



**T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**İKİ BOYUTLU MEDİKAL GÖRÜNTÜLERDE HİSTOGRAM
TABANLI KÜME KESTİRİM METODU İLE YARI OTOMATİK
SEGMENTASYON YÖNTEMLERİNİN GELİŞTİRİLMESİ**

Enver KÜÇÜKKÜLAHLI

**DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**DANIŞMAN
Doç. Dr. Pakize ERDOĞMUŞ**

DÜZCE, 2018

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**İKİ BOYUTLU MEDİKAL GÖRÜNTÜLERDE HİSTOGRAM
TABANLI KÜME KESTİRİM METODU İLE YARI OTOMATİK
SEGMENTASYON YÖNTEMLERİNİN GELİŞTİRİLMESİ**

Enver KÜÇÜKKÜLAHLI tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından
Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik ve Bilgisayar
Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Pakize ERDOĞMUŞ

Düzce Üniversitesi

Eş Danışman

Doç. Dr. Kemal POLAT

Abant İzzet Baysal Üniversitesi

Jüri Üyeleri

Doç. Dr. Pakize ERDOĞMUŞ

Düzce Üniversitesi

Prof. Dr. Resul KARA

Düzce Üniversitesi

Prof. Dr. Ömer ÖNBAŞ

Düzce Üniversitesi

Doç. Dr. Mehmet Recep BOZKURT

Sakarya Üniversitesi

Dr.Öğr.Üyesi Ahmet Reşit KAVSAOĞLU

Karabük Üniversitesi

Tez Savunma Tarihi: 27/06/2018

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

27 Haziran 2018

Enver KÜÇÜKKÜLAHLI

TEŞEKKÜR

Doktora öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç. Dr. Pakize ERDOĞMUŞ'a en içten dileklerimle teşekkür ederim.

Tez çalışmam boyunca değerli katkılarını esirgemeyen eş danışmanım Doç. Dr. Kemal POLAT'a da şükranlarımı sunarım.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili eşim Sibel KÜÇÜKKÜLAHLI, oğlum Mehmet Uras KÜÇÜKKÜLAHLI'ya ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

27 Haziran 2018

Enver KÜÇÜKKÜLAHLI

İÇİNDEKİLER

	<u>Sayfa No</u>
ŞEKİL LİSTESİ	VII
ÇİZELGE LİSTESİ	IX
KISALTMALAR.....	XI
SİMGELER	XII
ÖZET	XIII
ABSTRACT	XV
EXTENDED ABSTRACT	XVI
1. GİRİŞ.....	1
1.1. BÖLÜTLEME.....	1
1.1.1. Eşikleme Tabanlı Bölütleme	2
1.1.2. Kümeleme Tabanlı Bölütleme	5
1.1.3. Kenar Tabanlı Bölütleme	7
1.1.4. Bölge Tabanlı Bölütleme	7
1.1.5. Havza Tabanlı Bölütleme	8
1.1.6. Kısmi Diferansiyel Denklem (KDD) Tabanlı Bölütleme	8
1.1.7. Yapay Sinir Ağı (YSA) Tabanlı Bölütleme.....	9
1.2. HİSTOGRAM	9
1.3. LİTERATÜR TARAMASI.....	12
1.4. TEZ ORGANİZASYONU	20
2. MATERYAL VE YÖNTEM.....	21
2.1. MATERYAL	21
2.1.1. Multimodal Beyin Tümörü Görüntü Bölütleme Karşılaştırması (BRATS - Multimodal Brain Tumor Image Segmentation Benchmark) 2012 Veritabanı	21
2.1.2. Uluslararası Cilt Görüntüleme İşbirliği (ISIC- The International Skin Imaging Collaboration) Veritabanı.....	22

2.1.2.1. Lezyon Segmentasyonu	22
2.1.2.2. Dermoskopik Özellik Sınıflandırma	23
2.1.2.3. Dermoskopik Özellik Bölütleme	23
2.1.2.4. Hastalık Sınıflandırma	23
2.1.2.5. Maskeli Hastalık Sınıflandırma	24
2.2. YÖNTEM	24
2.2.1. Filtreleme	25
2.2.1.1. Ayrık Dalgacık Dönüşümü (DWT – Discrete Wavelet Transform)	25
2.2.1.2. Medyan Filtre	27
2.2.2. Histogram Tabanlı Küme Kestirimi (HBCE – Histogram Based Cluster Estimation)	28
2.2.3. Bölütleme	32
2.2.3.1. K-ortalama Algoritması	33
2.2.3.2. Biyocoğrafya Tabanlı Optimizasyon Algoritması (BBO)	35
2.2.3.3. Parçacık Sürü Optimizasyonu (PSO)	40
2.2.3.4. Darwinci Parçacık Sürü Optimizasyonu (DPSO)	45
2.2.3.5. Kesirli Düzen Darwinci Parçacık Sürü Optimizasyonu (FODPSO - Fractional Order Darwinian Partical Swarm Optimisation)	47
2.2.3.6. Merkez Tabanlı mPSO, mDPSO ve mFODPSO	48
2.2.3.7. Bölütleme İşlemi için Başlangıç Değerlerinin Atanması	49
2.2.4. Son İşleme	50
2.2.5. Kullanıcı Arayüzü	50
3. BULGULAR VE TARTIŞMA	56
3.1. PERFORMANS ÖLÇÜTLERİ	62
3.1.1. Rand Index (RI)	62
3.1.2. Global Consistency Error (GCE)	63
3.1.3. Dice Score, Jaccard indeksi, Doğruluk, Hassasiyet, Özgünlük	63
3.2. ÇALIŞMA ORTAMI	65
3.3. BULGULAR	66
4. SONUÇLAR VE ÖNERİLER	98
5. KAYNAKLAR	101
ÖZGEÇMİŞ	111

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1. Görüntü bölütleme teknikleri.....	2
Şekil 1.2. Statik eşik değerlendirme	3
Şekil 1.3. Histogram örnekleri, (a) bi-modal histogram, (b) multi-modal histogram	5
Şekil 1.4. K-ortalama kümeleme algoritmasının çalışma adımları	6
Şekil 1.5. Örnek bir görüntü (a) ve görüntünün histogramı (b)	9
Şekil 1.6. Histogram bilgi çıkartımı örneği	11
Şekil 2.1. Glioma hastalığına ait MR görüntüsü ve insan bölütleme sonucu	22
Şekil 2.2. Dermoskopik görüntü ve insan bölütleme sonucu.	23
Şekil 2.3. İşlem adımları için akış şeması.....	25
Şekil 2.4. Ayrık Dalgacık Dönüşümü alt bant ayrışımı.....	26
Şekil 2.5. Medyan filtrenin tek boyutlu test sinyallerinde çalıştırılması	28
Şekil 2.6. Kameraman görüntüsü ve bu görüntüye ait histogram.....	29
Şekil 2.7. Ortalama değişim miktarının üzerindeki tepe noktaları.	30
Şekil 2.8. Algoritma sonunda belirlenen küme başlangıç değerleri.	30
Şekil 2.9. Bölütlenmiş kameraman görüntüsü.	31
Şekil 2.10. Histogram Tabanlı Küme Kestirimi (HBCE) sözde kodu.....	32
Şekil 2.11. K-ortalama kümeleme algoritması akış şeması	34
Şekil 2.12. K-ortalama kümeleme işlemi örneği	35
Şekil 2.13. Basit BBO’da dışa ve içe göç eğrileri	36
Şekil 2.14. BBO algoritmasına ait sözde kod.	39
Şekil 2.15. Parçacık Sürü Optimizasyonu algoritması için akış şeması.	41
Şekil 2.16. Darwinci Parçacık Sürü Optimizasyonu algoritması için sözde kod.	46
Şekil 2.17. MATLAB™ GUI ortamında tasarlanmış kullanıcı arayüzü ekranı.	50
Şekil 2.18. Görüntü ve insan bölütleme sonucu bilgisi girildikten sonra ekran görüntüsü.....	51
Şekil 2.19. Histogramın filtrelenmesiyle elde edilen bilgilere ait ekran görüntüsü.	52
Şekil 2.20. HBCE adımından sonra elde edilen sonuçların görüntülenmesi.	53
Şekil 2.21. Bölütleme işleminden sonra elde edilen sonuçların görüntülenmesi.	54
Şekil 2.22. Düzenleme ve karşılaştırma işlemlerinden sonra elde edilen sonuçların görüntülenmesi.	55
Şekil 3.1. BRATS 2012 veritabanından alınan örnek görüntü ve bu görüntüye ait insan bölütleme sonucu	56
Şekil 3.2. Örnek MR görüntüsüne ait histogram bilgisi	57
Şekil 3.3. Ayıklanmış histogram görüntüsü.....	57
Şekil 3.4. DWT ile filtrelenmiş histogram görüntüsü.....	58
Şekil 3.5. HBCE sonucunda elde edilen tahmini küme merkezleri ve filtrelenmiş histogram görüntüsü.....	59
Şekil 3.6. Bölütleme işlemi sonrası elde edilen MR görüntüsü.....	60
Şekil 3.7. Bölütleme sonucunda elde edilen küme merkezleri ve filtrelenmiş histogram görüntüsü.....	60
Şekil 3.8. Bölütleme sonucunda elde edilen küme merkezleri, filtrelenmiş histogram görüntüsü ve HBCE sonuçlarının birlikte gösterimi.	61

Şekil 3.9. Bölütleme işlemi sonrası elde edilen MR görüntüsü.....	61
Şekil 3.10. Son işlem sonucunda elde edilen küme merkezleri, filtrelenmiş histogram görüntüsü ve bölütleme sonuçlarının birlikte gösterimi.....	62
Şekil 3.11. ISIC veritabanından alınan 5 örnek görüntü (a1-a5) ve bu görüntülere ait insan bölütleme sonuçları (b1-b5).....	67
Şekil 3.12. ISIC veritabanından alınan 5 örnek veri için siyah-beyaz görüntü (a1-a5), bu görüntülere ait histogram bilgisi(b1-b5), medyan filtre ile filtrelenmiş histogram (c1-c5), DWT ile filtrelenmiş histogram (d1-d5).	69
Şekil 3.13. HBCE algoritmasının yatay-dikey (a1-a5) ve dikey-yatay (b1-b5) çalıştırılmasıyla elde edilen sonuçlar.	73
Şekil 3.14. Örnek bir görüntü için (a) orijinal görüntü, (b) insan bölütleme sonucu.	75
Şekil 3.15. Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.....	76



ÇİZELGE LİSTESİ

Sayfa No

Çizelge 3.1. PSO, DPSO, FODPSO, mPSO, mDPSO ve mFODPSO algoritmaları için parametrelerin başlangıç değerleri	66
Çizelge 3.2. Küme sayısı bilgisi verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.	84
Çizelge 3.3. Tahmini küme merkezi bilgisi verilmeden (a) ve verilerek (b) çalıştırılan K-ortalama kümeleme algoritmasına ait ortalama performans sonuçları.	85
Çizelge 3.4. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan BBO algoritmasına ait ortalama performans sonuçları.	86
Çizelge 3.5. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan PSO algoritmasına ait ortalama performans sonuçları.	87
Çizelge 3.6. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan DPSO algoritmasına ait ortalama performans sonuçları.	88
Çizelge 3.7. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan FODPSO algoritmasına ait ortalama performans sonuçları.	89
Çizelge 3.8. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan mPSO algoritmasına ait ortalama performans sonuçları.	90
Çizelge 3.9. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan mDPSO algoritmasına ait ortalama performans sonuçları.	91
Çizelge 3.10. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan mFODPSO algoritmasına ait ortalama performans sonuçları.	92
Çizelge 3.11. Tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.....	93
Çizelge 3.12. Tahmini küme merkezi bilgisi tüm parçacıklar için verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.....	94
Çizelge 3.13. Tahmini küme merkezi bilgisi belirlenen ortalama aralıkta rastgele verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.	95

Çizelge 3.14. Tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan mDPSO bölütleme algoritmasının her bir test görüntüsüne ait ortalama performans sonuçları.	96
--	----



KISALTMALAR

BBO	Biyocoğrafya Tabanlı Optimizasyon (Biogeography Based Optimisation)
BRATS	Multimodal Beyin Tümörü Görüntü Bölütleme Karşılaştırması (Multimodal Brain Tumor Image Segmentation Benchmark)
BT	Bilgisayarlı Tomografi
DN	Doğru Negatif
DP	Doğru Pozitif
DPSO	Darwinci Parçacık Sürü Optimizasyonu
DWT	Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform)
EEG	Elektroensefalografi
FCM	Bulanık C-ortalama (Fuzzy C-Means)
FODPSO	Kesirli Düzen Darwinci Parçacık Sürü Optimizasyonu (Fractional Order Darwinian Partical Swarm Optimisation)
GCE	Global Consistent Error
HBCE	Histogram Tabanlı Küme Kestirimi (Histogram Based Cluster Estimation)
HSI	Habitat Uygunluk İndeksi (Habitat Suitability Index)
HU	Hounsfield Birimi (Hounsfield Unit)
h-v	Yatay – Dikey (Horizontal - Vertical)
ISIC	Uluslararası Cilt Görüntüleme İşbirliği (The International Skin Imaging Collaboration)
JAC	Jaccard indeksi
KDD	Kısmi diferansiyel denklem
mDPSO	Merkez Tabanlı Darwinci Parçacık Sürü Optimizasyonu
mFODPSO	Merkez Tabanlı Kesirli Düzen Darwinci Parçacık Sürü Optimizasyonu (Fractional Order Darwinian Partical Swarm Optimisation)
mPSO	Merkez Tabanlı Parçacık Sürü Optimizasyonu
MR	Manyetik Resonans
PSO	Parçacık Sürü Optimizasyonu
RI	Rand Index
SIVs	Uygunluk İndeks Değişkenleri (Suitability Index Variables)
v-h	Dikey – Yatay (Vertical - Horizontal)
YN	Yanlış Negatif
YP	Yanlış Pozitif
YSA	Yapay Sinir Ağı

SİMGELER

μ	Dışa Göç Verme
T	Eşik Değeri
Γ	Gama Fonksiyonu
L	Görüntünün yoğunluk seviyesi
H	Habitat
S	Habitattaki Türlerin Sayısı
λ	İçe Göç Alma
k	Küme sayısı
Smax	Maksimum Tür Sayısı
t	Zaman

ÖZET

İKİ BOYUTLU MEDİKAL GÖRÜNTÜLERDE HİSTOGRAM TABANLI KÜME KESTİRİM METODU İLE YARI OTOMATİK SEGMENTASYON YÖNTEMLERİNİN GELİŞTİRİLMESİ

Enver KÜÇÜKKÜLAHLI
Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı
Doktora Tezi

Danışman: Doç. Dr. Pakize ERDOĞMUŞ
Eş Danışman: Doç. Dr. Kemal POLAT
Haziran 2018, 110 sayfa

Medikal görüntülerin bilgisayar ya da bir uzman tarafından yorumlanmasından önce görüntünün anlamlı parçalara bölütlenerek sonraki görü işlemleri için hazırlanması doğru yönlendirme ve teşhis aşamasında son derece hayati önem taşımaktadır. Bölütleme algoritmalarının doğru sonuca ulaşabilmeleri ise küme sayısının doğru olarak tahmin edilmesine bağlıdır. Bu çalışmada iki boyutlu medikal görüntülerde küme sayısının ve tahmini küme merkezlerinin tespitinin yapılması ve yarı otomatik bir bölütleme algoritmasının geliştirilmesi hedeflenmiştir. Bu hedef doğrultusunda, Histogram Tabanlı Küme Kestirim (HBCE – Histogram Based Cluster Estimation) metodu geliştirilmiştir. Geliştirilen HBCE metodu ile küme sayısı ve tahmini küme merkezi bilgisi elde edilmiş ve bu bilgi K-ortalama, Biyocoğrafya Tabanlı Optimizasyon (BBO - Biogeography Based Optimisation), Parçacık Sürü Optimizasyonu (PSO), Darwinci Parçacık Sürü Optimizasyonu (DPSO), Fraksiyonel Düzen Darwinci Parçacık Sürü Optimizasyonu (FODPSO) ve bu algoritmaların merkez tabanlı olarak düzenlenmiş halleri olan mPSO, mDPSO, mFODPSO algoritmalarından herhangi birine verilerek bölütleme işlemi gerçekleştirilmiştir. Gerçekleştirilen çalışmanın test edilmesinde Uluslararası Cilt Görüntüleme İşbirliği (ISIC- The International Skin Imaging Collaboration) veritabanından elde edilen dermoskopi görüntüleri ve bu görüntülere ait insan bölütleme sonuçları kullanılmıştır. Bölütleme işlemi daha anlaşılır ve kolay hale getirmek için MATLAB™ ortamında bir arayüz geliştirilmiştir. Bölütleme performanslarını test etmek için, Rand Index (RI), Global Consistency Error (GCE), Dice Skoru, Jaccard indeksi, Doğruluk, Hassasiyet, Özgünlük değerleri kullanılmıştır. Yapılan karşılaştırmalı testler sonucunda HBCE algoritmasından alınan küme sayısı bilgisi ve tahmini küme merkezi değerlerinin bir parçacık için bölütleme algoritmalarına verilmesinin hem bölütleme algoritmasının performansını artırdığı hem de algoritma kararlılığını iyileştirdiği gözlemlenmiştir. Ayrıca merkez tabanlı hale getirilen mDPSO algoritmasının diğer algoritmalara nazaran daha iyi sonuçlara (%94,677 doğruluk) ulaştığı sonucuna varılmıştır.

Anahtar sözcükler: Görüntü Bölütleme, Küme Sayısı Belirleme, Merkez Tabanlı Darwinci Parçacık Sürü Optimizasyonu.



ABSTRACT

DEVELOPMENT OF SEMI-AUTOMATIC SEGMENTATION METHODS BY HISTOGRAM BASED CLUSTER ESTIMATION METHOD IN TWO- DIMENSIONAL MEDICAL IMAGES

Enver KÜÇÜKKÜLAHLI

Düzce University

Graduate School of Natural and Applied Sciences, Department of Electrical -
Electronics and Computer Engineering

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Pakize ERDOĞMUŞ

Co-supervisor: Assoc. Prof. Dr. Kemal POLAT

June 2018, 110 pages

Before the medical images have been interpreted by a computer or an expert, the preparation of medical images for subsequent image processing by segmenting the images into meaningful pieces is of vital importance in the correct orientation and diagnosis phase. The fact that segmentation algorithms can achieve the correct result depends on the correct estimation of the number of clusters. In this study, it is aimed to determine the number of clusters, to estimate cluster centers and to improve a semi-automatic image segmentation algorithm. With this aim, a novel algorithm called Histogram Based Cluster Estimation(HBCE) was developed in the extend of this thesis. Based on the developed HBCE method, cluster numbers and estimated cluster center information were obtained. Segmentation was realized sending with this information as an initial values for K-means, Biogeography Based Optimization (BBO), Particle Swarm Optimization (PSO), Darwinian Particle Swarm Optimization (DPSO), Fractional Order Darwinian Particle Swarm Optimization (FODPSO). The cluster number and cluster centers were also sent to modified versions of the PSO, DPSO, and FODPSO called as respectively mPSO, mDPSO and mFODPSO. The International Skin Imaging Collaboration (ISIC) and its human segmentation results were used for testing. To make the segmentation process easier and more intuitive, an interface has been created in the MATLAB™ environment. Rand Index, Global Consistency Error (GCE), Dice Score, Jaccard Index, Accuracy, Sensitivity, Originality values were used to test the segmentation performances. As a result of the comparative tests, that the number of clusters and the estimated cluster center values obtained from the HBCE algorithm are given to the segmentation algorithms for only one particle, was observed to increase the performance and stability of the segmentation algorithm. Moreover, the center-based mDPSO algorithm gives better results (94,677% accuracy) than the other algorithms.

Keywords: Center-Based Darwinian Particle Swarm Optimization, Determining Number of Clusters, Image Segmentation.

EXTENDED ABSTRACT

DEVELOPMENT OF SEMI-AUTOMATIC SEGMENTATION METHODS BY HISTOGRAM BASED CLUSTER ESTIMATION METHOD IN TWO- DIMENSIONAL MEDICAL IMAGES

Enver KÜÇÜKKÜLAHLI

Düzce University

Graduate School of Natural and Applied Sciences, Department of Electrical -
Electronics and Computer Engineering

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Pakize ERDOĞMUŞ

Co-supervisor: Assoc. Prof. Dr. Kemal POLAT

June 2018, 110 pages

1. INTRODUCTION

Before the medical images were interpreted by a computer or an expert, the preparation of medical images by segmenting the images into meaningful pieces for subsequent image processing is of vital importance at the diagnosis phase. The objects and the background in a picture or the borders of the organs in the medical images can be determined with image segmentation. With the segmented 2d images, the area of the lesion can be specified. With the segmented 3d images, the volume of the organs or lesions can also be specified. So the exact segmentation of the images increases the accuracy of the image processing and computer vision applications.

The fact that segmentation algorithms can achieve the correct result depends on the correct estimation of the number of clusters. The number of clusters and some similar results can be estimated by looking at the histogram information.

In this study, it is aimed to develop a histogram based cluster estimation method and a semi - automatic segmentation algorithm to determine the cluster number and estimated cluster centers in two - dimensional medical images.

2. MATERIAL AND METHODS

In the development stage of the algorithm, The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS) database and the human segmentation results from these images were used. The dermoscopy images obtained from the International Skin Imaging Collaboration (ISIC) database and the human segmentation results from these images were used to benchmarking the results.

Some pre-processing steps are required to ensure that the segmentation process can be performed in a healthy manner. First of these steps is extracting the histogram information from the image. With this step the histogram information of the image to be segmented is extracted and prepared for filtering. The filtering process aims to reduce the excess oscillation and noises on the histogram depending on the image. Discrete Wavelet Transform (DWT) or Median filter can be selected for filtering. Obtained histogram information after filtering is given to Histogram-Based Cluster Estimation (HBCE) algorithm. The number of clusters and possible cluster centers are determined by selecting one of the Vertical - Horizontal ($v - h$) or Horizontal - Vertical ($h - v$) methods in the HBCE algorithm and a threshold value is determined when necessary. The obtained cluster centers and cluster number information are given to the optimization algorithms for segmentation together with the image and the segmentation process is performed. For the segmentation process, it can be chosen one of these algorithms; K-means, Biogeography Based Optimization (BBO), Particle Swarm Optimization (PSO), Darwinian Particle Swarm Optimization (DPSO), Fractional Order Darwinian Particle Swarm Optimization (FODPSO) and mPSO, mDPSO, mFODPSO algorithms, which are the center-based states of these algorithms. In the final process, joining or editing is performed on the segments and the segmented image is obtained. An interface has been developed in the MATLAB TM environment and the entire work has been integrated into this interface so that the work can be easily applied by other users.

3. RESULTS AND DISCUSSIONS

The 30 images obtained from The International Skin Imaging Collaboration (ISIC) were run 30 times for each algorithm and the results were recorded for comparison. The human segmentation results obtained from the ISIC database were used for comparison with the obtained results. Rand Index, Global Consistency Error (GCE), DICE Score,

Jaccard Index, Accuracy, Sensitivity, Specificity and time comparisons were used for comparison.

The segmentation algorithms executed with the priori knowledge of the number of clusters received from HBCE are evaluated in terms of accuracy performances. It is observed that mPSO 94.556%, mDPSO 94.593% and mFODPSO 94.595% have better results than the other algorithms. When the operating times of the algorithms are considered, the HBCE algorithm completes the process in 0.52 ms even in the worst case.

The obtained mean values of the Accuracy results show that when the estimated cluster center information from the HBCE has been given for only one particle in the mDPSO algorithm outperforms the other with 94.677%.

4. CONCLUSION AND OUTLOOK

Experiments were carried out on the sample images mentioned in the material section and the experiments of the algorithms with the screen images belonging to these experiments were exhibited. The performed studies were compared with human segmentation results using various performance criteria and the results were displayed. As a result of the comparative tests, it is observed that both the number of clusters and the estimated cluster center values obtained from the HBCE algorithm are given to the segmentation algorithms for a particle, both improving the performance of the segmentation algorithm and improving the algorithm stability. Moreover, the center-based mDPSO algorithm achieved better results than the other algorithms.

1. GİRİŞ

Günümüzde son derece önemli hale gelen makine görmesi birçok alanda hayati önem taşımaktadır. Medikal görüntü işleme de bu alanlardan bir tanesidir ve görüntünün bilgisayar ya da uzman tarafından yorumlanmadan önce bazı ön işlemlerden geçmesi gerekmektedir. Bu ön işlem adımlarından olan bölütleme, görüntünün anlamlı parçalara ayrılarak sonraki görü işlemleri için hazırlanmasını sağlamaktadır. Tezin giriş kısmında bölütlemeyi ve bölütleme işleminde kullanılan yöntemlerden bahsedilecektir.

1.1. BÖLÜTLEME

Bilgisayar teknolojisindeki gelişmelerle birlikte görüntü işleme teknikleri çok çeşitli uygulamalarda oldukça önemli hale gelmiştir. Görüntü bölütleme en önemli görüntü işleme sürecidir [1]. Bölütleme, nesne tanıma, sahne ve görüntü anlamlandırmada ilk adım ve kilit konu olarak görülebilir. Uygulamalar, endüstriyel kalite kontrolünden tıbbi görüntülemeye, robot navigasyonu, jeofiziksel keşif, askeri uygulamalar ve medikal görüntü işlemeye kadar uzanır. Bütün bu alanlarda nihai sonucun kalitesi büyük ölçüde bölütlemenin niteliğine bağlıdır [2].

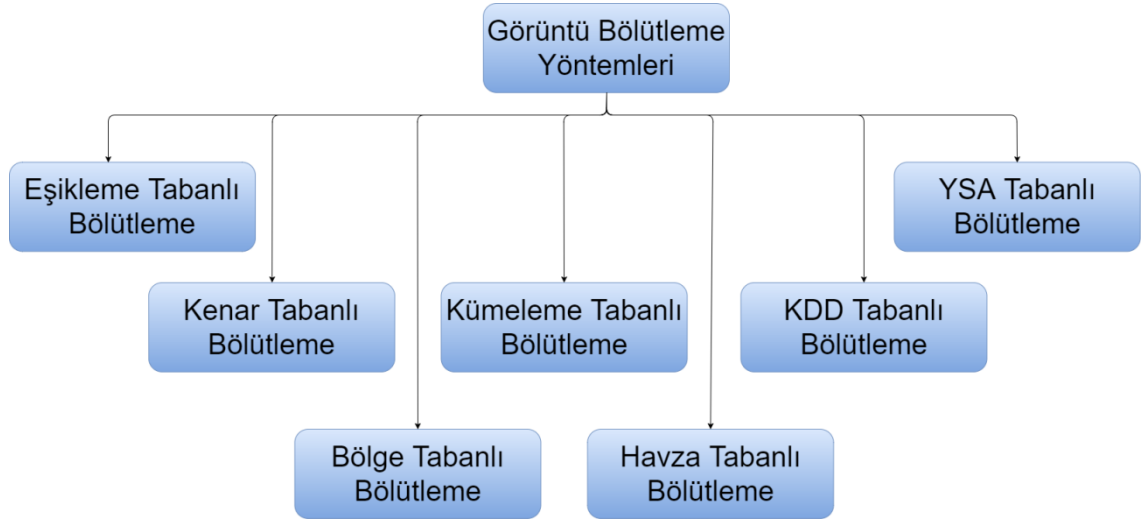
Bölütleme genellikle bir görüntünün birbirine bağlı bölgelere bölünmesi anlamında kullanılır. Bu tanımla bölgelerin üretimi, sınıflamanın ön aşaması olarak vurgulanır. Diğer tanımlamalar, tıbbi görüntülemenin en yaygın uygulaması olan sağlıklı anatomik yapılar ile patolojik doku arasındaki ayrımcılığa odaklanmaktadır. Tanımı gereği, bölütleme sonucu her zaman bölgesel soyutlanma seviyesinde bulunur. Bölütleme işlemi, özellik çıkartma düzeyine bağlı olarak; piksel, kenar, doku veya bölgeye dayalı olabilmektedir. Buna ek olarak, çeşitli yöntemlerin kombinasyonu ile elde edilen hibrid yaklaşımlar da bulunmaktadır [3].

Görüntü bölütleme için bir kural tabanı şu şekilde belirtilebilir [4], [5]:

- Her görüntü noktası bir bölgeye ait olmalıdır. Bu durum her nokta işlenene kadar bölütlemenin sonlandırılmaması gerektiği anlamına gelir.

- Bölütlemelerde kesişen bölgelerin olmaması gerekir ve bölgeler bitişik piksellerden oluşur.
- Bölütlenmiş bölgelerin ne tür özelliklere sahip olması gerektiğini, tek biçimli gri seviyeleri belirler ve bölütlemedeki her bölgenin maksimumluğunu gösterir.

Görüntü bölütleme, görüntüleri bölümlere ayırmak için çeşitli teknikler kullanır. Her tekniğin kendine özgü özellikleri vardır. Görüntü bölütleme için kullanılan teknikler şunlardır: eşikleme tabanlı teknikler, kenar tabanlı teknikler, bölge tabanlı teknikler, kümeleme tabanlı teknikler, havza tabanlı teknikler, kısmi diferansiyel denklem tabanlı ve yapay sinir ağı tabanlı teknikler vb. Görüntü bölütlemeye dair teknikler Şekil 1.1’de verilmiştir.



Şekil 1.1. Görüntü bölütleme teknikleri.

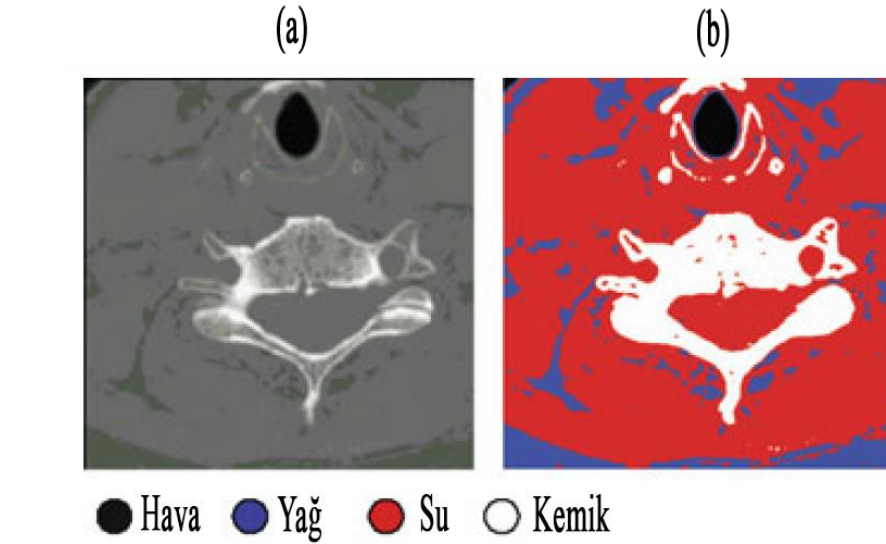
1.1.1. Eşikleme Tabanlı Bölütleme

Gerçekleştirilebilecek en temel bölütleme işlemi bir görüntünün eşik değerlemesidir. Bu yöntemde, pikselin ait olduğu sınıfı belirlemek için her piksel ilişkili ölçümünün bir veya daha fazla eşiğiyle karşılaştırılır. Renk veya basit bir doku tanımlayıcı kullanılabilir olsa da, özellik genellikle gri seviyededir. Eşikler sabit değerler halinde görüntüye uygulanabilir veya eşik değeri görüntü boyunca dinamik olarak değişecek şekilde uygulanabilir.

Kontrollü koşullar altında, bölünecek nesnelerin veya bölgelerin yüzey yansıtma düzgünlüğü iyiye, arka plandan farklıysa ve sahne eşit şekilde aydınlatılıyorsa, sonuçta ortaya çıkan görüntü, iyi tanımlanmış sınırlara sahip homojen bölgeler içerecektir ve bu

sınırlar genelde bi-modal veya multi-modal histogram olarak belirlenir.

Piksel yoğunluklarının atanması belli bir doku türü için iyi biliniyorsa ve sabitse, statik eşikler uygulanabilir. Statik bir eşik, benzer görüntü dizisindeki tek örnekten bağımsızdır. Örneğin, Bilgisayarlı Tomografi (BT)'deki kemik veya yumuşak doku pencereleri Hounsfield Biriminde (HU) statik eşiklerle gerçekleştirilebilir. Şekil 1.2'de Statik eşik değeri verilmiştir [3].



Şekil 1.2. Statik eşik değeri [6].

BT'deki piksel tabanlı bölütleme, farklı doku türleri için pencerelerin tanımlanmasına izin veren Hounsfield Birimlerine dayanır. Bununla birlikte, pek çok durumda arka plan düzeyi sabit değildir ve nesnelerin kontrastı görüntü içinde değişir. Bu gibi durumlarda, görüntünün bir alanında iyi çalışan bir eşik, diğer alanlarda kötü çalışabilir. Bu nedenle, görüntüdeki pozisyonun fonksiyonunda yavaş yavaş değişen bir eşik kullanmak uygundur. Chow ve Kaneko [7] tarafından görüntünün dikkörtgen alt resimlere bölünmesi ve her alt resimde eşik hesaplanması için bir dinamik eşik önerilmiştir. Bununla birlikte, alt-görüntü, gri-seviyeli histogramı bi-modal değilse ve bu durumda bu gibi alt-görüntüler komşu alt-görüntülerden interpolasyon eşikleri alıyorsa, eşik erişemeyebilir. Son olarak, tüm görüntü, her bir alt-görüntü için ayrı eşikler kullanılarak eşiklendirilir [8].

Bu yaklaşımın başarısı, uygun eşiklerin var olup olmadığı ve görüntü histogramından çıkarılıp çıkarılmadığına bağlıdır. Uygun bir eşik belirlemek için çeşitli yöntemler önerilmiştir. Bununla birlikte, bu sadece endüstriyel uygulamalarda olduğu gibi

kontrollü bir çevre varsayımı altında sınırlı bir görüntü setiyle de mümkündür. Aydınlanma, gürültü ve doku çeşitliliği bulunan doğal görüntülerde, yöntemler bu sorunlarla başa çıkmada yetersiz kalabilmektedir.

Eşikleme yöntemi genel olarak üç kategoriye ayrılmıştır.

1) Küresel Eşikleme: Küresel Eşikleme, uygun bir eşik değeri T 'nin seçilmesine dayanır. T sabit bir değerdir ve çıkış görüntüsü bu T değerine bağlıdır.

$$q(x, y) = \begin{cases} 1, & p(x, y) > T \\ 0, & p(x, y) \leq T \end{cases} \quad (1.1)$$

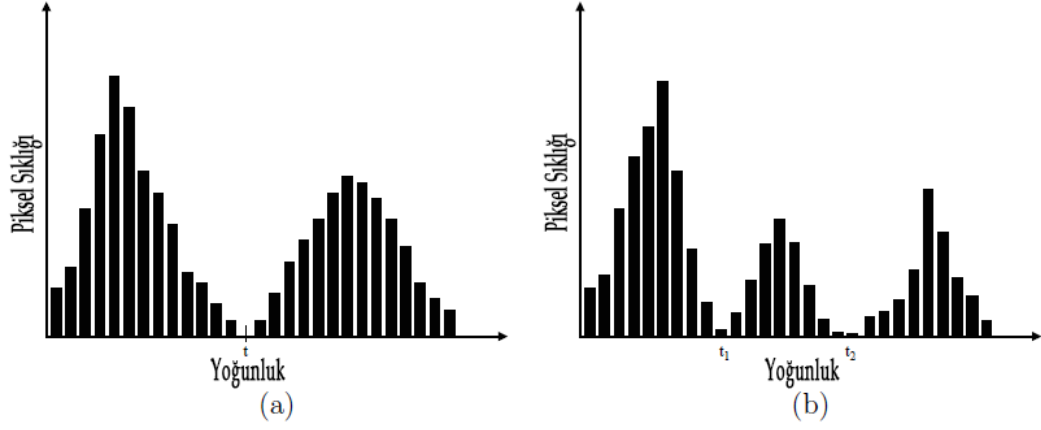
2) Değişken Eşikleme: Bu yöntemde, T değeri görüntüye göre değişir. Ayrıca iki kategoriye ayrılmıştır:

- Yerel Eşik: Burada T değeri x ve y komşuluklarına bağlıdır.
- Uyarlanabilir Eşik: T değeri x ve y 'ye bağlı bir fonksiyondur.

3) Çoklu Eşikleme: Bu eşik türünde, T_0 ve T_1 gibi çoklu eşik değerleri vardır. Bu çıktılar kullanılarak görüntü aşağıdaki gibi hesaplanabilir:

$$q(x, y) = \begin{cases} m, & p(x, y) < T_0 \\ n, & T_0 \leq p(x, y) < T_1 \\ o, & T_1 \leq p(x, y) \end{cases} \quad (1.2)$$

Eşik değerleri histogramın zirveleri yardımıyla hesaplanabilir. Daha koyu bir zeminde parlak bir nesneyle oluşturulmuş bir görüntüde histogram Şekil 1.3 (a)'da gösterilen örneğe benzer şekilde bi-modaldır. İki zirve nesnenin içindeki ve dışındaki nispeten çok sayıda noktaya karşılık gelir. Zirveler arasındaki dalma, cismin kenarı etrafındaki nispeten az sayıda noktaya karşılık gelir. Eşik her iki doruk arasındaki vadiye yerleştirilir, daha sonra gri seviyesi eşğin T 'sinden yüksek olan pikseller nesneyle ilişkilendirilirken geri kalan pikseller arka plana bağlanır. Şekil 1.3 (b), multi-modal bir histogramı göstermektedir.



Şekil 1.3. Histogram örnekleri, (a) bi-modal histogram, (b) multi-modal histogram [8].

1.1.2. Kümeleme Tabanlı Bölütleme

Kümeleme, bir veri setinin, benzer özelliklere sahip veri noktalarının aynı küme içine yerleştirilmesiyle gerçekleştirilen bir sınıflandırmadır. Resim parçalamasını aynı renk ve/veya aynı dokuya sahip olan pikselleri gruplandırarak kümeleme olarak düşünmek doğaldır. Kümeleme yöntemleri iki temel türe ayrılabilir: hiyerarşik ve bölümlü kümeleme. Her bir türde, kümeleri bulmak için bir alt sınıf türleri ve farklı algoritmalar bulunmaktadır.

Hiyerarşik kümeleme, daha küçük kümeleri daha büyük kümelere (aglomeratif algoritmalar) birleştirerek veya daha büyük kümeleri bölerek (bölücü algoritmalar) ardışık olarak ilerlemektedir. Kümeleme yöntemleri, hangi iki küçük kümenin birleştiği veya hangi büyük kümenin ayrıldığına karar verme kuralında farklılık gösterir. Algoritmanın sonucu, kümelerin nasıl ilişkili olduğunu gösteren dendogram adlı bir küme ağıdır. Dendogramı istenen seviyede keserek veri öğelerinin birbirine ayrılmış gruplar halinde kümelendirilmesi sağlanır.

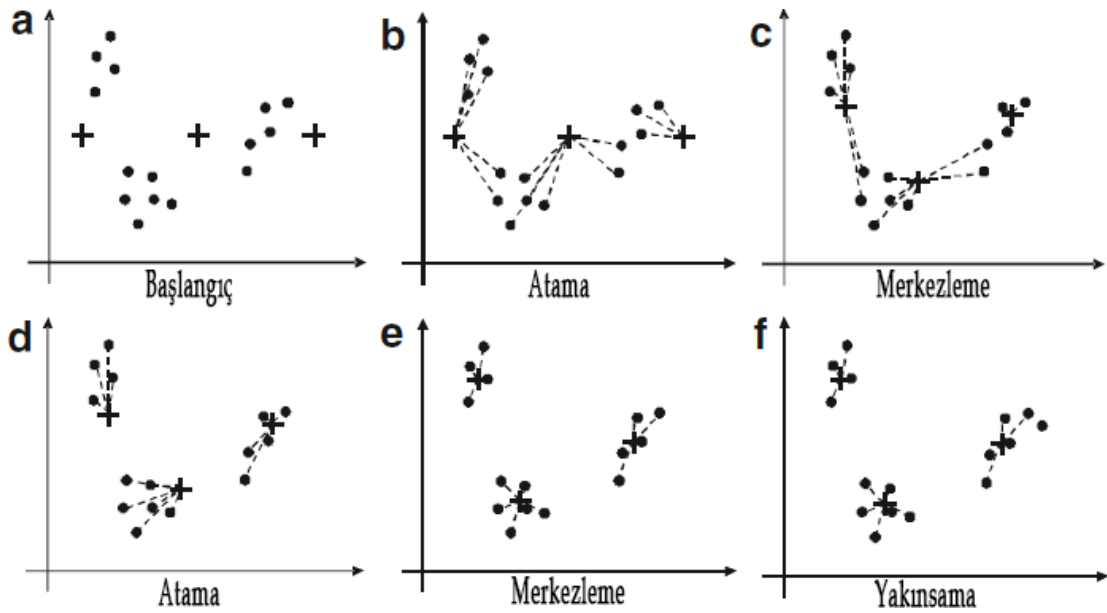
Öte yandan, bölümlü kümeleme, veri kümesini ayık kümeler grubuna doğrudan dönüştürmeye çalışır. Amaç fonksiyonu, bir gösterimin ne kadar iyi olduğunu ifade eder ve daha sonra, kümeleme algoritması en iyi temsilini elde etmek için bu işlevi en aza indirmeye çalışır. Kriter fonksiyonu, olasılık yoğunluk fonksiyonundaki piklere veya küresel yapıya küme atayarak verilerin yerel yapısını vurgular. Genel küresel kriterler, farklı kümeler arasındaki benzerliği en üst düzeye getirirken, her küme içindeki örnekler için farklılık ölçüsünü en aza indirmeyi içerir. En yaygın kullanılan kümeleme yöntemi, kriter fonksiyonu en yakın küme merkezinden karekök uzaklığı olan K-ortalama algoritması [9]'dır.

