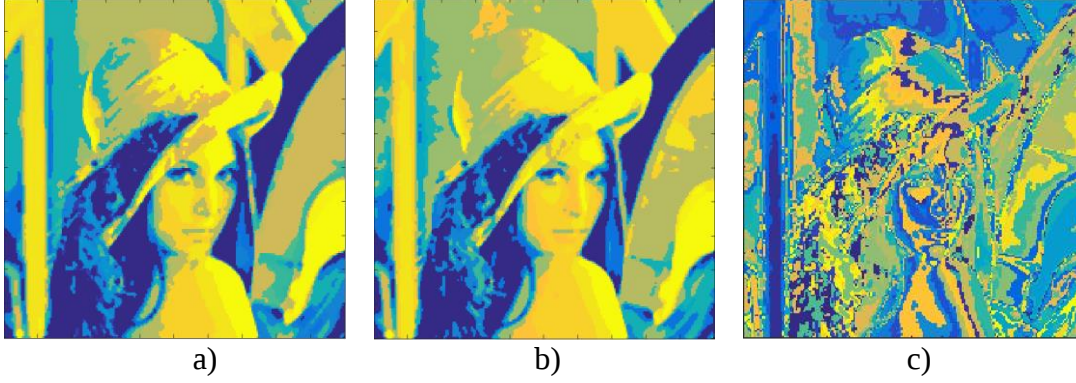
Şekil 5.6. Lena, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=7$ c) FCM, $c=8$, $t=8,150s$



Şekil 5.7. Lena, $m=2$ a) Otsu, $c=19$ b) Kapur $c=20$ c) FCM, $c=20$, $t=11,013s$



Şekil 5.8. Lena, $m=3$ a) Otsu, $c=32$ b) Kapur, $c=30$, c) FCM, $c=32$, $t=19,605s$

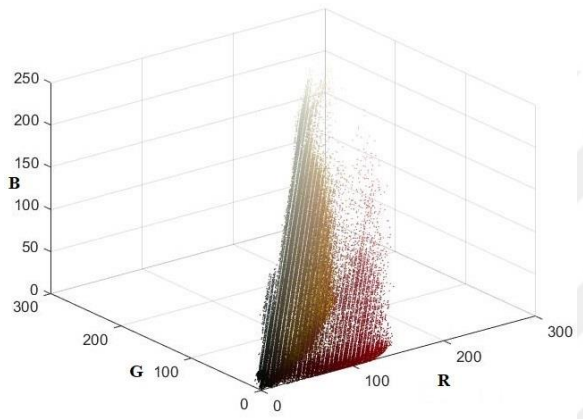


Şekil 5.9. Lena, $m=4$ a) Otsu, $c=51$ b) Kapur, $c=48$ c) FCM, $c=51$, $t=25,863s$

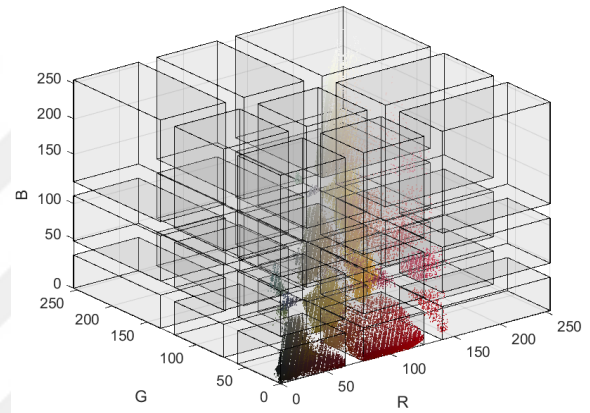
Geliştirilen algoritmada eşik değerleri renk kanallarına ait histogramlardan elde edilmekte ve akabinde renk uzayının bölütlenmesinde kullanılmaktadır. Lena görüntüsü ile yapılan deney sonuçları daha önce verilmiştir. Ancak algoritmanın performansın daha detaylı incelemek için farklı histogram karakteristikleri ve renk dağılımlarına sahip görüntüler ile de testlerin yapılması önemlidir. Bu nedenle Şekil 5.10 (a)' da orijinali, Şekil 5.10 (b)' de renk dağılımı, Şekil 5.10 (c)' de, iki eşikle renk uzayının bölütlenmesi ve Şekil 5.10 (d)' de histogram bilgileri verilen Ayakkabı görüntüsü için ilgili deneyler tekrarlanmıştır. Daha önce olduğu gibi doğrusal, OOA ve PSO teknikleri ile tahmin edilen eşikler ve hesaplama süreleri Çizelge 5.5 de verilmiştir. Lena da olduğu gibi bazı kümeler boş kalmıştır. Ayırıştırma performans çıktıları ise Çizelge 5.6 da gösterilmiştir.



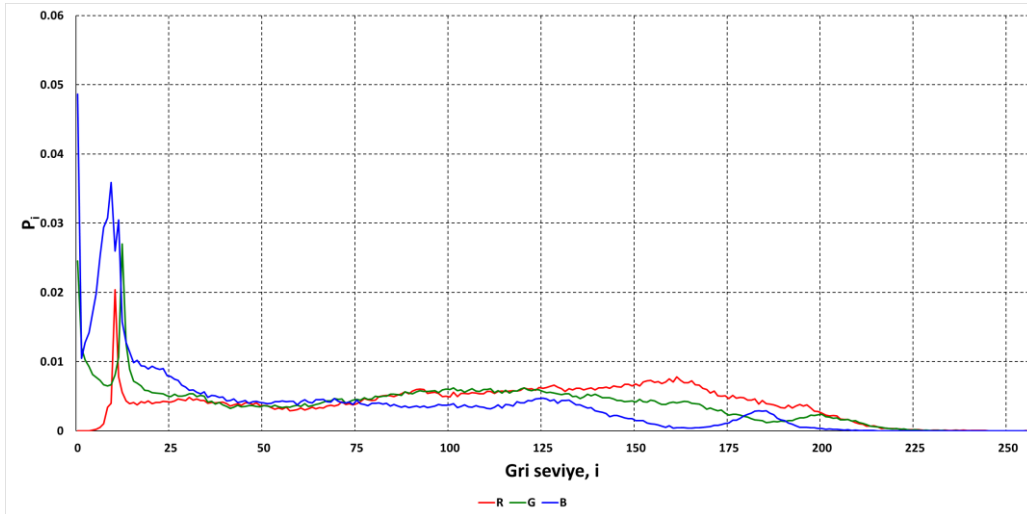
a)



b)



c)



d)

Şekil 5.10. Ayakkabı(481x321) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram

Çizelge 5.5. Ayakkabı görüntüsü için bulunan eşikler

Otsu						Kapur						
Doğrusal			OOA		PSO	Doğrusal		OOA		PSO		
			t ₁					t ₁				
R	103		103		103	110		110		110		
G	84		84		84	111		113		113		
B	73		73		73	86		86		86		
Süre,t (s)	0,018		0,235		0,236	0,032		0,302		0,309		
Küme:c			8/8					6/8				
	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂
R	71	137	71	137	71	137	81	146	81	145	81	146
G	61	131	61	131	61	131	46	145	77	146	77	145
B	47	111	47	111	47	111	67	137	68	135	68	135
Süre,t (s)	5,486		1,125		1,201	6,108		1,051		1,212		
Küme:c			19/27					18/27				
	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	60	112	157	60	112	155	60	118	168	66	117	169
G	47	99	149	47	99	149	57	113	169	57	112	170
B	40	92	148	40	92	149	40	90	141	41	90	140
Süre,t (s)	4,923			1,329		1,498	6,710		1,297		1,981	
Küme:c			29/64						30/64			
	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄
R	-	-	-	-	55	100	137	171	55	100	137	162
G	-	-	-	-	40	83	122	163	39	82	121	162
B	-	-	-	-	31	68	108	154	30	67	107	154
Süre,t (s)					1,222			1,357			1,689	1,236
Küme:c					52/125			51/125			49/125	

Çizelge 5.6. Ayakkabı performans ölçütleri: F, F' ve Q

Eşikleme Metodu	Otsu		Kapur		FCM
Algoritma	PSO	OOA	PSO	OOA	-
Eşik Sayısı/Küme	m = 1 / C = 8		m = 1 / C = 6		C = 8
F	0,002458	0,002458	0,002553	0,002553	0,000248
F'	0,000246	0,000246	0,000255	0,000255	0,000025
Q	0,009645	0,009645	0,010441	0,010441	0,000658
Eşik Sayısı/Küme	m = 2 / C = 19		m = 2 / C = 18		C = 19
F	0,001214	0,001214	0,001373	0,001373	0,000326
F'	0,000121	0,000121	0,000137	0,000137	0,000033
Q	0,003632	0,003632	0,004275	0,004275	0,000594
Eşik Sayısı/Küme	m = 3 / C = 29		m = 3 / C = 30		C = 30
F	0,000480	0,000480	0,000715	0,000715	0,000165
F'	0,000048	0,000048	0,000071	0,000071	0,000065
Q	0,001183	0,001183	0,001940	0,001940	0,000335
Eşik Sayısı/ Küme	m = 4 / C = 51	m = 4 / C = 52	m = 4 / C = 49		C = 52
F	0,000357	0,000375	0,000418	0,000413	0,000045
F'	0,000039	0,000044	0,000042	0,000041	0,000004
Q	0,000821	0,000870	0,000936	0,000922	0,000029

Tek eşik kullanıldığından, Otsu amaç fonksiyonu ile alınan sonuç Şekil 5.11 (a)' da, Kapur ile alınan sonuç Şekil 5.11 (b)' deki gibidir. Şekil 5.11 (c)' ise FCM algoritmasının tepkisini göstermektedir. Otsu ile 8 küme aktif olur iken, Kapur ile 6 prizma dolmuştur. Sınıf sayısı 8 seçildiğinde, FCM algoritması 19,524 saniyede tepki vermiştir. Otsu ve Kapur ile beyaz ve sarı ayakkabıların zeminden ayrıldığı söylenebilir. Ancak FCM de ayakkabılar için gereksiz yere fazla sınıf ataması yapıldığı söylenebilir. İki eşikli Otsu ile alınan sonuç Şekil 5.12 (a)' da, entropi ile alınan tepki ise Şekil 5.12 (b)' de gösterilmiştir.

Ayrıca FCM algoritmasının çıktısı ise Şekil 5.12 (c)' de verilmiştir. Eşiklemede ayakkabıların taban ve üst kısımları ayrı ayrı kümelerle atanmıştır. FCM de ise ayakkabılar için ortalama olarak sınıf ayrılmıştır. Bir bakıma aşırı ayrıştırma yapılmıştır. Eşik sayısı ile küme sayısının artacağı beklenen bir durumdur. Görüntülerde farklı aydınlık şiddetine maruz kalmış birçok nesne var ise küme sayısının artırılması iyi bir yaklaşımdır. Ancak çok fazla nesne mevcut değil ise aşırı ayrıştırma kaçınılmaz olacaktır. Örneğin eşik sayısı 3 olarak seçildiğinde, maksimum 64 küme ataması yapılabilir. Ancak hepsi için piksel ataması garanti değildir. Ayakkabı deneyinde, Otsu yöntemi ile alınan sonuç Şekil 5.13 (a)' da Kapur yaklaşımı ile bulunan sonuç Şekil 32 (b)' de ve FCM algoritmasının çıktısı ise Şekil 5.13(c)' de gösterilmiştir. Otsu yöntemi ile aktif olan küme sayısı, 29 iken, Kapur algoritması ile 30 prizma ya atama yapılmıştır. FCM algoritması $c=30$ seçilerek çalıştırıldığında, 101,154 saniyelik süre harcanmıştır. Buna rağmen oluşan sınıfların nesneleri temsil etme yeteneği oldukça azalmıştır. Çok seviyeli eşiklemede ise detaylara doğru bir kayma gözlemlenmiştir.

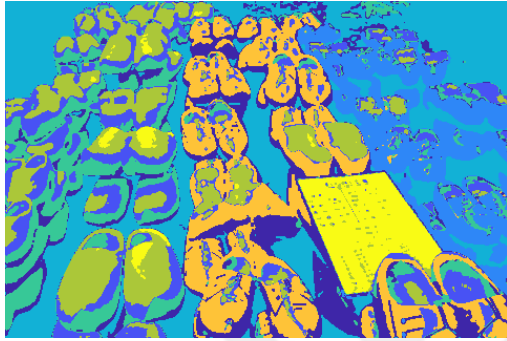
Son deneyde eşik sayısı $m=4$ olduğunda, Otsu yaklaşımı ile elde edilen sonuç Şekil 5.14 (a)' da Kapur ile bulunan sonuç Şekil 5.14 (b)' de ve FCM yönteminin çıktısı ise Şekil 5.14 (c)' de gösterilmiştir. Otsu ile aktif olan alt küp sayısı, 52 iken, Kapur algoritması ile 49 prizma doldurulmuştur. Aynı şekilde sınıf sayısı, $c=52$ seçilerek FCM algoritması çalıştırıldığında geçen süre 123,738 saniye olmuştur. Otsu ve Kapur yöntemlerinin çıktılarında ayakkabı bağcıklarının tespit edildikleri açıktır. Diğer taraftan bazı bölgelerin alanları çok küçük olduklarından fark edecek düzeyde değildirler. 4 eşikli sınıflandırma ile oluşan sınıfların pek anlamlı olduğu söylenemez. Aynı durum yüksek hesaplama maliyetine rağmen, FCM algoritmasının çıktıları içinde geçerlidir.



a)



b)



c)

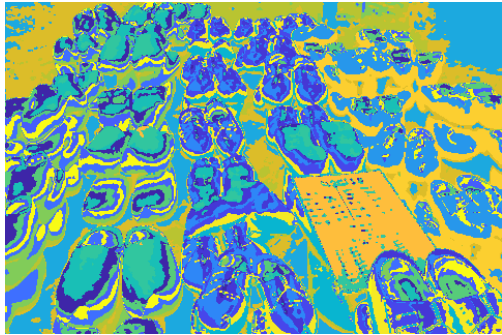
Şekil 5.11. Ayakkabı, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=6$ c) FCM, $c=8$, $t=19,524s$



a)

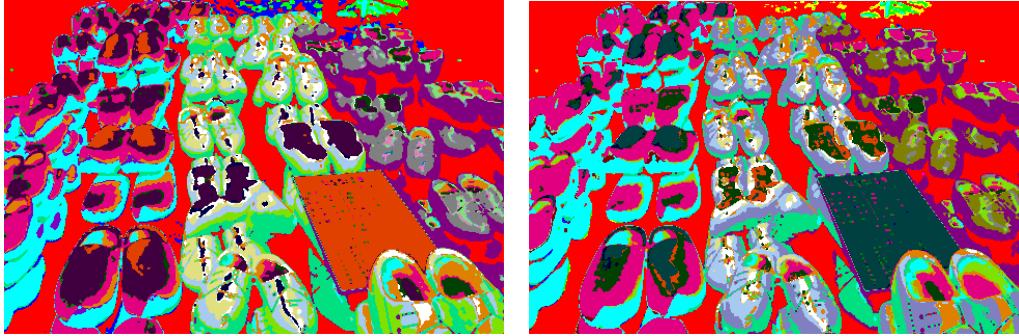


b)