K-ortalama kümelemede, algoritma elde edilecek kesim sayısı ile başlatılarak, ilk küme merkezi rastgele olacak şekilde algoritma tarafından yerleştirilir. Ardından, aşağıdaki iki adım, süreç tamamlanıncaya kadar yinelenerek tekrarlanır:

1. Her veri noktası en yakın küme merkezine hizalanır.
2. Geçerli atamaya dayanarak, kümeleme merkezleri yeniden hesaplanır.

Ortaya çıkan küme merkezlerinin yalnızca iterasyon sayısını ve dolayısıyla hesaplama süresini etkileyebilecek başlangıç konumlarından bağımsız oldukları matematiksel olarak kanıtlanabilir. Bununla birlikte, kümeleme sonucunu kesinlikle etkileyen sabit mesafe metrikleri (örn., Öklid mesafesi) veya bir veri uyarlamalı metrik (örn. Mahalanobis mesafesi) seçilebilir. Ayrıca önceden tanımlanmış küme merkezleri sayısı önemli bir parametredir.

K-ortalama kümeleme işlem adımları Şekil 1.4'te verilmiştir. Küme sayısı önceden tanımlanmıştır. Başlangıç (a) adımından sonra veri noktaları en yakın küme merkezine atanır (b). Daha sonra, merkezlerin pozisyonları yeniden hesaplanır (c) ve süreç, küme merkezlerine olan uzaklık minimuma inene kadar devam eder (d-e-f). Küme merkezlerinin son konumu ilk konumlarından etkilenmez. Bu, yalnızca yineleme sayısına etki edebilir [3].



Şekil 1.4. K-ortalama kümeleme algoritmasının çalışma adımları [10].

Eşikleme yöntemleri gibi kümeleme yöntemleri genel ve konumsal bilgileri koruyamaz. Bunun en büyük dezavantajı, bölütlemeye kastedilen şeylerin önemli bir parçası olan piksellerin mekansal olarak düzenlenmesinin değişmez olmasıdır. Ortaya çıkan bölümler bağlı değildir ve yaygın şekilde dağılabilir. Piksel koordinatlarını özellik olarak kullanarak bu tür bilgileri tanıtmak için bazı girişimlerde bulunulmuştur. Morfolojik filtreleme ile gerçekleştirilen son-işleme bu girişimlere örnek verilebilir.

1.1.3. Kenar Tabanlı Bölütleme

Bu bölütleme türü, kenarların soyut seviyesine dayanmaktadır ve resimdeki kapalı çizgiler sayesinde nesneleri yakalamaya çalışmaktadır. Bu nedenle, kenar tabanlı bölütleme işlemleri, nesnelerin açıkça tanımlanmış sınırlar olarak temsil edildiği problemler için kullanılır. Biyolojik doku görüntülerinde bu nadiren oluşur. Bu özel durumlardan bir tanesi, bir radyografda görüntülenen metalik bir implanttır.

Genel olarak, kenar tabanlı bölütleme için görüntü işleme zinciri kenar çıkarma ve kenar tamamlama işlemlerinden oluşur. Kenar çıkartma, genellikle Kenar-bazlı öznitelik çıkarımı ile elde edilir. İşlemin sonraki adımları sadece kenar pikselleri ve kenarsız pikseller elde etmek için ikili hale getirme, gürültüyü azaltmak için morfolojik filtreleme ve son olarak kenarın iskeletinin hesaplanmasıdır. İkili konturların izlenmesi ve kapatılması, kenar tabanlı bölütlemenin temel görevleridir. Neredeyse sadece, sezgisel yöntemler kullanılır [10].

Bölgenin sınırında bulunan bağlı bir piksele kenar denir. Bir kenardaki bu pikseller kenar noktaları olarak tanımlanır [11]. Bir görüntü fonksiyonunun türevi bulunarak kenar hesaplanabilir. Bazı kenarların tespit edilmesi çok kolaydır. Bunlar: Adım kenarı, Rampa kenarı, Başak kenarı, Çatı kenarıdır. Adım kenarı, yoğunluk seviyesinde ani bir değişikliktir. Rampa kenarı, yoğunlukta kademeli bir değişimdir. Başak kenar yoğunluğunda hızlı bir değişiklik olur ve bundan sonra hemen orijinal yoğunluğa döner. Çatı kenarı kısa bir mesafeden anlık değildir. Kenar tabanlı görüntü bölütleme yöntemi yapısal teknikler altında kalmaktadır [12].

1.1.4. Bölge Tabanlı Bölütleme

Bölgeye dayalı bölütlemenin bir avantajı olarak sadece bağlı kesimler üretilir ve morfolojik son işlemeden kaçınılmaktadır. Aglomeratif (aşağıdan yukarıya) ve bölücü (yukarıdan aşağıya) yaklaşımlar vardır. Tüm yaklaşımlar, komşu piksellerin veya

bölgelerin atamasını yönlendirmek için belli bir mesafe veya benzerlik ölçüsüne dayanır [10].

Bu yöntem, piksellerin benzer özellikleri temelinde bölümlere ayrılmış bir görüntüye dayanır. Bölge tabanlı bölütleme yöntemi ayrıca iki kategoriye ayrılır:

1) Bölge büyütme yöntemleri: Bölgeye göre büyüyen bölütleme yöntemleri, tohumun büyümesine (ilk pikseller) bağlı olarak görüntüyü çeşitli bölgelere ayıran yöntemlerdir. Bu tohumlar elle (önceki bilgilere göre) veya otomatik olarak (belirli uygulamaya göre) seçilebilir. Daha sonra tohumların büyümesi, pikseller arasındaki bağlantı ile kontrol edilir ve problemin önceki bilgisinin yardımıyla, bu durdurulabilir.

2) Böl ve Birleştir: Başlangıçta, tek bir bölge olarak alınan tüm görüntü, daha fazla bölünme mümkün olmayana kadar tekrar tekrar bölünür. Ardından bitişik ve benzer bölgeler birleştirilir. Daha fazla birleşme mümkün olmayana kadar tekrarlanır [1].

1.1.5. Havza Tabanlı Bölütleme

Havza bazlı yöntem, topolojik yorumlama kavramını kullanır. Yoğunluk, suyun döküleceği yerdeki minimum deliklere sahip havzaları temsil eder. Su, havzanın sınırına ulaştığında, bitişik havzalar birleştirilir. Havza barajları arasındaki ayrımı korumak için gereklidir ve bölütleme bölgesinin sınırları vardır. Bu barajlar dilatasyon kullanılarak inşa edilmiştir. Havza yöntemleri, görüntü gradyanını topografik yüzey olarak düşünmektedir. Daha fazla gradyana sahip pikseller, sürekli sınırlar olarak temsil edilir [13].

1.1.6. Kısmi Diferansiyel Denklem (KDD) Tabanlı Bölütleme

Kısmi diferansiyel denklem (KDD) tabanlı yöntemler, hızlı bölütleme yöntemleridir. Bunlar kritik zaman uygulamaları için uygundur. Temel iki KDD yöntemi vardır: doğrusal olmayan izotropik difüzyon filtresi (kenarları güçlendirmek için kullanılır) ve ikinci dereceden kuadratik olmayan değişim restorasyonu (gürültüyü gidermek için kullanılır). KDD yönteminin sonuçları, yakın operatörler kullanılarak kaydırılabilen bulanık kenarlar ve sınırlardır. Görüntüden gelen gürültüyü azaltmak için dördüncü dereceden KDD yöntemi kullanılır, kenarları ve sınırları daha iyi tespit etmek için ikinci derece KDD yöntemi kullanılır [14].

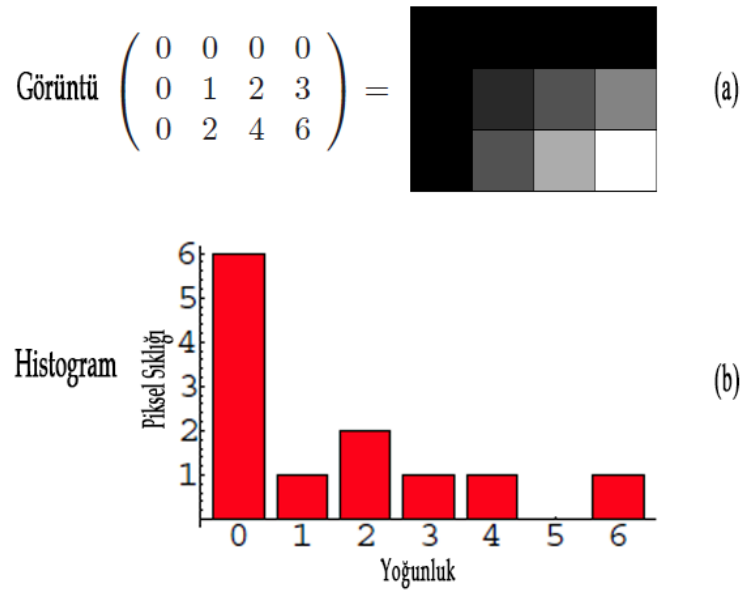
1.1.7. Yapay Sinir Ağı (YSA) Tabanlı Bölütleme

Bir sinir ağı öğrenme sürecini simüle etmeye çalışan insan beyninin yapay bir temsilidir [15], [16]. Yapay sinir ağı (YSA) bazlı bölütleme yöntemleri karar verme amacıyla insan beyninin öğrenme stratejilerini örnek alır. Birbirine bağlı ağlar çok sayıda bağlanmış düğümden yapılır ve her bir bağlantı belirli bir ağırlığa sahiptir. Günümüzde bu yöntem çoğunlukla medikal görüntülerin bölütlemesi için kullanılmaktadır. Gerekli görüntüyü arka plandan ayırmak için kullanılır [1].

1.2. HİSTOGRAM

Kontrast yayma, pencereleme, eşikleme, bölme vb. gibi gri değer haritalama işlemleri her zaman en az bir parametreye sahiptir. Bu parametreler için pratik bir değer oluşturmak için kullanışlı bir araç, bir görüntüde oluşan gri değerlerin frekans grafiği olan görüntü histogramıdır.

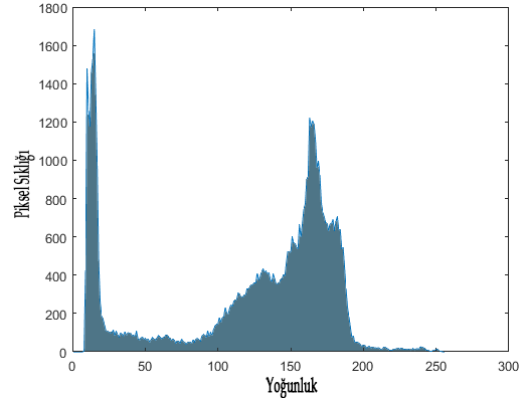
Şekil 1.5 (a)'da örnek görüntü üzerinde siyahtan beyaza gidecek şekilde piksel değerleri [0-6] aralığında tanımlanmıştır. Bu tanımlamaya göre 0 değeri görüntüde 6 kez görünmekteyken, 2 değerine sahip pikseller 2 tanedir. Bu şekilde aynı renk değerine sahip pikseller sayılarak oluşturulan grafik görüntünün histogramı olarak adlandırılır ve Şekil 1.5 (b)'de verilmiştir.



Şekil 1.5. Örnek bir görüntü (a) ve görüntünün histogramı (b) [17].

Histogram bilgisine bakarak birçok çıkarımda bulunulabilmektedir. Şekil 1.6 (a)'da verilen kameraman görüntüsü üzerinde;

- Kameraman'ın siyah paltosu, en yüksek karanlık gri değerlerden oluşur. Bu nedenle, $x = 17$ piksel zirvesinin bu görüntü alanına karşılık gelmesi beklenen bir durumdur. Histogramda $[0, 25]$ aralığı incelenmek üzere görüntüden kesildiğinde, Şekil 1.6 (b) görüntüsü elde edilmektedir.
- Orijinal görüntüde yoğun bir şekilde karşılaşılan renk değeri çimenlerdeki gri değerlerdir. Histogram üzerinde $[100,150]$ aralığında en yüksek pik değeri $x=134$ değeridir, bu yüzden bu tepenin bahsi geçen çimenlere karşılık gelmesi beklenir. Histogramdan kesilen $[100,150]$ aralığı Şekil 1.6 (c)'de de görülebileceği gibi çimenlere denk gelmektedir. Ayrıca orijinal görüntüden de anlaşılabilceği gibi görüntünün sağ tarafı nispeten daha koyudur ve bu alana dahil olmuştur. Çimenlerin üzerinde görünen beyaz çiçekler ise bu alana dahil olamamıştır.
- Orijinal görüntüde en yoğun karşılaşılan renk değeri gökyüzündeki gri değerlerdir. Histogram üzerinde bu alana ait yüksek pik değeri $x=163$ değeridir, bu yüzden bu tepenin bahsi geçen gökyüzüne karşılık gelmesi beklenir. Histogramdan kesilen $[150,170]$ aralığı Şekil 1.6 (d)'de de görülebileceği gibi gökyüzünün tamamına denk gelmemektedir. Orijinal görüntü incelendiğinde gökyüzünün orta kısmının bir miktar daha parlak olduğu gözlemlenmektedir. Dolayısıyla, gökyüzünün geri kalanının $x = 180$ 'deki zirveye karşılık gelmesi beklenmektedir ki bahsi geçen nokta bu alanda 2. En büyük pik noktasıdır. Histogramdan elde edilen $[170,200]$ aralığına ait görüntü Şekil 1.6 (e)'de verilmektedir.



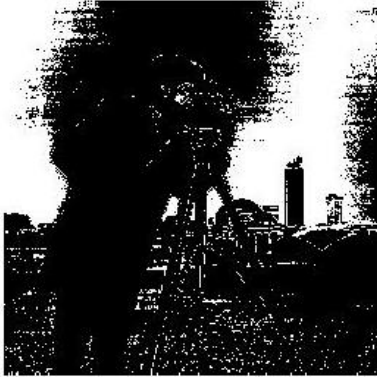
(a)



(b)



(c)



(d)



(e)

Şekil 1.6. Histogram bilgi çıkartımı örneği (a) Siyah-Beyaz bir fotoğraf ve bu fotoğrafa ait histogram, (b)Histogramdan çıkartılan [0,25] aralığı, (c)Histogramdan çıkartılan [100,150] aralığı, (d)Histogramdan çıkartılan [150,170] aralığı, (e)Histogramdan çıkartılan [170,200] aralığı.

1.3. LİTERATÜR TARAMASI

Görüntü işlemede filtreler sıklıkla kullanılmaktadır [18]–[21] ve genellikle görüntü üzerinde bulunan gürültülerin azaltılması için tercih edilmektedir [22]–[24]. Bir ön işleme adımı olarak histogramın filtrelenmesi bu işlemlerden farklı bir yaklaşımdır. Nabizadeh ve arkadaşları otomatik beyin tümörü algılama için Histogram Tabanlı Yerçekimsel Optimizasyon algoritması geliştirmişlerdir. Muhtemel küme merkezlerini kestirmek için histogram bilgisinden faydalanmışlar ve görüntü üzerindeki gürültüleri azaltmak için alçak geçiren gaus filtresi kullanırken histogramı düzgünleştirmek için Local Weighted Averaging Technique kullanmışlardır [25]. Ayrık Dalgacık Dönüşümü (DWT – Discrete Wavelet Transform) filtreleme işleminde sıkça kullanılmaktadır [26], [27]. Sharma ve diğerleri, odak ve odak olmayan Elektroensefalografi (EEG) sinyallerini sınıflandırmak için DWT tabanlı teknikler kullanmışlardır. Daubechies sıra 4 (db4) dalgacıklarını kullanarak EEG sinyallerinin altıncı seviyede ayrışmasını gerçekleştirmiş ve entropiler kullanarak dalgacık katsayılarının göreceli enerjisi için karmaşıklığını hesaplamışlardır [28]. Medyan filtresi genellikle gürültüyü gidermek için kullanılan doğrusal olmayan bir dijital filtreleme tekniğidir. Böyle bir gürültü azaltma, örneğin bir görüntü üzerinde kenar algılama gibi daha sonraki işlemlerin sonuçlarını iyileştirmek için tipik bir ön işlem adımıdır. Medyan filtrelemede, bir çıkış pikselinin değeri, piksellerin ortalama değerlerinden ziyade, komşu piksellerin medyanı tarafından belirlenir. Medyan, ortalama filtreye göre aşırı değerlere daha az hassastır, bu nedenle görüntüdeki keskinliği azaltmadan aykırı değerleri daha iyi kaldırabilir. Bu filtre, Manyetik Rezonans (MR) görüntüsünün kalitesini artırır [29]. Medyan filtre çeşitli çalışmalarda gürültü azaltımı için sıklıkla tercih edilen metotlardandır [30], [31].

Djerou ve arkadaşları, ikili parçacık sürü optimizasyonuna dayalı bir otomatik çok seviyeli eşikleme yaklaşımı önermişlerdir. Önerdikleri bu yaklaşımda, gri seviye eşik değerlerini kullanan bir fonksiyonu optimize ederek optimum eşik sayısını otomatik olarak tespit etmiş ve optimum eşik değerlerini aramışlardır. Geliştirdikleri algoritma yüksek sayıda eşik değeriyle başlayıp, sonrasında amaç fonksiyonuna bağlı olarak eşik değerlerini rafine etmiştir [32]. Cuevas ve arkadaşları, Diferansiyel Evrim optimizasyonuna dayanan görüntüde otomatik çoklu eşikleme yaklaşımı önermişlerdir. Burada bölütleme işlemi bir optimizasyon problemi olarak düşünülmüştür. Çalışmada parametreleri diferansiyel evrim algoritması kullanılarak hesaplanan Gauss fonksiyonu kullanılarak görüntünün 1d histogram bilgisi doldurulmuştur [33]. Wen ve arkadaşları

3d cerebrovascular yapıyı Time of Flight Magnetic Resonance Angiography datasından çıkartmak için otomatik istatistiksel yoğunluk tabanlı bir yaklaşım sunmuşlardır. Gauss ve Rayleigh dağıtım fonksiyonlarıyla modellenmiş yapıların beyin MR görüntüsünün histogram yoğunluğunu belirlemek için Finite Mixture Model kullanmışlardır [34]. Güvenç ve arkadaşları komşu piksellerin benzerlik yüzdelerini renk benzerlik ölçümü tabanlı matematiksel bir yaklaşım kullanarak hesaplamışlardır. Önerdikleri bu yöntem görüntüde muhtemel var olan bölge sayısı hakkında herhangi bir ön bilgiye ihtiyaç duymamaktadır [35]. Demirci ve arkadaşları renkli görüntüleri otomatik olarak ayırtmak için Otsu, Kapur ve ortalama esaslı eşik seçim yöntemlerini her bir kanal için uygulamış ve ardından belirlenen eşik değerlerini kullanarak üç boyutlu renk uzayını yeniden yapılandırmıştır [36]. Qu ve arkadaşları deniz marulu görüntülerinin otomatik bölütlenmesi için RGB görüntüyü NTSC renk uzayına dönüştürüp histogram bilgisini elde etmişler ve histogramdaki iki yükselme noktasını K-ortalama algoritmasına küme merkezi olarak vermişlerdir [37]. Azami ve diğerleri Bee Colony ve Hierarchical Evolutionary Algorithm kullanarak bir çoklu eşikleme algoritması gerçekleştirmişler ve MR görüntüleri üzerine uygulamışlardır. Bu algorithmada küme sayısı bilgisinin önceden girilmesine ihtiyaç bulunmamaktadır. Çalışmada optimum eşik sayısını bulmak için histogram üzerinde Bee Colony algoritmasını çalıştırmışlardır [38]. Bhandari ve arkadaşları uydu görüntülerini bölütlemek için Kapur's entropy, Cuckoo Search ve Wind Driven Optimization algoritmalarından bir hibrit geliştirmişlerdir. Optimum eşik değerini bulmak için Cuckoo Search ve Wind Driven Optimization algoritmalarını kullanmışlardır [39]. Bir modelin küme sayısının tahmin edilmesine yönelik beş genel yaklaşım çapraz doğrulama, cezalandırılmış olabilirlik tahmini, permütasyon testleri, yeniden örnekleme ve bir hata eğrisi dizini bulmadır. Çapraz doğrulama teknikleri, verileri mümkün olduğunca doğru bir şekilde sığdırmaya çalışan modeller oluşturur. Monte Carlo çaprazlama [40], [41], aşırı uyumu (çok fazla sayıda küme) önlemek için başarıyla kullanılmıştır. Cezalı olabilirlik tahmini de, verileri olabildiğince doğru bir şekilde yansıtan bir model bulmaya çalışır, aynı zamanda modelin karmaşıklığını da en aza indirir. Permütasyon testleri [42], bir "rasgele" zaman serisinin göreceli değişimine yaklaşma hatasının göreceli değişimini karşılaştırarak, verilere aşırı uyan bir PLA oluşturulmasını önlemeye çalışır. Eğer hatalar benzer bir hızda değişiyorsa, daha fazla bölüt zaman serisinin altta yatan yapısına değil gürültüye uyacaktır. Yeniden örnekleme [43] ve Konsensüs Kümelenmesi [44], veri kümesinin birçok örneğini kümeleyerek doğru sayıda kümeyi bulmaya ve çeşitli örneklerin

kümelenmelerinin en “kararlı” olduğu kümelerin sayısını belirlemeye çalışır. Bir hata eğrisine ait “kırılma noktası” belirlemek, uygun sayıda kümeyi veya parçayı belirlemek için sıklıkla kullanılır, ancak özellikle iyi araştırılmış bir yöntem değildir. Hata eğrisindeki her noktayı istatistiksel olarak değerlendiren yöntemler vardır ve bazı işlevler, döndürülen küme sayısını en aza indiren veya en üst düzeye çıkartan noktayı kullanır. Bu yöntemler boşluk istatistiğini [45] ve tahmin gücünü [46] içerir. Bu yöntemler genellikle (hiyerarşik algoritmalar hariç), tüm kümelenme veya bölütleme algoritmasının k'nın her bir potansiyel değeri için çalıştırılmasını gerektirir. Bu yöntemlerin çoğunun, kümelerin en iyi sayısını belirlemesi uygulamada çok iyi çalışmayabilir. Çapraz doğrulama ve cezalandırılmış olabilirlik tahmini gibi model tabanlı yöntemler, hesaplama açısından pahalıdır ve çoğu zaman algoritmasının birkaç kez çalıştırılmasını gerektirir. Permütasyon testleri ve yeniden örnekleme son derece verimsizdir, çünkü tüm kümeleme algoritmasının yüzlerce hatta binlerce kez tekrar çalışmasını gerektirir. Bir hata eğrisinin kırılma noktasını bulmak için mevcut yöntemlerin çoğunluğu, kümelenme algoritmasının k'nın her potansiyel değeri için yeniden çalıştırılmasını gerektirir. Daha da kötüsü, bir veri kümesini değerlendirmek için kullanılan işlevlerin çoğunun çalışma zamanları, bir küme kümesini üretmek için harcanan zamandan daha da uzun sürebilmektedir. Bir eğrinin kırılma noktasını bulabilen çoğu yöntem, kümeler iyi ayrıldığında da iyi çalışır. Birkaç tane var olan kümeleme algoritması, kümelenme sayısını belirlemek için yerleşik mekanizmalara sahiptir. TURN [47], [48] algoritması, bir eğrinin kırılma noktasını, 2. türevin kullanıcı tarafından belirlenen bir eşiğin üzerine çıktığı noktaya göre konumlandırır. BIRCH [49] algoritmasının bir varyantı [48], Bayes Bilgilendirme Kriteri ve kümeler arası mesafe ile kümelerin sayısı arasındaki oran değişimini kullanır [50]. Pratikte, kümelerin optimal sayısını belirlemek için çeşitli çalışmalar yapılmıştır. Pei ve arkadaşları [51] küme sayısını seçmek için sınıflar arası ve sınıf içi farklılığa dayalı optimal bir yöntem önermişlerdir.

MR görüntülerinin bölütlenmesiyle ilgili gerçekleştirilen çalışmaların test edilmesi ve karşılaştırmaların yapılması bir diğer önemli sorundur. Cabria ve Gondra yaptıkları çalışmada otomatik K-ortalama algoritmasının başarımını test etmek için kamuya açık beyin tümörü görüntü bölütleme (BRATS) [52], [53] MRI veritabanını kullanmışlardır [54]. BRATS veritabanı tümürlü beyin MR görüntülerinin yanında manuel olarak gerçekleştirilmiş bölütleme sonuçlarını da erişime sunmaktadır. Uluslararası Cilt

Görüntüleme İşbirliği (ISIC) Arşivi [55], 900'ü eğitim ve 379'u test olmak üzere toplam 1279 deri lezyonu görüntüsünden oluşmaktadır. Bu set algoritma geliştirmelerinde ve geliştirilen algoritmaların test edilmesinde sıklıkla kullanılmaktadır [56]–[59].

Benaichouche ve arkadaşları, metasezgisel optimizasyon tekniği kullanılarak başlama adımlarının otomatik belirlendiği bir Bulanık C-ortalama (FCM – Fuzzy C-Means) algoritması geliştirmişlerdir [60]. Mao ve arkadaşları, sayfa segmentasyonu algoritmalarında başlangıç değerlerini seçmek için simpleks metodunu kullanan bir otomatik eğitim metodu sunmuşlardır [61]. Beyin görüntülerinde bölütleme için; Cohesion based Self Merging temelli kısmi K-ortalama kümeleme algoritması [62], kümeleme başarısını artırabilmek için FCM ile yapay sinir ağları [63], sürü temelli kümeleme algoritması [64], FCM temelli Harmony Search hibrit algoritması [65], bulanık mantık temelli genetik algoritma [66] kullanılmıştır. MR görüntülerindeki gürültü ve referans görüntünün bulunmaması nedeniyle oluşan zorlukların üstesinden gelebilmek için K-ortalama ve Support Vector Machine [67], görüntülerde gürültü azaltımı için dalgacık dönüşümü kullandıktan sonra bölütleme için K-ortalama kümeleme algoritması [68] kullanılmıştır. K-ortalama kümeleme algoritması ve K-ortalama'ya dayalı hibrit yöntemler bölütleme işlemi için sıklıkla tercih edilmektedir. MR görüntülerin bölütlenmesinde; watershed bölütleme algoritmasıyla K-ortalama, genetik algoritma ile K-ortalama, genetik algoritma ile C-ortalama ve geleneksel K-ortalama algoritmaları [69] kullanılmıştır. Raju ve Rao yaptıkları çalışmada, görüntülerin bölümlere ayrılmasıyla sonuçlanan imgeler üzerinde kümeleme temelli teknikler kullanmışlardır. Parçacık Sürü Optimizasyonu (PSO) tekniği ile entegre FCM performansını ve varyasyonlarını farklı uygulama alanlarında analiz etmişlerdir. Farklı alanlardaki teknikleri analiz etmek için Global Consistency Error (GCE), Probabilistic Rand Index ve Variation of Information gibi çeşitli metrikler kullanmışlardır. Fractional Order Darwinian PSO (FODPSO) ile birlikte FCM'nin, kullanılan diğer tekniklerin diğer varyasyonlarıyla karşılaştırıldığında, sınıflandırma doğruluğu açısından daha iyi performansa ulaştığını görmüşlerdir [70]. Agarwal ve arkadaşları, kümeleme tekniğini kullanarak cilt lezyon bölütleme yöntemi önermektedirler. Önerilen algorithmadan elde edilen deri lezyon bölütlemesinin sonuçları, açıklamalı görüntüler ile karşılaştırılmıştır. Sonuçlar örtüşen puan ve korelasyon katsayısı şeklinde ifade edilmiştir. Algorithmadan elde edilen örtüşen puan ve korelasyon katsayısının maksimum değerleri sırasıyla %96,75 ve %97,66'dır. Sonuçlar ikna edicidir ve önerilen çalışmanın bazı gerçek

zamanlı uygulamalar için kullanılabileceğini göstermektedirler [71]. Eltayef ve arkadaşları, görüntü işleme teknikleri kullanılarak deri kanserinin sınır tespiti için tam otomatik bir yöntem sunmuşlardır. Saç ve çeşitli gürültüler, sırasıyla yönlü filtreler ve bozulan görüntünün yeniden oluşturulması yönteminin bir birleşimi uygulanarak tespit edilmiş ve kaldırılmıştır. Görüntülerde lezyon alanının sınırını belirlemek için Parçacık Sürü Optimizasyonu ve Markov Rasgele Alan yöntemlerini birleştirerek bir hibrit yöntem geliştirmişlerdir. Yöntem, 200 dermoskopik görüntüden oluşan bir veri seti üzerinde test edilmiştir ve deney sonuçları, yöntemin, lezyon sınırlarının çizilmesindeki doğruluk açısından, alternatif yöntemlere göre daha üstün olduğunu göstermiştir [72].

Sulaiman ve Isa sadece medikal görüntülerde değil tüm görüntülerde çalışabilecek Adaptif Bulanık K-ortalama algoritması gerçekleştirmiş ve karşılaştırdıkları algoritmalara göre daha iyi sonuçlar elde etmişlerdir [73]. Moriya ve diğerleri, patoloji görüntülerinde bölütleme için hibrit bir denetimsiz öğrenme ve kümelenme yaklaşımı önermişlerdir. Yöntem iki aşamadan oluşmaktadır ve ilk aşamada, küresel K-ortalama kullanarak hedef görüntüden eğitim verilerini çıkartmışlardır. Bu aşamanın amacı, özellik çıkarma için filtre olarak kullanılabilecek küme merkezlerini elde etmektir. İkinci aşamada, merkez noktalarından çıkarılan temsillere geleneksel K-ortalama uygulamış ve ardından küme etiketlerini hedef görüntülere yansıtmışlardır. Yöntemlerini akciğer kanseri örneğinin patoloji görüntüleri üzerinde değerlendirmişler ve yapılan deneyler sonucunda önerilen yöntemin, geleneksel K-ortalama ve çoklu eşik Otsu yöntemini hem nicel hem de nitel olarak geride bıraktığını göstermişlerdir [74]. Modi ve arkadaşları gerçekleştirdikleri çalışmada, her bir algoritmanın çeşitli koşullar altında avantaj ve başarısızlıklarını test etmişler ve nihayetinde tümörün segmentasyonunda hangi algoritmanın mükemmel olduğunu keşfetmeye çalışmışlardır. K-ortalama ve FCM kümeleme algoritmalarını 35 görüntü üzerinde test etmiş ve sonuçları karşılaştırmışlardır. Elde ettikleri sonuçlara göre FCM %79,15 ve K-ortalama %94,72 oranında doğruluğa erişmiştir [75].

Literatürde Biyocoğrafya Tabanlı Optimizasyon (BBO) ile görüntü bölütleme üzerine çalışmalar bulunmaktadır. Bu çalışmalardan bazıları şu şekildedir: Chahal ve Sidhu, BBO algoritması kullanarak görüntü bölütleme işlemini gerçekleştirmişlerdir. İki farklı algoritma önermişlerdir. Birinci algoritmalarında, renkli bir görüntüde otomatik olarak piksellerine ayırmak için BBO tabanlı global bir optimizasyon algoritması önermişlerdir. İkinci algoritmalarında ise, görüntünün hesaplama maliyetini azaltmak

için Bulanık K-ortalama kümeleme algoritmasıyla birleştirmişlerdir [76]. Gupta, Sandhu ve Mohan tarafından yapılan çalışmada ise renkli bir görüntüyü ayrık homojen bölgelere bölerek otomatik gruplama işlemi yapan sistemi BBO ile gerçekleştirmişlerdir [77]. BBO, uçak bakım sensörü seçimi, Yuga-Uda'nın anten tasarımı, kaotik sistemlerin parametre kestirimi, delik oluşturma süreci ve rezervuarın optimal çalışmasında verimli olmayan zamanların çözümlenmesi gibi pek çok mühendislik problemini en iyi duruma getirmede başarısını kanıtlamıştır [78]–[82]. Güçlü evrensel uygulanabilirliğe sahip bir Biyocoğrafya Tabanlı Optimizasyon (BBO) algoritması elde etmek için Zhang ve diğerleri, BBO ve Gray Wolf Optimizer algoritmalarını birleştirerek HBBOG adlı yeni bir hibrit algoritma önermişlerdir. BBO algoritmasında mutasyon operatöründen kurtulup küresel arama yeteneğini geliştirmek için göç operatörüne bir diferansiyel mutasyon operasyonu eklemişler, orijinal arama işlemi yerel arama yeteneğini geliştirmek için çoklu taşıma işlemiyle değiştirmişlerdir. Gray Wolf Optimizer için, muhalif öğrenme tabanlı yaklaşımı, algoritmanın yerel optima'ya bir dereceye kadar düşmesini önlemek için birleştirilmiştir. Ardından, geliştirilmiş BBO ve muhalif öğrenme tabanlı Gray Wolf Optimizer, HBBOG'u formüle etmek için tek boyutlu ve tüm boyutlu alternatif strateji olarak adlandırılan yeni bir strateji ile hibrit hale getirilmiştir. Çeşitli test fonksiyonları ve CEC2014 test seti üzerinde çok sayıda deney gerçekleştirmiş ve HBBOG'u kümeleme optimizasyonuna uygulamışlardır. Deneysel sonuçlar, HBBOG'un son teknoloji ürünü algoritmalarından daha iyi performans elde etmiştir [83]. Adalarasan ve Malathi, göz küresi fundusunun sayısal olarak görselleştirilmesi ve otomatik bir görüntü analizi algoritması geliştirmek için BBO algoritmasını damar içeren ve içermeyen pikselleri ayırmakta kullanmışlardır [84].

Pham ve arkadaşları, manyetik rezonans görüntüleme beyin görüntülerinin bölütlenmesi için yeni bir kümeleme yöntemi sunmuşlardır. İlk olarak, Yerel Uzamsal Bilgi ve Önyargı Düzeltmesi ile çekirdekleştirilmiş bulanık entropi kümelenmesini kullanan yeni bir amaç fonksiyonu tasarlamışlardır. Daha sonra MR beyin görüntülerini daha iyi bölütlere ayırmak için yeni amaç fonksiyonu ile geliştirilmiş bir parçacık sürüsü optimizasyonuna dayanan yeni bir algoritma önermişlerdir. Önerilen algoritmanın performansını test etmek için, McConnell Beyin Görüntüleme Merkezinden simüle edilmiş MR beyin görüntülerini ve İnternet Beyin Segmentasyon Havuzundan gerçek MR beyin görüntülerini içeren birkaç kıyaslama görüntüsü kullanmışlardır. Ek olarak, önerilen algoritmanın, diğer beş tekniğe kıyasla sistematik bir karşılaştırması

sunulmuştur. Deneyisel sonuçlar, önerilen algoritmanın gürültü ve yoğunluk homojenliği olmayan görüntüler için tatmin edici bir performans sağlayabildiğini ve rakiplerinden daha iyi sonuçlar verdiğini göstermiştir [85]. Kumar ve arkadaşları, piksel-piksel değerlerine dayanan bir amaç fonksiyonu önermiş ve bu değerleri, Diferansiyel Evrim, PSO ve Genetik Algoritmalar gibi evrimsel algoritmalar aracılığıyla optimize etmişlerdir [86]. Na ve arkadaşları, medikal görüntü bölütlemesinin etkinliğini arttırmak için, Dinamik Atalet Ağırlıklı PSO algoritması ile optimal eşiği seçmek için bir yöntem önermişlerdir. Deneme verileri, Dinamik Atalet Ağırlıklı PSO'nun hesaplama verimliliği, görüntü eşgüdümünün etkisinin yanı sıra optimum eşik aramanın kesinliği ve stabilitesinde bariz bir avantajı olduğunu göstermiştir [87].

Darwinci Parçacık Sürü Optimizasyonu (DPSO) ilk olarak Tillet ve arkadaşları [88] tarafından PSO algoritmasını kullanarak daha iyi bir doğal seçim modeli arayışı sırasında ortaya atılmıştır. Algoritmanın yakınsaklık oranını kontrol etmek için DPSO'nun fraksiyonel hesabı kullanılarak bir uzantısı Couceiro ve arkadaşları [89] tarafından, Kesirli Düzen DPSO (FODPSO) olarak önerilmiştir. Yeni algoritma, Pires ve arkadaşları tarafından geleneksel DPSO'dan alınan Kesirli Düzen PSO ile başarılı bir şekilde karşılaştırılmıştır [90]. Hammouche ve arkadaşları, PSO tabanlı segmentasyon yöntemini Genetik Algoritma, Diferansiyel Değerlendirme, Karınca kolonisi optimizasyonu, benzetimli tavlama, tabu arama gibi diğer algoritmaları, çalışma zamanı, doğruluk ve hassasiyet gibi parametrelere dayalı olarak detaylıca karşılaştırmışlardır [91]. Sathya ve Kayalvizhi, PSO'nun Kapur ve Otsu fonksiyonları için CPU zamanı ve uygunluk değeri açısından Genetik Algoritma'nın üzerinde performans sergilediğini göstermiştir [92]. Jiang ve arkadaşları, yaptıkları çalışma sonuçlarıyla, PSO ailesi yöntemlerinin farklı ölçümlerde bir öğrenme operatörü ile Genetik Algoritma'dan daha iyi olduğunu göstermektedir [93]. Kulkarni ve Venayagamoorthy, PSO'nun BF ve ayrıntılı yöntemlerden önemli ölçüde daha hızlı olduğunu göstermişlerdir [94]. Görüntü işleme yöntemleri ile birlikte optimizasyon algoritmaları, kolay algı ve analiz için uydu görüntülerinde nesneleri sınıflandırmak için kullanılır. Venkata ve arkadaşları, bir uydu görüntüsünde nesnelerin en uygun sınıflandırmasını elde etmek için PSO, DPSO, HSO ve önerilen MFA optimizasyon algoritmaları gibi çeşitli optimizasyon tekniklerini karşılaştırmışlardır [95]. Murinto ve arkadaşları, hayvan görüntü segmentasyonu için DPSO yöntemini kullanmışlardır. Sonuçlar daha sonra Harmonik Arama Algoritması yöntemi ile karşılaştırılmıştır. Kullanılan segmentasyon yönteminin kalitesini ölçmek

için PSNR değerleri kullanılmıştır. DeneySEL sonuçlardan, bu çalışmada kullanılan DPSO ve Harmonik Arama Algoritması yöntemleri arasında çok az fark olduğu gözlemlenmiştir. Bununla birlikte, DPSO yönteminin bazı hayvan vakalarında Harmonik Arama Algoritması'ndan biraz daha iyi olduğu sonucuna varmışlardır [96].