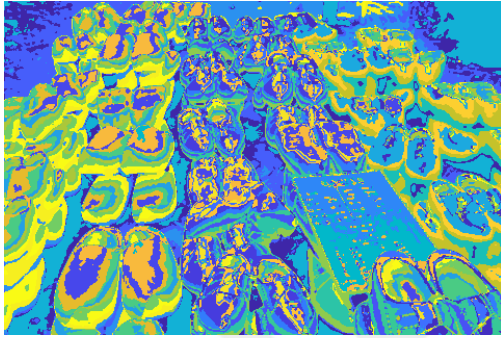
c)

Şekil 5.12. Ayakkabı, $m=2$ a) Otsu, $c=19$ b) Kapur, $c=18$ c) FCM, $c=19$, $t=54,051s$

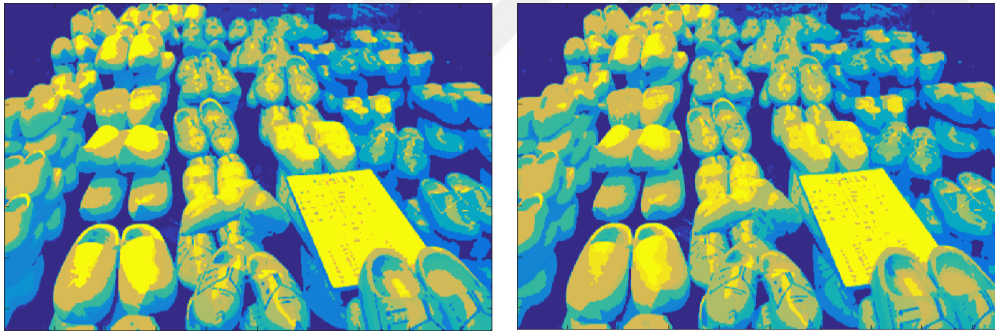


a)

b)

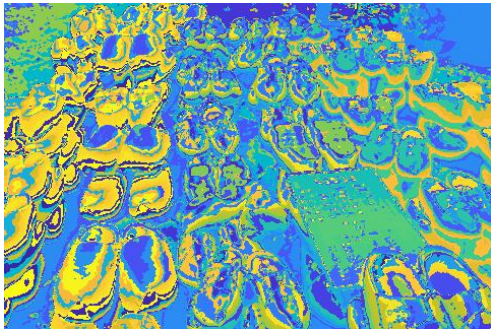


c)

Şekil 5.13. Ayakkabı, $m=3$ a) Otsu, $c=29$ b) Kapur, $c=30$ c) FCM, $c=30$, $t= 101,154s$ 

a)

b)



c)

Şekil 5.14. Ayakkabı, $m=4$ a) Otsu, $c=52$ b) Kapur, $c=49$ c) FCM, $c=52$, $t= 123,738 s$

Şekil 5.15 (a) mısır görüntüsünün orijinali göstermektedir. Ayrıca mısır görüntüsünün renk dağılımlı, iki eşikle renk uzayının bölütlenmesi ve RGB kanallarının histogramları sırasıyla Şekil 5.15 (b), Şekil 5.15 (c) ve Şekil 5.15 (d)' de verilmiştir. Çizelge 5.7 kaba kuvvet, OOA ve PSO yöntemleri ile mısır görüntüsü için tespit edilen eşik değerleri ile hesaplama sürelerini göstermektedir. Görüldüğü gibi optimizasyon yöntemleri ile iki eşikli doğrusal aramada yaklaşık 5 kat, üç eşikli doğrusal aramada ise yaklaşık 500 kat hesaplama maliyeti farkı vardır. Performans ölçütleri ise Çizelge 5.8' da verilmiştir. Küme sayısı 38 alınarak FCM çalıştırıldığında 89,561 saniyelik bir zaman dilimine ihtiyaç var iken, optimizasyon yöntemleri ile aynı sayıda sınıflandırma 1,119 saniyede tamamlanmıştır. Çok seviyeli eşiklemede her zaman aynı küme merkezlerini mümkün iken FCM de bunun bir garantisi yoktur.

Tek eşikli Otsu ve Kapur metotlarının neticeleri Şekil 5.16 (a) ve Şekil 5.16 (b) de gösterildiği gibidir. Ayrıca 8 küme için çalıştırılan FCM algoritmasının çıktısı Şekil 5.16(c) de gösterilmiştir. Önerilen yöntem sol ikinci mısır tanelerini düzgün bir şekilde ayırırken, FCM stratejisinde aşırı ayrıştırma problemi oluşmuştur. Bununla birlikte Otsu ve Kapur teknikleri diğer mısır tanelerinin izole edilmesinde zorlanırken, FCM algoritması daha iyi performans sergilemiştir. Ancak FCM nin hesaplama maliyeti Otsu ve Kapur yöntemlerinin yaklaşık 13 katıdır.

İki eşikli eşiklemede ise Otsu ve Kapur yöntemlerinin görsel tepkileri Şekil 5.17 (a) ve Şekil 5.17 (b)' de ve 21 küme için FCM algoritmasının çıktısı Şekil 5.17(c) de verilmiştir. Önerilen yöntem ve FCM algoritması da görüntüyü aynı küme sayısına bölütlese de, FCM nin çözüm süreci eşiklemenin yaklaşık 50 katıdır. Işık yansımından dolayı parlak olan mısırlar farklı ışık tonları ile temsil edilmektedirler. Dolayısıyla parlak olan mısır tanelerinin tek bir nesne olursa bile fazla sınıfa tahsis edilmeleri muhtemeldir.

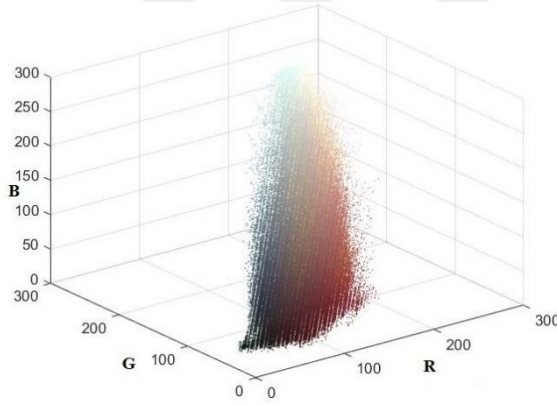
Üçlü eşikleme de Otsu yöntemi ile alınan sonuç Şekil 5.18(a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.18(b)' de ve FCM algoritmasının çıktısı ise Şekil 5.18(c)' de gösterilmiştir. Otsu amaç fonksiyonu ile doldurulan küme sayısı, 37 iken, Kapur algoritması ile 38 prizma aktif olmuştur.

Kümelerin homojenlikleri artınca ışık tonlarındaki farklılıklar çok küçük olsalar bile bir nesne için birden fazla sınıfın atanması muhtemeldir. Eşik sayısı dörde çıkarıldığında yukarda bahsedilen durum gözlenmiştir. Nitekim Şekil 5.19 (a) ve Şekil 5.19 (b)' deki

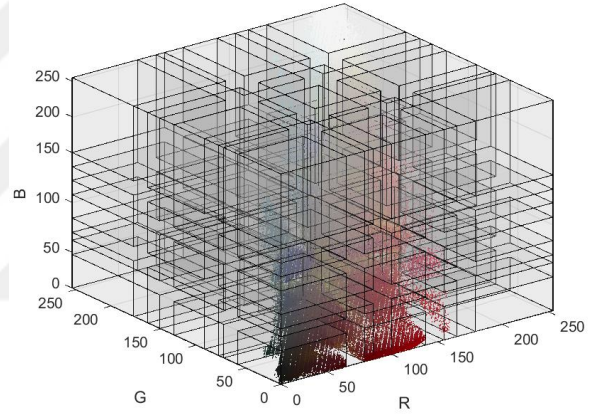
sonuçlar bu durumu teyit etmektedir. Şekil 5.19 (c)' de gösterilen ve FCM algoritmasından alınan sonuçlar tipik örnektir.



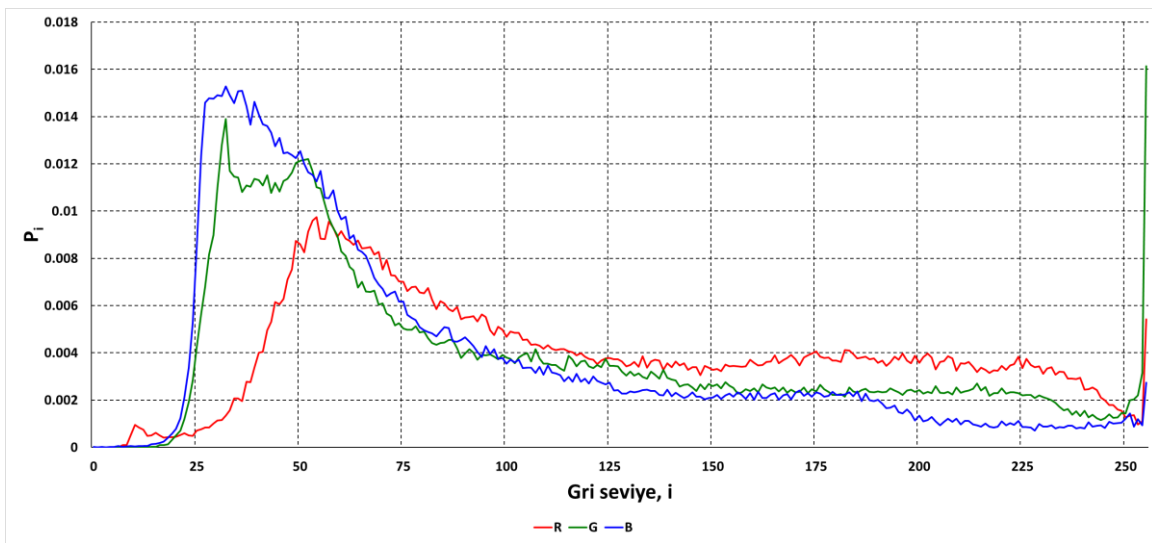
a)



c)



c)



d)

Şekil 5.15. Mısır(481x321) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram

Çizelge 5.7. Mısır görüntüsü için bulunan eşikler

Otsu						Kapur								
Doğrusal			OOA		PSO	Doğrusal		OOA		PSO				
			t ₁					t ₁						
R	134		134		134	136		137		137				
G	125		124		124	128		128		128				
B	113		112		112	125		126		126				
Süre,t (s)	0,075		0,234		0,241	0,032		0,267		0,266				
Küme:c			8/8					8/8						
	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂		
R	100	171	103	175	99	169	104	179	105	179	40	142		
G	88	168	93	169	85	160	93	169	93	169	93	169		
B	78	150	80	155	77	146	104	191	100	187	100	187		
Süre,t (s)	5,452		1,435		1,498		6,513		1,104		2,293			
Küme:c			21/27					19/27						
	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	83	136	193	82	135	191	83	135	191	87	142	190		
G	74	130	192	73	124	181	70	120	179	77	133	190		
B	65	115	177	63	109	170	65	115	177	79	137	196		
Süre,t (s)	510,549			1,157			1,476			670,834				
Küme:c				37/64					38/64					
	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄		
R	-	-	-	-	72	111	154	200	72	111	155	201		
G	-	-	-	-	61	100	146	195	60	98	144	193		
B	-	-	-	-	54	90	138	191	54	89	136	189		
Süre,t (s)					5,56					5,55				
Küme:c					59/125					61/125				
	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄		
R	-	-	-	-	72	111	154	200	-	-	-	-		
G	-	-	-	-	61	100	146	195	-	-	-	-		
B	-	-	-	-	54	90	138	191	-	-	-	-		
Süre,t (s)					5,56					5,55				
Küme:c					59/125					61/125				

Çizelge 5.8. Mısır performans ölçütleri: F, F' ve Q

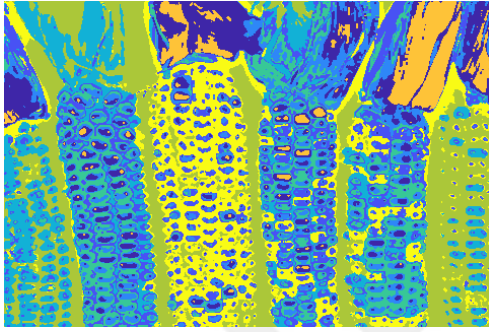
Eşikleme Metodu	Otsu		Kapur		FCM
Algoritma	PSO	OOA	PSO	OOA	-
Eşik Sayısı/Küme	m = 1 / C = 8		m = 1 / C = 8		C = 8
F	0,002104	0,002104	0,002135	0,002135	0,000483
F'	0,000210	0,000210	0,000213	0,000213	0,000048
Q	0,008994	0,008994	0,009450	0,009450	0,001281
Eşik Sayısı/Küme	m = 2 / C = 21		m = 2 / C = 19		C = 21
F	0,001219	0,001219	0,001263	0,001263	0,000178
F'	0,000122	0,000122	0,000126	0,000126	0,000018
Q	0,004143	0,004143	0,004779	0,004779	0,000311
Eşik Sayısı/Küme	m = 3 / C = 37		m = 3 / C = 38		C = 38
F	0,000757	0,000757	0,000900	0,000900	0,000081
F'	0,000076	0,000076	0,000093	0,000093	0,000008
Q	0,002112	0,002112	0,002957	0,002957	0,000150
Eşik Sayısı/ Küme	m = 4 / C = 61		m = 4 / C = 52		C = 61
F	0,000465	0,000457	0,000748	0,000727	0,000084
F'	0,000048	0,000046	0,000075	0,000073	0,000008
Q	0,001061	0,001046	0,002427	0,002310	0,000096



a)

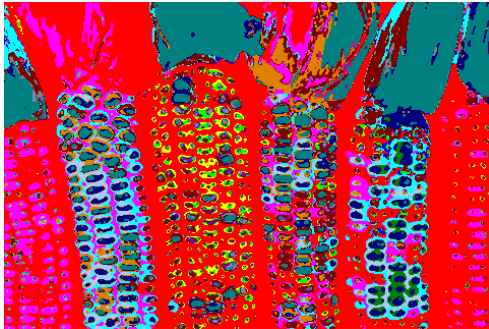


b)

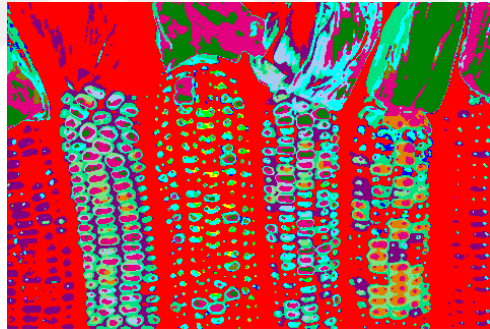


c)

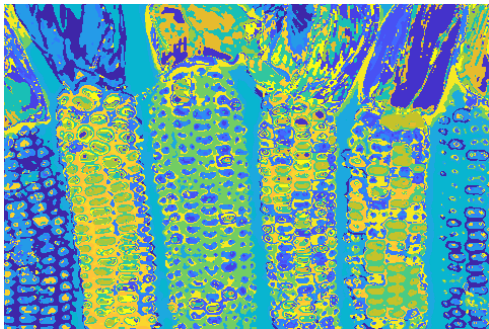
Şekil 5.16. Mısır, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=8$ c) FCM, $c=8$, $t=19,151s$



a)