FODPSO, Couceiro ve arkadaşları [89] tarafından sunulmuştur ve algoritmanın yakınsama oranını kontrol etmek için fraksiyonel analizin kullanıldığı DPSO'nun bir uzantısıdır. Fraksiyonel hesap birçok araştırmacının ilgilendiği bir konu olup hesaplamalı matematik, akışkanlar mekaniği ve diğer birçok farklı mühendislik dalında kullanılmaktadır [90], [97]. Akbar ve arkadaşları, FODPSO algoritmasını, monostatik Çoklu Giriş ve Çoklu Çıkış radarı senaryosunu temsil eden düzgün doğrusal diziye çarparak düzlem dalgalarının gelişi ve genlik tahminine yönelik bir metod tasarlamışlardır. Sistem modelinin tasarım parametrelerini, FODPSO algoritmasının gücünden yararlanarak optimize etmişlerdir. Önerilen FODPSO tabanlı optimizasyon mekanizmasının değeri, monostatik Çoklu Giriş ve Çoklu Çıkış radar sistemleri için her üç senaryoda tutarlı bir şekilde en uygun performans metrikleri değerlerine ulaşarak oluşturulmuştur [98]. Lahmiri, beyin MR görüntülerinde gliomayı saptamak için üç otomatik teşhis sistemini karşılaştırmıştır. Beyin MR görüntüsünü bölmek için PSO, DPSO ya da FODPSO algoritmalarını kullanmıştır [99]. Fu ve arkadaşları, kooperatif hedefin özellik noktalarının tanınmasını sağlamak için FODPSO algoritmasına dayalı bir görüntü bölütleme yöntemi önerilmiştir. Bu özellik noktalarının kooperatif hedeflerinin koordinatları, yakalanan görüntülerden bölgesel elips alanlarının merkezini çıkararak elde edilmiştir [100]. Sandeli ve Batouche, genelleştirilmiş ada modeline dayalı çok düzeyli eşikleme ile görüntü bölütlemesi için yeni bir yöntem sunmuşlardır. Bu model, PSO, Genetik Algoritma ve Yapay Arı Kolonisi olmak üzere üç metaheuristik işbirliği ile karakterize edilmiştir. İş birliği, yerel optimada takılıp kalmamak için bir göç paterninin kullanılması ile sağlanmıştır. Önerilen yöntemin etkinliği değerlendirmek için ayrı ayrı kullanıldığında GIM modelinin algoritmalarının her biri ile karşılaştırılmış ve DPSO, FODPSO gibi literatürden diğer yöntemler kullanılmıştır. Önerilen yöntemin diğer yöntemlere göre daha iyi sonuçlar verdiği sonucuna ulaşmışlardır [101].

1.4. TEZ ORGANİZASYONU

Tezin materyal bölümünde, bu çalışmada kullanılan örnek görüntüler ve test için kullanılan veriler hakkında bilgi verilmiş, yöntem bölümünde işlem adımları sırasına göre çalışmada kullanılan yöntemler ayrıntılarıyla anlatılmıştır. Ayrıca geliştirilen yöntemler hakkında bilgi verilmiş ve çalışma için geliştirilen MATLAB™ GUI arayüzü ekran görüntüleriyle tanıtılmıştır. Bulgular bölümünde her bir işlem adımında elde edilen sonuç görüntüler verilmiş, gerçekleştirilen testler sonucunda elde edilen veriler tablolar halinde sunulmuştur. Sonuç bölümünde bulgulardan elde edilen sonuçlar tartışılmış ve bundan sonraki çalışmalar için önerilerde bulunulmuştur.



2. MATERYAL VE YÖNTEM

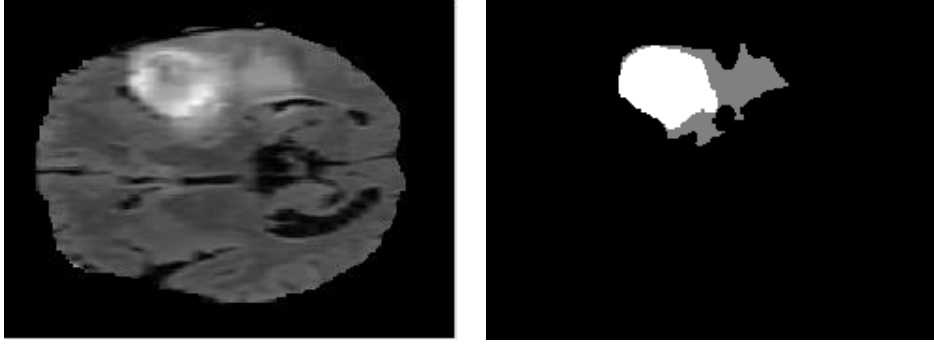
Bu bölümde öncelikle Histogram Tabanlı Küme Kestirimi (HBCE – Histogram Based Cluster Estimation) ve bölütleme algoritmalarının geliştirilmesinde ve testlerinde kullanılan insan bölütleme sonuçlarına sahip veritabanlarından bahsedilmiştir. Yöntem bölümünde ise bölütleme işleminin gerçekleştirilebilmesi için gerekli olan adımlar sırasıyla anlatılmıştır.

2.1. MATERYAL

Bölütleme işlemi için kullanılacak algoritmanın geliştirilmesi aşamasında Multimodal Beyin Tümörü Görüntü Bölütleme Karşılaştırması (BRATS - Multimodal Brain Tumor Image Segmentation Benchmark) 2012 veritabanından faydalanılmış sonrasında geliştirilen algoritmanın test aşamasında ise Uluslararası Cilt Görüntüleme İşbirliği (ISIC- The International Skin Imaging Collaboration) veritabanından alınan görüntüler kullanılmıştır.

2.1.1. Multimodal Beyin Tümörü Görüntü Bölütleme Karşılaştırması (BRATS - Multimodal Brain Tumor Image Segmentation Benchmark) 2012 Veritabanı

Multimodal Brain Tumor Image Segmentation Benchmark (BRATS) 2012 veritabanı, düşük ve yüksek dereceli glioma hastalarına ait 55 tanesi gerçek hastalardan alınmış, 65 tanesi sentetik olarak üretilmiş MR görüntülerinden oluşmaktadır. Bu 120 MR görüntüsünden 80 tanesinin insan segmentasyon sonuçları mevcut olup eğitim amaçlı kullanılmaktadır. BRATS veritabanından elde edilmiş MR görüntüsü ve bu görüntüye ait insan bölütleme sonucu Şekil 2.1’de verilmiştir.



Şekil 2.1. Glioma hastalığına ait MR görüntüsü ve insan bölütlemesi sonucu [53].

İnsan bölütlemesi sonuçları; gerçek hastalarda tümörlü bölge nekroz, kontrast tutan tümör, artmayan tümör ve ödem olmak üzere dört bölgeye ayrılmıştır. Sentetik datalarda ise tümör çekirdeği ve ödem olmak üzere iki bölgeye ayrılmıştır [102].

2.1.2. Uluslararası Cilt Görüntüleme İşbirliği (ISIC- The International Skin Imaging Collaboration) Veritabanı

ISIC, melanoma teşhisini iyileştirmek için dermoskopi görüntülerinden oluşan veri setlerini toplamayı ve herkes tarafından erişilebilir hale getirmeyi amaçlayan yakın zamanda başlamış uluslararası bir oluşumdur [55]. ISIC veri arşivindeki dermoskopik cilt görüntüleri uluslararası alanda önde gelen klinik merkezlerden toplanan 10.000'in üzerinde görüntüden oluşmaktadır. Resimler mahremiyet ve kalite güvencesi açısından ilişkili melanoma uzmanları tarafından incelenmiştir.

Bu oluşumun ana hedefi, dermoskopik görüntülerden otomatik melanoma teşhis algoritmalarının geliştirilmesini desteklemek için ISIC Arşivinden "anlık görüntü" sağlamaktır. Problem lezyon analizinin her aşamasına karşılık gelen 3 kısma bölünmüştür: lezyon segmentasyonu, lezyon dermoskopik özellik tespiti ve lezyon sınıflaması. İkinci ve üçüncü kısımlar ayrıca iki değişkenden oluşmaktadır: hastalık sınıflandırma ve maskeli hastalık sınıflandırma [103].

2.1.2.1. Lezyon Segmentasyonu

Bu görev, dermoskopik görüntülerden lezyon bölütlemelerinin otomatik maskeler şeklinde tahminini içermektedir. Lezyon bölütleme eğitim verileri uzman kişilerce önceden bölütlenmiş ve ikili bir maske biçiminde kaydedilmiş görüntülerden oluşmaktadır. Lezyonlu pikseller "1" ve lezyon dışında kalan diğer pikseller "0" ile ifade edilmektedir. Şekil 2.2'de örnek bir lezyonlu bölge ve 0-1'lerden oluşan insan bölütlemesi sonucu verilmiştir. Eğitim verileri, 900 görüntü ve bunlara ait insan

bölütleme sonuçlarını içermektedir.



Şekil 2.2. Dermoskopik görüntü ve insan bölütleme sonucu.

2.1.2.2. Dermoskopik Özellik Sınıflandırma

Bu görev, klinik olarak tanımlanmış dermoskopik özellik olan globüller ve çizgilerin otomatik olarak belirlenmesini içermektedir. Desen tespiti hem lokalizasyonu hem de sınıflandırmayı içermektedir. Mekansal özellik notlarının değişkenlik ve boyutsallığını azaltmak için, lezyon görüntüleri SLIC algoritması kullanılarak süper piksellere bölünmüştür [104]. Lezyon dermoskopik özellik verileri orijinal lezyon görüntüsünü ve buna karşılık gelen bir süper-piksel maskesini içermekte ve "globül" ve "çizgiler" dermoskopik özelliklerin varlığı ve yokluğunun süper-piksel uzman ek açıklamaları ile eşleştirilmektedir. Eğitim verileri, 335'i test veri seti olarak belirlenmiş 807 görüntüden oluşmaktadır.

2.1.2.3. Dermoskopik Özellik Bölütleme

Bu bölüm, tahminlerin her dermoskopik özellik için ikili maskeler biçiminde olması dışında, dermoskopik özellik sınıflandırma ile aynıdır. Bu ek bölüm, lezyon dermoskopik model algılamanın amacı için ikinci bir algoritma geliştirme ve değerlendirme mekanizmasını keşfetmek ve karşılaştırmak için sağlanmıştır.

2.1.2.4. Hastalık Sınıflandırma

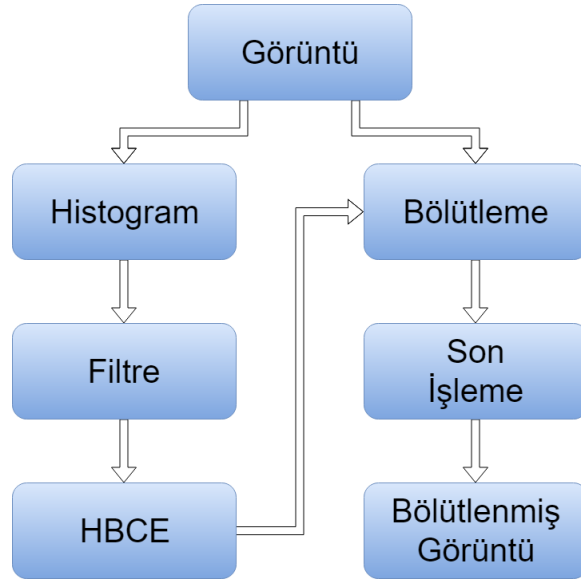
Bu görev, görüntülerin iyi huylu ya da hastalıklı olarak sınıflandırılmasına dayanmaktadır. Tahmin skorları "0" iyi huylu ve "1" hastalıklı olacak şekilde belirlenmiş ve skorlar bu aralığa göre normalize edilmiştir. Lezyon sınıflandırma verileri, hem kesin kötü huylu tanısı hem de insan bölütleme sonuçları ile eşleştirilmiş orijinal görüntüyü içermektedir. Eğitim için 900 görüntü ve ilgili insan bölütleme sonuçları kullanılmıştır.

2.1.2.5. Maskeli Hastalık Sınıflandırma

Bu bölüm, hastalık sınıflandırmaya ek olarak insan bölütleme sonuçlarından elde edilen görüntü maskeleri içermektedir.

2.2. YÖNTEM

Bölütleme işleminin sağlıklı bir şekilde gerçekleştirilebilmesi için bazı ön işleme adımlarından geçmesi gerekmektedir. Bu işlemlerden ilki histogram bilgisinin çıkartılmasıdır. Bölütlenmek üzere gelen görüntünün histogram bilgisi çıkartılarak filtreleme işlemi için hazırlanmış olur. Filtreleme işlemi, görüntüye bağlı olarak histogram üzerinde oluşan fazla salınım ve gürültüleri en aza indirmeyi hedeflemektedir. Bu hedef doğrultusunda herhangi bir filtre seçilmeyebilir, Ayrık Dalgacık Dönüşümü (DWT - Discrete Wavelet Transform) ya da Medyan filtre seçilebilmektedir. Filtreleme işleminden sonra elde edilen filtreli histogram bilgisi HBCE algoritmasına verilir. HBCE algoritmasında Dikey – Yatay (v-h – Vertical - Horizontal) ya da Yatay – Dikey (h-v – Horizontal - Vertical) yöntemlerinden biri ve gerekli görüldüğünde belirlenen bir eşik değeri seçilerek küme sayısı ve muhtemel küme merkezleri belirlenir. Elde edilen küme merkezi ve küme sayısı bilgisi görüntüyle birlikte bölütleme algoritmasına verilir ve bölütleme işlemi gerçekleştirilir. Bölütleme işlemi için K-ortalama, Biyocoğrafya Tabanlı Optimizasyon (BBO - Biogeography Based Optimisation), Parçacık Sürü Optimizasyonu (PSO), Darwinci Parçacık Sürü Optimizasyonu (DPSO), Kesirli Düzen Darwinci Parçacık Sürü Optimizasyonu (FODPSO) ve bu algoritmaların merkez tabanlı olarak düzenlenmiş halleri olan mPSO, mDPSO, mFODPSO algoritmalarından herhangi biri tercih edilir. Son işleme olarak elde edilen bölütlerde birleştirme ya da düzenleme işlemi gerçekleştirilir ve bölütlenmiş görüntü elde edilmiş olur. Bahsi geçen işlem adımları Şekil 2.3'te verilmiştir.



Şekil 2.3. İşlem adımları için akış şeması.

2.2.1. Filtreleme

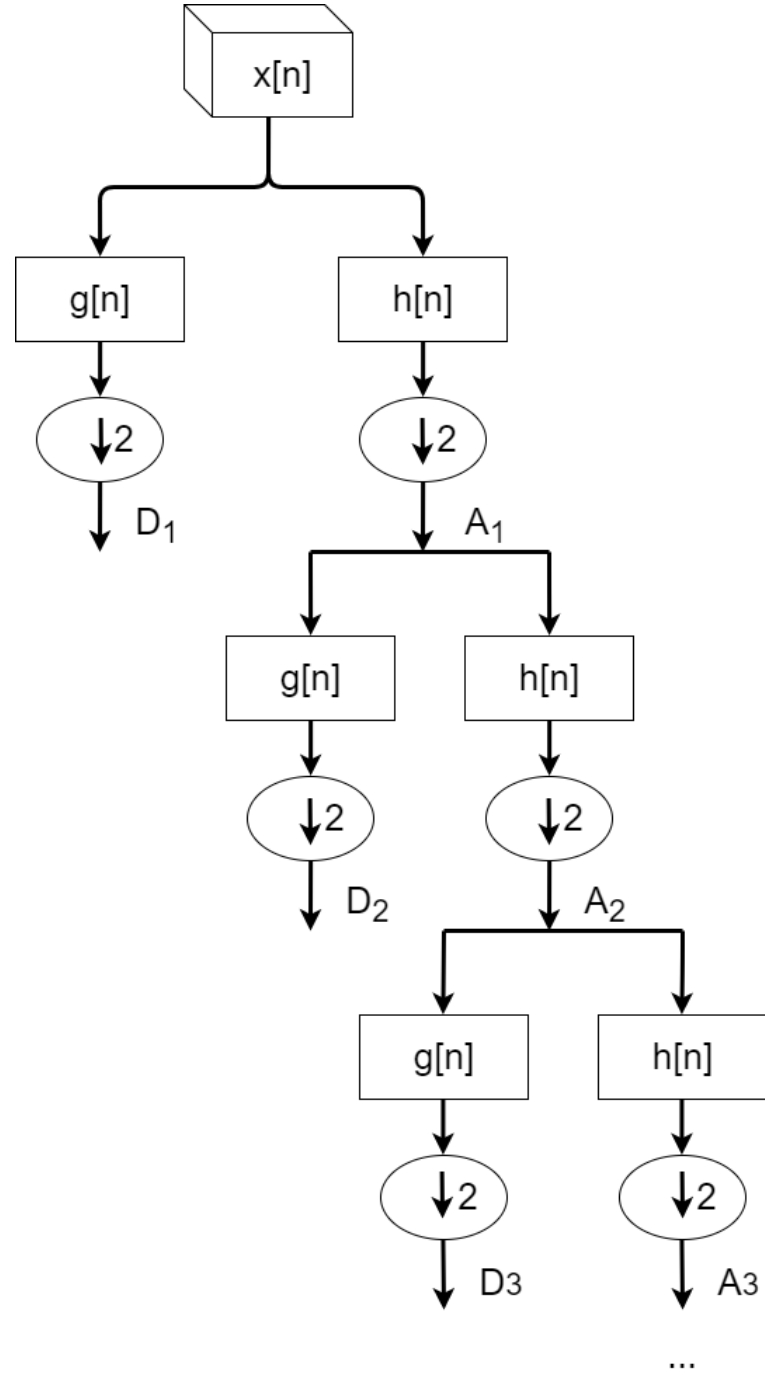
Sinyal işlemede, bir sinyalden, istenmeyen bazı bileşenleri veya özellikleri ortadan kaldırmak için kullanılan yöntemlere filtreleme denir. Süzme sinyal işlemenin bir sınıfıdır ve filtrenin tanımlayıcı özelliğidir. Süzme, sinyalin bazı yönlerinin tam veya kısmi baskılanmasıdır. Çoğu zaman bu bazı frekansları veya frekans bantlarını kaldırmak anlamına gelmektedir. Filtreler, elektronik ve telekomünikasyon, radyo, televizyon, ses kaydı, radar, kontrol sistemleri, müzik sentezi, görüntü işleme ve bilgisayar grafiklerinde yaygın olarak kullanılmaktadır.

2.2.1.1. Ayırık Dalgacık Dönüşümü (DWT – Discrete Wavelet Transform)

Fourier dönüşümü ile işaretlerin frekans analizi yapılırken, hangi zaman diliminde hangi frekansların yer aldığı belirlenemez. Kısa zamanlı fourier dönüşümü ise tüm zamanı eşit aralıklarla incelediği için değişken frekanslar içeren sinyallerin analizinde kullanışlı olamaz. Bu sebeple değişken zaman ölçeğine sahip bir dönüşüm yöntemine ihtiyaç duyulmuş ve dalgacık dönüşümü bu amaçla geliştirilmiştir. 1984 yılında Mallat tarafından önerilen çok çözünürlüklü sinyal analizi Dalgacık Dönüşümü olarak adlandırılmıştır [105].

Ayrık Dalgacık Dönüşümü (DWT – Discrete Wavelet Transform) yüksek frekanslar için dar, düşük frekanslar için geniş pencereler kullanarak tüm frekans aralıklarında optimum zaman frekans çözünürlüğü sağlayabilmektedir. Dalgacık Dönüşümünde işaret, çoklu çözünürlük analizi olarak isimlendirilen işlemle belirli sayıda alt bantlara

ayrılır. Şekil 2.4'te bir $x[n]$ işareti için alt bantlara ayırma işlemi verilmiştir. Burada $g[n]$ yüksek geçiren filtreyi, $h[n]$ alçak geçiren filtreyi temsil etmektedir. Yüksek geçiren filtreden çıkan D_1 , detay bandını ifade eder ve genel olarak yüksek frekans bileşenli gürültü katsayılarıdır. Alçak geçiren filtreden çıkan A_1 , yaklaşım bandını ifade eder ve giriş sinyalinden daha az gürültü içermektedir [106]. A_1 yaklaşım bandı, alt band seviyesi kadar tekrar ayrıştırılır ve işlemler tekrarlanır [107].

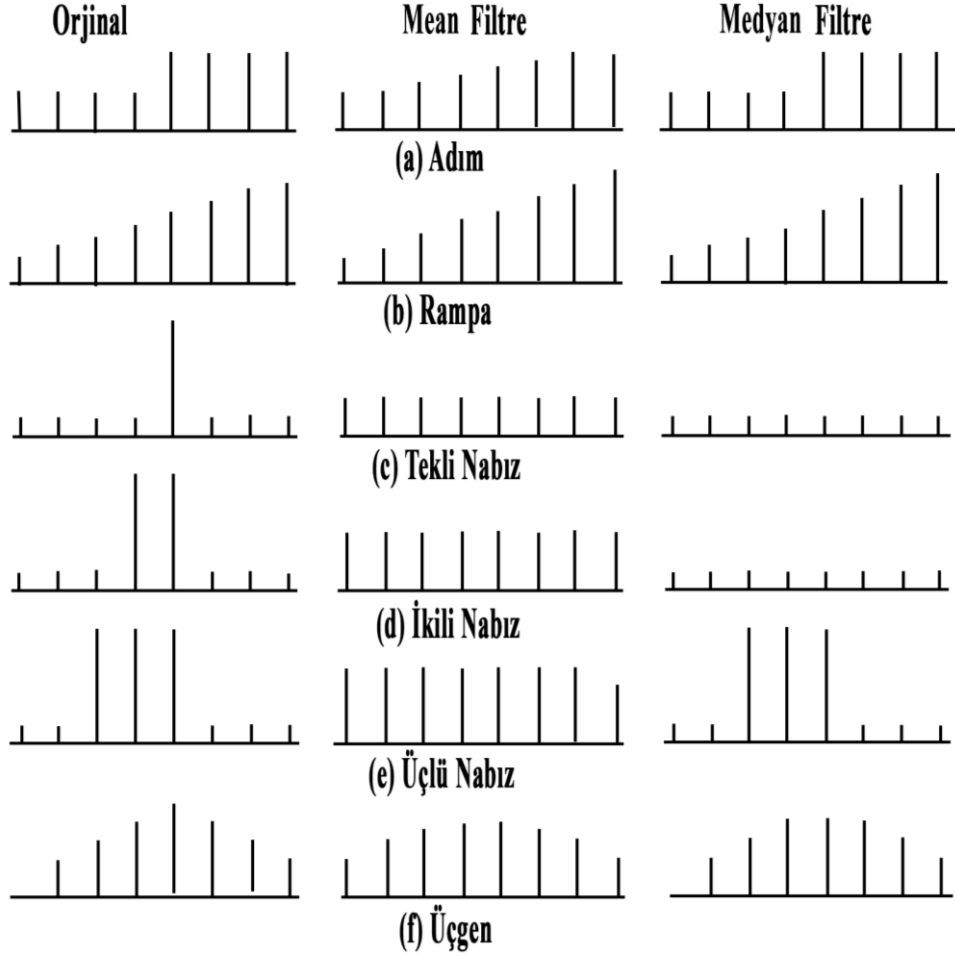


Şekil 2.4. Ayırık Dalgacık Dönüşümü alt bant ayrışımı.

2.2.1.2. Medyan Filtre

Medyan filtreleme, görüntülerde gürültü bastırma için yararlı olan ve Tukey [108] tarafından geliştirilen doğrusal olmayan bir sinyal işleme tekniğidir. Tek boyutlu formda medyan filtre, tek sayıda piksel içeren bir sürgülü pencereden oluşur. Penceredeki orta piksel, penceredeki piksellerin medyanı ile değiştirilir. N tek sayı olmak üzere ayrık bir dizinin ($a_1, a_2, \dots, a_N$) medyanı, dizinin $(N - 1) / 2$ ile tanımlanan elemanına küçük eşit ya da büyük eşit elemanı olarak belirlenir. Örneğin, bir pencerenin içindeki piksellerin değerleri (0,1, 0,2, 0,9, 0,4, 0,5) ise, merkez piksel, sıralanan dizinin medyan değeri olan 0,4 değeriyle değiştirilir (0,1, 0,2, 0,4, 0,5, 0,9). Bu örnekte, eğer 0,9 değeri, monotonik olarak artan bir sekansta bir gürültü artışı olsaydı, medyan filtre, kayda değer bir gelişme ile sonuçlanacaktı. Diğer taraftan, 0,9 değeri, geniş bantlı bir ışık sensörü için geçerli bir sinyal darbesini temsil edebilir ve sonuçta meydana gelen görüntü, bir miktar çözünürlük kaybına uğrayacaktır. Bu nedenle, bazı durumlarda, medyan filtre gürültü bastırma sağlarken, diğer durumlarda sinyal bastırmalarına neden olur.

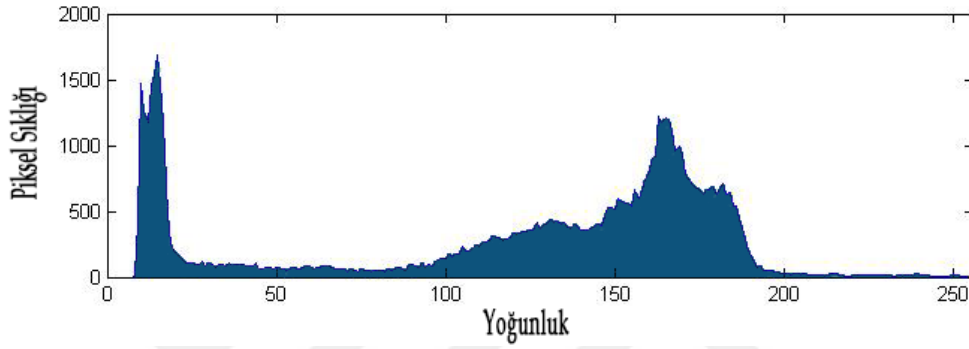
Medyan filtrenin çalışmasının bazı örnekleri ve mean filtre ile karşılaştırması Şekil 2.5'te verilmiştir. Örnekte rampa fonksiyonu, nabız fonksiyonu ve beş pikseli bir pencere ile üçgen fonksiyonu gösterilmektedir. Bu örneklerden, medyan filtrenin, adım fonksiyonlarını veya rampa fonksiyonlarını etkilememesinin genellikle arzu edilen özelliklere sahip olduğunu ifade ettiği söylenebilir. Süreleri, pencere genişliğinin yarısından daha az olan darbe işlevleri bastırılır. Fakat üçgenin tepesinin düzleştiği gözlemlenebilir [109].



Şekil 2.5. Medyan filtrenin tek boyutlu test sinyallerinde çalıştırılması [109].

2.2.2. Histogram Tabanlı Küme Kestirimi (HBCE – Histogram Based Cluster Estimation)

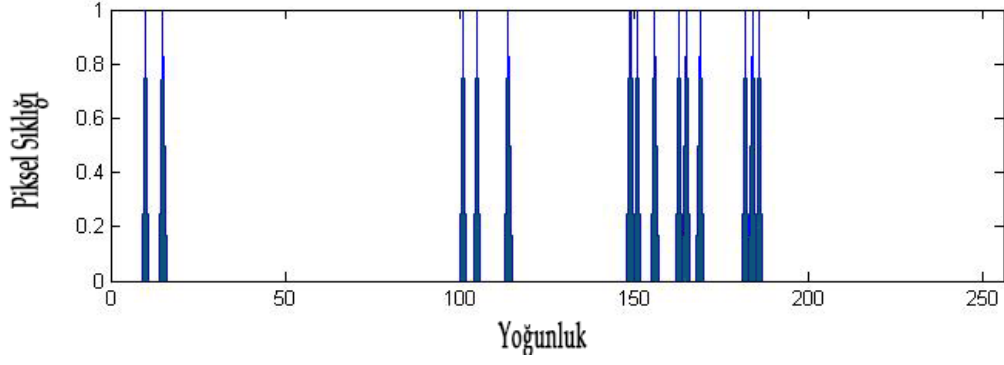
Kümeleme algoritmalarının işlevini doğru bir şekilde gerçekleştirebilmesi için görüntü bilgisinin yanı sıra görüntünün bölütleneceği küme sayısına da ihtiyaç duyulmaktadır. Gerçek hayat problemlerinde ise küme sayısına dair elimizde ön bir bilgi bulunmamaktadır [110]. Küme sayısının tespiti, veriseti hakkında bir önbilgiye sahip olmayı ya da çok uzun süren ve hata denetimini de içeren bir işlemi gerektirmektedir [50]. Bu tez çalışması kapsamında geliştirilen HBCE algoritması, görüntünün histogramını yorumlayarak küme sayısını kestirebilmektedir. Şekil 2.6’da kameraman görüntüsü ve bu görüntüye ait histogram bilgisi görünmektedir. Görüntünün histogramına bakarak görüntünün piksellerinin iki temel nokta etrafında yoğunlaşan renk değerlerine sahip olduğu söylenebilir.



Şekil 2.6. Kameraman görüntüsü ve bu görüntüye ait histogram.

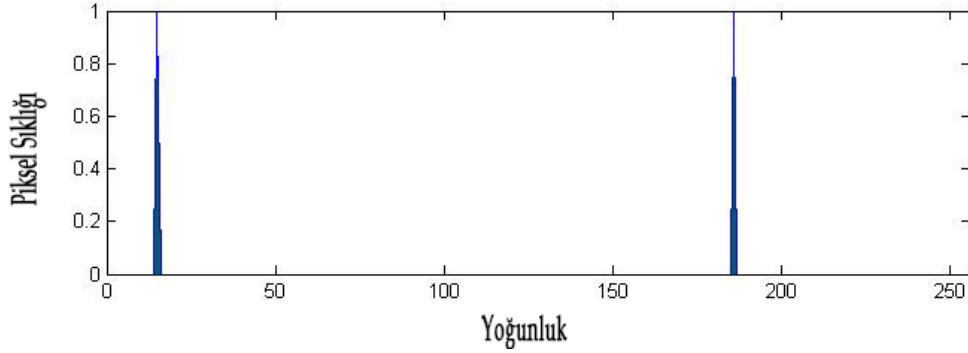
Histogram bilgisinin yorumlanabilmesi için öncelikle histogram üzerinde yükselen ve alçalan kenarların tespit edilmesi gerekmektedir. Böylelikle histogram üzerinde oluşan tepe ve çukur noktalarının işaretlenmesi ve bir sonraki işlem için hazırlanması sağlanmıştır.

Histogram üzerindeki tepelerin çukurlarla ve çukurların tepelerle olan mesafeleri hesaplanarak değişim miktarı tespit edilmiş ve ortalama değişim miktarı hesaplanmıştır. Ortalama değişim miktarının üstünde kalan değişim noktalarında, bu değişime neden olan tepe noktaları tespit edilmiştir. Ayrıca ortalama değeri için bir eşik değeri belirlemek de mümkündür. Bu şekilde ton farkı fazla olmayan görüntülerde bile küme belirleme işlemi gerçekleştirilebilmektedir. Şekil 2.7’de kameraman histogramındaki ortalama değişim miktarının üzerindeki tepe noktaları gösterilmiştir.



Şekil 2.7. Ortalama değişim miktarının üzerindeki tepe noktaları.

Histogramda yapılan incelemeyle tespit edilen tepe noktalarının birbirleri ile olan uzaklıkları hesaplanmıştır. Bu değerlerin ortalamalarından büyük olanlar kümeleme işleminde kullanılmak üzere küme başlangıç değerleri olarak, sayıları ise küme sayısı olarak kümeleme algoritmalarına gönderilebilmektedir. Şekil 2.8’de kümeleme algoritmalarına gönderilmek üzere belirlenen küme başlangıç değerleri ve küme sayısı görülebilmektedir.



Şekil 2.8. Algoritma sonunda belirlenen küme başlangıç değerleri.

Algoritmadan alınan değerlerin K-ortalama kümeleme algoritmasında kullanılmasıyla elde edilen sonuç Şekil 2.9’da görülmektedir.



Şekil 2.9. Bölütlenmiş kameraman görüntüsü.

Histogram yardımı ile küme merkezleri ve sayısının bulunması işlemi için kullanılan sözde kod Şekil 2.10’da verilmiştir.

```

    Img=getImg(İmgeyi al)
    Hist=histogram(img) (Histogramını elde et)
    'histogramdaki tepe ve çukur noktaların belirlenmesi
    FOR i=1:255
        IF değerlerde artıştan azalmaya geçildiyse ya da azalmadan artışa geçildiyse
            Tepe ya da Çukur Noktasını Kaydet
        END
    END
    'histogramın dikeyde taranması ve ortalamadan küçük değişimlerin
    'çıkartılması
    TepeveCukurlar'da her bir elemanın bir sonrakiyle arasındaki mesafelerin
    ortalamasını hesapla
    FOR i=1:length(TepeveCukurlar)
        IF tepe ve Çukurlardaki elemanın kendinden sonraki elemanla arasındaki mutlak
        fark ortalama*eşik'ten büyükse
            sadece tepe olan noktayı büyükTepeler'e kaydet
        END
    END
    'histogramın yatayda taranması ve ortalamadan küçük değişimlerin
    'çıkartılması
    Yerler=buyukTepeler elemanlarının histogramdaki yerlerini bul
    FOR i=1:length(yerler)
        toplam=her bir büyük tepe noktasının kendinden bir sonrakiyle olan mutlak
        mesafesi
    END
    ortalama=toplam/yerler'in uzunluğu
    sayac=1
    FOR i=1:length(yerler)
        IF her bir büyük tepe noktasının değerinin kendinden bir sonrakiyle olan
        değeri arasındaki mesafe ortalama*eşik'ten büyükse
            büyük olan değeri sonuç'a kaydet
        END
    END
    k= length(sonuç)
    kBaslangic=sonuç

```

Şekil 2.10. Histogram Tabanlı Küme Kestirimi (HBCE) sözde kodu.

Histogramın önce dikeyde sonra da yatayda taranmasıyla yapılan çalışma v-h olarak adlandırılırken, bu işlem önce yatayda gerçekleştirilip sonra dikeyde taramaya tabi tutulursa h-v şeklinde adlandırılmaktadır [111].

2.2.3. Bölütleme

Bölütlemenin amacı sadeleştirme, yani bir görüntüyü anlamlı ve kolay şekilde analiz edilebilir bir şekilde temsil etmektir. Görüntü bölütlemesi, görüntü analizinde ilk adımdır. Görüntü bölütlemenin ana amacı, bir görüntüyü benzer özelliklere sahip birkaç parçaya bölmektir. Görüntü bölütlemenin ana uygulamaları şunlardır: Tıbbi

görüntüleme, içerik tabanlı görüntü alma, otomatik trafik kontrol sistemleri, nesne algılama ve tanıma görevleri, vb.

Tezin bu kısmında kümeleme tabanlı bölütleme yöntemi olan K-ortalama, BBO ve Kümeleme ve eşik tabanlı yöntemlerin birleşimi olan PSO, DPSO ve FODPSO bölütleme yöntemleri incelenecektir.

2.2.3.1. *K-ortalama Algoritması*

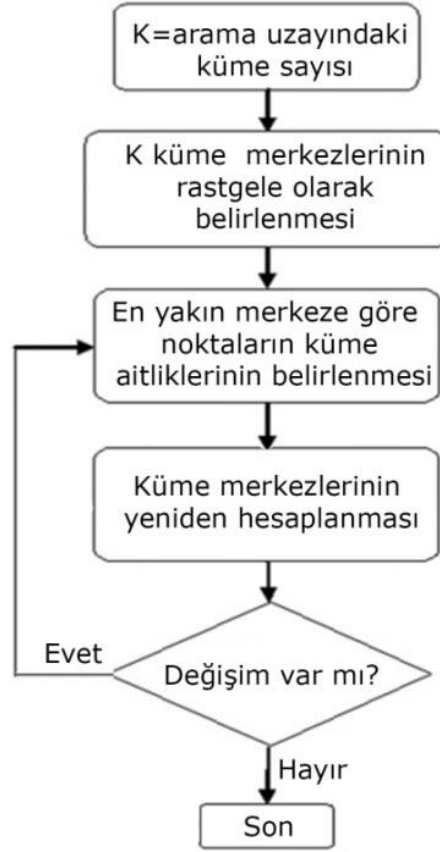
K-ortalama algoritması kümeleme problemlerinin çözümünde kullanılan en basit ve etkili yollardan bir tanesidir [112]. Macqueen [9] tarafından 1967 yılında önerilmiş bir algoritmadır. Bulanık C-ortalama ile karşılaştırıldığında daha az hesaplama karmaşıklığına sahip basit bir kümeleme metodudur.

Küme elemanlarının ağırlık merkezlerinin kümeyi temsil etmesi fikrine dayanan bir metottur [113]. K-ortalama algoritması, kullanıcı tarafından sisteme girilen ve n tane veriden oluşan veri kümesini gene kullanıcı tarafından girilen k adet kümeye böler. Küme aitliklerini nesnelerin küme merkezlerine olan uzaklıkları belirler. Kümeleme işleminde bir kümeye ait elemanların benzerlikleri fazla, kümeler arasındaki benzerlik ise az olmalıdır. Bu nedenle kullanıcı tarafından belirlenen küme sayısı dikkatli seçilmelidir.

K-ortalama kümeleme algoritmasının, işlevini hızlı bir şekilde gerçekleştirebilmesi için görüntü bilgisinin yanı sıra görüntünün bölütleneceği küme sayısına da ihtiyaç duymaktadır.

İyi bir başlangıç seti oluşturmak için belirli bir standart yoktur. Başlangıç değerlerinin rastgele olarak atanmasından ötürü her çalıştırılışında farklı sonuçlar alınabilmektedir.

K-ortalama kümeleme algoritması için akış şeması Şekil 2.11’de verilmiştir.



Şekil 2.11. K-ortalama kümeleme algoritması akış şeması [111].

K-ortalama, merkez tabanlı bir kümeleme algoritmasıdır. "k" küme sayısını temsil eder ve aynı zamanda bir giriş parametresidir. Veri kümesindeki her öge, kendisine en yakın küme merkezine atanır. Kümelenme görevini başarmak için iki temel adım vardır: birinci adımda doğru ya da yanlış olan küme merkezleri bulunur; ikinci adımda her bir eleman merkeze olan uzaklığına göre kendine en yakın kümeye atanır. Bu iki adım, algoritma tek bir optimum değere ulaşana kadar tekrarlanır.

K-ortalama kümeleme algoritması, uygun bir mesafe tanımı ve makul bir k değeri ayarlanması şartıyla, genellikle makul sonuçlara ulaşır. Şekil 2.12, $k = 3$ olan standart K-ortalama kümeleme algoritmasının sürecini göstermektedir.



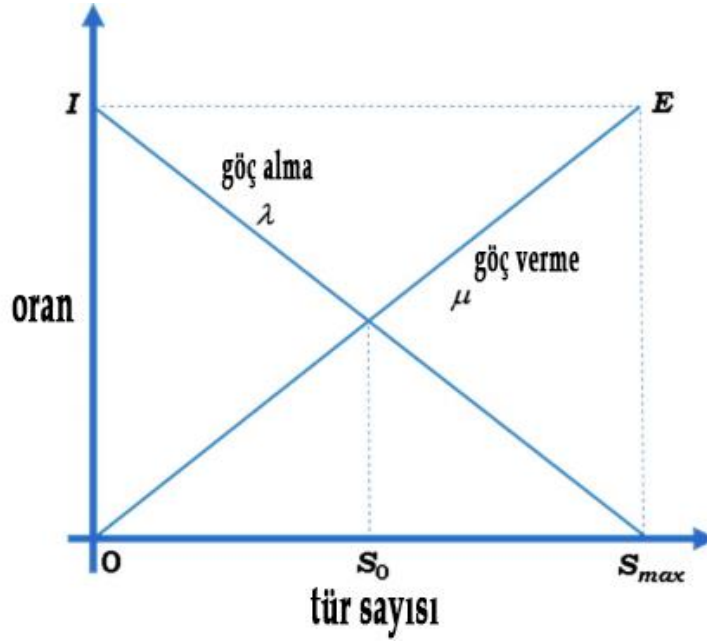
Şekil 2.12. K-ortalama kümeleme işlemi örneği [114].

2.2.3.2. Biyocoğrafya Tabanlı Optimizasyon Algoritması (BBO)

Biyocoğrafya, biyolojik organizmaların coğrafi dağılımını inceleyen popüler bir çalışma yöntemi olarak bilinir ve bu konudaki ilk eserler Alfred Wallace ve Charles Darwin tarafından verilmiştir. Çeşitli canlı türlerinin bir adadan başka bir adaya göçü, yeni türlerin gelişi ve mevcut bazı türlerin yok oluşunu tarif eden çeşitli matematiksel modelleri mevcuttur. Son zamanlarda, biyocoğrafyanın doğal yapısını esas alan ve nüfus tabanlı bir optimizasyon yöntemi olan Biyocoğrafya Tabanlı Optimizasyon (BBO) olarak adlandırılan yöntem geliştirilmiştir.

Biyocoğrafya modellerinde bir coğrafi alanın (ada veya yaşama alanı) uygunluğu, Habitat Uygunluk İndeksi (HSI) temelinde değerlendirilir. Yüksek HSI değerine sahip bir habitatın, canlı türleri için daha uygun olduğu; düşük HSI değerine sahip bir habitatın canlı türleri için daha az uygun olduğu şeklinde yorumlanır. HSI değerinin daha yüksek olduğu habitatlarda muhtemelen daha fazla sayıda canlı türü olacaktır. Bir habitatın yaşanabilirliğinin karakteristikleri olan yağış, bitki örtüsü, sıcaklık gibi değerler Uygunluk İndeks Değişkenleri (SIVs) olarak isimlendirilirler. Canlıların farklı habitatlar arasındaki hareket dinamikleri, içeriye göç alma (λ) ve dışarıya göç verme (μ) olarak iki değişken üzerinden incelenir ve hesaplanır. Şekil 2.13, tek bir habitatteki canlı türlerinin yaşam modelini göstermektedir. Şekildeki eğri, maksimum göç alma ve maksimum göç verme oranlarının eşit olduğu özel bir durumu göstermektedir. Ancak, gerçekte böyle bir sınırlama yoktur ve eşitlik durumu kolayca aşılabılır. Bu egrideki bir habitatın barındırabileceği maksimum tür sayısı (S_{max}) olarak isimlendirilmiştir. Bir habitatteki türlerin sayısı (S) küçük olduğunda, türlerin komşu habitatlardan buraya göç etme olasılığı daha fazla; buradan komşu habitatlara göç verme olasılığı ise daha düşüktür. Canlı türlerinin nüfustaki artışıyla beraber o habitatın göç alma olasılığı

düşerken; göç verme olasılığı yükselir. Göç alma ve göç verme durumları, Şekil 2.13'teki iki eğriyle ifade edilmektedir. Belirtilen bir zamanda $S=S_0$ olduğunda denge durumuna erişilmiş olup, bir habitata göç eden ve habitattan göç eden canlı türlerinin sayısının eşit olduğu değerlendirilir [115].