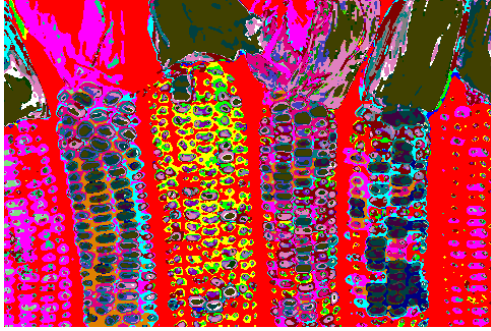


b)



c)

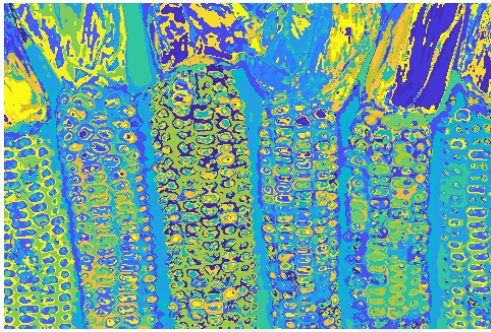
Şekil 5.17. Mısır, $m=2$ a) Otsu, $c=21$ b) Kapur, $c=19$ c) FCM, $c=21$, $t=55,350s$



a)

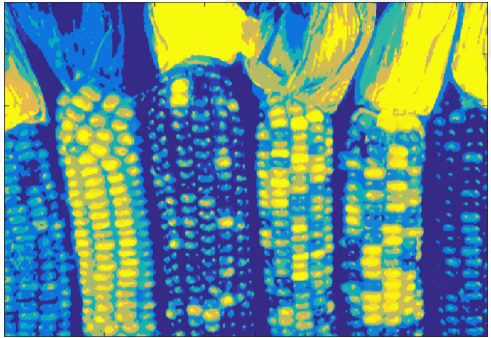


b)

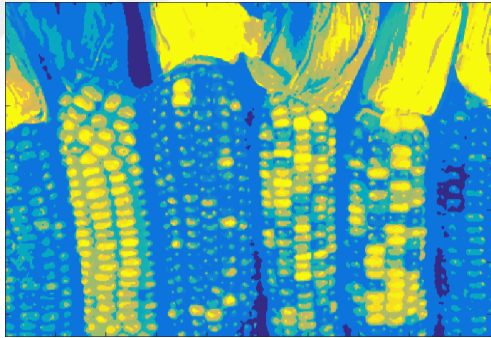


c)

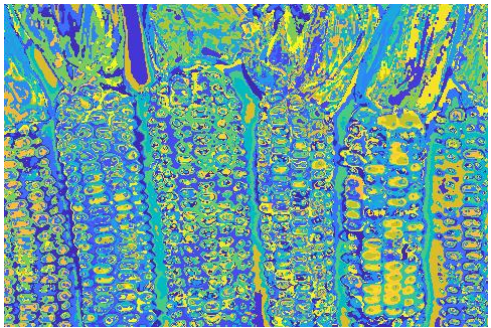
Şekil 5.18. Mısır, $m=3$ a) Otsu, $c=37$ b) Kapur, $c=38$ c) FCM, $c=38$, $t=89,561s$



a)



b)



c)

Şekil 5.19. Mısır, $m=4$ a) Otsu, $c=61$ b) Kapur, $c=52$ c) FCM, $c=61$, $t=133,562s$

Biber görüntüsünün orijinali, renk dağılımı, iki eşikle renk uzayının bölütlenmesi ve histogram bilgileri sırasıyla Şekil 5.20 (a), Şekil 5.20 (b), Şekil 5.20 (c) ve Şekil 5.20 (d)' de gösterilmiştir. Ayrıca Çizelge 5.9' da doğrusal, OOA ve PSO yöntemleri ile tespit edilen eşikler ve hesaplama süreleri yer almıştır. Histogram bilgileri kullanılarak hesaplanan eşikler Şekil 5.20 (b)' deki renk dağılımı gruplandırmak için kullanılmıştır. Bazı prizmaların boş kaldığı aktif küme sayılarından görülmektedir. Çizelge 5.10' de ise F , F' ve Q gibi ayrıştırma performans parametreleri yer almıştır.

Tek eşikli Otsu, Kapur sonuçları ve FCM algoritmasının tepkileri sırasıyla Şekil 5.21(a), Şekil 5.21 (b) ve Şekil 5.21 (c)' de gösterildiği gibidir. Otsu yaklaşımın tepkisinde ahşap kutunun tek bir parça halinde ayırt edilmesi barizdir. Aynı zamanda fiyat kâğıdı da bir nesne olarak tasnif edilmiştir. Ancak aynı durumun varlığını Kapur ve FCM yöntemleri için söylemek mümkün değildir. Ayrıca Otsu ve Kapur algoritmaları biberler için bir sınıf tahsis eder iken, FCM gereksiz yere fazla kümeleme yapmıştır.

Otsu, Kapur stratejilerinin iki eşikli çıktıları ve FCM tekniğinin ürettiği sonuçlar sırasıyla Şekil 5.22 (a), Şekil 5.22 (b) ve Şekil 5.22' de gösterilmiştir. Görüntüdeki detaylar daha belirgin olmuştur. Örneğin fiyat etiketi üzerindeki rakamlar üç algoritmada da başarıyla bölütlenmiştir. Ancak Otsu ve Kapur da ahşap kutu için aşırı ayrıştırma problemi görülmeye başlanmıştır. FCM de ise söz konusu problem netleşmiştir. Aktif küme sayısının 32 olduğu Otsu metodu ile aktif küme sayısının 27 olduğu entropi tekniğinin sonuçları sırasıyla Şekil 5.23(a) da ve Şekil 5.23 (b)' de görülmektedir. Şekil 5.23 (c) ise Biber görüntüsü FCM 'nin ürettiği sonuçları temsil etmektedir. Küme sayısı 32 iken FCM 89,572 saniyede sonuç vermiştir ve üstelik aşırı ayrıştırma problemini beraberinde getirmiştir.

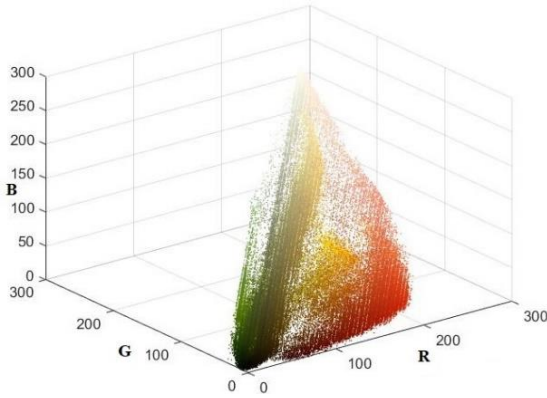
Eşik sayısı 4 olduğunda maksimum 125 küme oluşacaktır. Çok özel uygulamalar hariç, herhangi bir görüntünün 125 farklı sınıf içirmesi pek pratik değildir. Bununla birlikte dört eşik kullanıldığında, Otsu ile Şekil 5.24 (a)' da verilen tepki, Kapur ile Şekil 5.24 (b) deki çıktı üretilmiştir. Otsu da 69 aktif küme olduğundan, aynı küme sayısı ile FCM çalıştığında 157,186 saniyelik zamana ihtiyaç duyulmuştur ve elde edilen çıktı Şekil 5.24 (c)' de sunulmuştur.

Otsu ve Kapur yöntemlerinin çıktılarında bazı bölgeler çok küçük olduğundan insan gözü ile fark edilmeyecek bir düzeydedir. Ayrıca ışık tonları farklı olduğundan bazı biberler için 2 veya 3 sınıfa ayrılmışlardır. Diğer taraftan FCM metodunun çıktılarında anlamsal

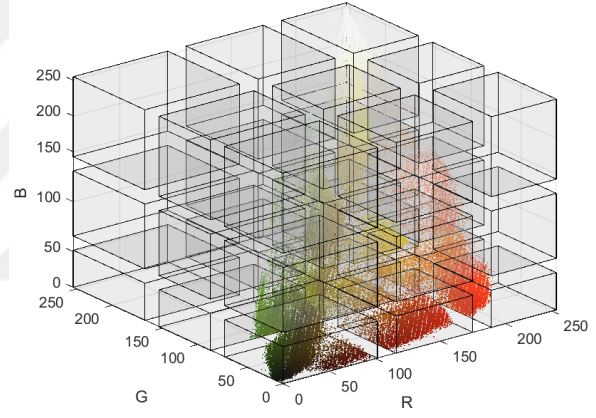
bütünlüğün bozulduğu söylenebilir. Bunlara ilave olarak F , F' ve Q performans değerlendirme ölçütleri hem önerilen hem de FCM algoritmasında sıfıra yakın olmuştur.



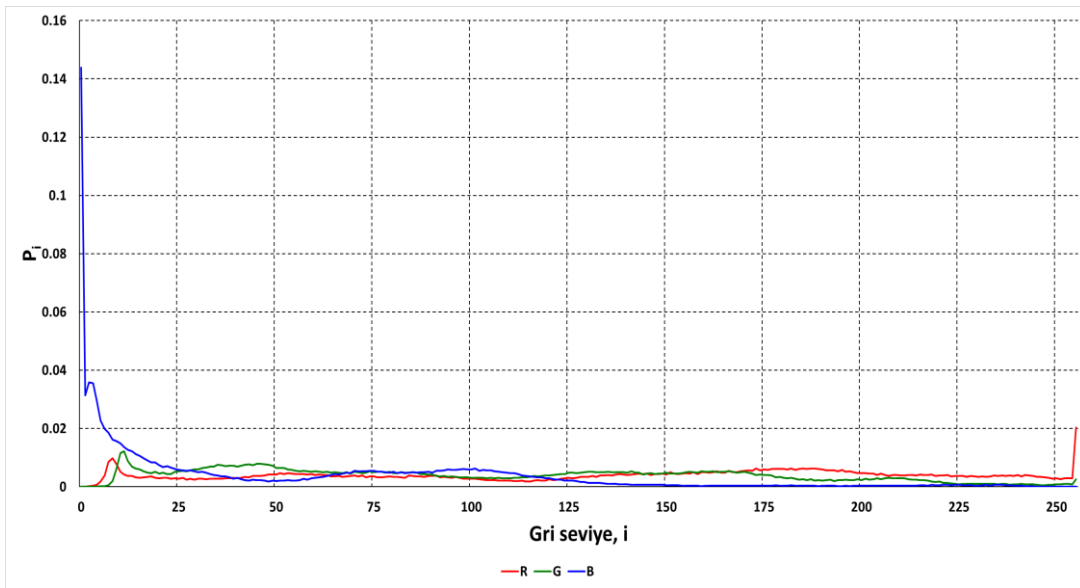
a)



b)



c)



d)

Şekil 5.20. Biber (321x481) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram

Çizelge 5.9. Biber için bulunan eşikler

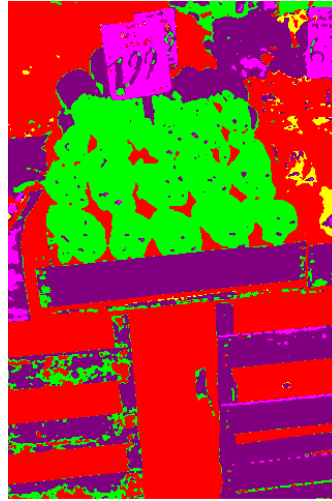
Otsu						Kapur																		
Doğrusal			OOA			PSO			Doğrusal			OOA			PSO									
			t ₁									t ₁												
R	124		124			124			132		132			132										
G	108		108			108			132		132			132										
B	58		58			58			132		133			133										
Süre,t (s)	0,012		0,198			0,205			0,031		0,213			0,221										
Küme:c				8/8									6/8											
t ₁ t ₂			t ₁ t ₂			t ₁ t ₂			t ₁ t ₂			t ₁ t ₂			t ₁ t ₂									
R	102	185	97	176		98	177		87	169	88	170	88	170										
G	82	154	80	151		80	151		91	178	91	177	91	177										
B	52	143	52	143		52	142		78	144	79	146	79	146										
Süre,t (s)	5,498		1,301			1,403			6,628		1,268			2,271										
Küme:c				18/27									16/27											
t ₁ t ₂ t ₃			t ₁ t ₂ t ₃			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	63	129	196	60	124	189	61	125	189	71	133	194	72	134	194	72	134	195						
G	59	116	178	60	116	174	59	115	175	66	125	184	68	126	183	68	125	182						
B	40	90	157	39	89	156	40	90	156	76	133	188	76	132	189	53	130	187						
Süre,t (s)	509,673			1,633			1,772			674,297			1,683			2,141								
Küme:c				32/64									27/64											
t ₁ t ₂ t ₃ t ₄			t ₁ t ₂ t ₃ t ₄			t ₁ t ₂ t ₃ t ₄			t ₁ t ₂ t ₃ t ₄			t ₁ t ₂ t ₃ t ₄			t ₁ t ₂ t ₃ t ₄									
R	-	-	-	-	54	111	162	206	53	109	161	205	-	-	-	-	56	102	152	202	56	102	152	202
G	-	-	-	-	54	101	145	188	54	101	146	189	-	-	-	-	65	120	173	220	65	120	173	220
B	-	-	-	-	19	54	94	158	19	54	94	158	-	-	-	-	34	83	134	189	34	84	134	189
Süre,t (s)				5,84			5,73						5,55			5,68								
Küme:c				69/125			67/125						54/125											

Çizelge 5.10. Biber performans ölçütleri: F, F' ve Q

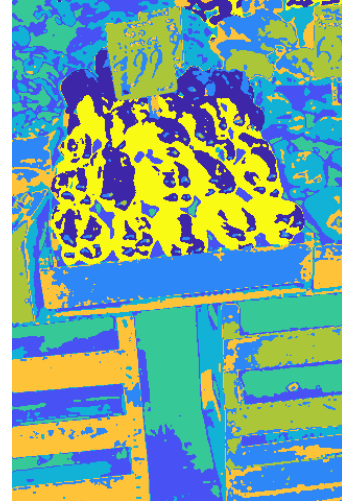
Eşikleme Metodu	Otsu		Kapur		FCM
Algoritma	PSO	OOA	PSO	OOA	-
Eşik Sayısı/Küme	$m = 1 / C = 8$		$m = 1 / C = 6$		$C = 8$
F	0,007621	0,007621	0,007227	0,007227	0,000666
F'	0,000762	0,000762	0,000723	0,000723	0,000067
Q	0,030131	0,030131	0,028407	0,028407	0,001851
Eşik Sayısı/Küme	$m = 2 / C = 18$		$m = 2 / C = 16$		$C = 18$
F	0,003008	0,003008	0,003523	0,003523	0,000647
F'	0,000301	0,000301	0,000352	0,000352	0,000065
Q	0,009102	0,009102	0,010593	0,010593	0,001192
Eşik Sayısı/Küme	$m = 3 / C = 32$		$m = 3 / C = 27$		$C = 32$
F	0,001787	0,001787	0,002313	0,002313	0,000555
F'	0,000179	0,000179	0,000231	0,000231	0,000056
Q	0,004321	0,004321	0,006095	0,006095	0,000814
Eşik Sayısı/ Küme	$m = 4 / C = 67$		$m = 4 / C = 54$		$C = 69$
F		0,001390	0,001462	0,001468	0,000111
F'		0,000148	0,000152	0,000155	0,000011
Q		0,003009	0,003415	0,003432	0,000101



a)



b)



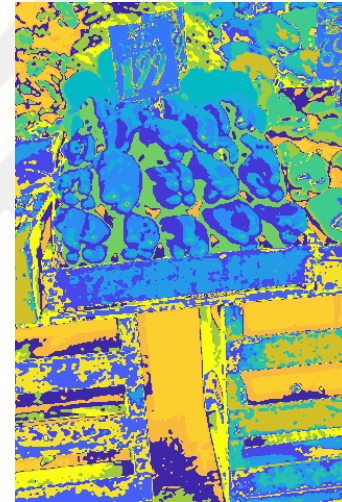
c)

Şekil 5.21. Biber, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=6$ c) FCM, $c=8$, $t=17,175$ s

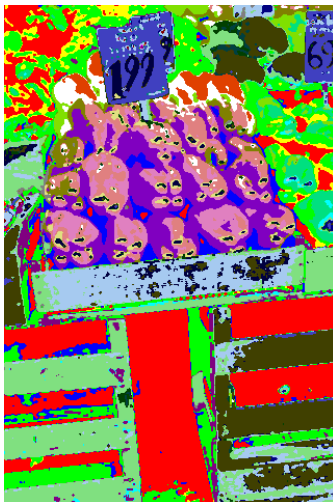
a)



b)



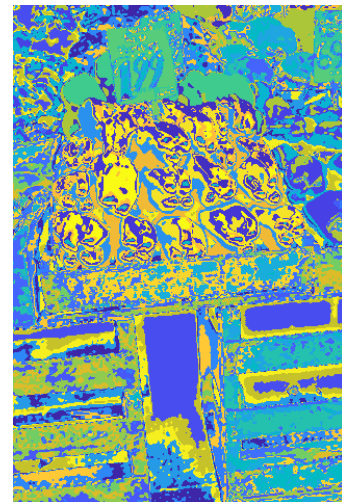
c)

Şekil 5.22. Biber, $m=2$ a) Otsu, $c=18$ b) Kapur, $c=16$ c) FCM, $c=18$, $t=48,723$ s

a)

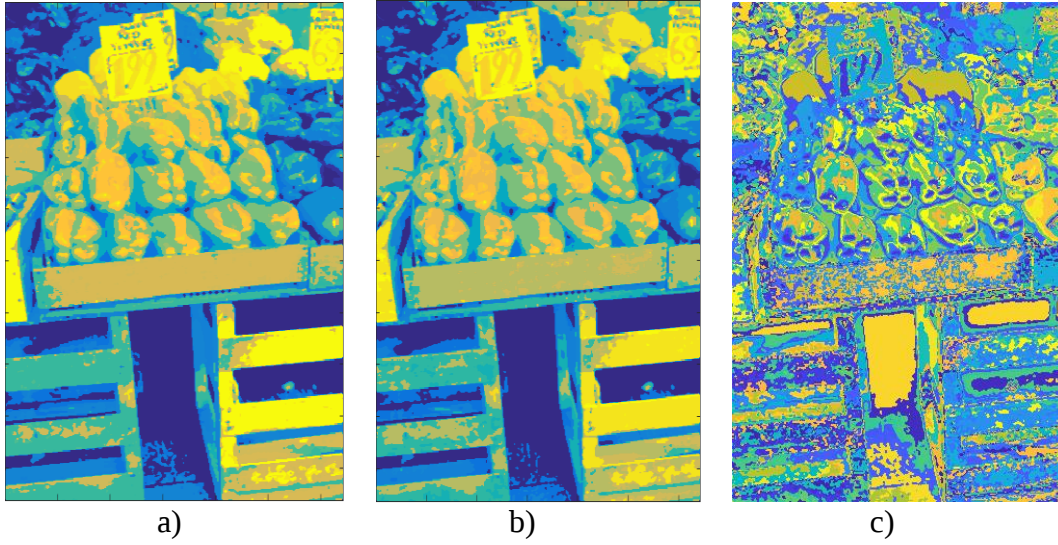


b)



c)

Şekil 5.23. Biber, $m=3$ a) Otsu, $c=32$ b) Kapur, $c=27$ c) FCM, $c=32$, $t=89,572$ s



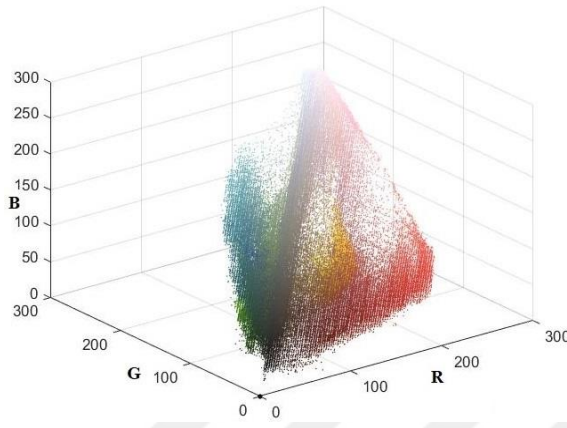
Şekil 5.24. Biber, $m=4$ a) Otsu, $c=69$ b) Kapur, $c=54$ c) FCM, $c=69$, $t=157,186s$

Geliştirilen algoritma görüntü işleme alanında referans olarak kullanılan ve Şekil 5.25 (a) verilen Papağan görüntüsü içinde test edilmiştir. Söz konusu olan görüntünün renk uzayındaki dağılımı, iki eşikle renk uzayının bölütlenmesi ve histogram bilgileri sırayla Şekil 5.25 (b), Şekil 5.25 (c) ve Şekil 5.25 (d)' de gösterilmiştir. Üç ayrı eşik hesaplama algoritması ile elde edilen eşik değerleri ve hesaplama süreleri Çizelge 5.11 de verilmiştir. Şekil 5.25 (b)' de verilen grafikte görüldüğü gibi Papağan görüntüsüne ait pikseller renk uzayında daha geniş alana yayılmıştır. Boş kümelerin oluşması kaçınılmaz olmakla birlikte, doluluk oranının daha fazla olacağı açıktır. Nitekim iki eşikte 24, üç eşikte 47 ve dört eşikte 84 kümeye piksel ataması yapılmıştır. F , F' ve Q performans değerlendirme ölçütleri de ise Çizelge 5.12 de mevcuttur.

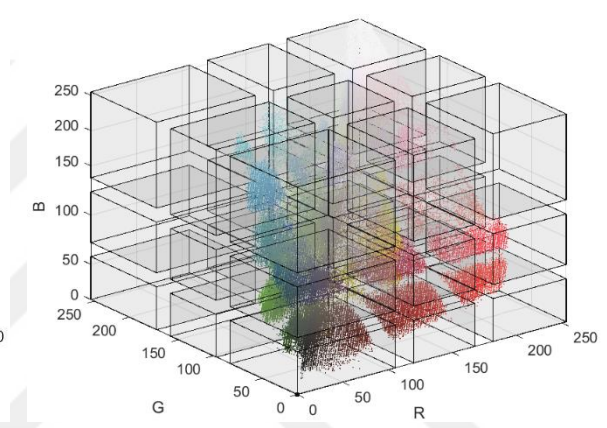
Eşik sayısı "1" verildiğinde, Otsu yaklaşımının çıktısı Şekil 5.26 (a)' da, Kapur metot ile üretilen sonuç Şekil 5.26 (b)' de ve FCM algoritmasından alınan çıktı ise Şekil 5.26(c) de sunulmuştur. Görüldüğü gibi her iki yöntemde de papağanlar arka plandan ayırt edilmiştir. Ayrıca yüz bölgesindeki çizgiler ve gözler net bir şekilde sınıflandırılmışlardır. Ancak FCM algoritmasında papağanların gözlerin de kayıplar olmuştur. Otsu ve Kapur yöntemleri iki eşikli çalıştırıldığında sırasıyla Şekil 5.27 (a) ve Şekil 5.27 (b)' deki sonuçlar alınmıştır. Görüldüğü gibi anlamsal bütünlük bozulmadan daha detaylı bir sınıflandırma yapılmıştır. Diğer taraftan Şekil 5.27(c)' de görüldüğü gibi 24 kümeli FCM de anlamsal bütünlük büyük ölçüde bozulmuştur.



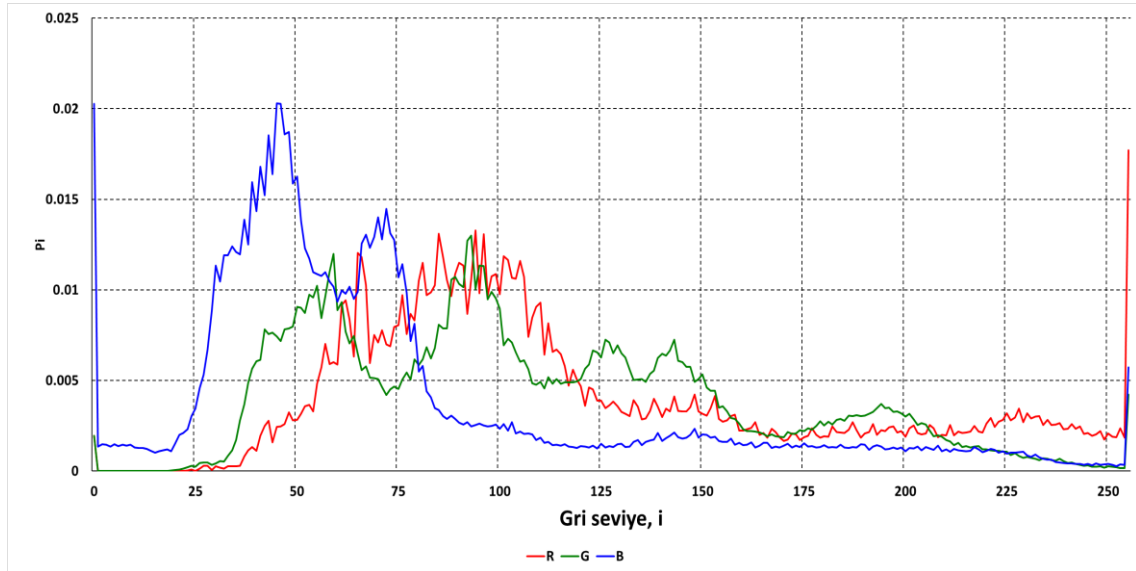
a)



b)



c)



d)

Şekil 5.25. Papağan (768x512) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram

Çizelge 5.11. Papağan için bulunan eşikler

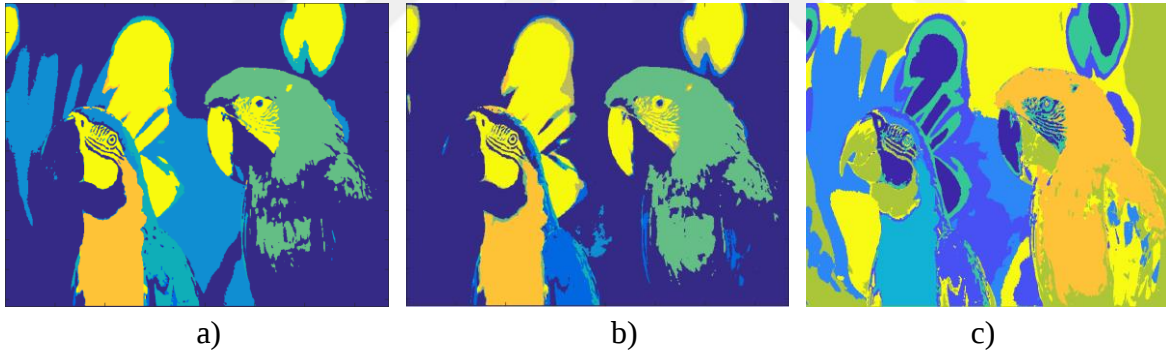
Otsu						Kapur										
Doğrusal			OOA		PSO	Doğrusal		OOA		PSO						
			t ₁					t ₁								
R	147		147		147	135		135		135						
G	120		120		120	154		154		154						
B	115		115		115	106		106		106						
Süre,t (s)	0,015		0,195		0,198	0,034		0,199		0,224						
Küme:c			8/8					8/8								
	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂	t ₁	t ₂				
R	108	178	102	169	102	169	118	184	118	184	118	184				
G	96	161	92	156	92	156	102	162	102	162	102	162				
B	61	133	61	130	61	130	88	164	85	164	85	164				
Süre,t (s)	5,602		1,135		1,421		6,234		1,137		2,051					
Küme:c			24/27				23/27									
	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	130	192	85	128	186	85	128	186	112	160	206				
G	76	118	170	76	118	169	76	118	169	100	155	208				
B	57	109	177	57	107	172	57	107	172	83	138	192				
Süre,t (s)	507,833			1,095			1,742			670,419	1,197	1,482				
Küme:C	47/64									45/64						
	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄				
R	-	-	-	-	82	119	162	207	82	119	162	207				
G	-	-	-	-	73	108	138	177	73	108	138	177				
B	-	-	-	-	53	86	129	182	53	86	129	182				
Süre,t (s)					5,06					5,12						
Küme:c					84/125								69/125			

Çizelge 5.12. Papağan performans ölçütleri: F, F' ve Q

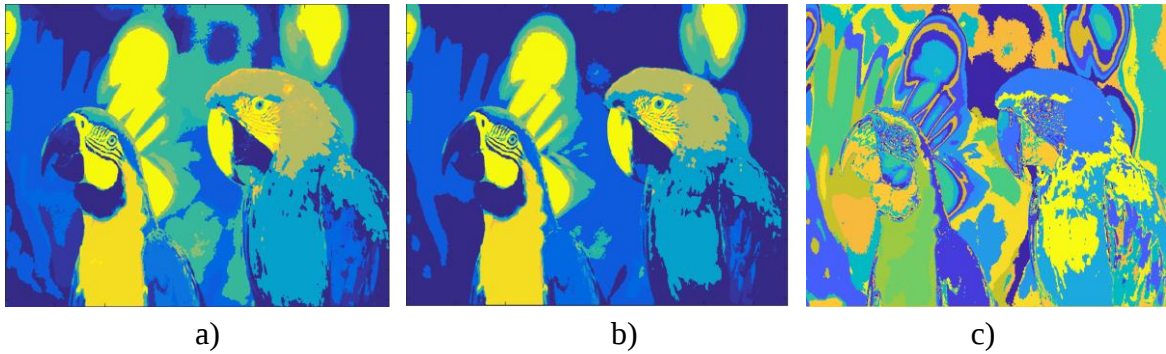
Eşikleme Metodu	Otsu		Kapur		FCM
Algoritma	PSO	OOA	PSO	OOA	-
Eşik Sayısı/Küme	m = 1 / C = 8		m = 1 / C = 8		C = 8
F	0,003166	0,002111	0,002747	0,002747	0,000660
F'	0,000317	0,000211	0,000275	0,000275	0,000066
Q	0,019512	0,013008	0,019274	0,019274	0,002660
Eşik Sayısı/Küme	m = 2 / C = 24		m = 2 / C = 23		C = 24
F	0,001439	0,000959	0,001621	0,001621	0,000179
F'	0,000144	0,000096	0,000162	0,000162	0,000001
Q	0,005399	0,003599	0,008425	0,008425	0,000489
Eşik Sayısı/Küme	m = 3 / C = 47		m = 3 / C = 45		C = 47
F	0,000862	0,000575	0,001716	0,001716	0,000164
F'	0,000086	0,000057	0,000181	0,000181	0,000016
Q	0,002697	0,001798	0,008960	0,008960	0,000317
Eşik Sayısı/ Küme	m = 4 / C = 84		m = 4 / C = 69		C = 84
F	0,000670	0,000446	0,000631	0,000631	0,000132
F'	0,000069	0,000046	0,000068	0,000068	0,000013
Q	0,001913	0,001275	0,002129	0,002129	0,000217