Şekil 2.13. Basit BBO'da dışa ve içe göç eğrileri [116].

Bir habitatın t zamanında S adet canlı türü içirme ihtimalinin P_s olduğu varsayılacak olursa, P_s aşağıdaki diferansiyel denklemlerle ifade edilir.

$$\dot{P}_s = \begin{cases} -(\lambda_s + \mu_s)P_s + \mu_{s+1}P_{s+1} & S = 0 \\ -(\lambda_s + \mu_s)P_s + \lambda_{s-1}P_{s-1} + \mu_{s+1}P_{s+1} & 1 \leq S \leq S_{max} - 1 \\ -(\lambda_s + \mu_s)P_s + \lambda_{s-1}P_{s-1} & S = S_{max} \end{cases} \quad (2.1)$$

Denklemdaki λ_s , habitatta S canlı türü olduğu andaki göç alma oranını; μ_s ise habitattaki S canlı türü olduğu andaki göç verme oranını ifade eder. $n=S_{max}$ olduğu varsayılacak olursa, habitatta S_{max} sayıda canlı türü olabilme olasılığının maksimum olduğu durumda habitattaki canlı türü sayısının olasılık denklemleri aşağıdaki matris ile ifade edilir.

$$\begin{bmatrix} \dot{P}_0 \\ \dot{P}_1 \\ \vdots \\ \dot{P}_{S_{max}} \end{bmatrix} = \begin{bmatrix} -(\lambda_0 + \mu_0) & \mu_1 & 0 & \cdots & 0 \\ \lambda_0 & -(\lambda_1 + \mu_1) & \mu_2 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \lambda_{n-2} & -(\lambda_{n-1} + \mu_{n-1}) & \mu_n \\ 0 & \cdots & 0 & \lambda_{n-1} & -(\lambda_n + \mu_n) \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{S_{max}} \end{bmatrix} \quad (2.2)$$

Biyocoğrafyanın bu temel prensipleri, birçok mühendislik ve optimizasyon problemlerinin çözümünde kullanılabilecek olan popülasyon tabanlı optimizasyon prosedürlerinin tasarımında kullanılmıştır. Biyocoğrafyanın matematiksel modeli ve popülasyon temelli optimizasyon algoritmalarının arasında birçok benzerlik bulunmasından dolayı, araştırmacıların doğadaki biyocoğrafya olayının çözümüne ilişkin bir optimizasyon yöntemi geliştirmeye ilgi duymaları kaçınılmazdı. Simon tarafından sunulan BBO algoritmasına göre her bir habitat (H) ($HcSIV^m$ olmak üzere) $m \times 1$ vektör çözümünde tasarlanmış ve her bir H 'nin m tane SIV'den oluştuğu varsayılmıştır. Her bir H habitatı için popülasyona dayalı algoritmalarındaki fitness fonksiyonuna karşılık gelen HSI değerleri bulunur. Daha yüksek HSI değerine sahip habitat, optimum çözüm için daha iyi bir aday olduğunu gösterir. Bir ekosistemin n tane habitat içerdiği düşünülür. Örneğin H^n adet habitat popülasyon tabanlı optimizasyon algoritmaları bağlamında, toplam olası n aday çözüm var demektir. Temel BBO algoritmasında, dışarıdan içeriye göç eden veya içeriden dışarıya göç edenlerin oranı Şekil 2.13'de görüldüğü gibi tür sayısı ile doğrusal olarak değişir ve aşağıdaki denklemlerdeki gibi ifade edilir:

$$\lambda_s = \lambda \times \left(1 - \frac{S}{S_{max}}\right) \quad (2.3)$$

$$\mu_s = \frac{\mu \times S}{S_{max}} \quad (2.4)$$

Şekil 2.14, temel BBO algoritmasını açıklar. 1. adımda kullanılacak olan maksimum jenerasyon sayısı (gen), S_{max} , λ , μ , maksimum göç alma oranı (m_{max}), elitizm (n_{elit}) gibi değişkenler tanımlanır ve başlangıç değerleri verilir. Bu adım aynı zamanda SIV'lere ve habitatlara yönelik problem çözümlerinin Denklem (2.1) ve (2.2) ile uyumlu bir şekilde hesaplama işlemlerinin planlanmasını gerektirir. Bu adımdan sonra, optimizasyon probleminin aday çözümlerine karşılık gelen bir dizi habitatın başlatılması yapılır. Bu rasgele başlatılan çözümler, büyük olasılıkla en son optimum çözümü üretmek için ince ayar gerektirecektir. Bu amaca ulaşmak için, habitatların nüfusu yeni yaşam alanlarını geliştirmek için “Göç” ve “Mutasyon” prosedürlerine tabidir. Yaşam alanlarını “Göç” prosedürüne tabi tutma ve sonrasında da “Mutasyon” prosedürüne tabi tutma işlemleri, sonlandırma durumuna ulaşana kadar nesilden nesile devam eder. Bu durum, maksimum jenerasyon sayısına ulaşılması veya fitness fonksiyonunun arzu edilen minimum değerini veren en iyi habitatın HSI değerini elde etmek olabilir. BBO algoritmasına ait sözde kod Şekil 2.14’te verilmiştir.



BAŞLA

S_{max} , λ , μ , m_{max} , gen , n_{elit} gibi BBO parametrelerine başlangıç değerlerini ver
Rastgele bir habitatlar kümesi başlat

REPEAT

FOR $i=1 - n$

H_i habitatının HSI_i değerlerini S_i , λ_s ve μ_s parametrelerine ata

END

n_{elit} sayıdaki en iyi olarak görülen habitatları korumak için elitizm uygula

FOR $i=1$ To $(n-n_{elit})$

λ_i ile orantılı olasılıklı habitat H_i 'yi seç

IF H_i seçildiyse

FOR $j=1 - n$

μ_j ile orantılı olasılıklı habitat H_j 'yi seç

IF H_j seçildiyse

H_j 'den rastgele bir $SIV(\alpha)$ seçin

H_i 'nin rastgele bir SIV 'sini α ile değiştir

END

END

Değişikliğe uğrayan H_i 'nin HSI_i 'sini tekrar hesapla

END

END

n_{elit} sayıdaki en iyi olarak görülen habitatları korumak için elitizm uygula

FOR $i=1$ to $(n-n_{elit})$

H_i habitatı için, tür sayısının olasılığı olan P_i 'yi, λ_i ve μ_i 'yi kullanarak

güncelle

FOR $j=1-m$

P_i 'ye oransal olasılıklı $H_i(j)$ 'nin SIV 'sini seç

IF $H_i(j)$ seçildiyse

$H_i(j)$ 'yi, rastgele üretilmiş olan $SIV(\beta)$ ile değiştir

END

END

Değişikliğe uğrayan H_i 'nin HSI_i 'sini tekrar hesapla

END

UNTIL sonlandırma kriterinin sağlanması

BİTİR

Şekil 2.14. BBO algoritmasına ait sözde kod.

Sonra her bir habitatın HST 'si, karşılık gelen tür sayısı S ve onun karşılık geldiği λ_s ve μ_s değerleriyle eşleştirilir. Habitatları bir "Göç" prosedürüne tabi tutmadaki temel amaç, her bir çözüm olasılığının içeri göç ve dışarı göç oranları, λ_s ve μ_s değerleri üzerinden habitatlar arasındaki bilgi paylaşımıdır. Burada elit olmayan habitatların her birinde bir habitat değişiklik olasılığı ($Phabit$) temelinde potansiyel olarak değişiklik yapılabilir. Her bir seçilen habitat için, bu habitatın SIV 'lerinden hangilerinin değiştirilmesi gerektiğine olasılıksal olarak karar verilir. Habitatın her SIV 'si için rastgele seçilmiş komşu habitatlardan (α) alınan bir SIV 'nin göç etme ve yer değiştirme işlemi yapılır.

Belirtildiği gibi, bu “Göç” prosedürü sadece aday olmayan elit habitat adayları için uygulanır. Bir elitizm prosedürü, önceki jenerasyondaki habitatların en iyilerinden bazılarını korumak için kullanılır. Bu, en iyi çözümlerin göç sürecinden dolayı bozulmasının önlenmesine yardımcı olur. “Göç” prosedürünün sonunda her bir habitat için HSI değerleri tekrar hesaplanır. Böylelikle, “Göç” prosedürünün uygun habitatlar/çözümler üretme konusunda yardımcı olup olmadığı ortaya çıkar.

"Göç" aşaması bittiğinde, elit olmayan her bir habitat potansiyel olarak “Mutasyon” prosedürüne tabi tutulabilir. Bu prosedürü uygulamak için ilk önce Denklem (2.1) ve (2.2)’yi kullanarak türlerin sayısının olasılıkları λ_s ve μ_s ’ler de kullanılarak hesaplanır. Mutasyon prosedürü, popülasyon içerisindeki çeşitliliğin artmasına neden olur ve dolayısıyla daha yeni ve uygun çözümlerin ortaya çıkması umut edilir. Şekil 2.13’ün derin analizi, canlı türü sayısının çok büyük veya çok küçük sayılara ulaşması olasılığının çok düşük olduğunu ortaya koymaktadır. Orta büyüklükte canlı türü sayısına ulaşılması durumu ihtimali çok daha fazladır. Bu nedenle düşük olasılıklı bir çözümün hayatta kalması pek olası değildir ve her ihtimalde yeni bir çözüme dönüşecektir. Öte yandan, yüksek olasılıklı bir çözümün mutasyona uğrama imkânı ise daha küçüktür. Bu durum mutasyon oranı $mut(S)$ olarak isimlendirilir ve Denklem (2.5)’deki gibi ifade edilir.

$$mut(S) = m_{max} \left(\frac{1-P_S}{P_{max}} \right) \quad (2.5)$$

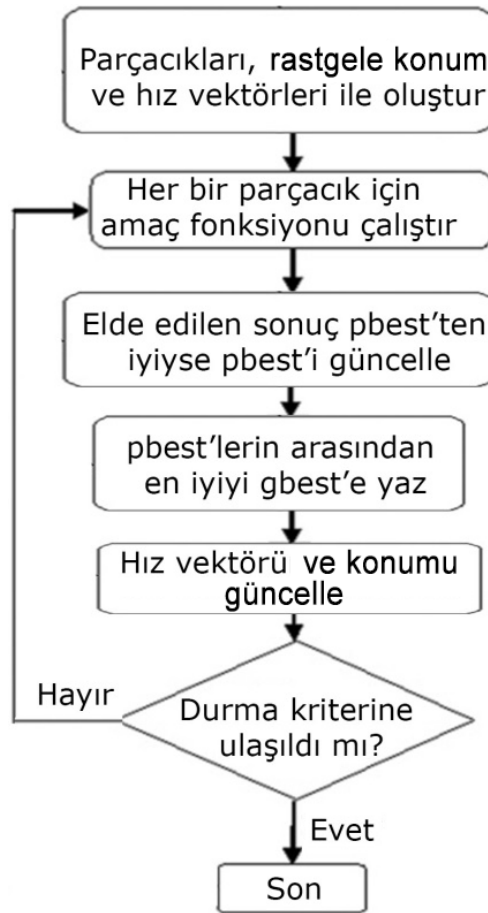
Burada, elit olmayan her habitat için, tür sayısının olasılığıyla orantılı bir olasılık temelinde habitatın rastgele bir SIV’si seçilir ve habitatın seçilen SIV’si rastgele üretilmiş (β) olan bir SIV ile yer değiştirilir. Bu işlem, elit olmayan her bir habitat için tekrarlanır. Yani “Göç” prosedüründe olduğu gibi “Mutasyon” prosedüründe de elitleştirme mekanizması uygulanmaktadır [116].

2.2.3.3. Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO) Kennedy ve Eberhart tarafından, kuş ve balık sürülerinin iki boyutlu hareketlerinden esinlenilerek geliştirilmiş popülasyon tabanlı bir stokastik optimizasyon tekniğidir [117]. Shi ve Eberhart tarafından optimizasyon modeline atalet ağırlığı eklenmiştir [118]. PSO temel olarak sürü zekası kavramından

yararlanır. Bir kuş sürüsündeki her bir kuşun bulunduğu yerdeki yem miktarı yoğunluğunda öttüğünü hayal edin. Aynı zamanda, her kuş, komşu kuşların konumunu algılayabilir ve komşu kuşlardan hangisinin en gürültülü öttüğünü söyleyebilir. Eğer her kuş şu üç yönergeyi takip ederse sürünün en yüksek yoğunlukta yeme sahip bir yer bulması için iyi bir şans vardır: (i) aynı yönde uçmaya devam et, (ii) şimdiye kadar en yüksek yem yoğunluğunun bulunduğu yere geri dön, (iii) en yüksek sesle öten komşu kuşa doğru ilerle [94].

Geleneksel PSO'da aday çözümlere parçacıklar denir. Bu parçacıklar en uygun çözümü bulmak için komşu parçacıklarla bilgi etkileşimi ve bilgi paylaşımı yaparak, arama alanına giderler. Ayrıca, prosedürün her aşamasında, tüm sürüde elde edilen en iyi küresel çözüm güncellenir. Tüm bu bilgileri kullanarak parçacıklar, başarının elde edildiği arama alanının yerini anlar ve bu başarılar tarafından yönlendirilir. Parçacık Sürü Optimizasyonu algoritması için akış şeması Şekil 2.15'te verilmiştir.



Şekil 2.15. Parçacık Sürü Optimizasyonu algoritması için akış şeması.

Algoritmanın her adımında, parçacık başarısını değerlendirmek için bir fitness fonksiyonu kullanılır. Sürüyü modellemek için, her bir parça n çok boyutlu uzayda yerel en iyi ($\tilde{x}_t^n$), komşu en iyi ($\tilde{n}_t^n$) ve küresel en iyi ($\tilde{g}_t^n$) bilgilerine bağlı konum (x_t^n) ve hız (v_t^n) bilgilerini kullanarak hareket ederler. Konum ve hız bilgileri için kullanılan genel formül Denklem (2.6) ve (2.7)'de verilmiştir [119].

$$v_{t+1}^n = wv_t^n + \rho_1 r_1 (\tilde{g}_t^n - x_t^n) + \rho_2 r_2 (\tilde{x}_t^n - x_t^n) + \rho_3 r_3 (\tilde{n}_t^n - x_t^n) \quad (2.6)$$

$$x_{t+1}^n = x_t^n + v_{t+1}^n \quad (2.7)$$

w , ρ_1 , ρ_2 ve ρ_3 katsayıları, yeni hızı belirlerken sırasıyla atalet etkisi, küresel en iyi, yerel en iyi ve komşu en iyi için ağırlıkları atar. Tipik olarak, atalet etkisi 1'den biraz daha düşük bir değere ayarlanır. ρ_1 , ρ_2 ve ρ_3 , "bilişsel" ve "sosyal" bileşenlerini temsil eden sabit tamsayı değerleridir. Bununla birlikte, her bileşen için farklı etkilerin tahsis edilmesiyle farklı sonuçlar elde edilebilir. Örneğin, bazı yöntemler komşu en iyiyi düşünmez ve ρ_3 sıfıra ayarlanır. Uygulamanın ve sorunun özelliklerine bağlı olarak, bu parametrelerin düzgün bir şekilde ayarlanması daha iyi sonuçlara yol açacaktır. r_1 , r_2 ve r_3 parametreleri, genellikle 0 ve 1 arasında rastgele sayılardır. Amaç, her bir parçacığın hız boyutuyla aynı bileşeni çarpmak yerine, hız boyutu başına yeni bir rastgele bileşen çarpmaaktır.

Başlangıçta, parçacıkların hızları sıfıra ayarlanır ve konumları, arama alanının sınırları içinde rastgele dağıtılır. Arama alanı, yoğunluk seviyelerinin sayısına bağlı olacaktır, yani, çerçeveler 8 bitlik görüntülerse, parçacıklar 0 ile 255 arasında dağıtılacaktır.

Yerel, komşu ve küresel en iyi değerler, problemin doğasını hesaba katarak mümkün olan en kötü değerlerle başlatılır. Ayarlanması gereken diğer parametreler, nüfus büyüklüğü ve durma kriterleridir. Kabul edilebilir bir zaman sınırı içinde genel olarak iyi bir çözüm elde etmek için popülasyon büyüklüğü çok önemlidir. Durma kriterleri, probleme bağlı olarak daha iyi sonuçlar ya da diğer kriterler elde edilmeden önceden tanımlanmış bir yineleme sayısı olabilir.

PSO, parçacıkların hızını ve sonuçtaki konumunu etkileyen komşu ve küresel bilgileri güncelleyerek parçacıklar arasındaki (yayıncılığa benzer) örtülü iletişimin bir etkisini

ortaya koymaktadır. Ayrıca, rastgele çarpanların (r_1 , r_2 ve r_3) tanıtımı nedeniyle stokastik bir keşif etkisi vardır. PSO, robotik [120]–[122], elektrik sistemleri [123], [124] gibi birçok uygulamada başarıyla kullanılmıştır.

Çok katmanlı bölütleme teknikleri, görüntü analizi yapmak için etkili bir yol sağlar. Bununla birlikte, sağlam bir optimum n-seviyesi eşiğin otomatik seçimi, görüntü bölütlemeye bir sorun olmaya devam etmektedir. Bu bölüm problemin daha kesin bir formülasyonunu sunmakta, bazı temel noktaları tanıtmaktadır.

L belirli bir görüntünün yoğunluk seviyeleri olsun ve bu seviyeler $\{0, 1, 2, \dots, L-1\}$ aralığında olsun.

$$P_i = \frac{h_i}{N}, \sum_{i=1}^N P_i = 1 \quad (2.8)$$

Burada i , belirli bir yoğunluk seviyesini temsil eder, yani $0 \leq i \leq L-1$; N , görüntüdeki toplam piksel sayısını temsil eder ve h_i yoğunluk seviyesi i için piksel sayısını gösterir. Başka bir deyişle, h_i , normalleştirilebilen ve olasılık dağılımı p_i olan görüntü histogramını temsil eder. Görüntünün toplam ortalaması (μ_T) aşağıdaki gibi kolayca hesaplanabilir:

$$\mu_T = \sum_{i=1}^L i P_i \quad (2.9)$$

İki seviyeli eşikleme, $n-1$ eşik seviyelerinin T_j , ($j = 1, \dots, n-1$) gerekli olduğu ve işlemin aşağıda belirtildiği gibi gerçekleştirildiği jenerik n seviyesinde eşiğe genişletilebilir:

$$F(x, y) = \begin{cases} 0, & f(x, y) \leq T_1 \\ \frac{1}{2}(T_1 + T_2), & T_1 < f(x, y) \leq T_2 \\ \vdots & \\ \frac{1}{2}(T_{n-2} + T_{n-1}), & T_{n-2} < f(x, y) \leq T_{n-1} \\ L, & f(x, y) > T_{n-1} \end{cases} \quad (2.10)$$

Burada x ve y , L yoğunluk seviyeleriyle $f(x, y)$ ile gösterilen $H \times W$ boyutundaki görüntünün genişlik (W) ve yükseklik (H) pikselleridir. Bu durumda, belirli bir görüntünün pikselleri, birden fazla nesneyi veya bu gibi nesneler üzerindeki belirli özellikleri (örn., Topolojik özellikler) temsil edebilen n sınıfa ($D_1, .., D_n$) bölünecektir.

Optimal eşiği (σ_B) elde etmenin en basit ve hesaplamalı olarak en verimli yöntemi, genellikle Denklem (2.11) tarafından tanımlanabilecek sınıflararası varyansı en üst düzeye çıkaran yöntemdir.

$$\sigma_B = \sum_{j=1}^n w_j (\mu_j - \mu_T)^2 \quad (2.11)$$

Burada j , w_j ve μ_j 'nin sırasıyla sınıf j 'nin oluşum ve ortalamasının olasılığı olduğu şekilde belirli bir sınıfı temsil eder. Sınıfların ($D_1, .., D_n$) ortaya çıkış olasılıkları Denklem (2.12)'de verilmektedir.

$$w_j = \begin{cases} \sum_{i=1}^{T_j} P_i, & j = 1 \\ \sum_{i=T_{j-1}+1}^{T_j} P_i, & 1 < j < n \\ \sum_{i=T_{j-1}+1}^L P_i, & j = n \end{cases} \quad (2.12)$$

Her sınıfın μ_j ortalaması şu şekilde hesaplanabilir:

$$\mu_j = \begin{cases} \sum_{i=1}^{T_j} \frac{iP_i}{w_j}, & j = 1 \\ \sum_{i=T_{j-1}+1}^{T_j} \frac{iP_i}{w_j}, & 1 < j < n \\ \sum_{i=T_{j-1}+1}^L \frac{iP_i}{w_j}, & j = n \end{cases} \quad (2.13)$$

Başka bir deyişle, n -seviyeli eşikleme problemi, genel olarak aşağıdaki gibi tanımlanan hedef işlevini (yani, fitness fonksiyonu) maksimize eden T_j eşiklerini aramak için bir optimizasyon problemine (φ) indirgenir:

$$\varphi = 1 < T_1 < \dots^{max} < T_{n-1} < L^{\sigma_B(T_j)} \quad (2.14)$$

Bu optimizasyon problemini hesaplamak, eşik seviyelerinin sayısı arttıkça çok daha büyük bir hesaplama çabası gerektirir [119].

2.2.3.4. *Darwinci Parçacık Sürü Optimizasyonu (DPSO)*

PSO ve diğer optimizasyon algoritmaları ile ilgili genel bir sorun, bir problem üzerinde iyi çalışabilmesi ancak başka bir problemde başarısız olabilecek şekilde yerel optimum noktada tuzağa düşebilmesidir. Bu sorunun üstesinden gelebilmek için pek çok araştırmacı bulanık mantığı PSO algoritması ile birleştiren başka ayarlamalar önermişlerdir. Pires ve arkadaşları PSO'nun yakınsama oranını kontrol etmek için fraksiyonel hesabı kullanmışlardır [125]. Yazarlar, hız türevinin sırasını değiştirmek için orijinal hız denklemini (2.6) yeniden düzenlemişlerdir. Alternatif olarak, birçok yazar, Diferansiyel Evrim'in yanında PSO algoritmasına seçim, mutasyon ve çapraz geçişi dahil etmeyi düşünmüştür. Burada temel amaç, ya parçacıkların birbirine çok yakın hareket etmesini engellemek, çarpışma ya da sıkıştırma faktörü, hızlanma sabitleri ya da eylemsizlik ağırlığı gibi parametrelere kendiliğinden adapte olmalarını engellemek suretiyle popülasyonun çeşitliliğini arttırmaktır [126].

PSO algoritmasını kullanarak doğal seçilimin daha iyi bir modelini ararken, Darwinci Parçacık Sürü Optimizasyonu (DPSO) formüle edilmiştir [88]. Her bir sürü sıradan bir PSO algoritması gibi ayrı ayrı doğal seleksiyonu simüle etmek için tasarlanmış sürülerin toplanmasını yöneten kurallar ile gerçekleştirilir. Rastgele oluşturulmuş popülasyon, fitness fonksiyon değerlendirmesi, popülasyon güncellemesi, rastgele tekniklerle optimizasyon arayışı ve başarıyı garanti etmeme gibi genetik algoritmalar ile arasındaki benzerliklere rağmen; PSO, çaprazlama ve mutasyon gibi genetik operatörleri kullanmaz, dolayısıyla evrimsel bir teknik olarak kabul edilmez. Öte yandan DPSO, doğal seçilimin (Darwin'in en iyinin hayatta kalma prensibi), PSO algoritmasının yerel optimadan kaçma yeteneğini artırabileceği fikriyle PSO'yu genişletir. Buradaki fikir, aynı test problemi ve basit bir seçim mekanizması üzerinde, her biri farklı bir sürü ile eşzamanlı paralel PSO algoritmaları çalıştırmaktır. Bir arama yerel bir optimum duruma geldiğinde, o alandaki arama basitçe atılır ve bunun yerine başka bir alan aranır. Bu yaklaşımda, her adımda, daha iyi olan sürüler ödüllendirilir (partikül ömrünü uzatır) ve durgun kalan sürüler cezalandırılır (sürünün ömrünü kısaltır veya parçacıkları yok eder).

Her sürünün genel durumunu analiz etmek için, tüm parçacıkların zindeliği değerlendirilir ve parçacıkların her birinin komşusu ve bireysel en iyi konumları güncellenir. Yeni bir küresel çözüm bulunursa, yeni bir parçacık ortaya çıkar. Sürü, tanımlanmış sayıda adımda en iyi durumunu bulmazsa bir parçacık silinir. İşlem adımlarına ait sözde kod Şekil 2.16’da verilmiştir.

Ana program döngüsü	Sürü evrimi algoritması
FOR koleksiyondaki her bir sürü için Sürüyü evrimleştir (sağ tarafta) Sürünün çoğalmasına izin ver Başarısız sürüyü sil	FOR sürüdeki her bir parçacık için Parçacıkların fitness değerlerini güncelle Parçacıkların en iyi değerlerini güncelle Parçacığı hareket ettir IF sürü iyiye gidiyorsa Sürüyü ödüllendir: parçacığı çoğalt: sürü ömrünü uzat IF sürü iyileşmemişse Sürüyü cezalandır: parçacığı sil: sürü ömrünü azalt

Şekil 2.16. Darwinci Parçacık Sürü Optimizasyonu algoritması için sözde kod.

Bir sürüyü silmek, parçacıkları silmek ve yeni bir sürü ve yeni bir parçacık ortaya çıkarmak için bazı basit kurallara uyulur: (i) sürü popülasyonu minimum sınırın altına düştüğünde, sürü silinir ve (ii) sürüdeki en kötü performans parçacığı, fitness fonksiyonunu geliştirmeden maksimum eşik sayısına (SC^{max}) ulaşıldığında silinir. Parçacıkların silinmesinden sonra, sıfıra ayarlanmak yerine sayaç, Denklem (2.15)’e göre eşik sayısına yaklaşan bir değere ayarlanır.

$$SC(N_{kill}) = SC^{max} \left[1 - \frac{1}{N_{kill}+1} \right] \quad (2.15)$$

Burada N_{kill} , fitness fonksiyonunda hiçbir iyileşme olmadığı zaman sürüden silinen parçacık sayısını temsil eder. Yeni bir sürü türetmek için, sürünün hiç silinmemiş bir partikülü bulunmamalıdır ve maksimum sürü sayısı aşılmamalıdır. Yine de, yeni sürü, sadece bir $p = f / NS$ olasılığı ile oluşturulur. Burada f , $[0,1]$ aralığında rastgele bir sayıyı ve NS sürülerin sayısını temsil eder. Bu faktör, çok sayıda sürü bulunması durumunda yeni sürüler oluşturulmasını önler. Ebeveyn sürüsü doğal kalır ve ebeveynin parçacıklarının yarısı çocuk sürüsü içinden, diğer yarısı da sürü koleksiyonu içinden rastgele seçilir. Sürünün ilk popülasyon numarası elde edilmezse, parçacıkların geri kalanı rastgele başlatılır ve yeni sürüye eklenir. Bir sürünün yeni bir küresel en iyi

duruma ulaşması ve bir sürünün maksimum tanımlanmış popülasyonuna ulaşamaması durumunda bir parçacık ortaya çıkar. PSO gibi, birkaç parametrenin de algoritmayı verimli bir şekilde çalıştıracak şekilde ayarlanması gerekir: (i) ilk sürü popülasyonu; (ii) maksimum ve minimum sürü popülasyonu; (iii) başlangıç sürü sayısı; (iv) maksimum ve minimum sürü sayısı ve (v) gebelik eşiği. Daha önce incelenen problemlerde [124] ve geliştirilen robotik araştırma stratejileri geliştirmelerinde [123], DPSO, üstün performans gösteren PSO ile başarılı bir şekilde karşılaştırılmıştır [119].

2.2.3.5. Kesirli Düzen Darwinci Parçacık Sürü Optimizasyonu (FODPSO - Fractional Order Darwinian Partical Swarm Optimisation)

FODPSO, Couceiro ve arkadaşları tarafından sunulmuştur ve algoritmanın yakınsama oranını kontrol etmek için fraksiyonel analizin kullanıldığı DPSO'nun bir uzantısıdır [127]. Fraksiyonel hesap birçok araştırmacının ilgilendiği bir konu olup hesaplamalı matematik, akışkanlar mekaniği ve diğer birçok farklı mühendislik dalında kullanılmaktadır. Sinyalin $x(t), \alpha \in \mathbb{C}$ ile fraksiyonel diferansiyeli ilkesine dayalı Grünwald–Letnikov ayrık zaman uygulaması şu şekilde tanımlanır:

$$D^\alpha[x(t)] = \frac{1}{C^\alpha} \sum_{k=0}^r \frac{(-1)^k \Gamma(\alpha+1) x(t-kC)}{\Gamma(k+1) \Gamma(\alpha-k+1)} \quad (2.16)$$

Burada Γ gama fonksiyonu, C örnekleme periyodu ve r kesir dizisidir. Grünwald–Letnikov'un ortaya koyduğu önemli bir özellik, tamsayı–düzen türevi sadece sınırlı bir diziyi temsil ederken, kesirli–düzen türevinin sonsuz sayıda terim gerektirmesidir. Bu nedenle, tamsayı türevleri “yerel” işlemcilerdir; kesirli türevler ise, tüm geçmiş olayların bir “belleğini” içerir. Bununla birlikte, geçmiş olayların etkisi zamanla azalır.

Fraksiyonel hesapların ortaya koyduğu özellikler, içsel bellek özelliği nedeniyle geri dönüşümsüzlük ve kaos gibi olguları tanımlamak için çok uygundur. Bu düşünce çizgisinde, parçacığın yörüngesinin dinamik fenomeni, fraksiyonel hesap araçlarının yeterince uyduğu bir durumu yapılandırır.

Denklem (2.6)'nın atalet etkisinin, $w = 1$ olarak düşünüldüğünde ve $T = 1$ olduğu varsayılarak Pires ve ark. [90] çalışmasına benzer şekilde Denklem (2.17)'deki gibi tanımlanabilir.

$$D^\alpha[v_{t+1}^n] = \rho_1 r_1(\tilde{g}_t^n - x_t^n) + \rho_2 r_2(\tilde{x}_t^n - x_t^n) + \rho_3 r_3(\tilde{n}_t^n - x_t^n) \quad (2.17)$$

Couceiro ve arkadaşları [89] yaptığı ön çalışmalarda $r \geq 4$ durumu için benzer sonuçlar sunmuştur. Ayrıca, hesaplama gereklilikleri r ile doğrusal olarak artar, yani FODPSO $O(r)$ bellek karmaşıklığına sahiptir. Bu nedenle Denklem (2.16) ve (2.6)'da verilen, sadece ilk $r = 4$ diferansiyel türev terimi kullanılarak Denklem (2.18) yazılabilir.

$$v_{t+1}^n = \alpha v_t^n + \frac{1}{2} \alpha v_{t-1}^n + \frac{1}{6} \alpha (1 - \alpha) v_{t-2}^n + \frac{1}{24} \alpha (1 - \alpha) (2 - \alpha) v_{t-3}^n + \rho_1 r_1(\tilde{g}_t^n - x_t^n) + \rho_2 r_2(\tilde{x}_t^n - x_t^n) + \rho_3 r_3(\tilde{n}_t^n - x_t^n) \quad (2.18)$$

DPSO, $\alpha = 1$ (belleksiz) olduğunda FODPSO'nun özel bir durumu olarak kabul edilir. Dolayısıyla, α değeri atalet parçacıklarını büyük ölçüde etkiler. Küçük bir α değeri ile, parçacıklar önceki faaliyetlerini görmezden gelirler, böylece sistem dinamiklerini göz ardı ederler ve yerel çözümlere takılıp kalmaya yatkın olurlar. Öte yandan, büyük bir α değeri ile, parçacıklar yeni çözümler keşfetmeye izin veren daha da çeşitlendirilmiş bir davranış sergileyecek ve böylece uzun vadeli performans artırımı sağlanacaktır. Bununla birlikte, arama seviyesi çok yüksekse, algoritmanın global çözümü bulması çok zaman alabilir. Bu çalışmada, Couceiro ve arkadaşları [89]'nın deneysel sonuçları temel alınarak $\alpha = 0.6$ fraksiyonel katsayısı tercih edilmiştir [119].

2.2.3.6. Merkez Tabanlı mPSO, mDPSO ve mFODPSO

PSO, DPSO ve FODPSO bölütleme algoritmaları eşik değerleri ile çalışacak şekilde tasarlanmıştır. Eşik değerleri arasında kalan pikseller o eşik aralığına ait kümeye atanarak bölütleme işlemi gerçekleştirilmektedir. Bu işlem Denklem (2.10)'da verilen şekilde gerçekleştirilmiştir.

Bölütleme algoritmalarının çalıştırılmasıyla elde edilen değerlere histogramın minimum ve maksimum noktaları da eklenerek histogramın tamamının kapsanması sağlanmıştır. Sonrasında her bir limit aralığı için iki limit değeri arasında kalan histogram bölgesinin ortalama değeri, o bölgedeki histogram yoğunluğu da göz önünde bulundurularak aşağıdaki formülle yeniden hesaplanmıştır.

$$M_i = \frac{\sum_{j=T_i}^{T_{i+1}} j \times h_j}{\sum_{j=T_i}^{T_{i+1}} h_j} + T_i \quad (2.19)$$

$$F(x, y) = \begin{cases} M_1, & f(x, y) \leq M_1 \\ M_2, & M_2 < f(x, y) \leq M_3, \quad |f(x, y) - M_2| \leq |f(x, y) - M_3| \\ M_3, & M_2 < f(x, y) \leq M_3, \quad |f(x, y) - M_2| > |f(x, y) - M_3| \\ \vdots & \\ M_{n-1}, & f(x, y) > M_{n-1} \end{cases} \quad (2.20)$$

Burada j , histogram yoğunluğunu, h_j ise o yoğunluğun piksel sayısını temsil eder. M_i her bir limit aralığı için iki limit değeri arasında kalan histogram bölgesine ait ortalama değeri temsil ederken piksellerin değerleri $F(x,y)$ kendilerine en yakın ortalama değere atanmaktadır.

2.2.3.7. Bölütleme İşlemi için Başlangıç Değerlerinin Atanması

HBCE algoritması küme sayısı bilgisini elde etmeye çalışırken aynı zamanda histogram üzerinde belirgin tepe noktalarını arayarak tahmini küme merkezi noktalarını da tespit etmektedir. Bu tahmini küme merkez noktası bilgileri bölütleme algoritmalarına verilerek asıl küme merkezleri bulunurken algoritmaya öncülük etmesi sağlanabilir. Bölütleme algoritmaları çalıştırılırken küme merkezi bilgisi olmadan (a), sadece bir parçacık için küme merkezi bilgisi ile (b), tüm parçacıklar için küme merkezi bilgisi verilerek (c) ve belirlenen ortalama aralıkta rastgele verilecek şekilde (d) tanımlamalar yapılmıştır.

$$E_i = \frac{\sum_{j=H_i}^{H_{i+1}} j \times h_j}{\sum_{j=H_i}^{H_{i+1}} h_j} + H_i \quad (2.21)$$

$$R(1..n) = \begin{cases} rand(0,255) & , a \\ R(1) = E_1, R(2..n) = rand(0,255) & , b \\ E & , c \\ rand(H_i, H_{i+1}), i = 1..(n-1) & , d \end{cases} \quad (2.22)$$

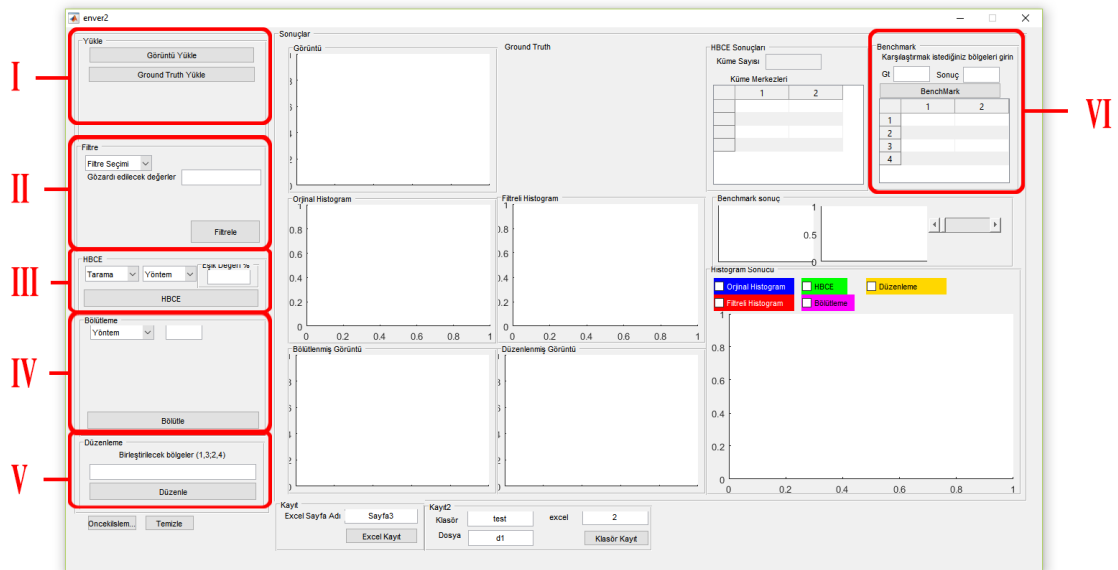
Burada H_i HBCE algoritmasından alınan tahmini küme merkezi bilgisini temsil ederken, E_i bu değere göre hesaplanmış tahmini eşik değerini ifade etmektedir. R ise parçacıklara ait başlangıç noktalarını ifade etmektedir.

2.2.4. Son İşleme

Önceki adımlarda elde edilen bölütlenmiş görüntü, karşılaştırma işlemleri ya da sonuç çıkartımı için son bir işleme ihtiyaç duyulabilmektedir. Bu adımda birleştirilmesi planlanan bölgeler işaretlenerek birleştirme işlemi gerçekleştirilir. Aynı zamanda karşılaştırma işlemi yapılacaksa hangi bölgelerin birbiri ile karşılaştırılacağı da bu adımda belirlenir.

2.2.5. Kullanıcı Arayüzü

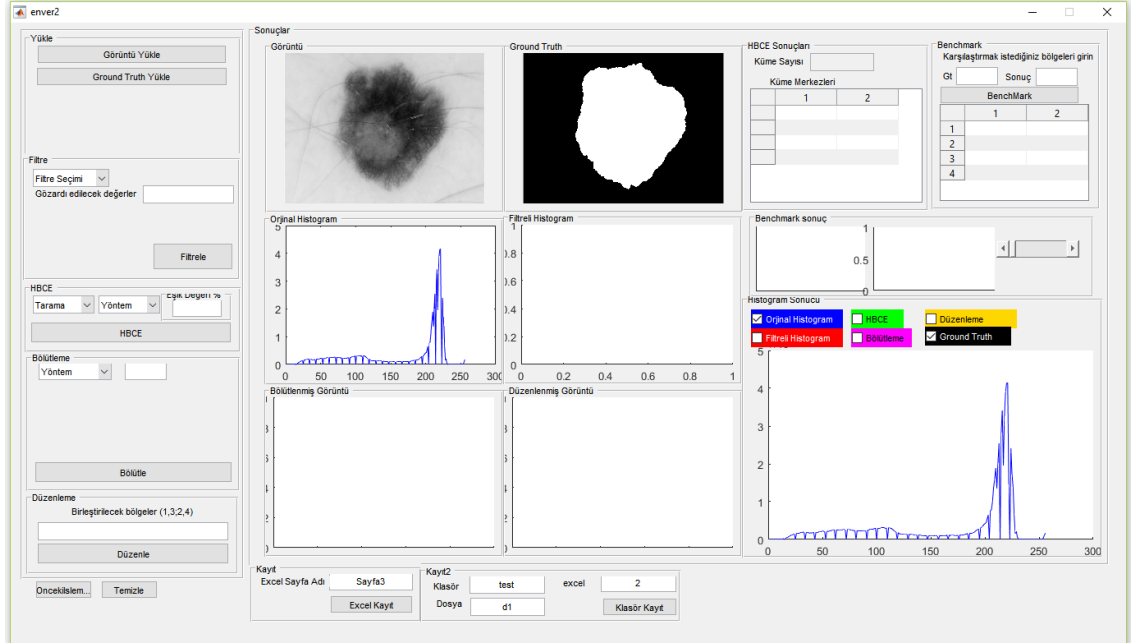
Çalışmanın hızlandırılması ve görselleştirilmesi için MATLAB™ ortamında bir arayüz hazırlanmıştır. MATLAB™ 2015 ortamında hazırlanan arayüz Şekil 2.17’de verilmiştir. Şekil üzerinde I. Bölüm Görüntü ve varsa insan bölütleme sonucunun yüklenmesi, II. Bölüm Filtre seçimi, III. Bölüm HBCE algoritmasının uygulanması, IV. Bölüm Bölütleme işlemi, V. Bölüm Düzenleme işlemi ve VI. Bölüm varsa insan bölütleme sonucu ile karşılaştırma işlemleri için kullanıcı seçim kutucuklarıdır.