Renkli görüntülerin çok seviyeli eşiklemesi için geliştirilen algoritma, üç eşik kullanılarak deneyler yapılmıştır. Otsu yöntemi ile alınan sonuç Şekil 5.28 (a)' da, Kapur yaklaşımı ile bulunan sonuç Şekil 5.28 (b)' de verilmiştir. FCM algoritmasının çıktısı ise Şekil 5.28 (c)' de gösterilmiştir. Karşılaştırma yapmak amacı ile küme sayısı, $c=47$ seçilerek FCM algoritması çalıştırıldığında geçen süre 247,708 saniyedir. Otsu ve Kapur eşikleme neticesinde papağanların gagaları net bir şekilde ayrılmış iken, FCM algoritmasının çıktılarında bazı bölgeler ilgisiz sınıflara atanmıştır

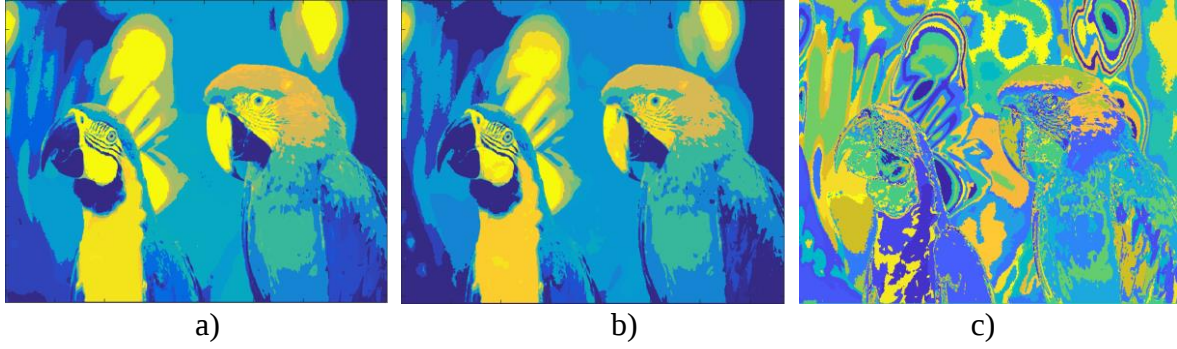
Son deneyde eşik sayısı $m=4$ olarak atanmış, Otsu algoritması ile Şekil 5.29 (a)' daki, Kapur tekniği ile Şekil 5.29 (b)' deki sonuçlar alınmıştır. 84 küme sayısı ile FCM çalıştırıldığında ise Şekil 5.29 (c)' de verilen sonuç alınmıştır. FCM algoritması çalıştırıldığında geçen süre ise 492,019 saniyedir. Geliştirilen arayüzde her sınıfa atanan piksel sayıları sayılmaktadır. Diğer taraftan hem çok seviyeli eşiklemede hem de FCM' de oluşan bazı bölgelerin piksel sayısı çok az olduğundan fark edilmeleri güçtür. Ayrıca her iki yaklaşımda da aşırı ayırtırmaya doğru bir yönelme olmuştur.



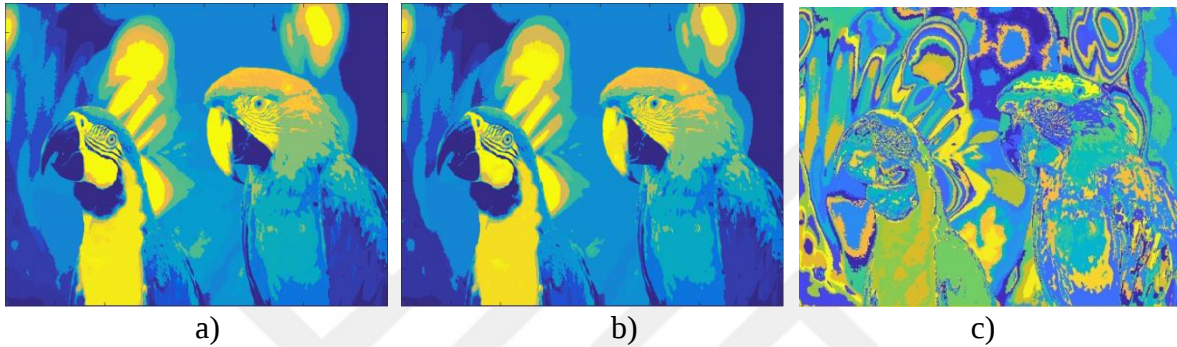
Şekil 5.26. Papağan, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=8$ c) FCM, $c=8$, $t=46,555s$



Şekil 5.27. Papağan, $m=2$ a) Otsu, $c=24$ b) Kapur, $c=23$ c) FCM, $c=24$, $t=125,738s$



Şekil 5.28. Papağan, $m=3$ a) Otsu, $c=47$ b) Kapur, $c=45$ c) FCM, $c=47$, $t= 247,708s$



Şekil 5.29. Papağan, $m=4$ a) Otsu, $c=84$ b) Kapur, $c= 69$ c) FCM, $c=84$, $t= 492,019s$

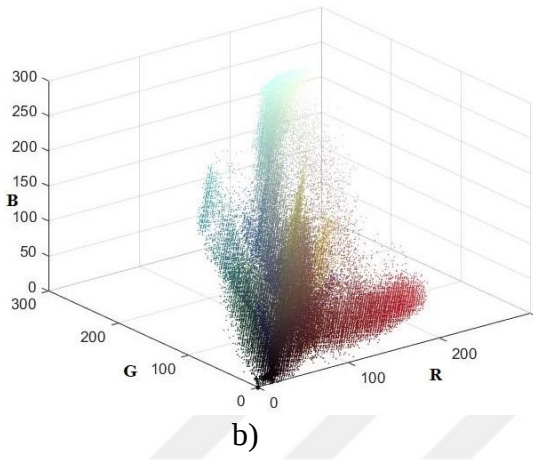
Görüntü işleme algoritmalarını denemek için kullanılan resimlerden biri de Şekil 5.30 (a)' gösterilen Kız Çocuğu görüntüsüdür. RGB renk uzayında geniş alana yayılmış olan renk dağılımı Şekil 5.30 (b)' de, iki eşikle renk uzayının bölütlenmesi Şekil 5.30 (c)' de, renk kanallarına ait histogramlarda Şekil 5.30 (d)' de verilmiştir. Renk uzayında oluşturulan prizmaların yarıya yakınına piksel ataması yapılmış olup, üç farklı optimizasyon algoritması ile tahmin edilen eşikler Çizelge 5.13' de listelenmiştir. Eşik sayısı artıkça ayırıştırma performans değerleri Çizelge 5.14' de gösterildiği gibi sıfıra doğru yaklaşmıştır.

Tek eşik kullanılarak yapılan sınıflandırmada 8 tane prizma ya atama yapılmış olup, Otsu yaklaşımının çıktısı Şekil 5.31 (a)' da, Kapur metot ile alınan sonuç Şekil 5.31 (b)' de verilmiştir. FCM ise Şekil 5.31 (c)' deki sonucu üretmiştir. Hem Otsu ve hem Kapur yöntemlerinde kız çocuğu arka plandan tamamen ayırt edilmiştir. Ancak FCM algoritmasının pek başarılı olduğu söylenemez.

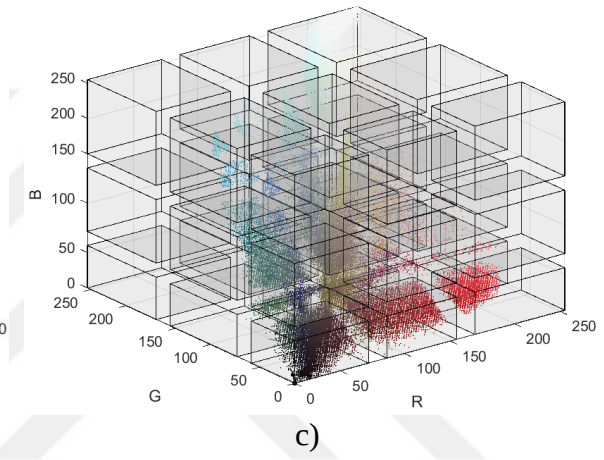
Eşik sayısı $m=2$ olarak alındığında, Otsu metodu ile elde edilen çıktı Şekil 5.32 (a)' da, Kapur ile elde edilen sonuç Şekil 5.32 (b)' de ve FCM algoritmasının çıktısı ise Şekil 5.32 (c)' de gösterilmiştir. Otsu ve Kapur yöntemleri anlamsal bütünlüğü bozmadan daha detaylı bir sınıflandırma yapsa bile bazı bölgelerde kayıplara sebep olmaktadır. FCM yönteminde ise hemen hemen bütün bölgelerde anlam bütünlüğü bulunmamaktadır.



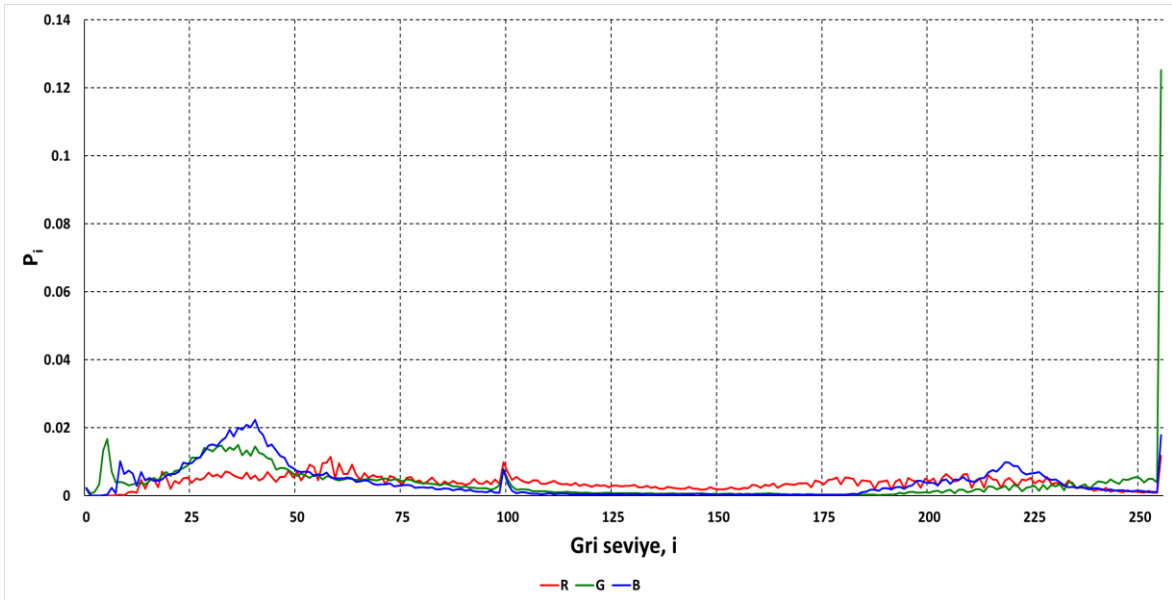
a)



b)



c)



d)

Şekil 5.30. Kız çocuğu (765x512) a) Orijinal b) RGB renk dağılımı c) Alt prizmalar ve kümeler d) RGB Histogram

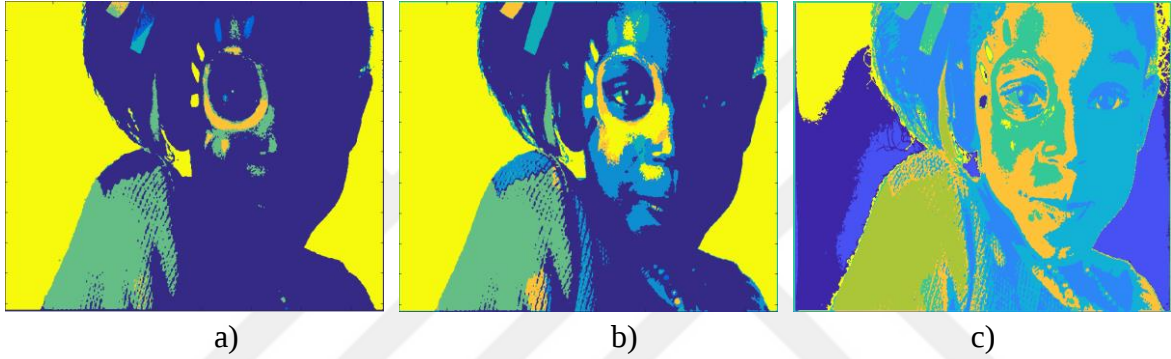
Çizelge 5.13. Kız Çocuğu için bulunan eşikler

Otsu							Kapur						
Doğrusal				OOA		PSO	Doğrusal				OOA		PSO
				t ₁						t ₁			
R	130			130		130	124			124		124	
G	142			142		142	69			69		69	
B	130			130		130	78			78		78	
Süre,t (s)	0,014			0,197		0,204	0,021			0,211		0,257	
Küme:c				8/8						8/8			
	t ₁	t ₂		t ₁	t ₂		t ₁	t ₂		t ₁	t ₂		
R	81	157		81	157		102	174		102	174		
G	62	157		62	157		107	196		107	196		
B	60	150		60	150		102	185		102	185		
Süre,t (s)	6,006			1,193		1,483	6,152			1,209		2,104	
Küme:c				20/27						20/27			
	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	74	135	193	74	135	193	72	130	190	72	130	190	
G	49	93	173	49	93	173	60	117	196	60	117	196	
B	48	87	164	48	87	164	59	105	185	59	105	185	
Süre,t (s)	517,026			1,127		1,391	656,248			1,174		2,004	
Küme:c				40/64						37/64			
	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄	t ₁	t ₂	t ₃	t ₄	
R	-	-	-	-	50	90	143	196	50	90	143	196	
G	-	-	-	-	36	68	116	184	36	68	116	184	
B	-	-	-	-	45	80	149	211	45	80	149	211	
Süre,t (s)					5,98		5,80			5,85		5,92	
`Küme:c`				60/125						64/125			