Şekil 2.17. MATLAB™ GUI ortamında tasarlanmış kullanıcı arayüzü ekranı.

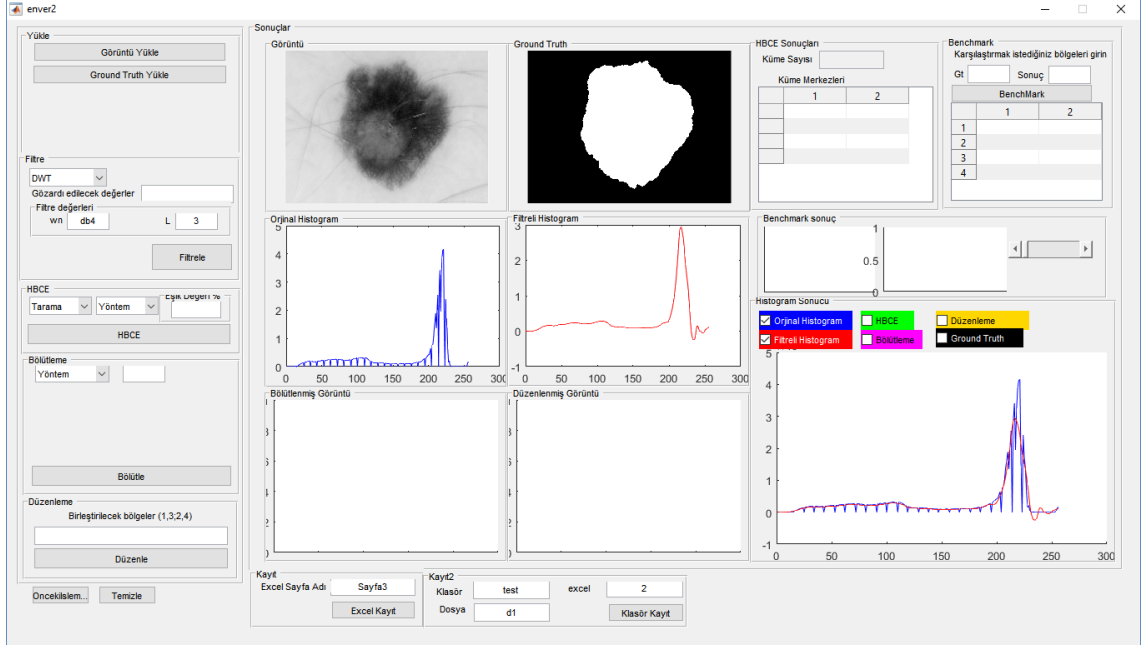
I. Bölümde, Arayüz üzerinden önce görüntü ve varsa insan bölütleme sonucu seçilir. Görüntüler (*.jpg, *.png, *.mha, *.dcm) formatlarında olabilmektedir. Görüntü ve insan

bölütleme sonucu seçildiğinde Sonuçlar panelinde Görüntü, insan bölütleme sonucu ve Orjinal Histogram bölümleri doldurulmaktadır. Aynı zamanda Histogram Sonucu paneli Orjinal görüntü histogramıyla doldurulur. Bu aşamaya ait ekran görüntüsü Şekil 2.18’de verilmiştir.



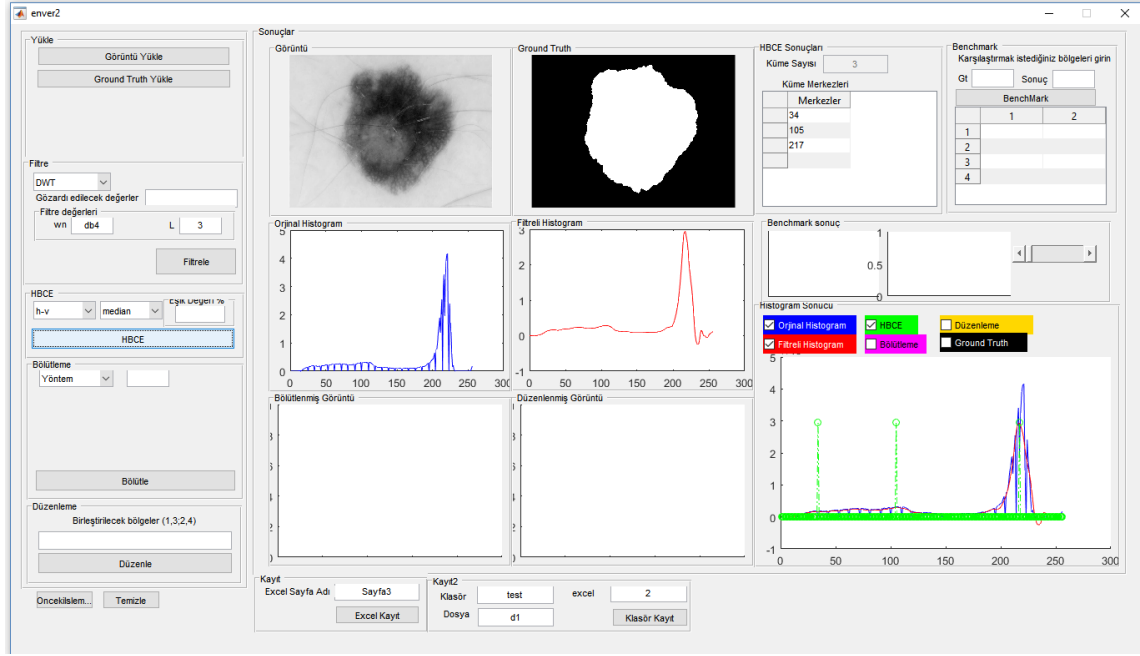
Şekil 2.18. Görüntü ve insan bölütleme sonucu bilgisi girildikten sonra ekran görüntüsü.

Sıradaki adım filtrenin seçilmesidir, II. Bölümde filtre yöntemi seçilir. Bu aşamada Filtre Yok, Medyan ya da DWT filtrelerden uygun olanı seçilir ve istendiği takdirde filtre parametreleri girilir. Filtreleme işleminden sonra Sonuçlar panelinde Filtreli Histogram bölümü ve Histogram Sonucu paneli filtrelenmiş histogram ile doldurulur. Bu adıma ait ekran görüntüsü Şekil 2.19’da verilmiştir.



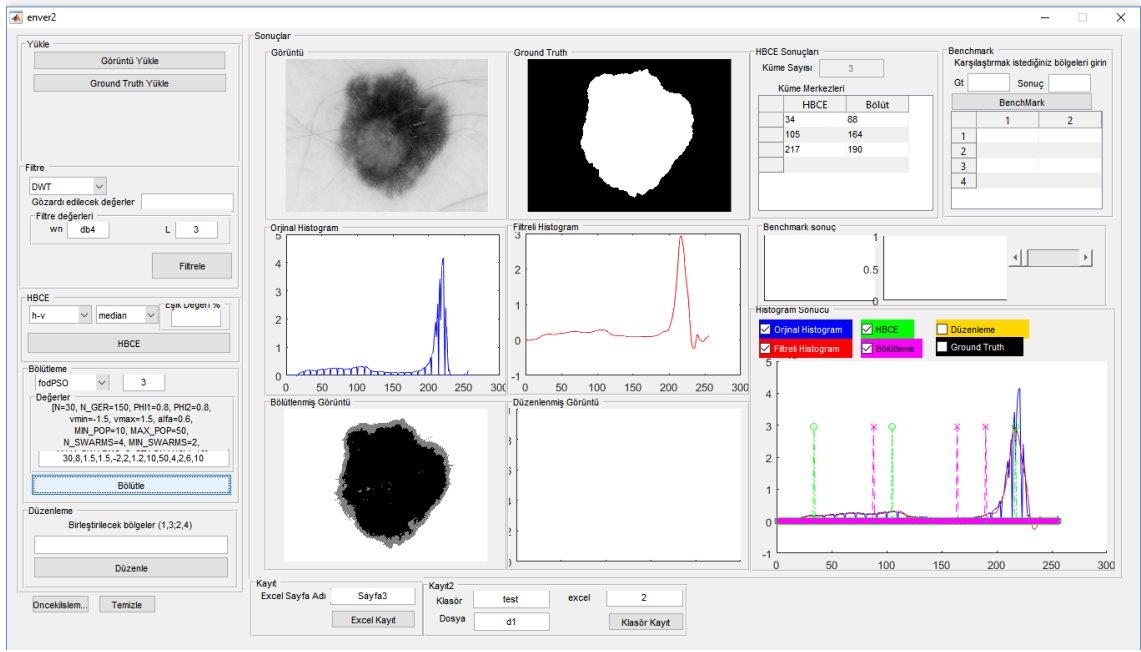
Şekil 2.19. Histogramın filtrelenmesiyle elde edilen bilgilere ait ekran görüntüsü.

Histogramın filtrelenmesiyle görüntü HBCE algoritması için hazırlanmış olur. III. Bölümde, HBCE algoritması için Tarama bilgisi v-h ya da h-v olarak seçilir ve varsa uygun eşik değeri belirlenir. HBCE Sonuçları panelinde HBCE algoritmasının belirlemiş olduğu küme sayısı ve bu kümelere ait tahmini merkez noktaları verilir. Aynı zamanda Histogram Sonucu panelinde belirlenen tahmini küme merkezleri işaretlenerek gösterilir ve bölütleme işlemi için hazırlanmış olur. HBCE adımına ait ekran görüntüsü Şekil 2.20’de verilmiştir.



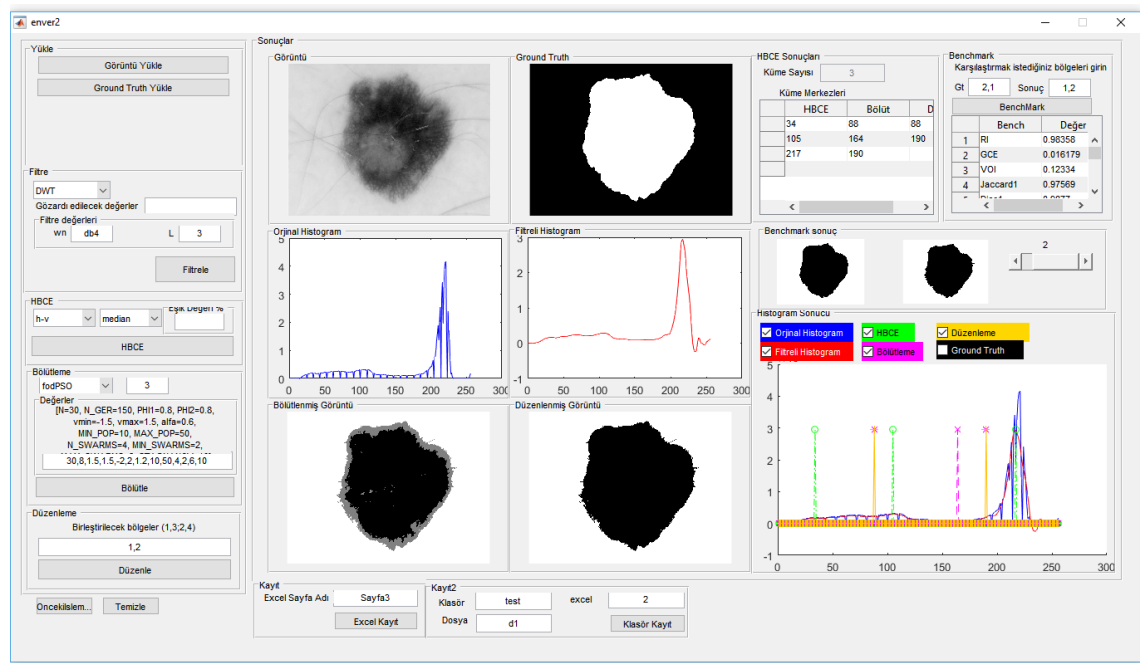
Şekil 2.20. HBCE adımıdan sonra elde edilen sonuçların görüntülenmesi.

IV. Bölüm bölütleme ileminin gerçekleştirildiği kısımdır. Bölütleme işleminde kullanılacak yöntem K-ortalama, BBO, PSO, DPSO, FODPSO ya da merkez tabanlı hale getirilmiş PSO, DPSO, FODPSO'dan herhangi biri olarak belirlenir. İstendiği takdirde yukarıda bahsi geçen bölütleme algoritmaları için parametre girişi yapılabilir. Parametre girişi yapılmadığı takdirde önceden belirlenmiş parametrelerle devam edilir. Bu aşamada HBCE'den alınan bilgilerin hangilerinin ne şekilde kullanılacağı da belirlenir. Yöntem 1'de sadece HBCE'den alınan küme sayısı bilgisi bölütleme işleminde kullanılır. Yöntem 2'de HBCE'den alınan küme sayısı bilgisiyle birlikte tahmini küme merkezlerine ait bilgiler optimizasyon algoritmasındaki sürülerden sadece bir örneklem için başlangıç değeri olarak belirlenir. Yöntem 3'te ise HBCE'den alınan küme sayısı bilgisiyle birlikte tahmini küme merkezlerine ait bilgiler optimizasyon algoritmasındaki sürülerin tamamı için başlangıç değeri olarak belirlenir. Yöntem 4'te HBCE'den alınan küme sayısı bilgisiyle birlikte tahmini küme merkezlerine ait bilgiler optimizasyon algoritmalarındaki sürülere belirlenen ortalama aralıkta rastgele atanacak şekilde başlangıç değerleri olarak verilir. İşlem gerçekleştirildiğinde Sonuçlar panelinde Bölütlenmiş Görüntü bölümü bölütlenmiş görüntüyle doldurulur. Bununla birlikte HBCE Sonuçları panelinde bölütleme işleminden sonra elde edilen küme merkezleri görüntülenirken, Histogram Sonucu panelinde de görsel olarak ifade edilir. Bölütleme işleminden sonra elde edilen ekran görüntüsü Şekil 2.21'de verilmiştir.



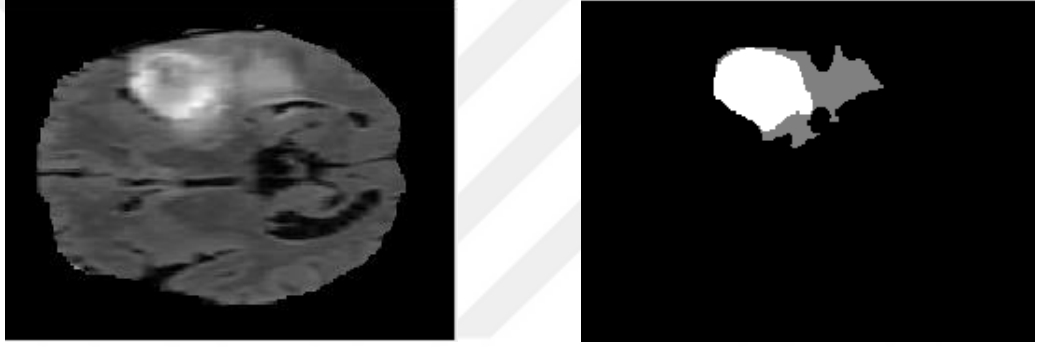
Şekil 2.21. Bölütleme işleminden sonra elde edilen sonuçların görüntülenmesi.

V. Bölüm Düzenleme bölümüdür. Bölütleme işleminden sonra elde edilen görüntüde birleştirilmek istenen bölgeler belirlenir ve düzenleme işlemi gerçekleştirilir. Bu aşamada numaralandırma koyudan açığa gidecek şekilde belirlenmiştir. Eğer insan bölütleme sonucu varsa karşılaştırma işlemi için ayarlamalar VI. Bölümde gerçekleştirilir. Karşılaştırılmak istenen bölgeler sırasıyla belirlenir ve gerekli yerlere girilerek karşılaştırma işlemi gerçekleştirilir. Düzenleme işleminden sonra elde edilen görüntü Sonuçlar panelinde Düzenlenmiş Görüntü bölümüne aktarılırken, Histogram Sonucu paneline de işlenir. Karşılaştırma sonuçları ise Benchmark panelinde görüntülenmektedir. Bu adıma ait ekran görüntüsü Şekil 2.22’de verilmiştir.



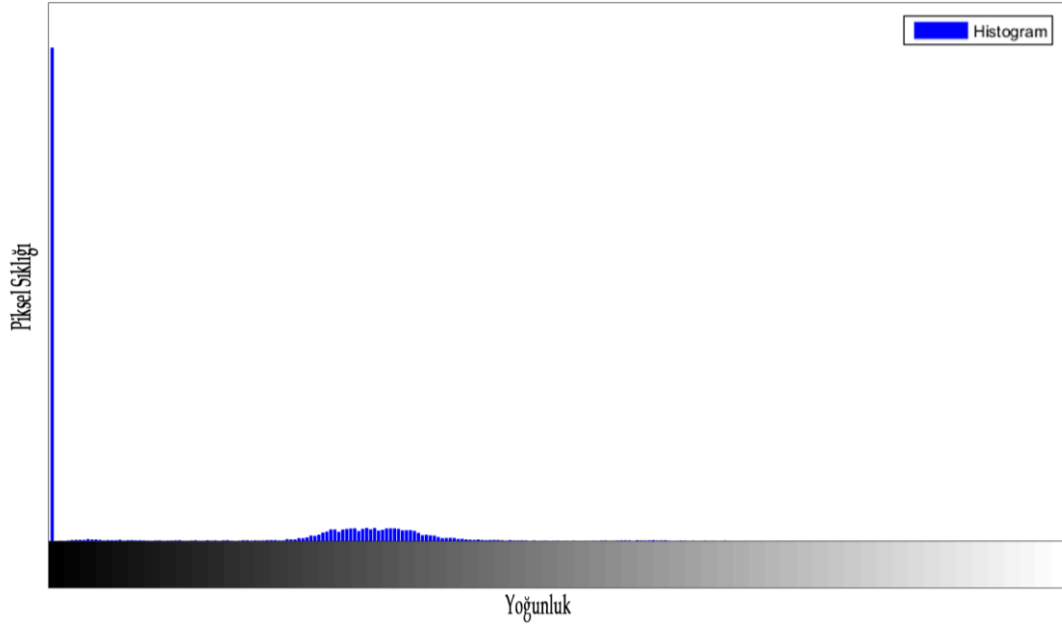
3. BULGULAR VE TARTIŞMA

Tezin bu kısmında geliştirilen yöntemin çalışma adımları örnek ekran görüntüleriyle anlatılmaktadır. Örnek çalışma BRATS 2012 veritabanından alınan görüntü üzerinde gerçekleştirilmiş ve çalışma adımları özetlenmiştir. Örnek Manyetik Resonans (MR) görüntüsü üzerinde tümör çekirdeği ve ödem olmak üzere bölütlenmek istenen iki bölge bulunmaktadır. Şekil 3.1’de BRATS 2012 veritabanından alınan örnek görüntü ve bu görüntüye ait insan bölütleme sonucu verilmiştir.



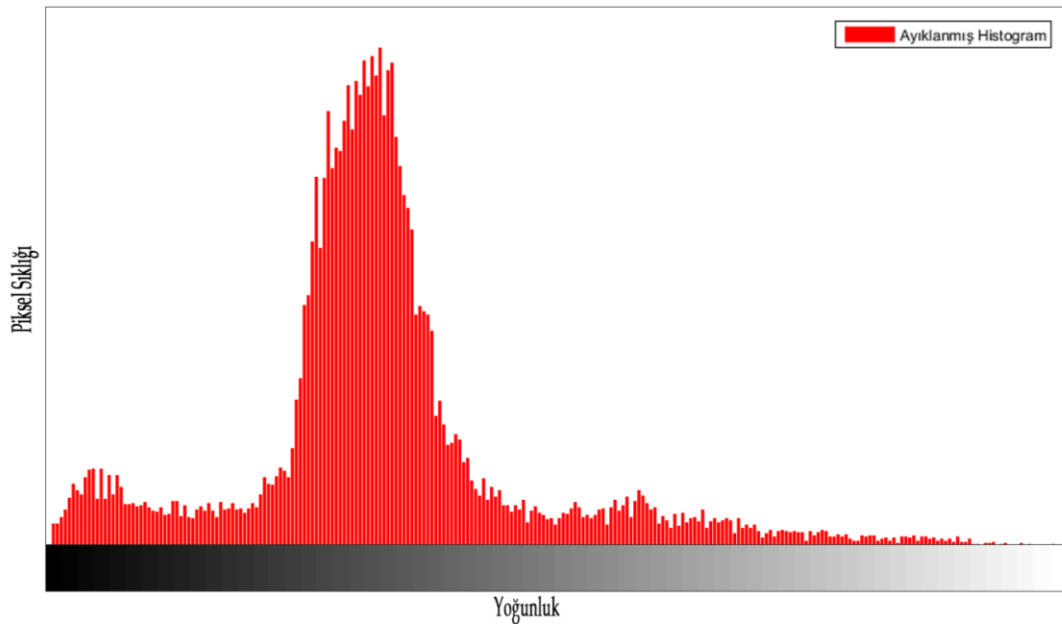
Şekil 3.1. BRATS 2012 veritabanından alınan örnek görüntü ve bu görüntüye ait insan bölütleme sonucu [53].

Görüntü elde edildikten sonraki ilk işlem görüntüye ait histogramın çıkartılmasıdır. Şekil 3.2’de MR görüntüsüne ait histogram verilmiştir. Görüntüden de anlaşılabileceği gibi siyah renk değerine sahip piksellerin sayısı oldukça fazladır ve görüntü histogramında karar verme işlemini olumsuz şekilde etkileyeceği bir gerçektir.



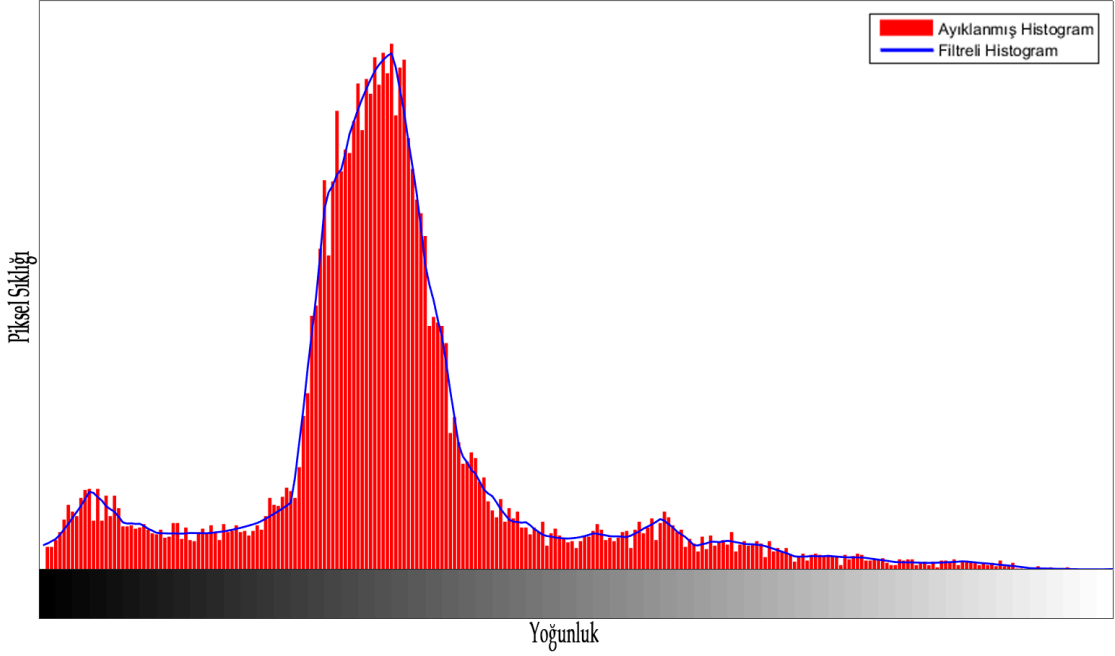
Şekil 3.2. Örnek MR görüntüsüne ait histogram bilgisi.

Histogramda görülen siyah renk değerine sahip piksellerin çalışmanın doğruluğunu engellememesi için çıkartılması gerekmektedir. Bu tür görüntülerde istenmeyen piksellerin görüntü histogramından ayıklanması için kullanıcı panelinde gözardı edilecek piksellerin seçimi için bir alan bırakılmıştır. Bu örnekte sıfır renk değerine ait bilgiler histogramdan ayıklanmıştır. Ayıklanmış histogram görüntüsü Şekil 3.3'te verilmiştir.



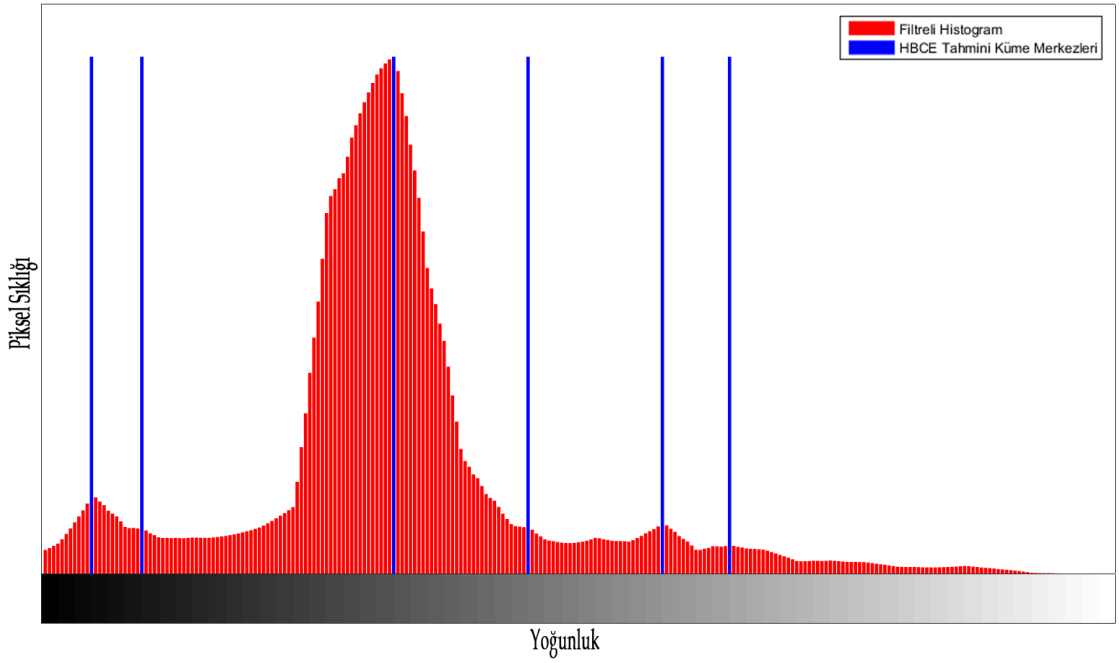
Şekil 3.3. Ayıklanmış histogram görüntüsü.

Histogram verisi incelendiğinde çok fazla salınıma sahip olduğu görülebilmektedir. Histogram verisine uygulanacak bir filtre yardımıyla bu salınımlar temizlenebilmektedir. Örnek çalışmadan DWT filtre ile filtrelenmiş histogram görüntüsü Şekil 3.4’te verilmiştir.



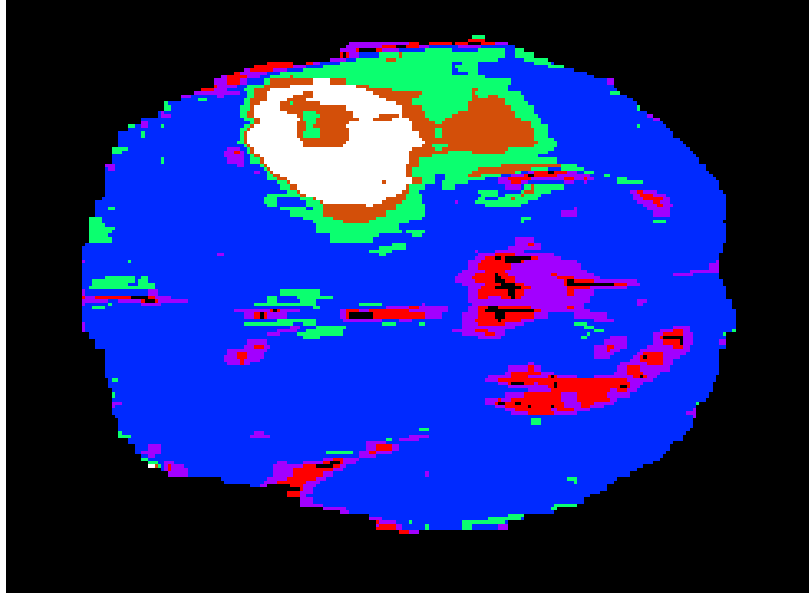
Şekil 3.4. DWT ile filtrelenmiş histogram görüntüsü.

Gerekli ön işlemlerden geçirildikten sonra elde edilen son histogram bilgisi HBCE için hazır hale getirilmiştir. HBCE işlemi için önce yatay sonra dikey tarama tercih edilmiş ve herhangi bir eşik değeri belirlenmeden ortanca hesabı kullanılarak işlem gerçekleştirilmiştir. HBCE işlemiyle birlikte küme sayısı “6” tahmini küme merkezleri ise “12-24-84-116-148-164” olarak belirlenmiştir. HBCE işlemiyle elde edilen tahmini küme merkezleri filtrelenmiş histogram bilgisiyle birlikte Şekil 3.5’te görüntülenmektedir.



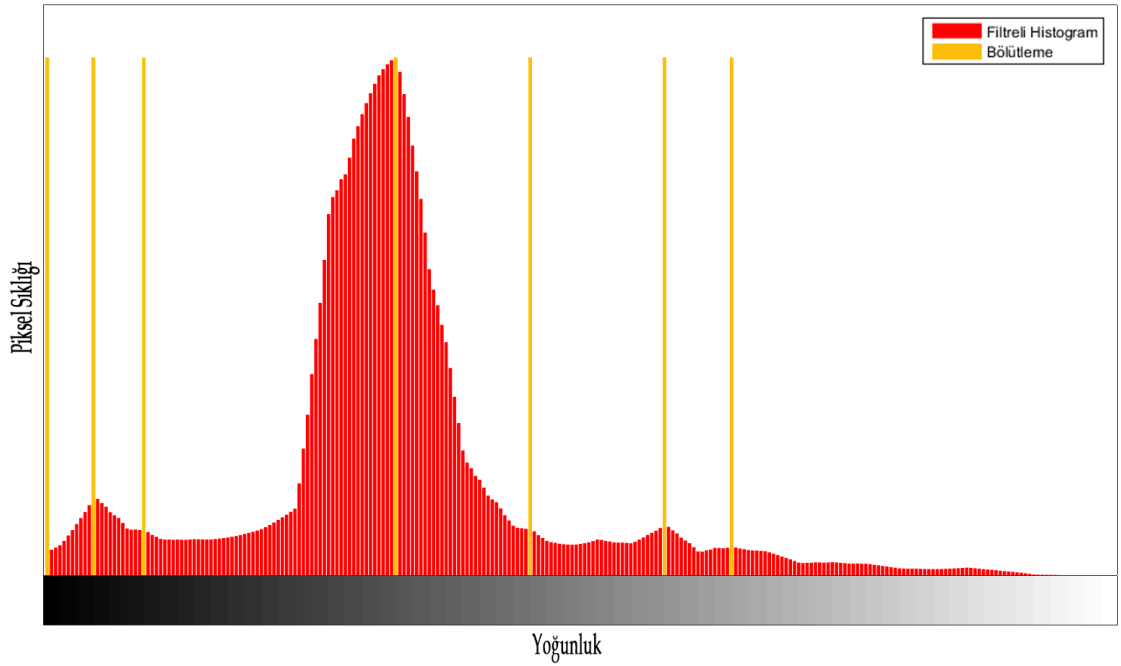
Şekil 3.5. HBCE sonucunda elde edilen tahmini küme merkezleri ve filtrelenmiş histogram görüntüsü.

HBCE işlemiyle elde edilen küme sayısı ve küme merkezleri bilgisi bölütleme işleminde kullanılmak üzere kaydedilmiştir. Örnek çalışmada bölütleme işlemi için BBO algoritması tercih edilmiştir. BBO bölütleme algoritmasına HBCE işlemiyle elde edilen küme sayısı bilgisi verilmiştir. Küme sayısı bilgisine ek olarak HBCE işleminden elde edilen tahmini küme merkezi bilgileri de BBO bölütleme algoritmasında rastgele oluşturulan sürülerin tamamına başlangıç değeri olarak verilmiştir. Görüntünün orijinal hali verilerek çalıştırılan BBO algoritması sonucunda elde edilen bölütlenmiş görüntü Şekil 3.6’da verilmiştir. Filtreleme işleminden önce histogram üzerinde gözardı edilen siyah pikseller bölütleme işleminde hesaba eklenerek işlemin doğruluğu sağlanmıştır.

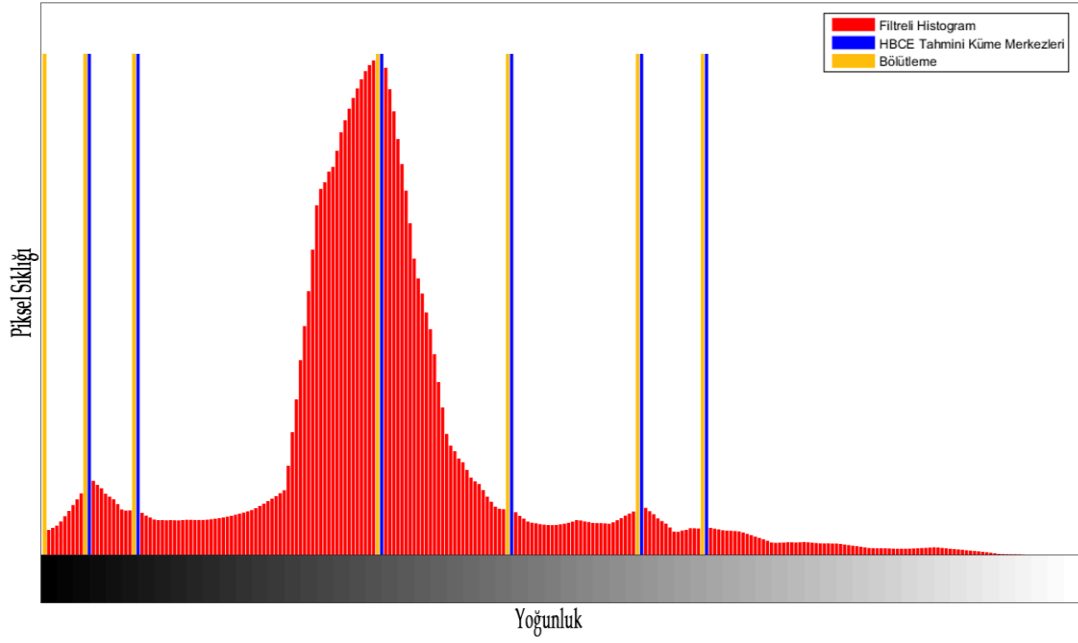


Şekil 3.6. Bölütleme işlemi sonrası elde edilen MR görüntüsü.

Bölütleme işleminden sonra elde edilen bölüt merkez noktaları filtreli histogram ile birlikte Şekil 3.7’de, filtreli histogram ve HBCE sonuçları ile birlikte Şekil 3.8’de verilmiştir.

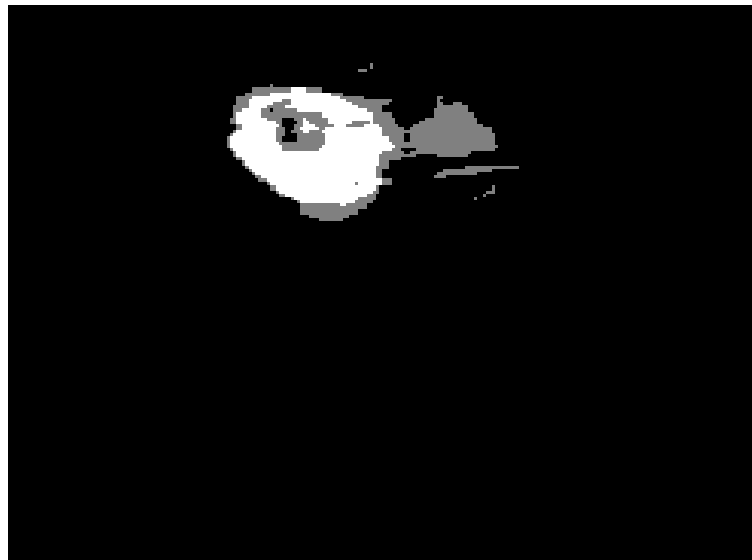


Şekil 3.7. Bölütleme sonucunda elde edilen küme merkezleri ve filtrelenmiş histogram görüntüsü.

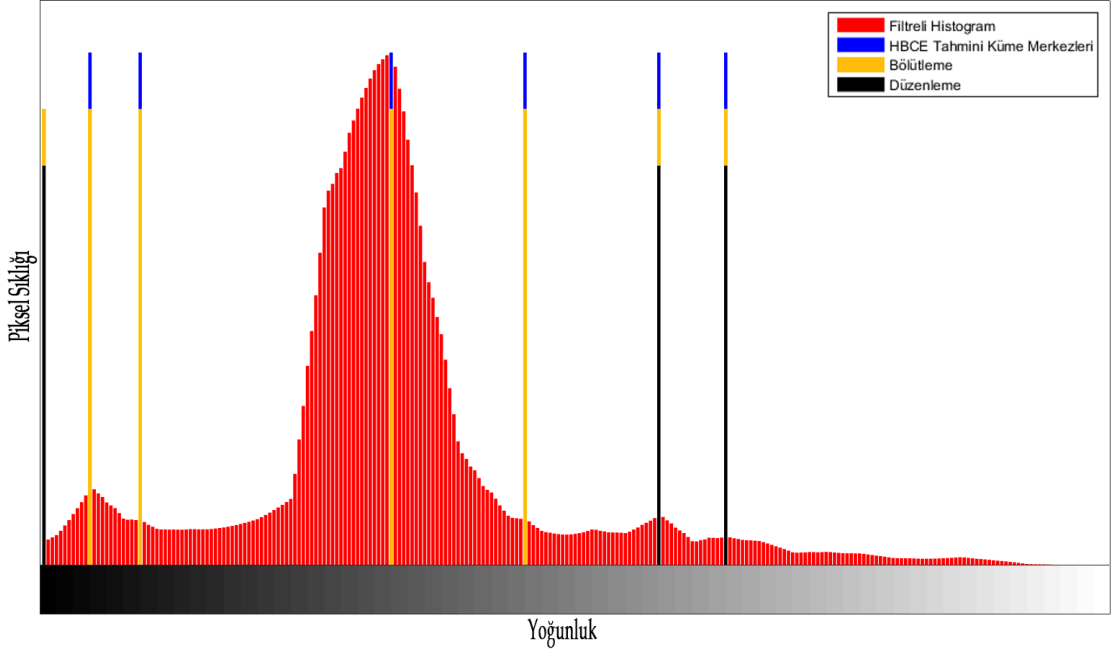


Şekil 3.8. Bölütleme sonucunda elde edilen küme merkezleri, filtrelenmiş histogram görüntüsü ve HBCE sonuçlarının birlikte gösterimi.

Bölütleme işleminden sonra fazlalık bölütlerin çıkartılması için son işleme adımının gerçekleştirilmesi gerekmektedir. Bu örnekte son iki bölüt üzerinde işlem yapılacağından önceki bölütler çalışmadan çıkartılmıştır. Son işleme adımından sonra elde edilen bölütlenmiş görüntü Şekil 3.9'da, son işleme adımında çıkartılan bölgelerden sonra elde edilen düzenlenmiş histogram görüntüsü Şekil 3.10'da verilmiştir.



Şekil 3.9. Bölütleme işlemi sonrası elde edilen MR görüntüsü.