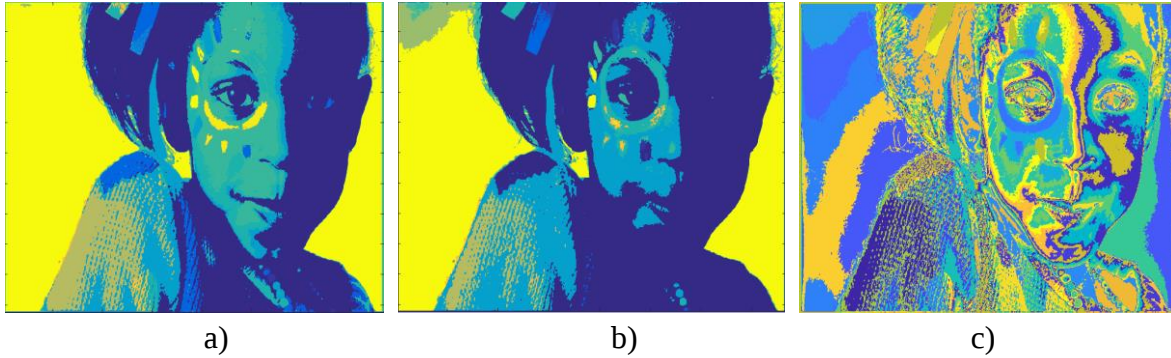
Çizelge 5.14. Kız Çocuğu performans ölçütleri: F, F' ve Q

Eşikleme Metodu	Otsu		Kapur		FCM
Algoritma	PSO	OOA	PSO	OOA	-
Eşik Sayısı/Küme	m = 1 / C = 8		m = 1 / C = 8		C = 8
F	0,004117	0,002745	0,001910	0,001910	0,000342
F'	0,000412	0,000274	0,000191	0,000191	0,000034
Q	0,028652	0,019101	0,011416	0,011416	0,001306
Eşik Sayısı/Küme	m = 2 / C = 20		m = 2 / C = 20		C = 20
F	0,002139	0,001426	0,002468	0,002468	0,000070
F'	0,000214	0,000143	0,000247	0,000247	0,000007
Q	0,011420	0,007613	0,015096	0,015096	0,000157
Eşik Sayısı/Küme	m = 3 / C = 40		m = 3 / C = 37		C = 40
F	0,001471	0,000981	0,001078	0,001078	0,000028
F'	0,000150	0,000100	0,000110	0,000110	0,000003
Q	0,007241	0,004827	0,005424	0,005424	0,000053
Eşik Sayısı/ Küme	m = 4 / C = 60		m = 4 / C = 64		C = 60
F	0,000833	0,000555	0,001013	0,001013	0,000011
F'	0,000085	0,000057	0,000116	0,000116	0,000001
Q	0,003287	0,002191	0,004790	0,004790	0,000015

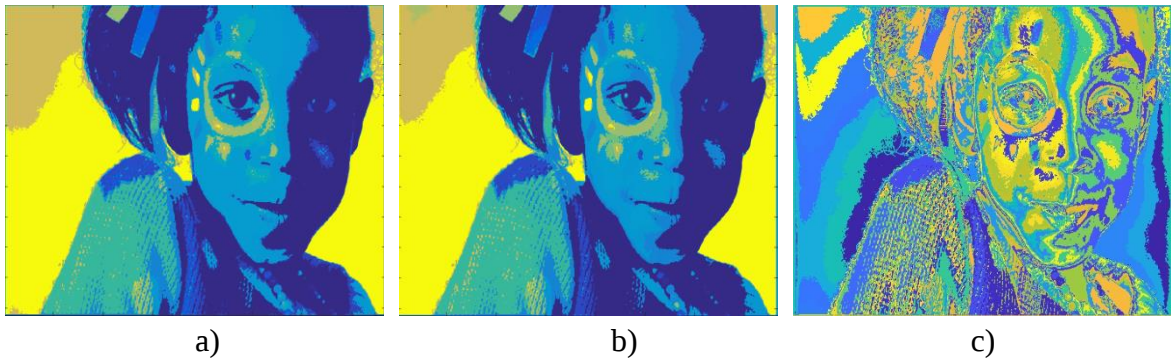
Her bir renk kanalı için 3 tane eşik seçildiğinde, ayrıştırma hassasiyetinin artacağı malumdur. Şekil 5.33 (a)' Otsu ile alınan sonucu gösterir iken, Şekil 5.33 (b)' entropi tekniğinin çıktısını temsil etmektedir. Geleneksel FCM algoritmasının tepkisi ise Şekil 5.33 (c) de gösterilmiştir. Eşikleme yöntemlerinde daha kararlı sonuç alınmıştır. Ancak FCM de oluşan kümelerin ise anlamsal bütünlüğünün olduğunu söylemek zordur. Son olarak eşik sayısı $m=4$ seçildiğinde, Otsu algoritmasının tepkisi Şekil 5.34 (a)' da, Kapur yönteminin çıktısı Şekil 5.34 (b)' de ve FCM yönteminin kümeleme performansı Şekil 5.34 (c)' de sunulmuştur. Sınıf sayısı $c=60$ olduğunda FCM algoritması için geçen süre 362,153 saniyedir.



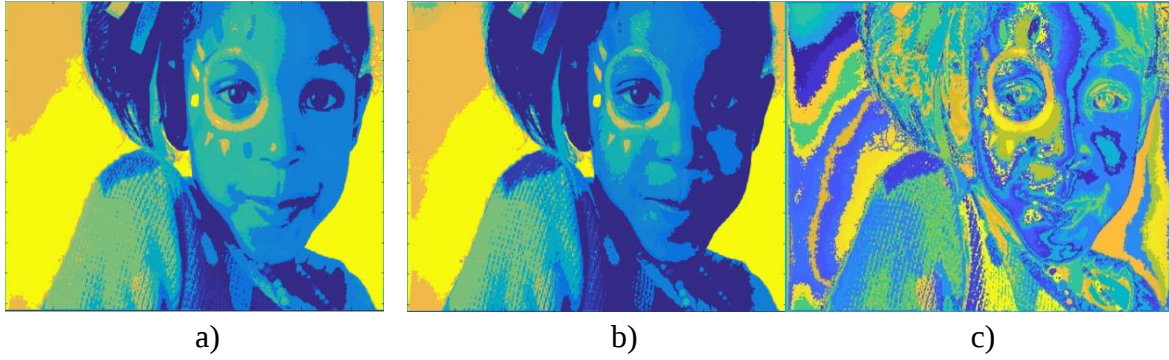
Şekil 5.31. Kız Çocuğu, $m=1$ a) Otsu, $c=8$ b) Kapur, $c=8$ c) FCM, $c=8$, $t=31,696s$



Şekil 5.32. Kız Çocuğu, $m=2$ a) Otsu, $c=20$ b) Kapur, $c=20$ c) FCM, $c=20$, $t=152,067s$



Şekil 5.33. Kız Çocuğu, $m=3$ a) Otsu, $c=40$ b) Kapur, $c=37$ c) FCM, $c=37$, $t=215,78s$



Şekil 5.34. Kız Çocuğu, $m=4$ a) Otsu, $c=60$ b) Kapur, $c=64$ c) FCM, $c=60$, $t=362,153s$

Arka plandaki renk tonları ışıktan dolayı farklı olduğu için homojenlik arttığında, arka plan iki farklı sınıfa ayrılmıştır. Bu durum Otsu ve Kapur yöntemin de net bir şekilde gözükmemektedir. Ancak FCM algoritmasında arka plan için gereksiz sınıflar tahsis edilmiştir. Dolayısıyla gözükmediği gibi hem arka plan ve hem Kız Çocuğu'nun yüz ve elbise kısımları aşırı ayrıştırmaya maruz kalmıştır. Kız Çocuğu görüntüsünde nesne sayısının az olduğu açıktır. Dolayısıyla önerilen yöntemde eşik sayısının veya FCM algoritmasında küme sayısının büyük olması aşırı ayrıştırmaya sebep olmuştur.

Daha önce anlatıldığı gibi önerilen algoritma hızlı bir şekilde renkli görüntülerin ayrıştırılmasını sağlamaktadır. Önerilen yaklaşım ile FCM algoritmasının ayrıştırma süreleri Çizelge 5.15 verilmiştir. Tipik olarak 765x512 boyutlarındaki Kız Çocuğu görüntüsünün ayrıştırma zamanları incelendiğinde, $m=1$ için FCM 31,696 saniye tüketmiştir. Diğer taraftan Otsu fonksiyonu ile PSO algoritması 0,204 saniyelik süre harcamıştır. Görüldüğü gibi önerilen algoritma 155 kat daha hızlıdır. Aynı şartlar için Çizelge 5.14' de verilen F , F' ve Q performans ölçütlerine bakıldığında, FCM algoritmasının F değerinin 12 kat (0,004117/0,000342) daha düşük olduğu görülmüştür. Kapur fonksiyonu ve PSO kombinasyonu ile karşılaştırıldığında, FCM tekniği 5,58 kat (0,001910/0,000342) daha başarılı olmuştur. Diğer taraftan aynı kombinasyon için Çizelge 5.15 den harcanan süreye bakıldığında, önerilen yöntemin 123 kat (31,696 /0,257) daha hızlı olduğu açıktır.

Çizelge 5.15. Test görüntüleri ve bölütleme süreleri (Saniye).

Görüntü	Boyutu	Eşik sayısı	Otsu			Kapur			FCM
Kameraman	300x292	$m=2$	Doğrusal	PSO	OOA	Doğrusal	PSO	OOA	0,891
		$m=3$	0,200	0,312	0,325	0,206	0,317	0,114	0,917
		$m=4$	164,713	0,427	0,472	243,758	0,422	0,123	1,186
		$m=4$	-	0,508	0,513	-	0,515	0,598	
Lena	200x200	$m=1$	0,008	0,244	0,244	0,029	0,333	0,294	8,150
		$m=2$	5,572	1,567	1,141	6,628	2,310	1,022	11,013
		$m=3$	512,833	1,818	1,163	676,992	2,630	1,133	19,605
		$m=4$	-	5,50	5,54	-	5,90	5,96	25,863
Ayakkabı	481x321	$m=1$	0,018	0,236	0,235	0,032	0,309	0,302	19,524
		$m=2$	5,486	1,201	1,125	6,108	1,212	1,051	54,051
		$m=3$	4,923	1,498	1,329	6,710	1,981	1,297	101,154
		$m=4$	-	1,357	1,222	-	1,236	1,689	123,738
Mısır	321x481	$m=1$	0,075	0,241	0,234	0,032	0,266	0,267	19,151
		$m=2$	5,452	1,498	1,435	6,513	2,293	1,104	55,350
		$m=3$	510,549	1,476	1,157	670,834	2,542	1,119	89,561
		$m=4$	-	5,55	5,56	-	5,75	5,69	133,562
Biber	321x481	$m=1$	0,012	0,205	0,198	0,031	0,221	0,213	17,175
		$m=2$	5,498	1,403	1,301	6,628	2,271	1,268	48,723
		$m=3$	509,673	1,772	1,633	674,297	2,141	1,683	89,572
		$m=4$	-	5,73	5,84	-	5,68	5,55	157,186
Papağan	768x512	$m=1$	0,015	0,198	0,195	0,034	0,224	0,199	46,555
		$m=2$	5,602	1,421	1,135	6,234	2,051	1,137	125,738
		$m=3$	507,833	1,742	1,095	670,419	1,482	1,197	247,708
		$m=4$	-	5,12	5,06	-	5,42	5,42	492,019
Kız çocuğu	765x512	$m=1$	0,014	0,204	0,197	0,021	0,257	0,211	31,696
		$m=2$	6,006	1,483	1,193	6,152	2,104	1,209	152,067
		$m=3$	517,026	1,391	1,127	656,248	2,004	1,174	215,78
		$m=4$	-	5,80	5,98	-	5,92	5,85	362,153

Aynı şekilde Çizelge 5.15' e göre, $m=4$ için FCM 362,153 saniyelik zaman harcar iken, Otsu fonksiyonu ile PSO algoritması 5,80 saniye, Kapur fonksiyonu ve OOA algoritması 5,85 saniyelik zaman harcamıştır. Görüldüğü gibi önerilen algoritma 62.5 kat daha hızlıdır. Aynı görüntü için Çizelge 5.14' de verilen F metriği incelendiğinde ise, FCM algoritmasının F değerinin 75 kat (0,000833/0,000011) daha düşük olduğu görülmüştür. Yine Kapur fonksiyonu ve OOA algoritması ikilisi için FCM algoritmasının F değeri 92 kat (0,001013/0,000011) daha düşük olmuştur.

Görüldüğü gibi FCM algoritması görüntüdeki bütün pikselleri tekrarlı olarak defalarca taraması gerektiğinden, büyük boyutlu görüntülerin ayrıştırmasında çok fazla zaman alacaktır. Dolayısıyla FCM algoritmasının gerçek zamanlı uygulamalar için yetersiz kalması muhtemeldir. Diğer taraftan F , F' ve Q ölçütleri açısından FCM algoritması önerilen algoritmadan daha başarılı olmuştur. Ancak FCM tekniğinde her çalıştırmada rastgele küme merkezi atadığından, söz konusu başarının her zaman sağlanacağına garanti bulunmamaktadır.

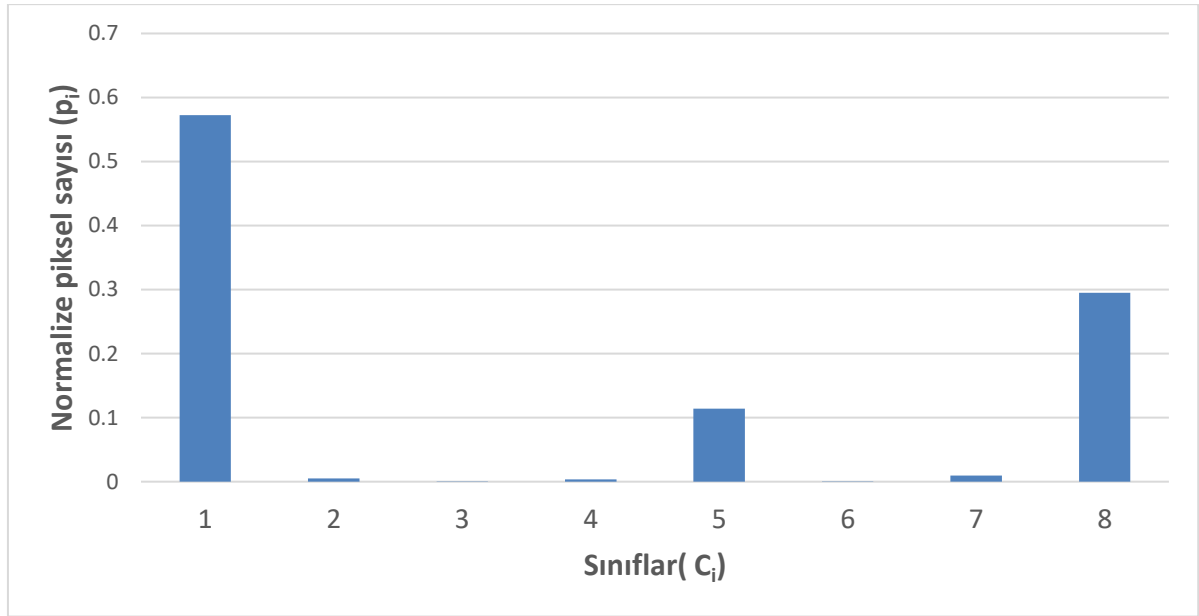
Eşikleme yöntemleri ise 256 elemandan oluşan tek boyutlu bir dizi üzerinde işlem yapar ve görüntü boyutundan tamamen bağımsız çalışır. Dolayısıyla her zaman FCM den hızlı olmuştur.