Şekil 3.10. Son işlem sonucunda elde edilen küme merkezleri, filtrelenmiş histogram görüntüsü ve bölütleme sonuçlarının birlikte gösterimi.

Gerçekleştirilen işlemlerin doğruluğunu test etmek için elde edilen sonuç görüntü, insan bölütleme sonuçlarıyla karşılaştırılmıştır. Karşılaştırma için çeşitli performans ölçülerinden faydalanılmıştır. Bu performans ölçütleri sırasıyla Rand Index (RI), Global Consistency Error (GCE), Dice Score (DICE), Jaccard Index, Doğruluk, Hassasiyet ve Özgünlük'tür. Aynı zamanda algoritmaların çalışma zamanları da kaydedilmiş ve performans karşılaştırmasına eklenmiştir.

3.1. PERFORMANS ÖLÇÜTLERİ

Görüntü bölütleme sonuçlarının doğruluğunu test etmek için sıklıkla kullanılan performans ölçütleri iki görüntü arasındaki benzerliği çeşitli yöntemlerle ölçerek sonucun doğruluğu konusunda bilgi edinmemizi sağlamaktadır. Bu performans ölçütlerinden bazıları şu şekilde sıralanabilir:

3.1.1. Rand Index (RI)

Rand Index (RI) güvenilir verilerle (y) deney sonucu elde edilen veriler (x) arasındaki eşleşen çiftleri sayar. Temel olarak RI iki veri kümesi arasındaki benzerliği ölçer. n elemandan ve 2 bölüm olarak tanımlanmış S 'den oluşan bir veri seti için;

a: X'te ve Y'de aynı sette olan S içerisindeki eşleşen elemanların sayısı, b: X'te ve Y'de farklı sette olan S içerisindeki eşleşen elemanların sayısı, c: X'te aynı ve Y'de farklı sette olan S içerisindeki eşleşen elemanların sayısı, d: X'te farklı ve Y'de aynı sette olan S içerisindeki eşleşen elemanların sayısı olmak üzere.

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \quad (3.1)$$

a+b, X ve Y arasındaki benzerlik; c+d, X ve Y arasındaki farklılık olarak tanımlanmıştır. Rand Index 0 ve 1 arasında bir değere sahiptir. 0 değeri iki veri seti arasında hiç benzerlik olmadığını, 1 değeri ise tam tamına aynı olduğunu belirtir.

3.1.2. Global Consistency Error (GCE)

Global Consistency Error (GCE) hangi bölütlemenin diğerine göre daha rafine sonuç verdiğini ölçer. Aynı görüntünün bölümlerini temsil eden birbiriyle ilişkili bölütlemeler tutarlı kabul edilir. Eğer bir bölüm diğer bölümün uygun bir alt kümesi ise o piksel rafine edilmiş bir alanda bulunuyor demektir ve hata 0 olmalıdır. Eğer alt küme ilişkisi yoksa bu durumda iki bölge tutarsız bir şekilde üst üste gelir.

GCE formülü Denklem (3.2)'de belirtildiği gibi tanımlanır. İki bölütleme sonucunu $s1$ ve $s2$ olarak alır ve segmentasyon hatasını ölçer. Hata ölçümü sonucunda 0 ve 1 aralığında bir değer üretilir. Burada 0 hata olmadığına işaret eder. Belirli bir piksel için pi , $s1$ ve $s2$ tarafından içerildiğini belirler.

$$GCE = \frac{1}{n} \min\{\sum_i E(s1, s2, pi), \sum_i E(s2, s1, pi)\} \quad (3.2)$$

3.1.3. Dice Score, Jaccard indeksi, Doğruluk, Hassasiyet, Özgünlük

Örtüşme indeksi olarak da adlandırılan Zar katsayısı [128] (DICE), tıbbi hacim bölütlemelerini doğrulamak için en çok kullanılan metriktir. Algoritmik ve insan bölütleme sonuçları arasında doğrudan karşılaştırmaya ek olarak, tekrarlanabilirliği ölçmek için DICE kullanmak yaygındır. Zou ve diğerleri [129], DICE'ı, bölütlerin aynı MR görüntüsüne açıklamalı olarak eklendiği, ardından tekrarlanan bölümlere ait

çakışma örtüsünün istatistiksel olarak hesaplandığı bir sistem olarak kullanmışlardır. Denklem (3.3)'te DICE denklemi verilmiştir.

$$DICE = \frac{2|S_g^1 \cap S_t^1|}{|S_g^1| + |S_t^1|} = \frac{2DP}{2DP + YP + YN} \quad (3.3)$$

Burada insan bölütleme sonucu $S_g = \{S_g^1, S_g^2\}$, işlem sonucu elde edilen bölütleme sonucu S_t olarak tanımlandığında; S_g^1 ve S_g^2 kümelenmiş bölgeleri tanımlamaktadır. İki net bölüm S_g ve S_t için karışıklık matrisi, Doğru Pozitif (DP), Yanlış Pozitif (YP), Yanlış Negatif (YN) ve Doğru Negatif (DN) gibi iki bölüm arasındaki örtüşmeyi yansıtan dört ortak temelden oluşur.

- DP = S_t ve S_g 'de bulunan doğru değerler örtüşmektedir,
- YP = S_t 'de bulunan doğru değer, S_g 'de bulunan yanlış değer ile örtüşmektedir,
- YN = S_t 'de bulunan yanlış değer, S_g 'de bulunan doğru değer ile örtüşmektedir,
- DN = S_t 'de bulunan yanlış değer, S_g 'de bulunan yanlış değer ile örtüşmektedir.

Jaccard endeksi (JAC) [130], iki set arasındaki kesişmenin birleşimlerine bölünmesi olarak tanımlanır ve Denklem (3.4)'de verilmiştir.

$$JAC = \frac{|S_g^1 \cap S_t^1|}{|S_g^1 \cup S_t^1|} = \frac{DP}{DP + YP + YN} \quad (3.4)$$

Doğruluk (ACC) rastlantısal ve sistematik hataların bir kombinasyonunu tanımlayarak doğruluğu ifade eder, dolayısıyla yüksek doğruluk, yüksek hassasiyet gerektirir. Doğruluk ölçütü formülü Denklem (3.5)'te verilmiştir.

$$Doğruluk = \frac{(DP + DN)}{(DP + YP + YN + DN)} \quad (3.5)$$

Hassasiyet ve Geri Çağırma olarak da adlandırılan Doğru Pozitif Oranı, değerlendirilen bölütlemelerde pozitif olarak tanımlanan bölgelerle insan bölütleme sonucunun pozitif

bölgerlerinin örtüşme miktarını ölçer. Aynı şekilde, Özgünlük olarak da adlandırılan Doğru Negatif Oranı, değerlendirilen bölütlemelerde negatif olarak tanımlanan bölgelerle insan bölütleme sonucundaki negatif bölgelerin örtüşme miktarını ölçer. Denklem (3.6) ve (3.7)'de Hassasiyet ve Özgünlük formülleri verilmiştir [131].

$$Hassasiyet = \frac{DP}{DP+YN} \quad (3.6)$$

$$Özgünlük = \frac{DN}{DN+YP} \quad (3.7)$$

3.2. ÇALIŞMA ORTAMI

Test ortamı için kullanılan bilgisayar Intel i7 2600k işlemci ve 16GB RAM ve 128GB SSD sürücüyü sahiptir.

BBO algoritması için kullanılan parametreler, popülasyon sayısı=10, maksimum mutasyon oranı=0.005, iterasyon sayısı=6, koruma oranı=0.2, $\alpha=0.9$, $\sigma=0.2$ olarak seçilmiştir [132].

PSO, DPSO, FODPSO, mPSO, mDPSO ve mFODPSO algoritmaları için başlangıç değerleri ise Çizelge 3.1'de verilmiştir.

Çizelge 3.1. PSO, DPSO, FODPSO, mPSO, mDPSO ve mFODPSO algoritmaları için parametrelerin başlangıç değerleri [119].

Parametre	PSO	DPSO	FODPSO	mPSO	mDPSO	mFODPSO
İterasyon Sayısı	8	8	8	8	8	8
Popülasyon	200	30	30	200	30	30
P_1	1.5	1.5	1.5	1.5	1.5	1.5
P_2	1.5	1.5	1.5	1.5	1.5	1.5
W	1.2	1.2	1.2	1.2	1.2	1.2
V_{max}	2	2	2	2	2	2
V_{min}	-2	-2	-2	-2	-2	-2
X_{max}	255	255	255	255	255	255
X_{min}	0	0	0	0	0	0
Minimum Pop.	-	10	10	-	10	10
Maksimum Pop.	-	50	50	-	50	50
Sürü Sayısı	-	4	4	-	4	4
Minimum Sürü	-	2	2	-	2	2
Maksimum Sürü	-	6	6	-	6	6
Durgunluk	-	10	10	-	10	10
Fraksiyonel Katsayı	-	-	0.75	-	-	0.75

3.3. BULGULAR

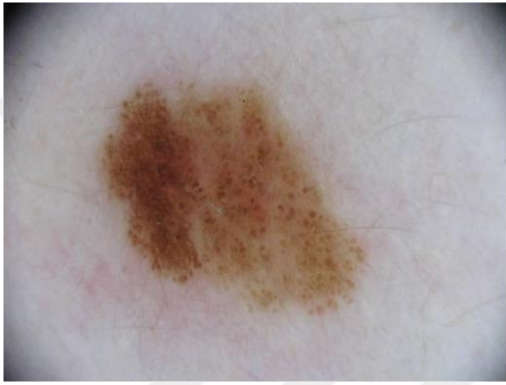
The International Skin Imaging Collaboration (ISIC)'dan elde edilen 30 adet görüntü her algoritma için 30 defa çalıştırılarak sonuçlar elde edilmiş ve karşılaştırma işlemi için kaydedilmiştir. Elde edilen sonuçların karşılaştırılması için ISIC veritabanından elde edilen insan bölütleme sonuçları kullanılmış ve RI, GCE, DICE, Jaccard, Doğruluk, Hassasiyet, Özgünlük ve zaman karşılaştırmaları kullanılmıştır. Şekil 3.11'de ISIC veritabanından alınan görüntülerden beş tanesi ve bu görüntülere ait insan bölütleme sonuçları verilmiştir.



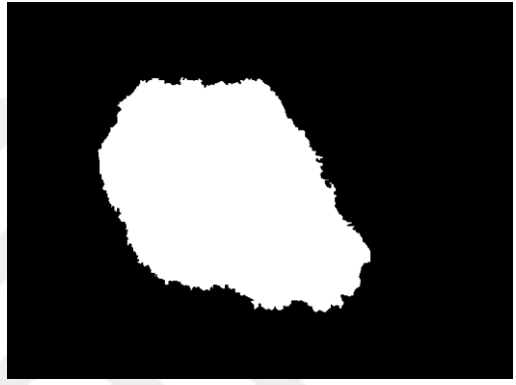
a(1)



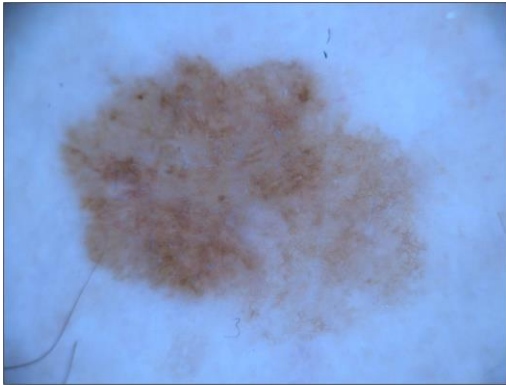
b(1)



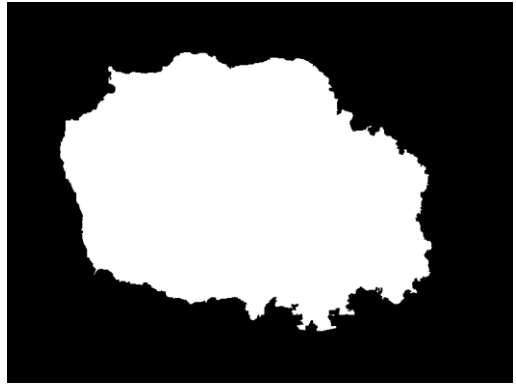
a(2)



b(2)



a(3)



b(3)

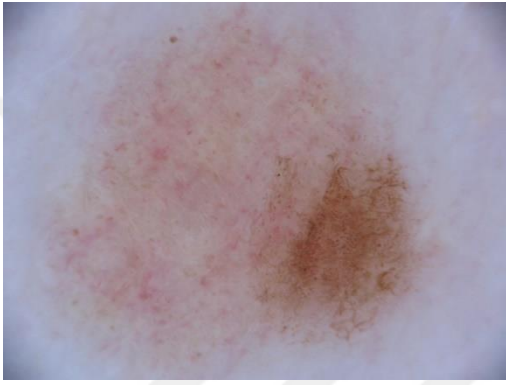
Şekil 3.11. ISIC veritabanından alınan 5 örnek görüntü (a1-a5) ve bu görüntülere ait insan bölütleme sonuçları (b1-b5).



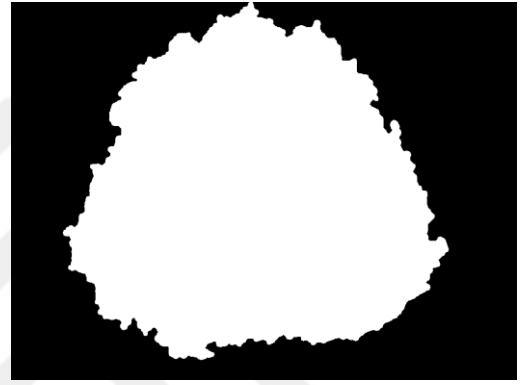
a(4)



b(4)



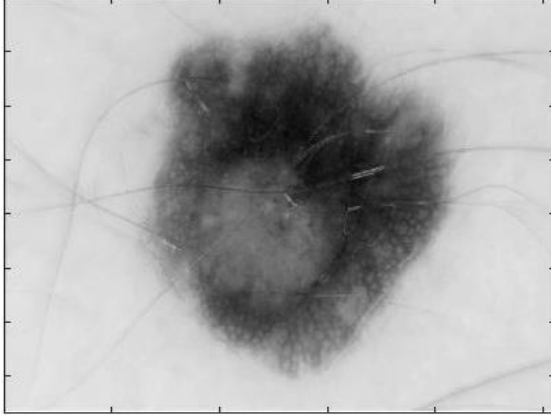
a(5)



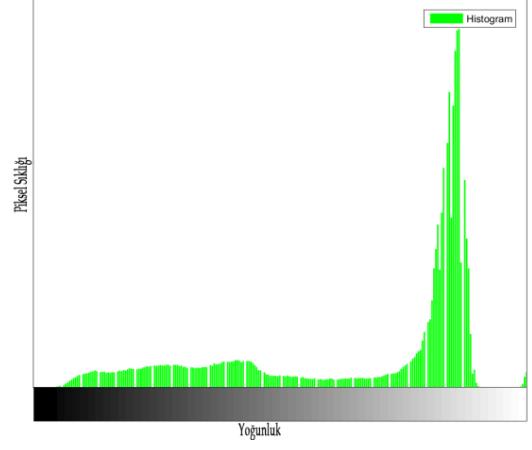
b(5)

Şekil 3.11. (devam). ISIC veritabanından alınan 5 örnek görüntü (a1-a5) ve bu görüntülere ait insan bölütleme sonuçları (b1-b5).

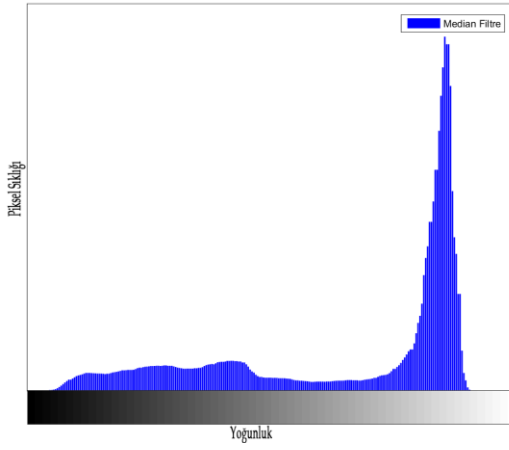
Gerçekleştirilen ön işleme adımında ISIC veritabanından alınan örnek görüntülerde DWT filtrenin ortanca filtreye göre daha iyi sonuç verdiği çeşitli denemeler sonucunda gözlemlenmiş ve DWT filtre tercih edilmiştir. Şekil 3.12’de örnek görüntü histogramları için gerçekleştirilen medyan ve DWT filtre sonuçları verilmiştir.



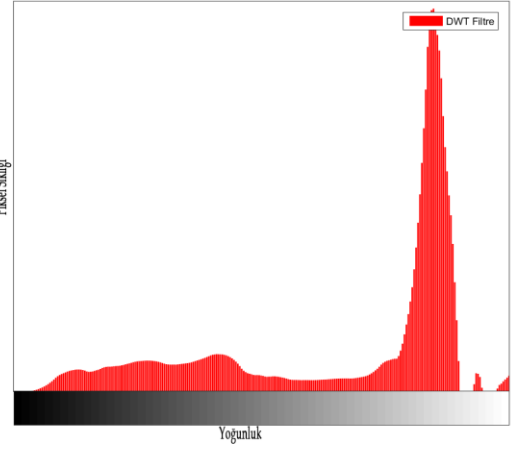
a(1)



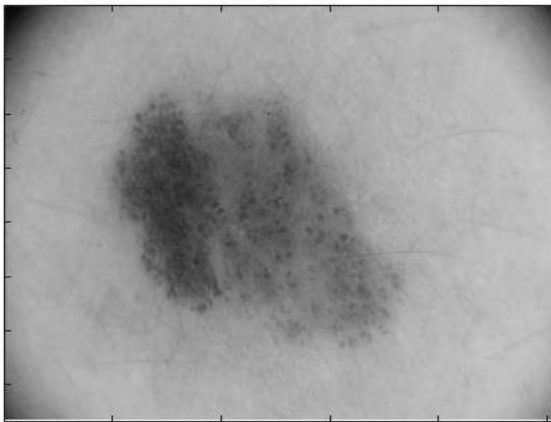
b(1)



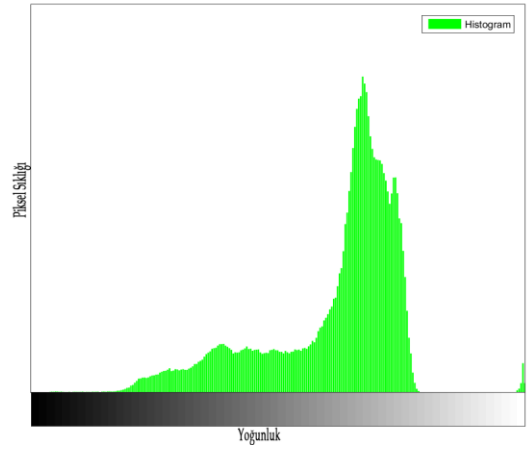
c(1)



d(1)

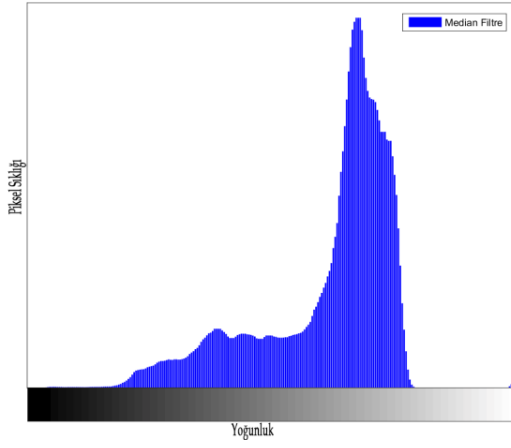


a(2)

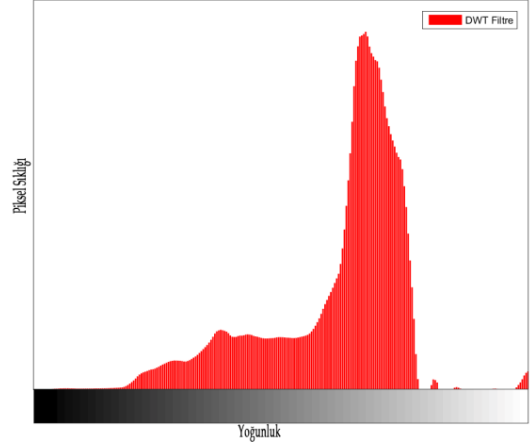


b(2)

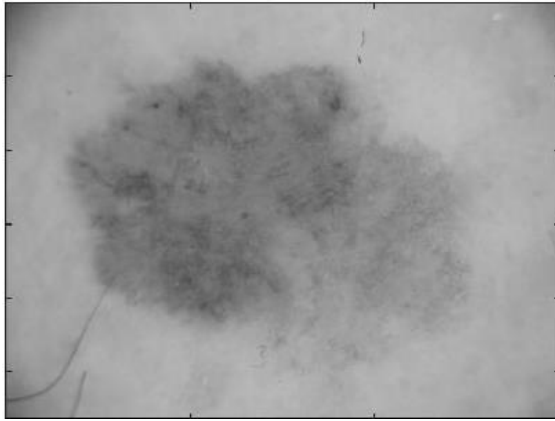
Şekil 3.12. ISIC veritabanından alınan 5 örnek veri için siyah-beyaz görüntü (a1-a5), bu görüntülere ait histogram bilgisi(b1-b5), medyan filtre ile filtrelenmiş histogram (c1-c5), DWT ile filtrelenmiş histogram (d1-d5).



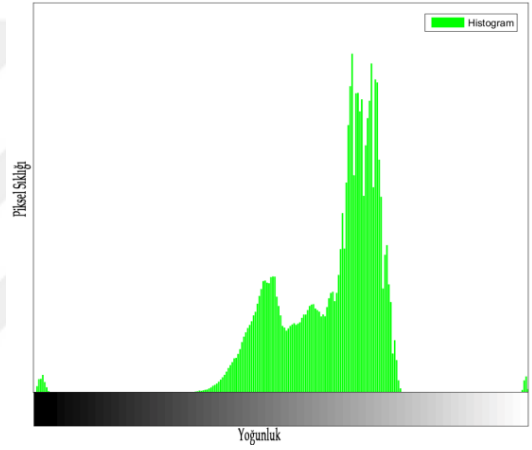
c(2)



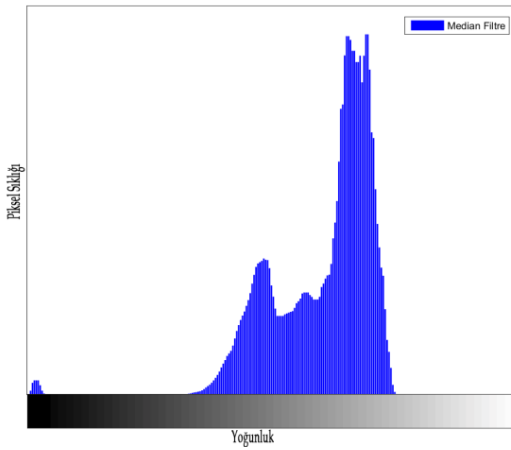
d(2)



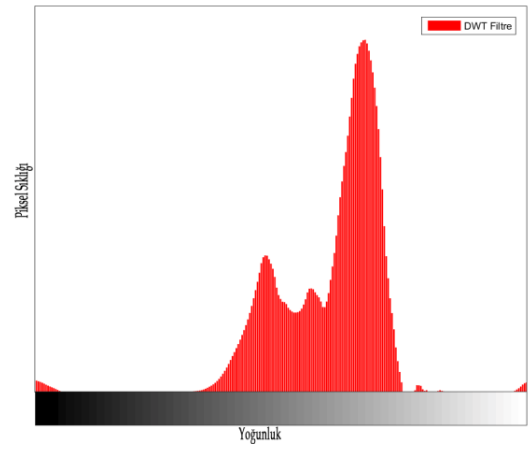
a(3)



b(3)

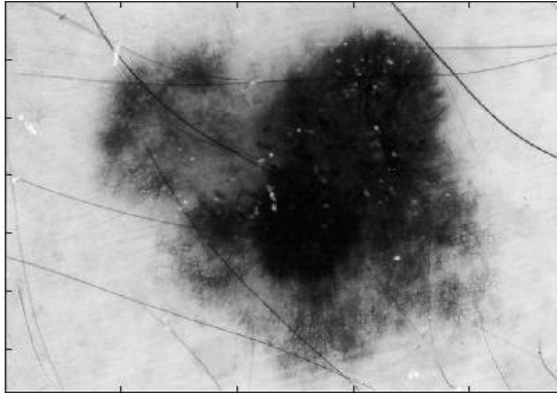


c(3)

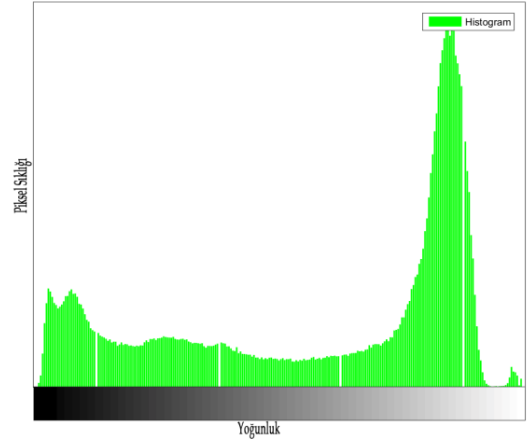


d(3)

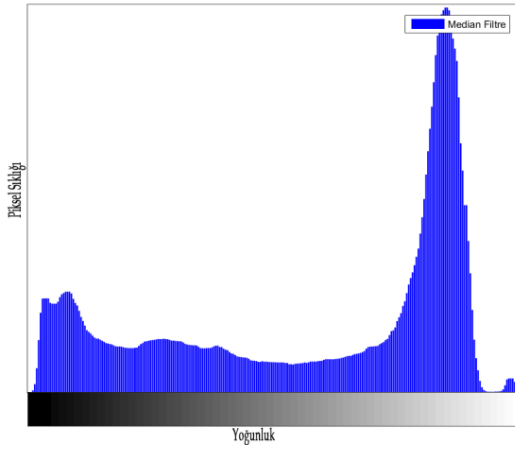
Şekil 3.12. (devam). ISIC veritabanından alınan 5 örnek veri için siyah-beyaz görüntü (a1-a5), bu görüntülere ait histogram bilgisi(b1-b5), medyan filtre ile filtrelenmiş histogram (c1-c5), DWT ile filtrelenmiş histogram (d1-d5).



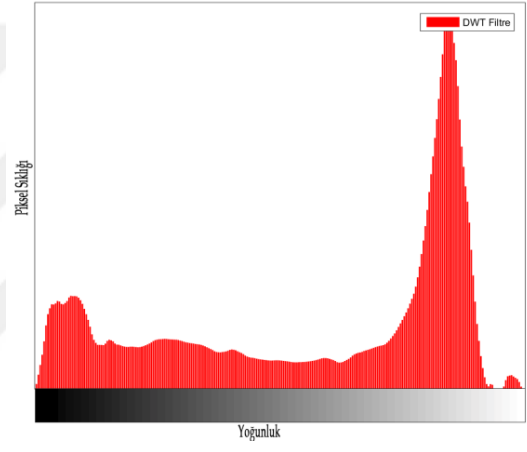
a(4)



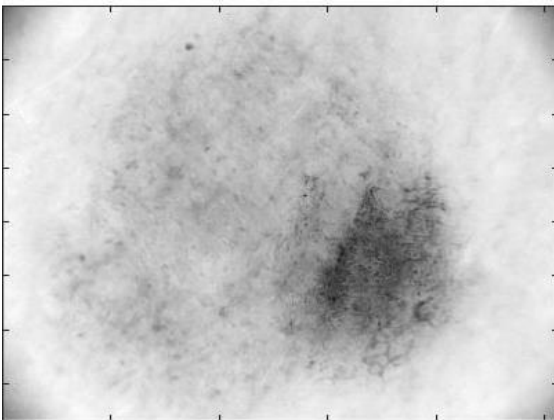
b(4)



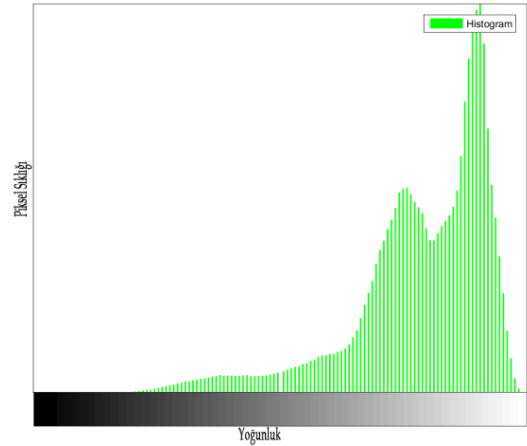
c(4)



d(4)

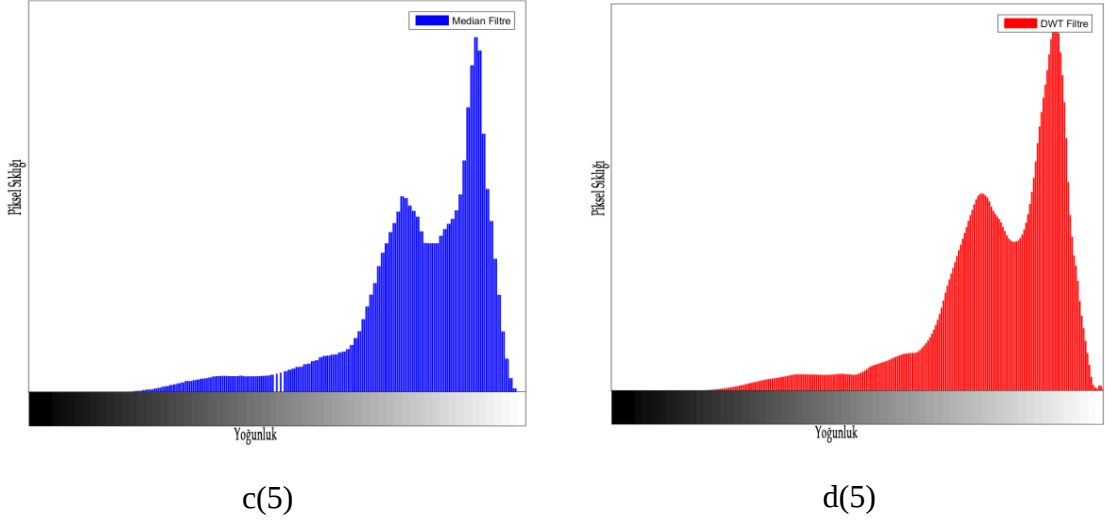


a(5)



b(5)

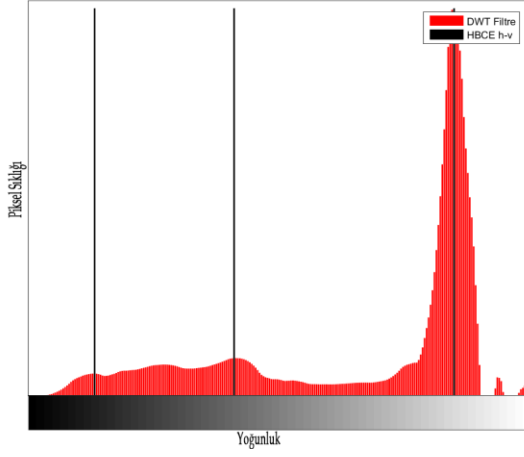
Şekil 3.12. (devam). ISIC veritabanından alınan 5 örnek veri için siyah-beyaz görüntü (a1-a5), bu görüntülere ait histogram bilgisi(b1-b5), medyan filtre ile filtrelenmiş histogram (c1-c5), DWT ile filtrelenmiş histogram (d1-d5).



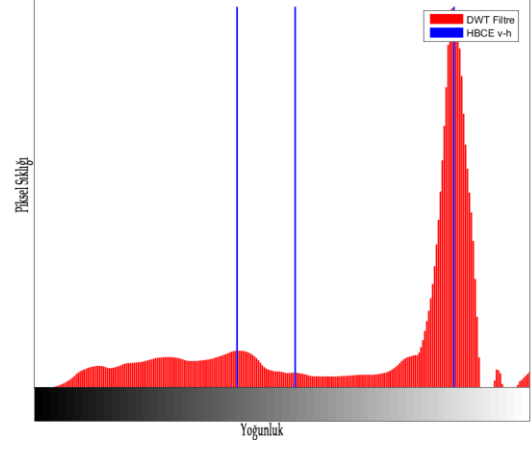
Şekil 3.12. (devam). ISIC veritabanından alınan 5 örnek veri için siyah-beyaz görüntü (a1-a5), bu görüntülere ait histogram bilgisi(b1-b5), medyan filtre ile filtrelenmiş histogram (c1-c5), DWT ile filtrelenmiş histogram (d1-d5).

Şekil 3.12’de görüldüğü gibi medyan filtre ile filtrelenmiş histogramda keskin tepe ve keskin çukur noktaları bulunurken DWT ile filtrelenmiş histogramda keskin tepe ve keskin çukurlara rastlanmamıştır. Bu durum HBCE algoritmasının çalıştırılmasıyla doğru sonuç elde edilmesinde oldukça önemlidir.

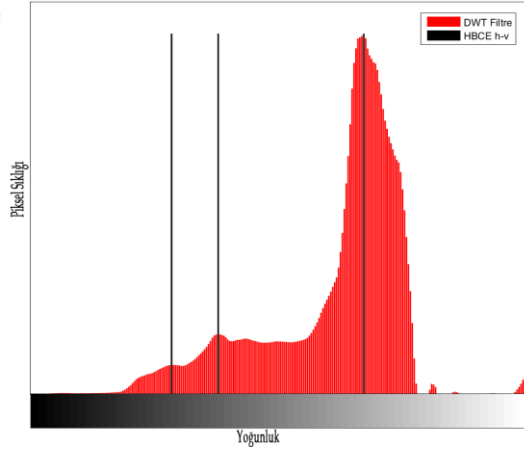
HBCE algoritması yatay-dikey (h-v) ve dikey-yatay (v-h) şekillerle eşik değeri girilmeden çalıştırılmış ve sonuçlar değerlendirilmiştir. Yatay-dikey olarak çalıştırılan HBCE algoritması kullanılan örnek görüntülerde küme sayısını hatalı kestirmiş ve istenilen sonuca ulaşamamıştır. Buna karşılık dikey-yatay olarak çalıştırılan HBCE algoritması herhangi bir eşik değeri girilmeden istenilen sonuca ulaşmayı başarmıştır. Şekil 3.13’te HBCE algoritmasının yatay-dikey ve dikey-yatay çalıştırılmasıyla elde edilen sonuçlar filtrelenmiş histogramla birlikte gösterilmektedir.



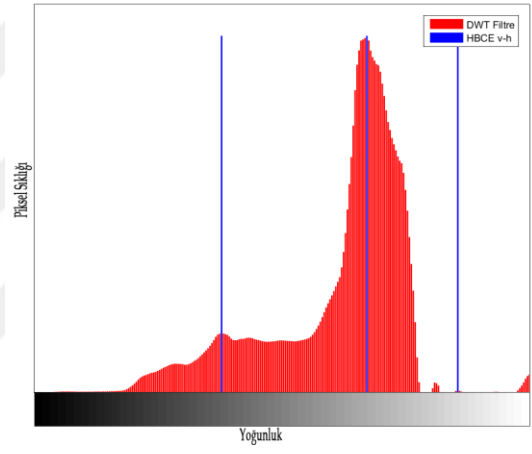
a(1)



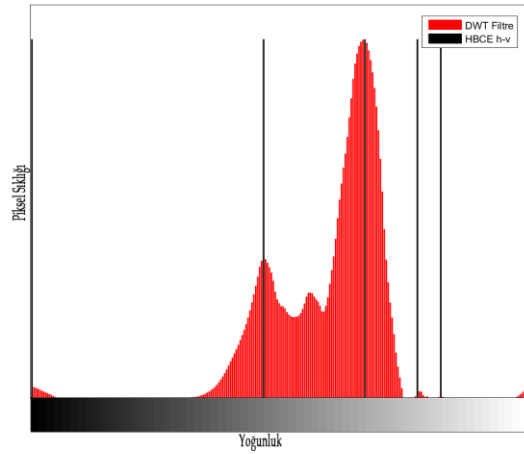
b(1)



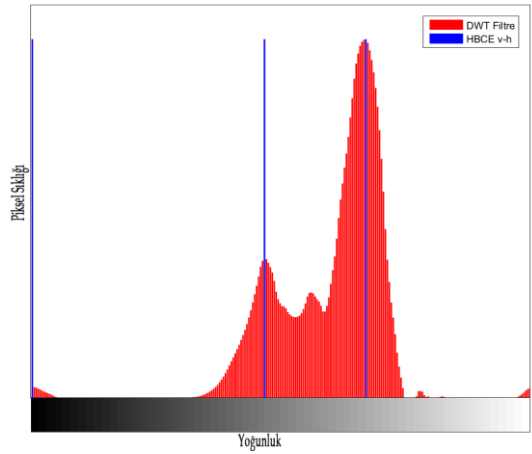
a(2)



b(2)

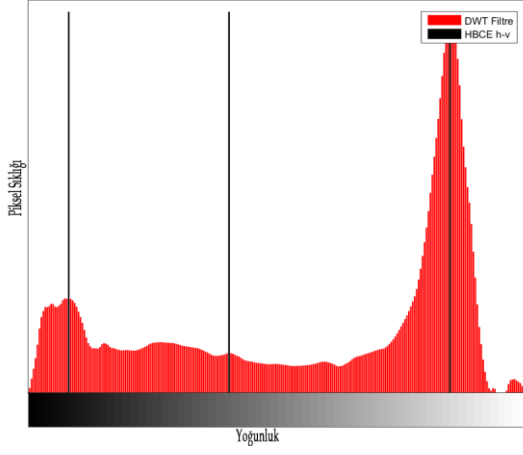


a(3)

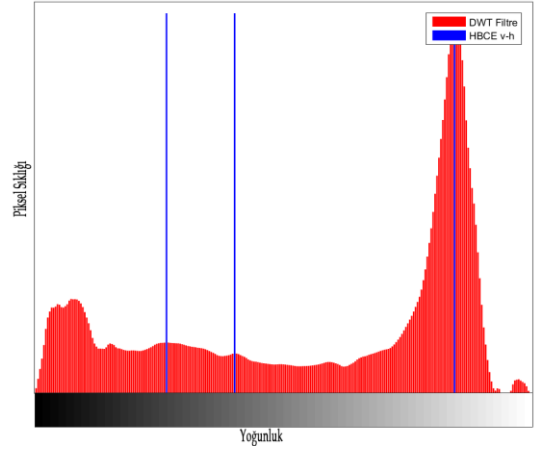


b(3)

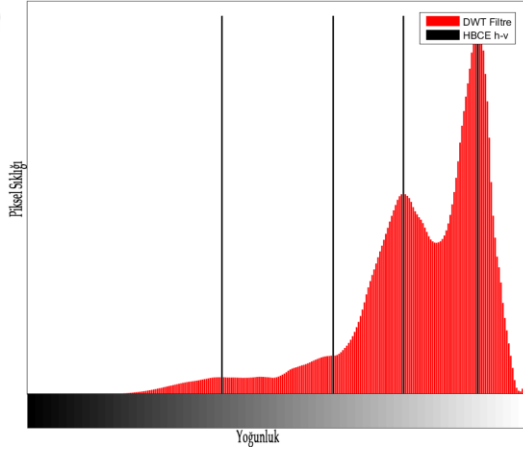
Şekil 3.13. HBCE algoritmasının yatay-dikey (a1-a5) ve dikey-yatay (b1-b5) çalıştırılmasıyla elde edilen sonuçlar.



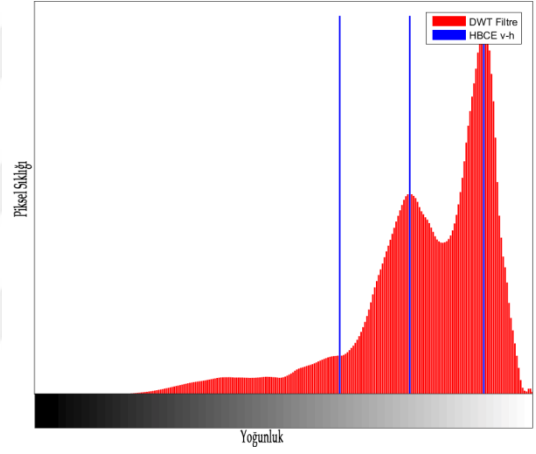
a(4)



b(4)



a(5)



b(5)

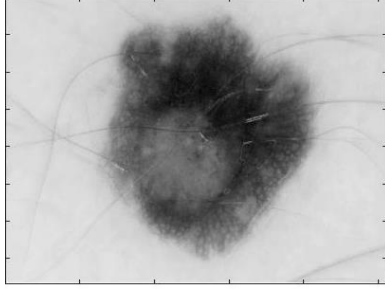
Şekil 3.13. (devam). HBCE algoritmasının yatay-dikey (a1-a5) ve dikey-yatay (b1-b5) çalıştırılmasıyla elde edilen sonuçlar.

Şekil 3.13'te görüldüğü gibi a(3) ve a(5) çözümlerinde yatay-dikey çalışan HBCE algoritması küme sayısını 4 olarak tespit etmiştir. Oysaki dikey-yatay çalıştırılan HBCE algoritması tüm örneklerde 3 küme sayısını doğru tahmin etmeyi başarmıştır.

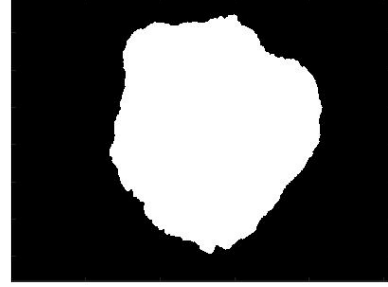
HBCE algoritmasının çalıştırılmasıyla küme sayıları ve tahmini küme merkezleri elde edilmiş bölütleme işlemi için hazırlanmıştır. Bölütleme işleminde sırasıyla K-ortalama, BBO, PSO, DPSO, FODPSO ve mPSO, mDPSO, mFODPSO algoritmaları kullanılmıştır. Bölütleme işleminde HBCE'den alınan küme sayıları tüm bölütleme algoritmalarına doğrudan verilmiştir.

ISIC veritabanından alınan 30 görüntü için her bir algoritma 30 defa çalıştırılmış ve sonuçlar kaydedilmiştir. Örnek bir görüntü için (a) orijinal görüntü, (b) insan bölütleme

sonucu Şekil 3.14'te; her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü Şekil 3.15'te verilmiştir.

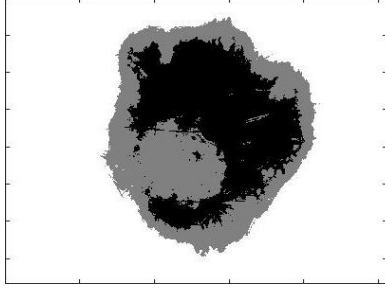


(a)

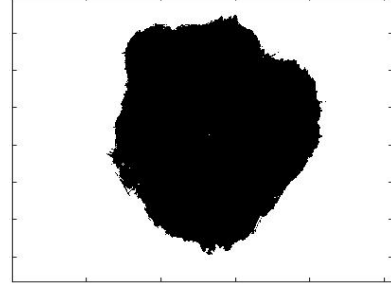


(b)

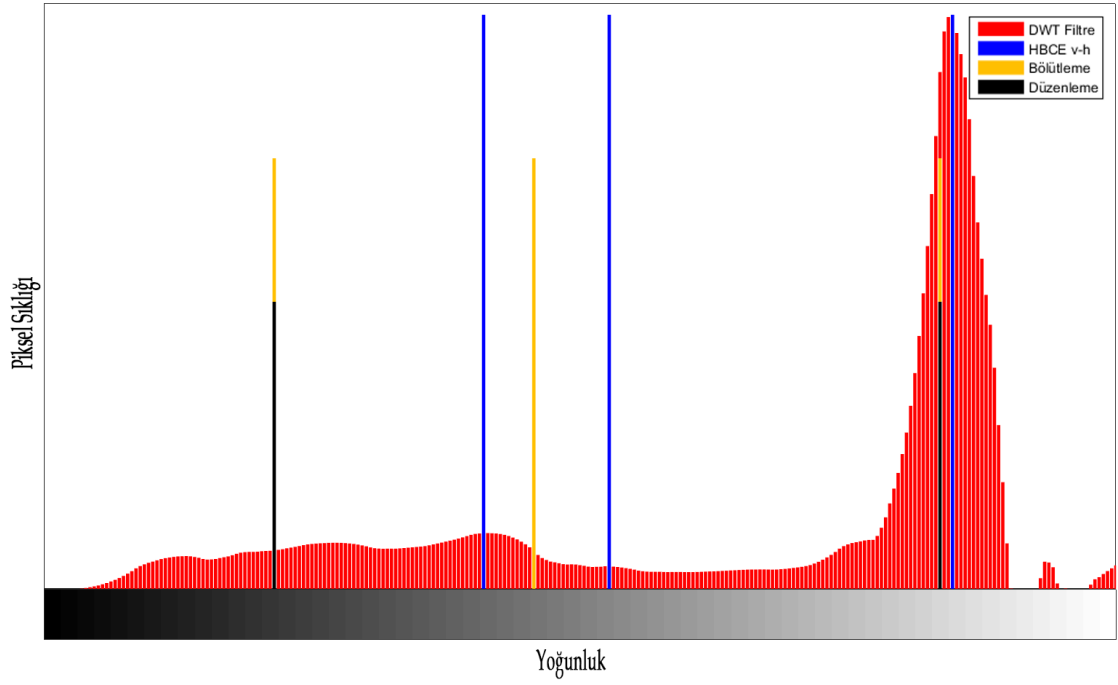
Şekil 3.14. Örnek bir görüntü için (a) orijinal görüntü, (b) insan bölütleme sonucu.



a(1)



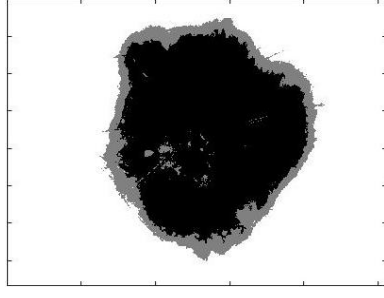
b(1)



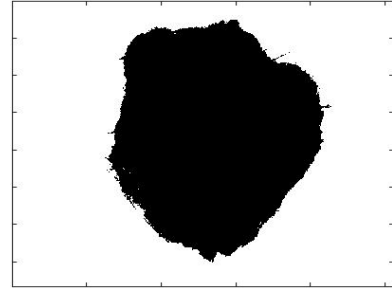
K-ortalama

c(1)

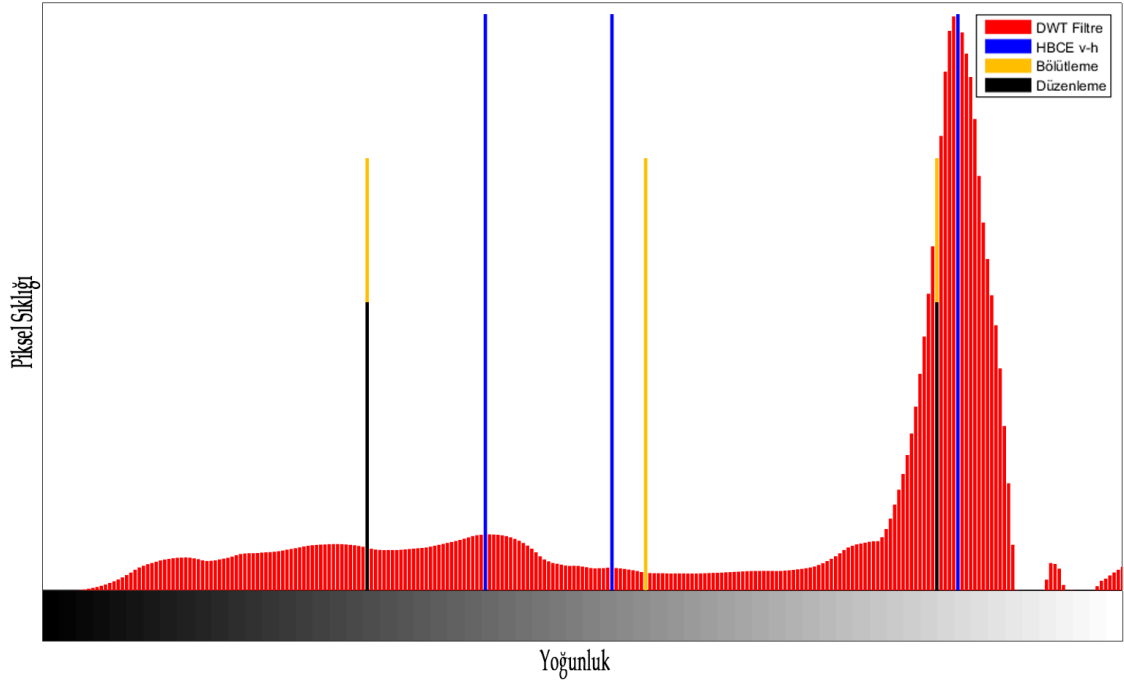
Şekil 3.15. Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.



a(2)



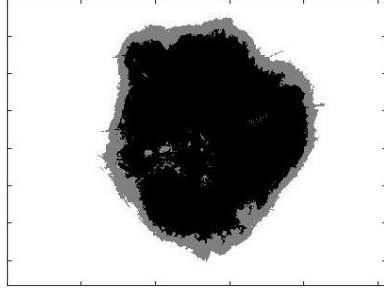
b(2)



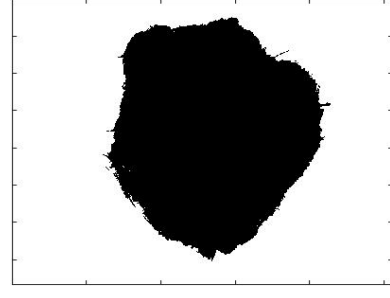
BBO

c(2)

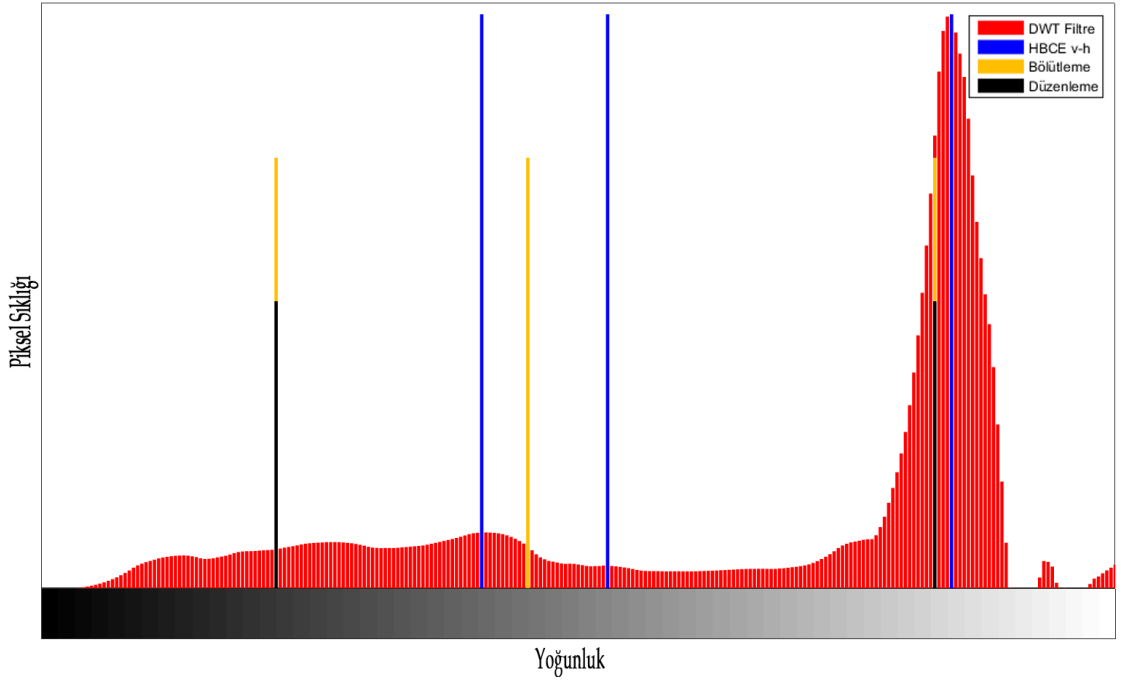
Şekil 3.15. (devam). Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.



a(3)



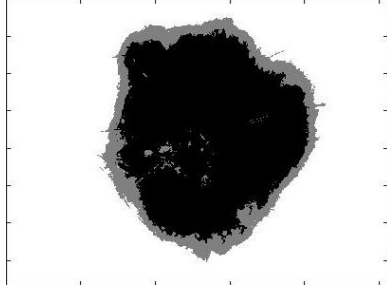
b(3)



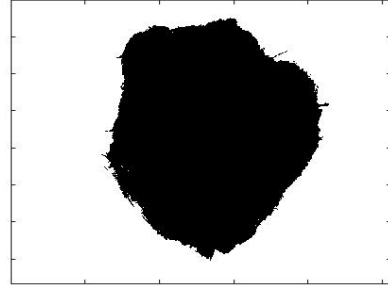
PSO

c(3)

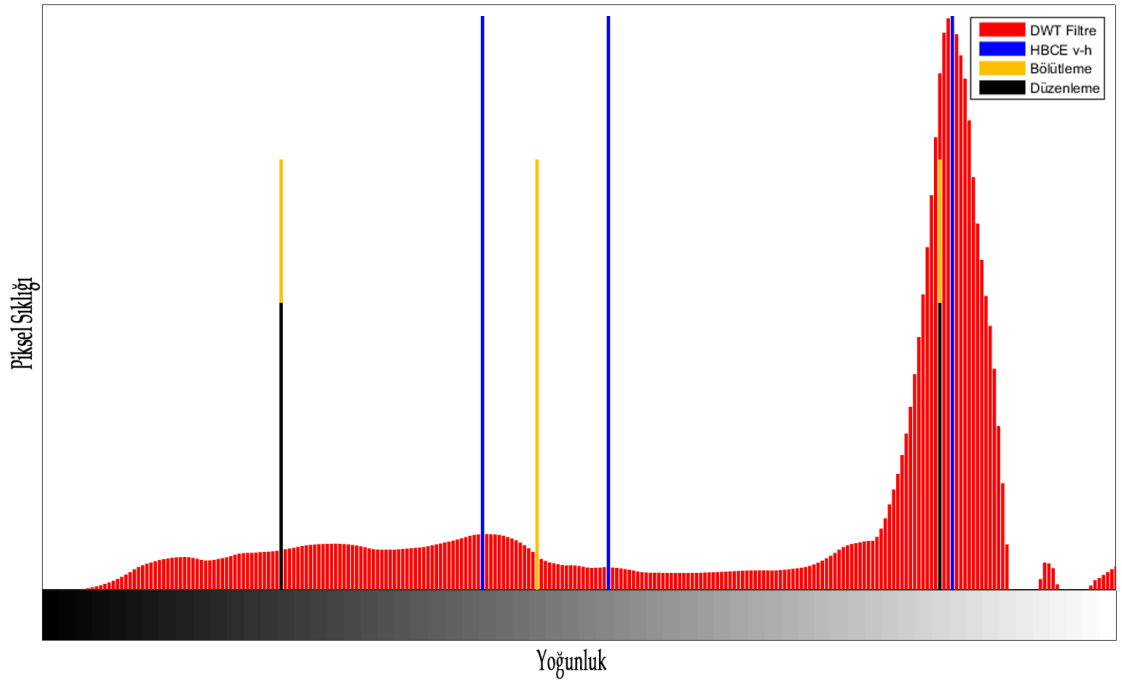
Şekil 3.15. (devam). Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.



a(4)



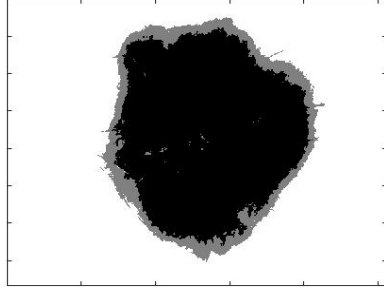
b(4)



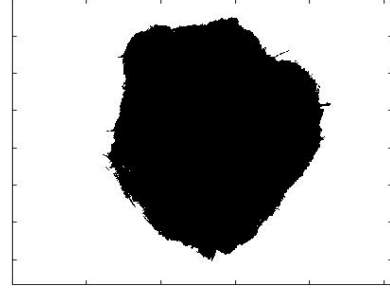
DPSO

c(4)

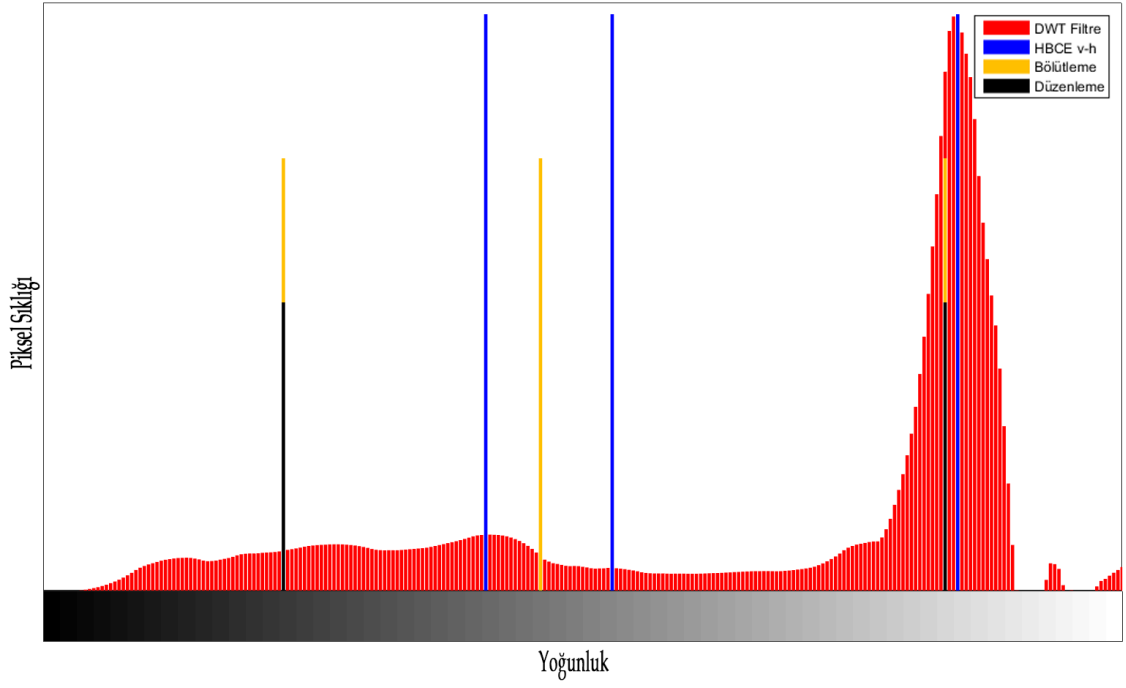
Şekil 3.15. (devam). Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.



a(5)



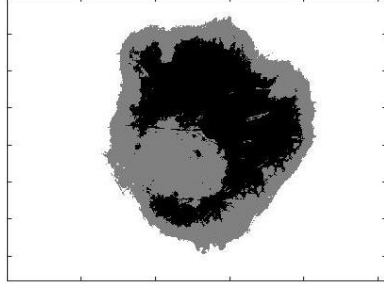
b(5)



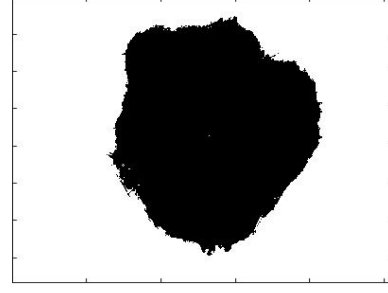
FODPSO

c(5)

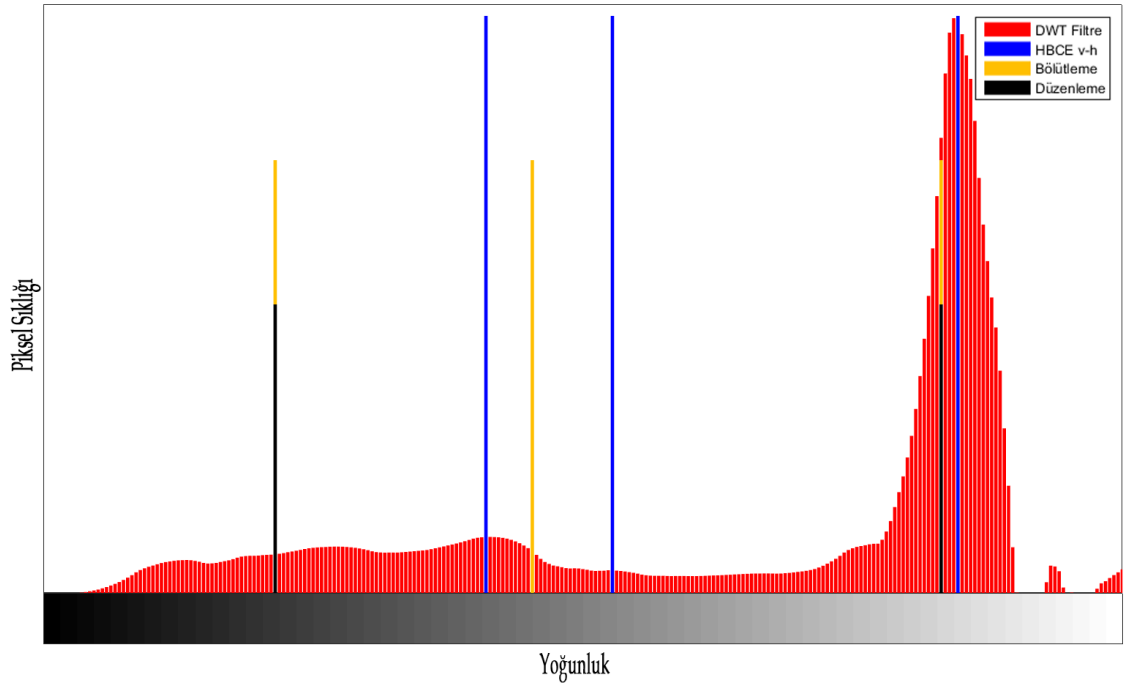
Şekil 3.15. (devam). Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.



a(6)



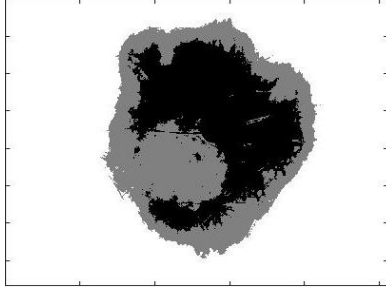
b(6)



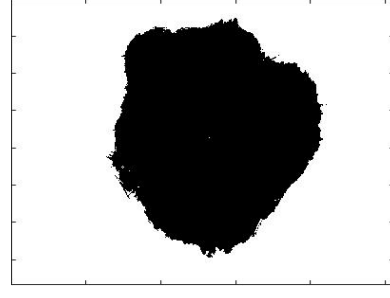
mPSO

c(6)

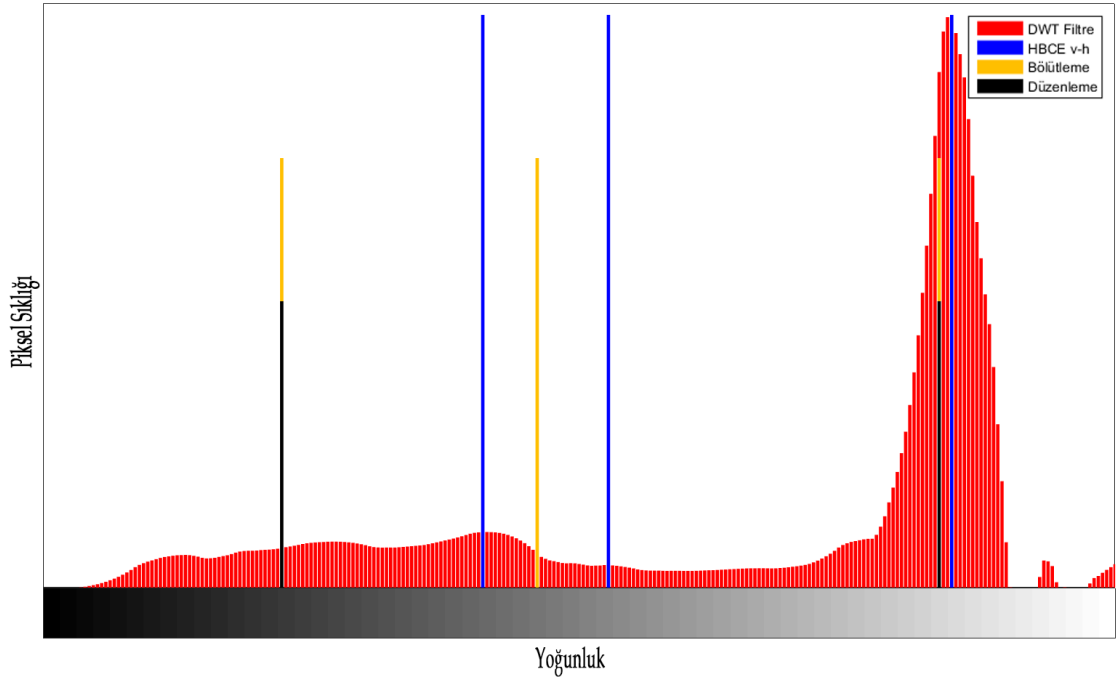
Şekil 3.15. (devam). Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.



a(7)



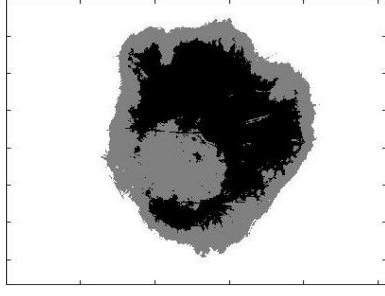
b(7)



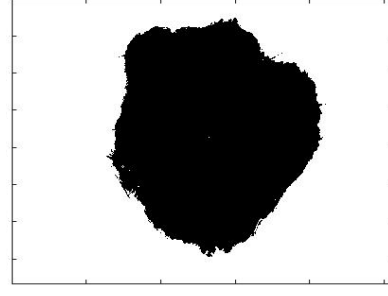
mDPSO

c(7)

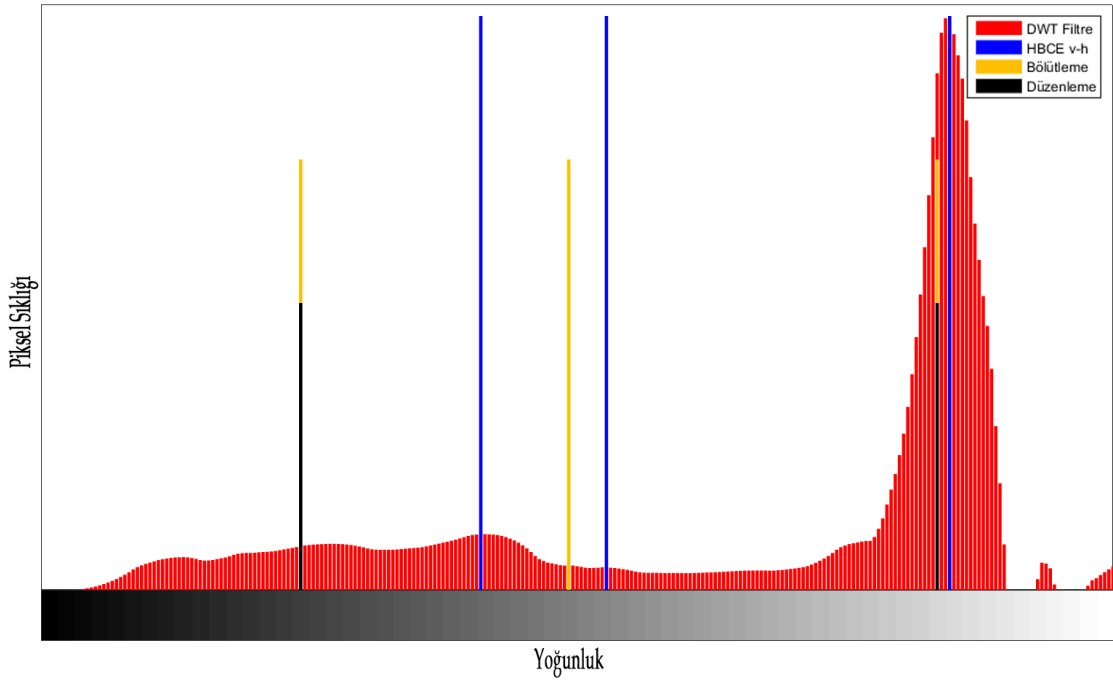
Şekil 3.15. (devam). Her bir algoritma için (a1-a8) bölütme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.



a(8)



b(8)



mFODPSO

c(8)

Şekil 3.15. (devam). Her bir algoritma için (a1-a8) bölütleme sonucunda elde edilen görüntü, (b1-b8) düzenleme sonucu, (c1-c8) histogram görüntüsü.

Küme sayısı bilgisi verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları Çizelge 3.2’de verilmiştir. Algoritmaların çalışma zamanları incelendiğinde en hızlı çözümlerin K-ortalama ve BBO algoritmalarına ait olduğu gözlenmiştir. PSO, DPSO ve FODPSO algoritmalarının düzenlenmiş versiyonlarının ise orjinallerine göre ortalama 0,2 saniye daha yavaş oldukları saptanmıştır. HBCE’den alınan küme sayısı bilgisiyle çalıştırılan bölütleme algoritmaları performansları açısından değerlendirildiğinde mPSO, mDPSO ve mFODPSO diğer algoritmalara nazaran daha iyi sonuçlara ulaştığı gözlemlenmektedir. Standart sapma değerleri göz

önünde bulundurulduğunda PSO, DPSO ve FODPSO ve bunların merkez tabanlı hale getirilmiş versiyonları K-ortalama ve BBO algoritmalarına göre daha kararlı sonuçlara ulaşmışlardır. Algoritmaların çalışma zamanlarına bakıldığında ise HBCE algoritması en kötü olduğu durumda bile 0,52 ms sürede işlemini tamamlamış, bölütleme işleminde ise en hızlı algoritma K-ortalama olmuştur.

Çizelge 3.2. Küme sayısı bilgisi verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.

Bölütleme Algoritması	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
K-ortalama	89,672 ±9,0	7,915 ±5,6	92,780 ±12,4	93,114 ±15,7	89,459 ±16,2	94,652 ±16,4	91,733 ±7,8	0,496 ±0,0	0,905 ±0,9
BBO	82,182 ±14,2	11,739 ±7,7	85,949 ±17,8	85,754 ±22,6	79,788 ±24,4	84,783 ±26,1	91,962 ±9,9	0,515 ±0,0	1,727 ±1,0
PSO	90,017 ±7,8	8,306 ±5,8	94,496 ±4,6	95,588 ±3,7	91,783 ±6,7	97,466 ±4,6	90,907 ±8,2	0,505 ±0,0	2,411 ±1,0
DPSO	90,059 ±7,9	8,254 ±5,7	94,507 ±4,7	95,598 ±3,7	91,806 ±6,7	97,259 ±5,0	91,312 ±8,0	0,500 ±0,0	2,461 ±1,0
FODPSO	90,028 ±7,9	8,248 ±5,7	94,483 ±4,8	95,583 ±3,8	91,782 ±6,7	97,204 ±5,1	91,315 ±8,0	0,505 ±0,0	2,478 ±1,0
mPSO	90,135 ±7,8	8,161 ±5,7	94,556 ±4,6	95,644 ±3,7	91,883 ±6,6	97,302 ±4,8	91,418 ±8,0	0,519 ±0,5	2,615 ±1,6
mDPSO	90,211 ±7,9	8,094 ±5,6	94,593 ±4,7	95,675 ±3,6	91,938 ±6,6	97,168 ±5,0	91,721 ±7,8	0,496 ±0,0	2,677 ±1,6
mFODPSO	90,211 ±7,8	8,137 ±5,6	94,595 ±4,7	95,664 ±3,6	91,922 ±6,6	97,171 ±5,0	91,697 ±7,8	0,494 ±0,0	2,695 ±1,6

HBCE algoritmasından alınan küme sayısı bilgisiyle birlikte tahmini küme merkezlerine ait bilgiler de bölütleme algoritmalarına verilerek çeşitli çalışmalar gerçekleştirilmiş ve elde edilen sonuçlar değerlendirilmiştir. HBCE'den alınan tahmini küme merkezi bilgisi K-ortalama algoritmasında başlangıç noktaları olarak verilmeden (a) ve verilerek (b) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.3'te verilmiştir. Zaman karşılaştırmaları incelendiğinde küme merkezi bilgilerinin verilmesi K-ortalama algoritmasını hızlandırdığı gözlemlenmiştir. Ancak elde edilen performans sonuçları değerlendirildiğinde K-ortalama algoritması küme merkez bilgisi verilmeden

çalıştırıldığında daha iyi sonuçlara ulaştığı gözlemlenmiştir.

Çizelge 3.3. Tahmini küme merkezi bilgisi verilmeden (a) ve verilerek (b) çalıştırılan K-ortalama kümeleme algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	89,672 ±9,0	7,915 ±5,6	92,780 ±12,4	93,114 ±15,7	89,459 ±16,2	94,652 ±16,4	91,733 ±7,8	0,496 ±0,0	0,905 ±0,9
b	89,777 ±7,5	7,978 ±6,1	89,337 ±21,0	89,351 ±23,9	85,848 ±23,8	91,570 ±24,6	90,868 ±8,3	0,495 ±0,0	0,670 ±0,4

HBCE'den alınan tahmini küme merkezi bilgisi BBO algoritmasında başlangıç noktaları olarak verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.4'te verilmiştir. Yöntemlerin çalışma zamanları incelendiğinde en hızlı yöntemin küme merkezi bilgisi verilmeden (a) çalıştırılan BBO algoritması olduğu gözlemlenmiştir. Ancak elde edilen performans sonuçları değerlendirildiğinde BBO algoritması küme merkez bilgisi sadece bir parçacık için verilerek (b) çalıştırıldığında en iyi Doğruluk, DICE, Jaccard ve Hassasiyet; tüm parçacıklar için verilerek (c) çalıştırıldığında ise en iyi RI ve GCE değerlerine ulaştığı gözlemlenmiştir.

Çizelge 3.4. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan BBO algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	82,182 ±14,2	11,739 ±7,7	85,949 ±17,8	85,754 ±22,6	79,788 ±24,4	84,783 ±26,2	91,962 ±9,9	0,515 ±0,0	1,727 ±1,0
b	87,497 ±9,9	9,518 ±6,2	90,399 ±15,6	91,086 ±17,7	86,610 ±18,7	91,451 ±19,6	91,752 ±8,4	0,512 ±0,0	1,766 ±1,1
c	88,540 ±7,9	8,158 ±5,3	86,877 ±23,8	86,200 ±28,3	82,635 ±27,8	88,092 ±29,2	90,203 ±8,5	0,512 ±0,0	1,813 ±1,1
d	88,323 ±9,5	8,446 ±6,3	87,146 ±23,2	86,847 ±26,7	83,046 ±26,9	88,605 ±28,1	90,475 ±9,3	0,513 ±0,0	1,815 ±1,1

HBCE’den alınan tahmini küme merkezi bilgisi PSO algoritmasında başlangıç noktaları olarak verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.5’te verilmiştir. Yöntemlerin çalışma zamanları incelendiğinde en hızlı yöntemin küme merkezi bilgisi tüm parçacıklar için verilerek (c) çalıştırılan PSO algoritması olduğu gözlemlenmiştir. Ancak elde edilen performans sonuçları değerlendirildiğinde PSO algoritması küme merkez bilgisi sadece bir parçacık için verilerek (b) çalıştırıldığında en iyi RI, Doğruluk, DICE, Jaccard ve Hassasiyet değerlerine ulaştığı gözlemlenmiştir.

Çizelge 3.5. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan PSO algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	90,017 ±7,8	8,306 ±5,8	94,496 ±4,6	95,588 ±3,7	91,783 ±6,7	97,466 ±4,6	90,907 ±8,2	0,505 ±0,0	2,411 ±0,9
b	90,097 ±7,7	8,276 ±5,8	94,550 ±4,5	95,623 ±3,6	91,843 ±6,6	97,539 ±4,4	90,882 ±8,2	0,505 ±0,0	2,405 ±0,9
c	88,500 ±7,9	8,186 ±5,3	86,852 ±23,8	86,191 ±28,3	82,619 ±27,8	88,173 ±29,2	90,035 ±8,6	0,508 ±0,1	2,401 ±0,9
d	89,463 ±7,7	8,530 ±5,9	91,345 ±16,2	92,108 ±17,6	88,256 ±17,9	94,247 ±18,3	90,382 ±8,2	0,505 ±0,0	2,414 ±0,9

HBCE'den alınan tahmini küme merkezi bilgisi DPSO algoritmasında başlangıç noktaları olarak verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.6'da verilmiştir. Yöntemlerin çalışma zamanları incelendiğinde en hızlı yöntemin küme merkezi bilgisi tüm parçacıklar için verilerek (c) çalıştırılan DPSO algoritması olduğu gözlemlenmiştir. Ancak elde edilen performans sonuçları değerlendirildiğinde DPSO algoritması küme merkez bilgisi sadece bir parçacık için verilerek (b) çalıştırıldığında en iyi RI, Doğruluk, DICE, Jaccard ve Hassasiyet değerlerine ulaştığı gözlemlenmiştir.

Çizelge 3.6. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan DPSO algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	90,059 ±7,9	8,254 ±5,7	94,507 ±4,7	95,598 ±3,7	91,806 ±6,7	97,259 ±4,9	91,312 ±8,0	0,500 ±0,0	2,461 ±1,0
b	90,139 ±7,7	8,229 ±5,7	94,565 ±4,6	95,637 ±3,7	91,870 ±6,6	97,259 ±4,9	91,490 ±7,9	0,497 ±0,0	2,453 ±0,9
c	88,500 ±7,9	8,186 ±5,3	86,852 ±23,8	86,191 ±28,3	82,619 ±27,8	88,173 ±29,2	90,035 ±8,6	0,497 ±0,0	2,430 ±0,9
d	87,097 ±16,1	8,685 ±6,2	91,718 ±14,6	92,633 ±15,7	88,625 ±16,4	94,360 ±16,7	90,827 ±8,0	0,497 ±0,0	2,461 ±0,9

HBCE'den alınan tahmini küme merkezi bilgisi FODPSO algoritmasında başlangıç noktaları olarak verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.7'de verilmiştir. Yöntemlerin çalışma zamanları incelendiğinde en hızlı yöntemin küme merkezi bilgisi verilmeden (a) çalıştırılan FODPSO algoritması olduğu gözlemlenmiştir. Ancak elde edilen performans sonuçları değerlendirildiğinde FODPSO algoritması küme merkez bilgisi sadece bir parçacık için verilerek (b) çalıştırıldığında en iyi RI, Doğruluk, DICE, Jaccard ve Hassasiyet değerlerine ulaştığı gözlemlenmiştir.

Çizelge 3.7. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan FODPSO algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	90,028 ±7,9	8,248 ±5,7	94,483 ±4,8	95,583 ±3,8	91,782 ±6,7	97,204 ±5,1	91,315 ±8,0	0,505 ±0,0	2,478 ±0,9
b	90,119 ±7,8	8,239 ±5,7	94,542 ±4,7	95,617 ±3,8	91,842 ±6,7	97,240 ±5,0	91,420 ±7,9	0,505 ±0,0	2,504 ±1,0
c	88,500 ±7,9	8,186 ±5,3	86,852 ±23,8	86,191 ±28,3	82,619 ±27,8	88,173 ±29,2	90,035 ±8,6	0,505 ±0,0	2,519 ±1,1
d	87,256 ±15,8	8,633 ±6,2	91,820 ±14,4	92,722 ±15,6	88,739 ±16,2	94,464 ±16,5	90,713 ±8,0	0,505 ±0,0	2,553 ±1,1

HBCE'den alınan tahmini küme merkezi bilgisi mPSO algoritmasında başlangıç noktaları olarak verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.8'de verilmiştir. Yöntemlerin çalışma zamanları incelendiğinde en hızlı yöntemin küme merkezi bilgisi verilmeden (a) çalıştırılan mPSO algoritması olduğu gözlemlenmiştir. Ancak elde edilen performans sonuçları değerlendirildiğinde mPSO algoritması küme merkez bilgisi sadece bir parçacık için verilerek (b) çalıştırıldığında en iyi RI, Doğruluk, DICE, Jaccard ve Hassasiyet değerlerine ulaştığı gözlemlenmiştir.