Önerilen çok seviyeli eşikleme yönteminde her zaman aynı sonuçları almak mümkün iken, her seferinde farklı küme merkezleri ile çalıştığından FCM algoritmasında aynı sonucu almanın garantisi bulunmamaktadır. Ayrıca önerilen yöntemin süresi küme sayısından çok fazla etkilemez iken, FCM tekniğinin sonuç üretme süresi küme sayısına bağlıdır.

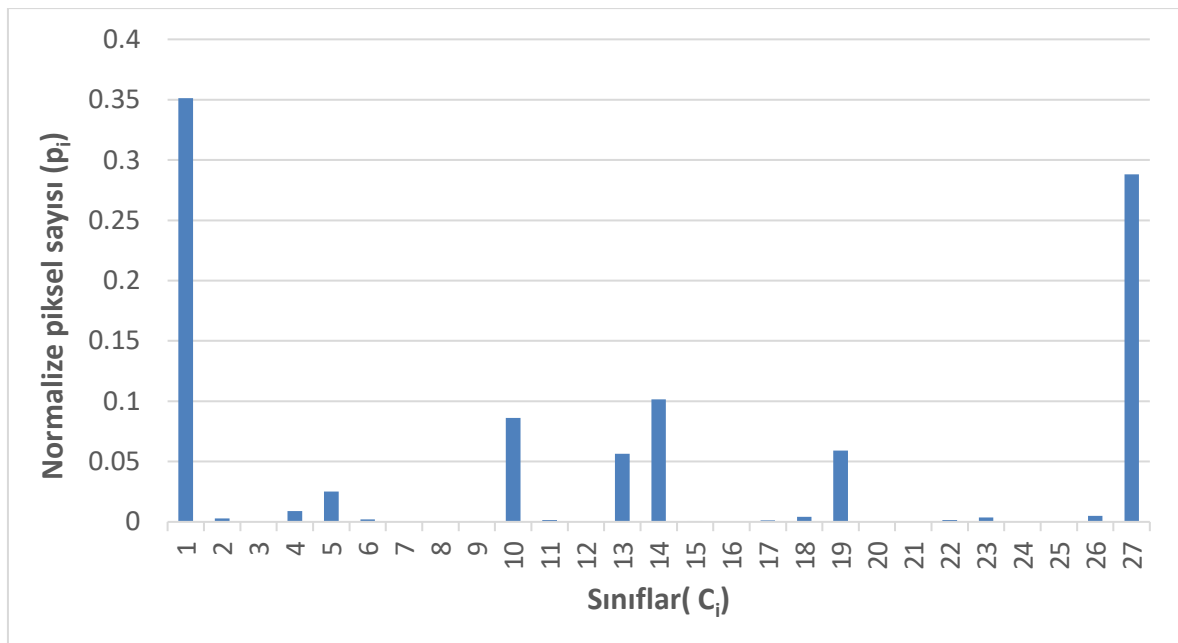
Önerilen algoritmada eşik sayısı arttıkça sınıf sayısı da artmaktadır. Bununla birlikte renk uzayında küme tahsis edildiği halde doldurulamayan ya da boş kalan küme sayısında artmıştır. Bazı kümelere atanan piksel sayısı sıfır olmamakla birlikte, görüntünün tamamına oranla çok küçük olduğundan oluşan bölgelerin gözle fark edilmesi zorlaşmıştır. Renkli görüntülerin üç kanalı ve üç ayrı histogramı olmasına rağmen, Kılıçaslan ve diğerleri renkli görüntülerin tek boyutlu histogramını oluşturmak için yeni bir yaklaşım geliştirmişlerdir [87]. Söz konusu yaklaşımda yatay ekseninde sınıf etiketi, dikey ekseninde ise her sınıfa atanan piksel sayısı gösterilmiştir. Örneğin Şekil 5.35 Kız Çocuğu görüntüsünün Otsu yöntemi ve tek eşik kullanılarak sınıflandırıldığında elde edilen histogramı göstermektedir. Tek eşik ile maksimum 8 küme tahsis edilebilmektedir. Görüldüğü gibi 1, 5 ve 8 nolu kümelere atanan piksel sayıları daha belirgin iken, diğer sınıflara atanan piksel sayıları çok az kalmıştır. İki eşikte ise renk uzayında oluşan küp sayısı maksimum 27 olup, ancak 20 tanesine atama yapılabilmektedir. Şekil 5.36' da görüldüğü gibi diğerleri boş kalmıştır.

Otsu yöntemi ve üç eşik kullanılarak elde edilen histogram ise Şekil 5.37' de gösterilmiştir. Şekil 5.33 de görüldüğü gibi üç eşik ile üretilen dominant bölgeler kolaylıkla fark edilmektedir. Ancak diğer sınıflara atanan piksel sayıları %1 in altında olduğundan gözle

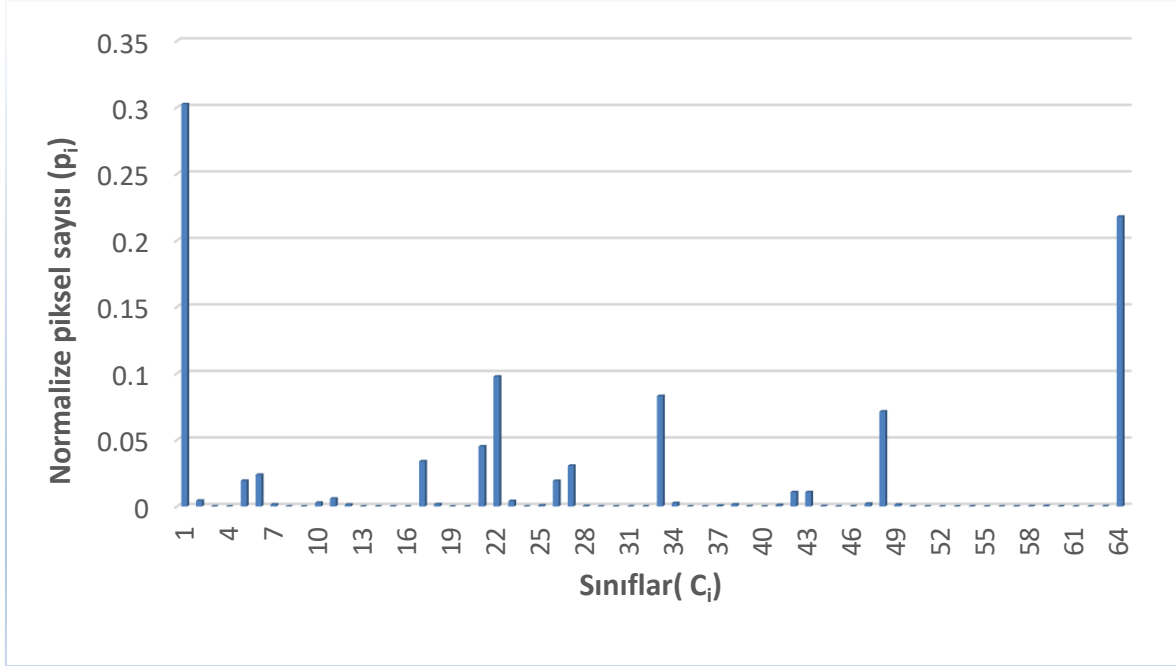
fark edilmeleri zordur. Ayrıca sınıflandırma sürecinde aynı gruba atanan piksellerin görüntü üzerinde yakın konumlarda olma zorunluluğu bulunmadığından, aynı gruba atanan pikseller görüntü üzerinde dağılmış durumdadır. Şekil 5.38 Kız Çocuğu görüntüsünde 4 eşik kullanıldığında oluşan sınıfların histogramını göstermektedir. Maksimum sınıf sayısı 125 iken, sadece 60 sınıfa piksel atanmıştır. Yaklaşık 5 tane kümeye atanan piksel sayısı belirgin iken, diğerlerine atanan piksel sayısı oldukça küçük kalmıştır. Bir bakıma aşırı ayrıştırma olmuştur.



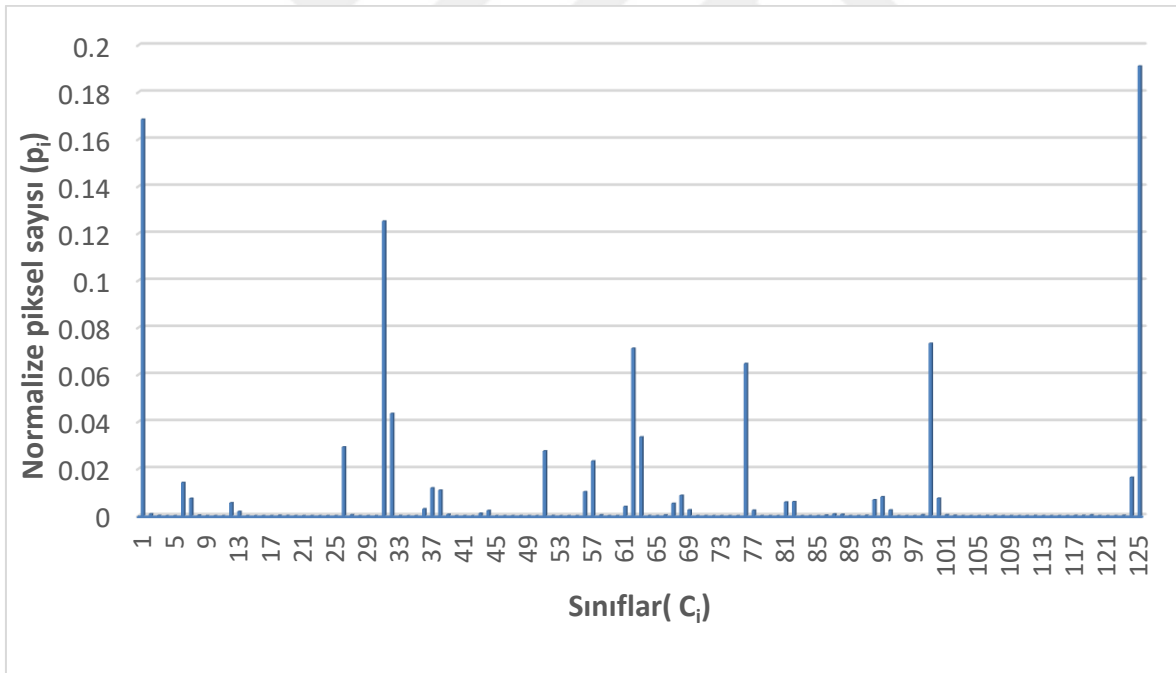
Şekil 5.35. Kız Çocuğu, $m=1$, Otsu sınıfların histogramı



Şekil 5.36. Kız Çocuğu, $m=2$, Otsu sınıfların histogramı



Şekil 5.37 Kız Çocuğu, $m=3$, Otsu sınıfların histogramı



Şekil 5.38 Kız Çocuğu, $m=4$, Otsu sınıfların histogramı



6. SONUÇ

Görüntü ayrıştırma, görüntünün homojen bölgelere ayrıştırılmasını veya başka bir ifade ile aynı bölgede olan piksellerin çok benzer veya aynı olmasını gerektirir. Ayrıca farklı bölgelerin piksellerinin ise diğer bölgelerin piksellerinden farklı olması önemlidir. Bir görüntünün onu oluşturan alt bölgeler ya da nesneler şeklinde tasnif edilmesinin ardından, nesne tanıma süreçleri otomatik tanıma yapabilmelidir. Bu nedenle sayısal görüntü işleme alanındaki en temel ve en zor konulardan birisi görüntü ayrıştırma işlemidir.

Bu çalışmada, renkli görüntülerin ayrıştırılması için yeni bir yöntem geliştirilmiştir. Önerilen algoritma renkli görüntülerin ayrıştırmasını hızlı ve otomatik olarak yapabilmektedir. Görüntüdeki piksel bilgileri yerine histogram bilgileri kullanıldığından, önerilen yaklaşım görüntü boyutlarından bağımsızdır. Sayısal görüntüdeki bilgiler genelde 256 elemanlı tek boyutlu dizi ile temsil edilmektedirler. Her bir ayrıştırma işlemi için sadece histogram bilgisi kullanıldığından, geliştirilen algoritma farklı boyutlardaki görüntülere rahatlıkla uygulanabilir. Dolayısıyla geleneksel kullanıcı bağımlı FCM algoritmasından daha hızlı bir algoritma tasarlanmıştır. Ayrıca söz konusu yöntemin, kullanıcı tarafından belirlenen küme sayısı veya iterasyon sayısı gibi parametrelere ihtiyaç duymaması en önemli avantajıdır.

Sadece gri ölçekli görüntülerin çok seviyeli eşiklenmesinde kullanılan Otsu ve Kapur metotları renkli görüntüler içinde kullanılabilir hale getirilmiştir. Bu kapsamda öncelikle her bir renk kanalı için m tane eşik hesaplanmaktadır. Ardından elde edilen eşik değerleri ile üç boyutlu RGB renk uzayının bölümlenmesi yapılmıştır. Her kanaldan elde edilen eşik sayısı ile $(m+1)^3$ adet küme, başka bir ifade ile renk küpü oluşturulmuştur. Renk uzayında oluşturulan her bir renk küpü içerisinde kalan pikseller aynı kümeye atanmıştır. Önerilen algoritmada elde edilecek maksimum küme sayısı ile eşik sayısı arasında bir ilişki mevcuttur. Bununla birlikte oluşan bütün kümeler piksel atanması garanti değildir. Tanımlanan kümeler hiçbir pikselin atanmama ihtimali de mevcuttur. Çünkü oluşan kümelerin doldurulabilmesi renkli görüntünün RGB renk uzayındaki renk dağılımına bağlıdır. Diğer taraftan herhangi bir küpün boş kalması veya boş küme oluşması görüntü ayrıştırma süreci açısından problem teşkil etmemektedir.

Oluşacak kümelerin veya bölgelerin homojenliği eşik sayısı ile doğru orantılı iken, renk uzayında oluşturulan alt küplerin hacimleri ile ters orantılıdır. Söz konusu renk küplerinin hacimleri düştükçe homojenlik artmaktadır. Ancak eşik sayısının çok fazla olması aşırı

ayırıştırma durumunun ortaya çıkmasına neden olabilmektedir. Görüntü üzerinde çok fazla çeşit nesne var ise eşik sayısının fazlalığı avantaj iken, nesne sayısı çok az olduğunda gereksiz kümelerin oluşmasına neden olabilmektedir. Ortaya çıkması muhtemel bu durum eşik sayısı azaltılarak ikinci bir ayırıştırma işlemi ile giderilebilir. Ancak bu çalışmada iki aşamalı eşiklemeye dair sonuçlar alınmamıştır.

Önerilen yöntemde her bir kanal için aynı sayıda eşik kullanılmıştır. Ancak renk kanallarına ait histogramların dar bir aralığa yığıldığı durumlarda fazla eşik kullanmak gereksiz olabilir. Dolayısıyla renkli görüntülerin histogram bilgileri ile her bir kanal için otomatik eşik sayısını belirleyen algoritmaların geliştirilmesi veya farklı kanallar için farklı sayıda eşikler kullanılarak sınıflandırma yapılması muhtemel çalışma alanlarıdır. Ayrıca çok küçük sayıda piksellerden oluşan sınıfların renk benzerliği en yakın olan sınıflara dahil edilerek bölge birleştirmesi muhtemel çalışma alanıdır.

KAYNAKLAR

1. Gonzalez, R. C., and Woods, R. E. (2008). *Digital image processing*: Pearson/Prentice Hall.
2. Niblack, W. (1986). *An introduction to digital image processing*: Prentice-Hall Englewood Cliffs.
3. Baxes, G. A. (1994). *Digital image processing: Principles and applications*: Wiley New York.
4. Fu, K.-S., and Mui, J. (1981). A survey on image segmentation. *Pattern recognition*, 13(1), 3-16.
5. Jensen, J. R., and Lulla, K. (1987). Introductory digital image processing: A remote sensing perspective. *Geocarto International*, 65-65.
6. Lehmann, T. M., Gonner, C., and Spitzer, K. (1999). Survey: Interpolation methods in medical image processing. *IEEE transactions on medical imaging*, 18(11), 1049-1075.
7. Narkhede, H. (2013). Review of image segmentation techniques. *International Journal of Science and Modern Engineering*, 1(8), 54-61.
8. İncetaş, M. O. (2014). *Tıbbi görüntülerin uyarlanabilir bölge genişletme algoritması*, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
9. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 62-66.
10. Kapur, J., Sahoo, P. K., and Wong, A. K. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3), 273-285.
11. Portes de Albuquerque, M., Esquef, I. A., Gesualdi Mello, A. R., and Portes de Albuquerque, M. (2004). Image thresholding using tsallis entropy. *Pattern Recognition Letters*, 25(9), 1059-1065.
12. Muppidi, M., Rad, P., Agaian, S. S., and Jamshidi, M. (2015). *Image segmentation by multi-level thresholding based on fuzzy entropy and genetic algorithm in cloud*. Paper presented at the System of Systems Engineering Conference (SoSE), 2015 10th.
13. Sarkar, S., Das, S., and Chaudhuri, S. S. (2015). A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution. *Pattern Recognition Letters*, 54, 27-35.
14. Yin, P.-Y. (2007). Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Applied mathematics and computation*, 184(2), 503-513.
15. Sezgin, M. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1), 146-168.
16. Wolpert, D. H., and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67-82.

17. Lai, C.-C., and Tseng, D.-C. (2004). A hybrid approach using gaussian smoothing and genetic algorithm for multilevel thresholding. *International Journal of Hybrid Intelligent Systems*, 1(3, 4), 143-152.
18. Sathya, P., and Kayalvizhi, R. (2011). Modified bacterial foraging algorithm based multilevel thresholding for image segmentation. *Engineering Applications of Artificial Intelligence*, 24(4), 595-615.
19. Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 13(6), 3066-3091.
20. 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.
21. 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.
22. Horng, M.-H., and Jiang, T.-W. (2010). *Multilevel image thresholding selection based on the firefly algorithm*. Paper presented at the Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2010 7th International Conference on.
23. Gao, H., Xu, W., Sun, J., and Tang, Y. (2010). Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *IEEE Transactions on Instrumentation and Measurement*, 59(4), 934-946.
24. Sathya, P., and Kayalvizhi, R. (2011). Amended bacterial foraging algorithm for multilevel thresholding of magnetic resonance brain images. *Measurement*, 44(10), 1828-1848.
25. Sathya, P. D., and Kayalvizhi, R. (2011). Modified bacterial foraging algorithm based multilevel thresholding for image segmentation. *Engineering Applications of Artificial Intelligence*, 24(4), 595-615.
26. Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., and Osuna, V. (2014). A multilevel thresholding algorithm using electromagnetism optimization. *Neurocomputing*, 139, 357-381.
27. Bhandari, A. K., Kumar, A., and Singh, G. K. (2015). Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using kapur's, otsu and tsallis functions. *Expert Systems with Applications*, 42(3), 1573-1601. doi:<https://doi.org/10.1016/j.eswa.2014.09.049>
28. Roy, S., Kumar, U., Chakraborty, D., Nag, S., Mallick, A., and Dutta, S. (2015). Comparative analysis of cuckoo search optimization-based multilevel image thresholding *Intelligent computing, communication and devices* (327-342): Springer.
29. Shi, Y., Gao, H., and Wu, D. (2015). *Multi-level image segmentation based on an improved differential evolution with adaptive parameter controlling strategy*. Paper presented at the Control and Decision Conference (CCDC), 2015 27th Chinese.
30. Li, J.-y., Zhao, Y.-d., Li, J.-h., and Liu, X.-j. (2015). Artificial bee colony optimizer with bee-to-bee communication and multipopulation coevolution for multilevel threshold image segmentation. *Mathematical Problems in Engineering*, 2015.

31. Pal, S. S., Kumar, S., Kashyap, M., Choudhary, Y., and Bhattacharya, M. (2016). *Multi-level thresholding segmentation approach based on spider monkey optimization algorithm*, New Delhi.
32. Kurugollu, F., Sankur, B., and Harmanci, A. E. (2001). Color image segmentation using histogram multithresholding and fusion. *Image and vision computing*, 19(13), 915-928.
33. Navon, E., Miller, O., and Averbuch, A. (2005). Color image segmentation based on adaptive local thresholds. *Image and vision computing*, 23(1), 69-85.
34. Kurban, T., Civicioglu, P., Kurban, R., and Besdok, E. (2014). Comparison of evolutionary and swarm based computational techniques for multilevel color image thresholding. *Applied Soft Computing*, 23, 128-143.
35. Bhandari, A., Kumar, A., and Singh, G. (2015). Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms. *Expert Systems with Applications*, 42(22), 8707-8730.
36. Coleman, G. B., and Andrews, H. C. (1979). Image segmentation by clustering. *Proceedings of the IEEE*, 67(5), 773-785.
37. Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3), 191-203.
38. Trivedi, M. M., and Bezdek, J. C. (1986). Low-level segmentation of aerial images with fuzzy clustering. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(4), 589-598.
39. Yambal, M., and Gupta, H. (2013). Image segmentation using fuzzy c means clustering: A survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(7).
40. Tan, K. S., Lim, W. H., and Isa, N. A. M. (2013). Novel initialization scheme for fuzzy c-means algorithm on color image segmentation. *Applied Soft Computing*, 13(4), 1832-1852.
41. Patel, B. C., and Sinha, G. (2010). An adaptive k-means clustering algorithm for breast image segmentation. *International Journal of Computer Applications*, 10(4), 35-38.
42. Yao, H., Duan, Q., Li, D., and Wang, J. (2013). An improved k-means clustering algorithm for fish image segmentation. *Mathematical and Computer Modelling*, 58(3), 790-798.
43. Moftah, H. M., Azar, A. T., Al-Shammari, E. T., Ghali, N. I., Hassanien, A. E., and Shoman, M. (2014). Adaptive k-means clustering algorithm for mr breast image segmentation. *Neural Computing and Applications*, 24(7), 1917-1928.
44. Sikka, K., Sinha, N., Singh, P. K., and Mishra, A. K. (2009). A fully automated algorithm under modified fcm framework for improved brain mr image segmentation. *Magnetic Resonance Imaging*, 27(7), 994-1004.
45. Khan, A., Ullah, J., Jaffar, M. A., and Choi, T.-S. (2014). Color image segmentation: A novel spatial fuzzy genetic algorithm. *Signal, Image and Video Processing*, 8(7), 1233-1243.
46. Wang, X.-Y., and Bu, J. (2010). A fast and robust image segmentation using fcm with spatial information. *Digital Signal Processing*, 20(4), 1173-1182.

47. Ozyurt, O., Dincer, A., x, ztu, x, and rk, C. (2009, 20-22 May 2009). *Brain mr image segmentation with fuzzy c-means and using additional shape elements*. Paper presented at the Biomedical Engineering Meeting, 2009. BIYOMUT 2009. 14th National.
48. Gong, M., Liang, Y., Shi, J., Ma, W., and Ma, J. (2013). Fuzzy c-means clustering with local information and kernel metric for image segmentation. *IEEE Transactions on Image Processing*, 22(2), 573-584. doi:10.1109/TIP.2012.2219547
49. Zhao, F., Jiao, L., and Liu, H. (2013). Kernel generalized fuzzy c-means clustering with spatial information for image segmentation. *Digital Signal Processing*, 23(1), 184-199.
50. Ma, J., Tian, D., Gong, M., and Jiao, L. (2014). Fuzzy clustering with non-local information for image segmentation. *International Journal of Machine Learning and Cybernetics*, 5(6), 845-859.
51. Adams, R., and Bischof, L. (1994). Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6), 641-647.
52. Fan, J., Yau, D. K., Elmagarmid, A. K., and Aref, W. G. (2001). Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, 10(10), 1454-1466.
53. Moghaddamzadeh, A., and Bourbakis, N. (1997). A fuzzy region growing approach for segmentation of color images. *Pattern recognition*, 30(6), 867-881.
54. Shih, F. Y., and Cheng, S. (2005). Automatic seeded region growing for color image segmentation. *Image and vision computing*, 23(10), 877-886.
55. Tang, J. (2010). *A color image segmentation algorithm based on region growing*. Paper presented at the Computer engineering and technology (iccet), 2010 2nd international conference on.
56. Trémeau, A., and Colantoni, P. (2000). Regions adjacency graph applied to color image segmentation. *IEEE Transactions on Image Processing*, 9(4), 735-744.
57. Felzenszwalb, P. F., and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2), 167-181.
58. Khadem, M. S. (2010). Mri brain image segmentation using graph cuts.
59. Camilus, K. S., and Govindan, V. (2012). A review on graph based segmentation. *International Journal of Image, Graphics & Signal Processing*, 4(5), 1-13.
60. 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.
61. Fisher, J. L., Hinds, S. C., and D'Amato, D. P. (1990). *A rule-based system for document image segmentation*. Paper presented at the Pattern Recognition, 1990. Proceedings., 10th International Conference on.
62. Tolia, Y. A., and Panas, S. M. (1998). On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system. *IEEE Signal Processing Letters*, 5(10), 245-247.
63. Karmakar, G. C., and Dooley, L. S. (2002). A generic fuzzy rule based image segmentation algorithm. *Pattern Recognition Letters*, 23(10), 1215-1227.

64. Demirci, R. (2006). Rule-based automatic segmentation of color images. *AEU-International Journal of Electronics and Communications*, 60(6), 435-442.
65. Demirci, R., Güvenç, U., and Kahraman, H. T. (2014). Görüntülerin renk uzayı yardımıyla ayrıştırılması. *İleri Teknoloji Bilimleri Dergisi*, 3(1), 1-8.
66. Ghaemi, M., and Feizi-Derakhshi, M.-R. (2014). Forest optimization algorithm. *Expert Systems with Applications*.
67. Ahirwal, B., Khadtare, M., and Mehta, R. (2007, 25-28 Nov. 2007). *Fpga based system for color space transformation rgb to yiq and ycbcr*. Paper presented at the 2007 International Conference on Intelligent and Advanced Systems.
68. Tabakov, S. D. (2013). Introduction to vision, colour models and image compression *Medical physics international*, 1(1), 50.
69. Burger, W., and Burge, M. J. (2009). *Principles of digital image processing*: Springer.
70. Cheng, H.-D., and Sun, Y. (2000). A hierarchical approach to color image segmentation using homogeneity. *Image Processing, IEEE Transactions on*, 9(12), 2071-2082.
71. Pratt, W. Digital image processing. 1991. *Publisher John Wiley & Sons*.
72. Shapiro, L., and Stockman, G. C. (2001). Computer vision. 2001. *ed: Prentice Hall*.
73. Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.
74. Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM computing surveys (CSUR)*, 31(3), 264-323.
75. Eberhart, R., and Kennedy, J. (1995). *A new optimizer using particle swarm theory*. Paper presented at the Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on.
76. Atashpaz-Gargari, E., and Lucas, C. (2007). *Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition*. Paper presented at the Evolutionary computation, 2007. CEC 2007. IEEE Congress on.
77. Karaboga, D., and Basturk, B. (2007). Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems *Foundations of fuzzy logic and soft computing* (pp. 789-798): Springer.
78. Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Retrieved from
79. Yang, X.-S. (2005). Engineering optimizations via nature-inspired virtual bee algorithms *Artificial intelligence and knowledge engineering applications: A bioinspired approach* (pp. 317-323): Springer.
80. Storn, R. (1996). *On the usage of differential evolution for function optimization*. Paper presented at the Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American.
81. Karaboğa, D. (2006). *Yapay zeka optimizasyon algoritmaları*: Nobel Akademik Yayıncılık.

82. Rahkar Farshi, T., Demirci, R., and Feizi-Derakhshi, M.-R. (2018). Image clustering with optimization algorithms and color space. *Entropy*, 20(4), 296.
83. Kahraman, A. S., Rahkar Farshi, T., and Demirci, R. Renkli görüntülerin çok seviyeli eşiklenmesi ve sınıflandırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 6(4), 846-859.
84. Monteiro, F. C., and Campilho, A. C. (2006). *Performance evaluation of image segmentation*, Berlin, Heidelberg.
85. Liu, J., and Yang, Y.-H. (1994). Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7), 689-700.
86. Borsotti, M., Campadelli, P., and Schettini, R. (1998). Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19(8), 741-747.
87. Kılıçaslan, M., Tanyeri, U., ve Demirci, R. (2018). Renkli görüntüler için tek boyutlu histogram. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*. doi:10.29130/dubited.413822

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : RAHKAR FARSHİ, Taymaz
 Uyruğu : İran
 Doğum Tarihi : 28.12.1985
 Doğum Yeri : İran
 E-Mail : taymaz.farshi@gmail.com



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi Bilgisayar Mühendisliği	Devam Ediyor
Yüksek Lisans	Karadeniz Teknik Üniversitesi	2013
Lisans	University College of Nabi Akram	2009

İş Deneyimi

Yıl	Yer	Görev
2017-Halen	Altınbaş Üniversitesi	Öğretim Görevlisi
2015-2017	İstanbul Aydın Üniversitesi	Öğretim Görevlisi
2014-2015	InfoDiff	Yazılım Geliştirici

Yabancı Dil

İngilizce, Farsça, Azerice

Yayınlar

1. Rahkar Farshi, T., Demirci, R., and Feizi-Derakhshi, M.-R. (2018). Image clustering with optimization algorithms and color space. *Entropy*, 20(4), 296.
2. 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

Dizi, Müzik, Kitap



GAZİ GELECEKTİR..