Çizelge 3.8. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan mPSO algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	90,135 ±7,8	8,161 ±5,7	94,556 ±4,6	95,644 ±3,7	91,883 ±6,6	97,302 ±4,8	91,418 ±8,0	0,519 ±0,6	2,615 ±1,6
b	90,218 ±7,7	8,144 ±5,7	94,619 ±4,5	95,686 ±3,6	91,951 ±6,5	97,364 ±4,6	91,431 ±8,0	0,496 ±0,0	2,616 ±1,6
c	87,349 ±11,7	9,583 ±6,7	88,727 ±19,2	89,595 ±19,9	84,994 ±21,7	88,922 ±22,6	93,833 ±6,8	0,495 ±0,0	2,807 ±1,7
d	89,651 ±7,9	8,317 ±5,9	91,361 ±16,5	92,082 ±17,9	88,303 ±18,2	94,038 ±18,6	90,880 ±8,0	0,496 ±0,0	2,643 ±1,6

HBCE'den alınan tahmini küme merkezi bilgisi mDPSO algoritmasında başlangıç noktaları olarak verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.9'da verilmiştir. Yöntemlerin çalışma zamanları incelendiğinde en hızlı yöntemin küme merkezi bilgisi sadece bir parçacık için verilerek (b) çalıştırılan mDPSO algoritması olduğu gözlemlenmiştir. Bununla birlikte, elde edilen performans sonuçları değerlendirildiğinde mDPSO algoritması küme merkez bilgisi sadece bir parçacık için verilerek (b) çalıştırıldığında en iyi RI, Doğruluk, DICE, Jaccard ve Hassasiyet değerlerine ulaşabilmiştir.

Çizelge 3.9. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan mDPSO algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	90,211 ±7,9	8,094 ±5,6	94,593 ±4,7	95,675 ±3,7	91,938 ±6,6	97,168 ±5,0	91,721 ±7,8	0,496 ±0,0	2,677 ±1,6
b	90,318 ±7,6	8,096 ±5,6	94,677 ±4,5	95,728 ±3,6	92,025 ±6,4	97,251 ±4,7	91,741 ±7,8	0,494 ±0,0	2,668 ±1,6
c	87,349 ±11,7	9,583 ±6,7	88,727 ±19,2	89,595 ±19,9	84,994 ±21,7	88,922 ±22,6	93,833 ±6,8	0,494 ±0,0	2,840 ±1,7
d	89,222 ±8,8	8,698 ±6,0	93,628 ±7,6	94,763 ±7,7	90,722 ±9,7	96,244 ±9,2	91,257 ±7,8	0,499 ±0,2	2,686 ±1,6

HBCE'den alınan tahmini küme merkezi bilgisi mFODPSO algoritmasında başlangıç noktaları olarak verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) denemeler yapılmış ve elde edilen sonuçların ortalama değerleri Çizelge 3.10'da verilmiştir. Yöntemlerin çalışma zamanları incelendiğinde en hızlı yöntemin küme merkezi bilgisi verilmeden (a) çalıştırılan mFODPSO algoritması olduğu gözlemlenmiştir. Ancak elde edilen performans sonuçları değerlendirildiğinde mFODPSO algoritması küme merkez bilgisi sadece bir parçacık için verilerek (b) çalıştırıldığında en iyi RI, Doğruluk, DICE, Jaccard ve Hassasiyet değerlerine ulaştığı gözlemlenmiştir.

Çizelge 3.10. Tahmini küme merkezi bilgisi verilmeden (a), sadece bir parçacık için verilerek (b), tüm parçacıklar için verilerek (c) ve belirlenen ortalama aralıkta rastgele verilerek (d) çalıştırılan mFODPSO algoritmasına ait ortalama performans sonuçları.

Küme Merkezi Bilgisi	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
a	90,211 ±7,8	8,137 ±5,6	94,595 ±4,7	95,664 ±3,7	91,922 ±6,6	97,171 ±5,0	91,697 ±7,8	0,494 ±0,0	2,695 ±1,6
b	90,266 ±7,7	8,111 ±5,6	94,631 ±4,6	95,694 ±3,7	91,974 ±6,6	97,186 ±4,9	91,752 ±7,8	0,497 ±0,1	2,704 ±1,6
c	87,349 ±11,7	9,583 ±6,7	88,727 ±19,2	89,595 ±19,9	84,994 ±21,7	88,922 ±22,6	93,833 ±6,8	0,494 ±0,0	2,860 ±1,7
d	89,031 ±9,1	8,757 ±6,0	93,416 ±8,2	94,569 ±8,3	90,473 ±10,3	96,014 ±9,9	91,240 ±7,8	0,494 ±0,0	2,715 ±1,6

Küme sayısı bilgisi ile birlikte tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları Çizelge 3.11’de verilmiştir. Algoritmaların çalışma zamanları incelendiğinde en hızlı çözümlerin BBO algoritmasına ait olduğu gözlenmiştir. PSO, DPSO ve FODPSO algoritmalarının düzenlenmiş versiyonlarının ise orjinallerine göre daha yavaş oldukları saptanmıştır. HBCE’den alınan küme sayısı bilgisi ile birlikte tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan bölütleme algoritmaları performansları açısından değerlendirildiğinde mDPSO diğer algoritmalarla nazaran daha iyi sonuçlara ulaştığı gözlemlenmektedir. Standart sapma değerleri göz önünde bulundurulduğunda mDPSO diğer algoritmalarla göre daha kararlı sonuçlara ulaşmıştır.

Çizelge 3.11. Tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.

Bölütleme Algoritması	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
BBO	87,497 ±9,9	9,518 ±6,2	90,399 ±15,6	91,086 ±17,7	86,610 ±18,7	91,451 ±19,6	91,752 ±8,4	0,512 ±0,0	1,766 ±1,1
PSO	90,097 ±7,7	8,276 ±5,8	94,550 ±4,5	95,623 ±3,6	91,843 ±6,6	97,539 ±4,4	90,882 ±8,2	0,505 ±0,0	2,405 ±0,9
DPSO	90,139 ±7,7	8,229 ±5,7	94,565 ±4,6	95,637 ±3,7	91,870 ±6,6	97,259 ±4,9	91,490 ±7,9	0,497 ±0,0	2,453 ±0,9
FODPSO	90,119 ±7,8	8,239 ±5,7	94,542 ±4,7	95,617 ±3,8	91,842 ±6,7	97,240 ±5,0	91,420 ±7,9	0,505 ±0,0	2,504 ±1,0
mPSO	90,218 ±7,7	8,144 ±5,7	94,619 ±4,5	95,686 ±3,6	91,951 ±6,5	97,364 ±4,6	91,431 ±8,0	0,496 ±0,0	2,616 ±1,6
mDPSO	90,318 ±7,6	8,096 ±5,6	94,677 ±4,5	95,728 ±3,6	92,025 ±6,4	97,251 ±4,7	91,741 ±7,8	0,494 ±0,0	2,668 ±1,6
mFODPSO	90,266 ±7,7	8,111 ±5,6	94,631 ±4,6	95,694 ±3,7	91,974 ±6,6	97,186 ±4,9	91,752 ±7,8	0,497 ±0,1	2,704 ±1,6

Küme sayısı bilgisi ile birlikte tahmini küme merkezi bilgisi tüm parçacıklar için verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları Çizelge 3.12’de verilmiştir. Algoritmaların çalışma zamanları incelendiğinde en hızlı çözümlerin BBO algoritmasına ait olduğu gözlenmiştir. PSO, DPSO ve FODPSO algoritmalarının düzenlenmiş versiyonlarının ise orjinallerine göre daha yavaş oldukları saptanmıştır. HBCE’den alınan küme sayısı bilgisi ile birlikte tahmini küme merkezi bilgisi tüm parçacıklar için verilerek çalıştırılan bölütleme algoritmaları performansları açısından değerlendirildiğinde düzenlenmiş mPSO, mDPSO ve mFODPSO algoritmalarının diğer algoritmalarla nazaran daha iyi sonuçlara ulaştığı gözlemlenmektedir. Standart sapma değerleri göz önünde bulundurulduğunda mPSO, mDPSO ve mFODPSO algoritmalarının diğerlerine nazaran daha kararlı çalıştığı gözlemlenmiştir.

Çizelge 3.12. Tahmini küme merkezi bilgisi tüm parçacıklar için verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.

Bölütleme Algoritması	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
BBO	88,540 ±7,9	8,158 ±5,3	86,877 ±23,8	86,200 ±28,3	82,635 ±27,8	88,092 ±29,2	90,203 ±8,6	0,512 ±0,0	1,813 ±1,1
PSO	88,500 ±7,9	8,186 ±5,3	86,852 ±23,8	86,191 ±28,3	82,619 ±27,8	88,173 ±29,2	90,035 ±8,6	0,508 ±0,1	2,401 ±0,9
DPSO	88,500 ±7,9	8,186 ±5,3	86,852 ±23,8	86,191 ±28,3	82,619 ±27,8	88,173 ±29,2	90,035 ±8,6	0,497 ±0,0	2,430 ±0,9
FODPSO	88,500 ±7,9	8,186 ±5,3	86,852 ±23,8	86,191 ±28,3	82,619 ±27,8	88,173 ±29,2	90,035 ±8,6	0,505 ±0,0	2,519 ±1,1
mPSO	87,349 ±11,7	9,583 ±6,7	88,727 ±19,2	89,595 ±19,9	84,994 ±21,7	88,922 ±22,6	93,833 ±6,8	0,495 ±0,0	2,807 ±1,7
mDPSO	87,349 ±11,7	9,583 ±6,7	88,727 ±19,2	89,595 ±19,9	84,994 ±21,7	88,922 ±22,6	93,833 ±6,8	0,494 ±0,0	2,840 ±1,7
mFODPSO	87,349 ±11,7	9,583 ±6,7	88,727 ±19,2	89,595 ±19,9	84,994 ±21,7	88,922 ±22,6	93,833 ±6,8	0,494 ±0,0	2,860 ±1,7

Küme sayısı bilgisi ile birlikte tahmini küme merkezi bilgisi belirlenen ortalama aralıkta rastgele verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları Çizelge 3.13'te verilmiştir. Algoritmaların çalışma zamanları incelendiğinde en hızlı çözümlerin BBO algoritmasına ait olduğu gözlenmiştir. PSO, DPSO ve FODPSO algoritmalarının düzenlenmiş versiyonlarının ise orjinallerine göre daha yavaş oldukları saptanmıştır. HBCE'den alınan küme sayısı bilgisi ile birlikte tahmini küme merkezi bilgisi belirlenen ortalama aralıkta rastgele verilerek çalıştırılan bölütleme algoritmaları performansları açısından değerlendirildiğinde mDPSO algoritmasının diğer algoritmalarla nazaran daha iyi sonuçlara ulaştığı gözlemlenmektedir. Standart sapma değerleri göz önünde bulundurulduğunda mDPSO diğer algoritmalarla göre daha kararlı sonuçlara ulaşmıştır.

Çizelge 3.13. Tahmini küme merkezi bilgisi belirlenen ortalama aralıkta rastgele verilerek çalıştırılan bölütleme algoritmalarına ait ortalama performans sonuçları.

Bölütleme Algoritması	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
BBO	88,323 ±9,5	8,446 ±6,3	87,146 ±23,2	86,847 ±26,7	83,046 ±26,9	88,605 ±28,1	90,475 ±9,3	0,513 ±0,0	1,815 ±1,1
PSO	89,463 ±7,7	8,530 ±5,9	91,345 ±16,2	92,108 ±17,6	88,256 ±17,9	94,247 ±18,3	90,382 ±8,2	0,505 ±0,0	2,414 ±0,9
DPSO	87,097 ±16,1	8,685 ±6,2	91,718 ±14,6	92,633 ±15,7	88,625 ±16,4	94,360 ±16,7	90,827 ±8,0	0,497 ±0,0	2,461 ±0,9
FODPSO	87,256 ±15,8	8,633 ±6,2	91,820 ±14,4	92,722 ±15,6	88,739 ±16,2	94,464 ±16,5	90,713 ±8,0	0,505 ±0,0	2,553 ±1,1
mPSO	89,651 ±7,9	8,317 ±5,9	91,361 ±16,5	92,082 ±17,9	88,303 ±18,2	94,038 ±18,6	90,880 ±8,0	0,496 ±0,0	2,643 ±1,6
mDPSO	89,222 ±8,8	8,698 ±6,0	93,628 ±7,6	94,763 ±7,7	90,722 ±9,7	96,244 ±9,2	91,257 ±7,8	0,499 ±0,2	2,686 ±1,6
mFODPSO	89,031 ±9,1	8,757 ±6,0	93,416 ±8,2	94,569 ±8,3	90,473 ±10,3	96,014 ±9,9	91,240 ±7,8	0,494 ±0,0	2,715 ±1,6

ISIC veritabanından alınan görüntüleri üzerinde gerçekleştirilen çeşitli testler sonucunda en iyi sonuçların DWT filtre ile filtrelenmiş histogram üzerinde çalıştırılan v-h HBCE algoritmasından elde edilen küme sayısı ve tahmini küme merkezi bilgisinin sadece bir parçacık için mDPSO bölütleme algoritmasına verilerek alındığı gözlemlenmiştir. mDPSO algoritmasıyla gerçekleştirilen testlere ait her bir görüntüden elde edilen ortalama değerler Çizelge 3.14’te verilmiştir.

Çizelge 3.14. Tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan mDPSO bölütleme algoritmasının her bir test görüntüsüne ait ortalama performans sonuçları.

Görüntü	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
1	98,667 ±0,1	1,322 ±0,1	99,329 ±0,1	99,499 ±0,0	99,003 ±0,1	99,857 ±0,0	98,269 ±0,2	0,498 ±0,0	1,446 ±1,5
2	80,190 ±3,7	14,998 ±2,1	88,779 ±2,4	92,098 ±1,9	85,406 ±3,2	85,464 ±3,3	99,791 ±0,8	0,486 ±0,0	1,603 ±2,2
3	73,971 ±3,1	10,618 ±0,5	84,551 ±2,2	90,800 ±1,5	83,181 ±2,4	83,181 ±2,4	99,999 ±0,0	0,495 ±0,0	1,645 ±3,1
4	98,373 ±0,4	1,587 ±0,4	99,179 ±0,2	99,452 ±0,1	98,912 ±0,3	98,933 ±0,3	99,935 ±0,1	0,481 ±0,0	1,405 ±1,1
5	97,089 ±0,3	2,846 ±0,3	98,523 ±0,2	99,127 ±0,1	98,269 ±0,2	99,166 ±0,1	94,995 ±1,5	0,511 ±0,0	3,161 ±3,4
6	97,911 ±0,2	2,038 ±0,2	98,944 ±0,1	99,319 ±0,1	98,647 ±0,2	98,856 ±0,3	99,253 ±0,6	0,495 ±0,0	2,882 ±2,9
7	80,463 ±0,0	18,452 ±0,0	89,027 ±0,0	91,455 ±0,0	84,256 ±0,0	94,801 ±0,0	79,631 ±0,0	0,492 ±0,0	3,464 ±1,4
8	82,580 ±0,7	15,773 ±0,6	90,358 ±0,4	88,644 ±0,4	79,607 ±0,7	98,383 ±0,0	85,390 ±0,7	0,502 ±0,0	3,156 ±3,3
9	84,252 ±0,3	14,167 ±0,2	91,383 ±0,2	92,997 ±0,1	86,912 ±0,2	98,975 ±0,0	80,982 ±0,4	0,488 ±0,0	3,217 ±1,8
10	80,448 ±0,4	17,875 ±0,3	89,017 ±0,3	89,853 ±0,2	81,576 ±0,4	98,263 ±0,1	79,961 ±0,6	0,513 ±0,0	3,334 ±2,5
11	97,456 ±0,4	2,493 ±0,4	98,711 ±0,2	99,023 ±0,2	98,066 ±0,3	98,213 ±0,4	99,702 ±0,2	0,495 ±0,0	3,016 ±2,3
12	96,828 ±0,8	3,029 ±0,7	98,386 ±0,4	98,901 ±0,3	97,827 ±0,6	97,849 ±0,6	99,938 ±0,2	0,483 ±0,0	3,149 ±2,6
13	86,372 ±0,5	12,548 ±0,4	92,645 ±0,3	93,139 ±0,2	87,159 ±0,4	100,000 ±0,0	85,313 ±0,5	0,493 ±0,0	1,234 ±1,9
14	97,892 ±0,7	2,073 ±0,6	98,934 ±0,3	99,181 ±0,3	98,377 ±0,5	98,553 ±0,6	99,660 ±0,2	0,478 ±0,0	1,362 ±1,8
15	97,505 ±0,6	2,404 ±0,5	98,736 ±0,3	99,154 ±0,2	98,323 ±0,4	99,955 ±0,0	95,243 ±1,2	0,479 ±0,0	5,669 ±6,7
16	97,144 ±0,2	2,806 ±0,2	98,551 ±0,1	99,101 ±0,1	98,219 ±0,2	99,051 ±0,4	96,462 ±1,1	0,491 ±0,0	5,689 ±6,3

Çizelge 3.14. (devam). Tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan mDPSO bölütleme algoritmasının her bir test görüntüsüne ait ortalama performans sonuçları.

Görüntü	RI (%)	GCE (%)	Doğruluk (%)	DICE (%)	Jaccard (%)	Hassasiyet (%)	Özgünlük (%)	HBCE Çalışma Zamanı (ms)	Bölütleme Çalışma Zamanı (s)
17	97,127 ±0,3	2,710 ±0,2	98,542 ±0,1	99,128 ±0,1	98,272 ±0,1	99,819 ±0,1	92,290 ±1,1	0,483 ±0,0	5,572 ±9,5
18	92,089 ±0,2	7,382 ±0,2	95,874 ±0,1	97,232 ±0,1	94,612 ±0,1	98,941 ±0,5	87,487 ±1,5	0,504 ±0,0	5,824 ±7,2
19	78,613 ±9,9	10,541 ±3,0	87,238 ±6,7	92,220 ±4,4	85,858 ±7,5	86,146 ±7,7	97,141 ±2,8	0,496 ±0,0	6,012 ±25,1
20	85,973 ±0,4	13,120 ±0,3	92,410 ±0,2	93,012 ±0,2	86,938 ±0,3	99,003 ±0,1	85,543 ±0,5	0,503 ±0,0	1,221 ±1,8
21	91,983 ±0,3	7,953 ±0,2	95,816 ±0,1	96,459 ±0,1	93,159 ±0,2	97,286 ±0,2	93,739 ±0,7	0,494 ±0,0	1,270 ±2,0
22	97,418 ±0,3	2,545 ±0,3	98,692 ±0,2	98,999 ±0,1	98,019 ±0,3	99,607 ±0,2	96,996 ±0,8	0,502 ±0,0	1,231 ±1,5
23	89,868 ±0,5	9,930 ±0,5	94,647 ±0,3	93,758 ±0,3	88,251 ±0,5	96,508 ±0,6	93,318 ±0,9	0,497 ±0,0	3,182 ±2,9
24	86,413 ±0,9	11,403 ±0,6	92,666 ±0,6	94,899 ±0,4	90,296 ±0,7	99,749 ±0,0	77,354 ±1,8	0,499 ±0,0	2,849 ±5,9
25	88,283 ±0,3	9,715 ±0,2	93,751 ±0,2	95,938 ±0,1	92,193 ±0,2	99,672 ±0,0	76,860 ±0,7	0,497 ±0,0	1,149 ±1,4
26	93,226 ±0,0	6,370 ±0,0	96,490 ±0,0	97,260 ±0,0	94,666 ±0,0	99,986 ±0,0	90,710 ±0,0	0,497 ±0,0	0,810 ±1,1
27	95,010 ±0,9	4,890 ±0,9	97,437 ±0,5	97,401 ±0,5	94,938 ±0,9	99,587 ±0,2	95,439 ±1,1	0,498 ±0,0	1,202 ±1,7
28	81,903 ±0,8	15,914 ±0,6	89,936 ±0,5	91,266 ±0,4	83,937 ±0,7	99,827 ±0,0	78,934 ±1,1	0,500 ±0,0	1,222 ±1,7
29	87,411 ±0,4	12,573 ±0,4	93,250 ±0,2	93,466 ±0,1	87,734 ±0,3	93,487 ±0,6	92,996 ±1,0	0,491 ±0,0	1,676 ±2,9
30	97,075 ±0,7	2,807 ±0,6	98,514 ±0,4	99,058 ±0,2	98,134 ±0,5	98,415 ±0,6	98,897 ±0,5	0,495 ±0,0	1,386 ±2,4

4. SONUÇLAR VE ÖNERİLER

Bu çalışmada iki boyutlu medikal görüntülerde optimizasyon tabanlı yarı otomatik bir bölütleme algoritması geliştirilmiştir. Görüntü bölütleme, görüntünün anlamlı bölgelere ayrılması işlemi olup, en önemli aşaması görüntüdeki bölge sayısının tespit edilmesidir. Başta K-ortalama ve Bulanık C-ortalama olmak üzere bölütleme amacı ile kullanılan birçok kümeleme algoritması girdi olarak küme sayısını istemektedir. Otomatik teşhis sistemleri için önemli aşamalardan bir tanesi bu görüntülerdeki optimum bölge sayısının bulunması problemidir ki bu problem bir optimizasyon problemidir. Bu çalışmada, görüntü bölütlemeye bölge sayısının ve bölge merkezlerinin otomatik olarak bulunması amacıyla histogram tabanlı bir küme kestirim metodu geliştirilmiştir. Bu algoritma türeve dayalı bir algoritmadır. Eğer histogram üzerinde bir noktada, artıştan azalmaya geçen bölgeler var ise bu bölgeler histogramın tepe noktaları olup, küme merkezleri olarak düşünülebilirler. Algoritma, bu yöntemle histogram üzerinde yatay ve dikeydeki farklara dayalı olarak tepe noktalarını tesbit etmektedir. Bu algoritma ile elde edilen tahmini küme merkezleri ve küme sayısı bilgileri, optimizasyon algoritmalarına girdi olarak verilerek, bölütleme işlemi gerçekleştirilmektedir. Ancak histogram üzerindeki küçük değişimlerin filtrelenmesi HBCE algoritmasının başarısında önemli paya sahiptir. Bu amaçla HBCE algoritması ile doğru küme sayısı ve küme merkezi bulunabilmesi için, görüntünün histogramı bir filtreden geçirilmektedir. Bu çalışmada istenilen filtrenin manuel olarak seçilmesi sağlanmıştır. Filtre seçildikten sonra, görüntü bölütleme için kullanılan algoritma seçimi de manuel olarak yapılmaktadır. Geliştirilen arayüz, yapılan bu manuel seçimlere bağlı olarak çalıştığı için, yarı otomatik bir bölütleme sistemi olarak adlandırılmıştır. Ayrıca gerçekleştirilen çalışmanın diğer görüntü türlerine de uygulanabileceği gözlemlenmiştir.

Çalışmanın birinci bölümünde konu hakkında genel bilgilere yer verilmiş, ikinci bölümde ise kullanılan materyaller anlatıldıktan sonra geliştirilen yöntem parçalar halinde tanıtılmıştır. Geliştirilen Histogram Tabanlı Küme Kestirim algoritması BRATS ve ISIC veritabanlarından elde edilen görüntülerle test edilmiş ve zorlu bir problem olan küme sayısı kestiriminde kullanılabilirliği sergilenmiştir. Tez kapsamında bölütleme amacı ile kullanılan K-ortalama, BBO, PSO, DPSO ve FODPSO algoritmaları kısaca

tanıtılmış ve çalışma kapsamında geliştirilen merkez tabanlı mPSO, mDPSO ve mFODPSO algoritmaları anlatılmıştır. Son olarak çalışmanın diğer kullanıcılar tarafından da kolayca uygulanabilir olması için MATLAB™ ortamında bir arayüz geliştirilmiş ve tüm çalışma bu arayüz içerisinde toplanmıştır.

Çalışmanın üçüncü bölümünde ISIC veritabanından elde edilen örnek görüntüler üzerinde denemeler yapılmış ve bu denemelere ait ekran görüntüleri ile algoritmaların çalışmaları sergilenmiştir. Gerçekleştirilen çalışmalar, çeşitli performans kriterleri kullanılarak insan bölütleme sonuçları ile karşılaştırılmış ve sonuçlar sergilenmiştir.

Yapılan karşılaştırmalı testler sonucunda;

- HBCE'den elde edilen küme sayısı bilgisiyle çalıştırılan bölütleme algoritmaları performansları açısından değerlendirildiğinde mPSO (%94,556 doğruluk), mDPSO (%94,593 doğruluk) ve mFODPSO (%94,595 doğruluk) algoritmalarının diğer algoritmalara nazaran daha iyi sonuçlara ulaştığı gözlemlenmiştir.
- HBCE algoritmasının en kötü durumunda bile 0,52ms sürede çözüme ulaştığı gözlemlenmiştir.
- HBCE'den elde edilen küme sayısı bilgisiyle birlikte tahmini küme merkezleri bilgisinin sadece bir parçacık için bölütleme algoritmalarına başlangıç değeri olarak verildiğinde tüm bölütleme algoritmalarının sonuçlarında iyileşme olduğu gözlemlenmiştir. Ayrıca tüm bölütleme algoritmalarının çözümlerinin daha kararlı hale geldiği sonucuna da varılmıştır.
- HBCE'den alınan küme sayısı bilgisi ile birlikte tahmini küme merkezi bilgisi sadece bir parçacık için verilerek çalıştırılan bölütleme algoritmaları performansları açısından değerlendirildiğinde mDPSO'nun diğer algoritmalara nazaran daha iyi sonuçlara (%94,677 doğruluk) ulaştığı gözlemlenmiştir. Standart sapma değerleri göz önünde bulundurulduğunda mDPSO'nun diğer algoritmalara göre daha kararlı sonuçlar elde ettiği gözlemlenmiştir.
- Algoritmaların çalışma zamanları göz önünde bulundurulduğunda, mDPSO algoritması için HBCE'den elde edilen tahmini küme merkezi bilgisi sadece bir parçacık için verildiğinde, hiç verilmeden kullanılan versiyonuna nazaran daha hızlı çalıştığı gözlemlenmiştir.

İlerideki çalışmalarda;

- HBCE algoritması, daha farklı görüntüler üzerinde de denenerek küme sayısı ve tahmini küme merkezi bilgisinin elde edilmesinde kullanılabilir.
- HBCE algoritmasının renkli görüntülerle de çalışabilen bir versiyonu geliştirilebilir.
- HBCE algoritmasının farklı bölütleme algoritmalarıyla çalışması ve performansı değerlendirilebilir.



5. KAYNAKLAR

- [1] E. Anjna and E. R. Kaur, "Review of image segmentation technique," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 4, pp. 36-39, 2017.
- [2] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.
- [3] T. M. Deserno, "Fundamentals of biomedical image processing," in *Biomedical Image Processing*, T. M. Deserno, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 27–36.
- [4] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," *Proceedings of the 2nd International Joint Conference on Pattern Recognition*, Copenhagen, Denmark, 1974, pp. 424–433.
- [5] T. Pavlidis, *Structural Pattern Recognition*, Heidelberg, Berlin: Springer, 2013.
- [6] T. M. Lehmann and E. Meyer zu Bexten, *Handbuch der Medizinischen Informatik*, Berlin, München: Hanser, 2005.
- [7] C. K. Chow and T. Kaneko, "Boundary detection and volume determination of the left ventricle from a cineangiogram," *Computers in Biology and Medicine*, vol. 3, no. 1, pp. 13–26, 1973.
- [8] X. Muñoz Pujol, "Image segmentation integrating colour, texture and boundary information," Ph.D. dissertation, Electronics Informatics and Automation, Universitat de Girona, Girona, Spain, 2002.
- [9] J. Macqueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, USA, 1967, pp. 281-297.
- [10] T. M. Deserno, *Biomedical Image Processing*, Heidelberg, Berlin: Springer, 2011, pp. 27-34.
- [11] W. Pratt, *Digital Image Processing*, 3rd ed., New York, USA: Oxford University Publication, 2001.
- [12] V. K. Dehariya, S. K. Shrivastava, and R. C. Jain, "Clustering of image data set using k-means and fuzzy k-means algorithms," in *2010 International Conference on Computational Intelligence and Communication Networks*, Bhopal, 2010, pp. 386–391.
- [13] W.-X. Kang, Q.-Q. Yang, and R.-P. Liang, "The comparative research on image segmentation algorithms," in *2009 First International Workshop on Education Technology and Computer Science*, Wuhan, Hubei, 2009, pp. 703–707.
- [14] M. Yambal and H. Gupta, "Image segmentation using fuzzy C means clustering: a survey," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 7, pp. 2927-2929, 2013.

- [15] T. F. Wang and D. Y. Li, "Automatic segmentation of medical ultrasound image using self-creating and organizing neural network," *Journal of Electronics*, vol. 21, no. 1, pp. 124–127, 1999.
- [16] Y. L. Huang and D. R. Chen, "Watershed segmentation for breast tumor in 2-D sonography," *Ultrasound in Medicine & Biology*, vol. 30, no. 5, pp. 625–632, 2004.
- [17] T. Maintz, Digital and medical image processing, *Lecture Notes*, Universiteit Utrecht, 2005.
- [18] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 192–207, 2018.
- [19] Z. Li and J. Zheng, "Single image de-hazing using globally guided image filtering," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 442–450, 2018.
- [20] H. M. Ali, "MRI medical image denoising by fundamental filters," *SCIREA Journal of Computer*, vol. 2, no. 1, pp. 2–26, 2017.
- [21] M. Agarwal and R. Mahajan, "Medical image contrast enhancement using range limited weighted histogram equalization," *Procedia Computer Science*, vol. 125, pp. 149–156, 2018.
- [22] C. T. Lu, M. Y. Chen, J. H. Shen, L. L. Wang, N. Y. Yen, and C. H. Liu, "X-ray bio-image denoising using directional-weighted-mean filtering and block matching approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 18, pp. 1–18, 2018.
- [23] A. Garg and V. Khandelwal, "Combination of spatial domain filters for speckle noise reduction in ultrasound medical images," *Advances in Electrical and Electronic Engineering*, vol. 15, no. 5, pp. 857–865, 2018.
- [24] Z. Kong, L. Han, X. Liu, and X. Yang, "A new 4-D nonlocal transform-domain filter for 3-D magnetic resonance images denoising," *IEEE transactions on medical imaging*, vol. 37, no. 4, pp. 941–954, 2018.
- [25] N. Nabizadeh, N. John, and C. Wright, "Histogram-based gravitational optimization algorithm on single MR modality for automatic brain lesion detection and segmentation," *Expert systems with applications*, vol. 41, no. 17, pp. 7820–7836, 2014.
- [26] C. R. Mohan and S. Kiran, "Image enrichment using single discrete wavelet transform multi-resolution and frequency partition," *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, Springer, Singapore, 2018, pp. 87–98.
- [27] A. Subhashini, "A new approach on denoising for 1d, 2d and 3d images based on discrete wavelet transformation and thresholding," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 1, pp. 708–710, 2018.
- [28] R. Sharma, R. Pachori, and U. Acharya, "An integrated index for the identification of focal electroencephalogram signals using discrete wavelet transform and entropy measures," *Entropy*, vol. 17, no. 12, pp. 5218–5240, 2015.
- [29] A. Chaudhary and S. S. Singh, "Lung cancer detection on ct images by using

- image processing,” in *2012 International Conference on Computing Sciences*, India, 2012, pp. 142–146.
- [30] A. Gupta and D. Singhal, “Analytical global median filtering forensics based on moment histograms,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 14, no. 2, pp. 1–23, 2018.
 - [31] N. Rauf, D. Y. Alam, M. Jamaluddin, and B. A. Samad, “Improve image quality of transversal relaxation time propeller and flair on magnetic resonance imaging,” *Journal of Physics: Conference Series*, vol. 979, no. 1, p. 012079, 2018.
 - [32] L. Djerou, N. Khelil, H. E. Dehimi, and M. Batouche, “Automatic multilevel thresholding using binary particle swarm optimization for image segmentation,” in *2009 International Conference of Soft Computing and Pattern Recognition*, Malacca, 2009, pp. 66–71.
 - [33] E. Cuevas, D. Zaldivar, and M. Pérez-Cisneros, “A novel multi-threshold segmentation approach based on differential evolution optimization,” *Expert Systems with Applications*, vol. 37, no. 7, pp. 5265–5271, 2010.
 - [34] L. Wen, X. Wang, Z. Wu, M. Zhou, and J. S. Jin, “A novel statistical cerebrovascular segmentation algorithm with particle swarm optimization,” *Neurocomputing*, vol. 148, pp. 569–577, 2015.
 - [35] U. Güvenç, Ç. Elmas, ve R. Demirci, “Renkli görüntülerin otomatik ayrıştırılması,” *Politeknik Dergisi*, c. 11, s. 1, ss. 9-12, 2008.
 - [36] R. Demirci, U. Güvenç, ve H. T. Kahraman, “Görüntülerin renk uzayı yardımıyla ayrıştırılması,” *İleri Teknoloji Bilimleri Dergisi*, c. 3, s. 1, ss. 1–8, 2014.
 - [37] L. Qu, X. Dong, and F. Guo, “Automatic K-means for color enteromorpha image segmentation,” in *2009 Third International Symposium on Intelligent Information Technology Application*, Nanchang, 2009, vol. 3, pp. 224–227.
 - [38] H. Azami, M. Azarbad, and S. Sanei, “Segmentation of medical images based on hierarchical evolutionary and bee algorithms,” in *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Southampton, 2013, pp. 1–6.
 - [39] A. K. Bhandari, V. K. Singh, A. Kumar, and G. K. Singh, “Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur’s entropy,” *Expert Systems with Applications*, vol. 41, no. 7, pp. 3538–3560, 2014.
 - [40] P. Smyth, “Clustering using monte carlo cross-validation,” in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, 1996, vol. 1, pp. 26–133.
 - [41] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “Wavecluster: A multi-resolution clustering approach for very large spatial databases,” in *Proc. 24th VLDB Conference*, 1998, vol. 98, pp. 428–439.
 - [42] K. T. Vasko and H. T. T. Toivonen, “Estimating the number of segments in time series data using permutation tests,” in *2002 IEEE International Conference on Data Mining*, 2002, pp. 466–473.
 - [43] V. Roth, T. Lange, M. Braun, and J. Buhmann, *A resampling approach to cluster validation*, Compstat, Heidelberg: Physica-Verlag HD, 2002, pp. 123–128.

- [44] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, no. 1/2, pp. 91–118, 2003.
- [45] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.
- [46] R. Tibshirani and G. Walther, "Cluster validation by prediction strength," *Journal of Computational and Graphical Statistics*, vol. 14, no. 3, pp. 511–528, 2005.
- [47] A. Foss and O. R. Zaiane, "A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets," in *2002 IEEE International Conference on Data Mining*, Japan, 2002, pp. 179–186.
- [48] T. Chiu, D. Fang, J. Chen, Y. Wang, and C. Jeris, "A robust and scalable clustering algorithm for mixed type attributes in large database environment," in *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '01*, 2001, pp. 263–268.
- [49] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *ACM Sigmod Record*, 1996, vol. 25, no. 2, pp. 103–114.
- [50] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *16th IEEE International Conference on Tools with Artificial Intelligence*, 2004, pp. 576–584.
- [51] J. Pei, L. Zhao, X. Dong, and X. Dong, "Effective algorithm for determining the number of clusters and its application in image segmentation," *Cluster Computing*, vol. 20, no. 4, pp. 2845–2854, 2017.
- [52] B. H. Menze *et al.*, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2004, 2015.
- [53] MICCAI BRATS 2012. (2018, May 03). [Online]. Available: <http://www2.imm.dtu.dk/projects/BRATS2012/>
- [54] I. Cabria and I. Gondra, "Automated localization of brain tumors in mri using potential-k-means clustering algorithm," in *Computer and Robot Vision (CRV), 2015 12th Conference*, Halifax, NS, 2015, pp. 125–132.
- [55] ISIC Archive. (2018, May 01). [Online]. Available: <https://isic-archive.com/>
- [56] N. C. F. Codella *et al.*, "Skin lesion analysis toward melanoma detection: a challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)," *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium*, Washington, DC, 2017, pp. 168–172.
- [57] N. Codella, J. Cai, M. Abedini, R. Garnavi, A. Halpern, and J. R. Smith, "Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images," *International Workshop on Machine Learning in Medical Imaging*, Cham, 2015, pp. 118–126.

- [58] A. Esteva *et al.*, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [59] M. A. Marchetti *et al.*, “Results of the 2016 International Skin Imaging Collaboration International Symposium on Biomedical Imaging challenge: Comparison of the accuracy of computer algorithms to dermatologists for the diagnosis of melanoma from dermoscopic images,” *Journal of the American Academy of Dermatology*, vol. 78, no. 2, p. 270–277, 2018.
- [60] A. N. Benaichouche, H. Oulhadj, and P. Siarry, “Improved spatial fuzzy c-means clustering for image segmentation using PSO initialization, Mahalanobis distance and post-segmentation correction,” *Digit. Signal Processing*, vol. 23, no. 5, pp. 1390–1400, 2013.
- [61] S. Mao and T. Kanungo, “Automatic training of page segmentation algorithms: An optimization approach,” in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Barcelona, 2000, vol. 4, pp. 531–534.
- [62] S. Koley and A. Majumder, “Brain MRI segmentation for tumor detection using cohesion based self merging algorithm,” in *011 IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, 2011, pp. 781–785.
- [63] M. T. Rostami, J. Ghasemi, and R. Ghaderi, “Neural network for enhancement of FCM based brain MRI segmentation,” in *2013 13th Iranian Conference on Fuzzy Systems (IFSC)*, Qazvin, 2013, pp. 1–4.
- [64] T. Si, A. De, and A. K. Bhattacharjee, “Brain MRI segmentation for tumor detection using grammatical swarm based clustering algorithm,” in *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]*, Nagercoil, 2014, pp. 1196–1201.
- [65] O. Moh’d Alia, R. Mandava, and M. E. Aziz, “A hybrid harmony search algorithm for MRI brain segmentation,” *Evolutionary Intelligence*, vol. 4, no. 1, pp. 31–49, 2011.
- [66] M. Hasanzadeh and S. Kasaei, “Multispectral brain MRI segmentation using genetic fuzzy systems,” in *2007 9th International Symposium on Signal Processing and Its Applications*, Saharjah, 2007, pp. 1–4.
- [67] J. Liu and L. Guo, “A new brain MRI image segmentation strategy based on K-means clustering and SVM,” in *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, 2015, vol. 2, pp. 270–273.
- [68] J. Liu and L. Guo, “A new brain MRI image segmentation strategy based on wavelet transform and K-means clustering,” in *2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Ningbo, 2015, pp. 1–4.
- [69] K. Sinha and G. R. Sinha, “Efficient segmentation methods for tumor detection in MRI images,” in *Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference*, Bhopal, 2014, pp. 1–6.
- [70] N. G. Raju and P. A. N. Rao, “Particle swarm optimization methods for image segmentation applied in mammography,” *International Journal of Engineering Research and Applications*, vol. 3, no. 6, pp. 1572–1579, 2013.
- [71] A. Agarwal, A. Issac, M. K. Dutta, K. Riha, and V. Uher, “Automated skin lesion

- segmentation using K-Means clustering from digital dermoscopic images,” in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, Barcelona, 2017, pp. 743–748.
- [72] K. Eltayef, Y. Li, and X. Liu, “Lesion segmentation in dermoscopy images using particle swarm optimization and markov random field,” in *IEEE International Symposium on Computer-Based Medical Systems*, Thessaloniki, 2017, pp. 739–744.
 - [73] S. N. Sulaiman and N. A. M. Isa, “Adaptive fuzzy-K-means clustering algorithm for image segmentation,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2661–2668, 2010.
 - [74] T. Moriya *et al.*, “Unsupervised pathology image segmentation using representation learning with spherical k-means,” in *Medical Imaging 2018: Digital Pathology*, vol. 10581, pp. 10581 - 10588, 2018.
 - [75] H. Modi, N. Baraiya, and H. Patel, “Comparative analysis of segmentation of tumor from brain MRI images using Fuzzy C-Means and K-Means,” *Fuzzy Systems*, vol. 10, no. 1, pp. 14–18, 2018.
 - [76] S. K. Chahal and D. S. Sidhu, “Image segmentation using biogeography based optimization (BBO),” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 8, pp. 2436–2440, 2014.
 - [77] S. Gupta, K. Bhuchar, and P. S. Sandhu, “Implementing color image segmentation using biogeography based optimization,” in *2011 International Conference on Software and Computer Applications IPCSIT*, Singapore, 2011, vol. 9, pp. 79–86.
 - [78] O. B. Haddad, S.-M. Hosseini-Moghari, and H. A. Loáiciga, “Biogeography-based optimization algorithm for optimal operation of reservoir systems,” *Journal of Water Resources Planning and Management*, vol. 142, no. 1, p. 4015034, 2015.
 - [79] D. Simon, “Biogeography-based optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
 - [80] S. Singh, S. Tayal, and G. Sachdeva, “Evolutionary performance of BBO and PSO algorithms for Yagi-Uda antenna design optimization,” in *2012 World Congress on Information and Communication Technologies*, Trivandrum, 2012, pp. 861–865.
 - [81] M. Tamjidy, S. Paslar, B. T. H. T. Baharudin, T. S. Hong, and M. K. A. Ariffin, “Biogeography based optimization (BBO) algorithm to minimise non-productive time during hole-making process,” *International Journal of Production Research*, vol. 53, no. 6, pp. 1880–1894, 2015.
 - [82] L. Wang and Y. Xu, “An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems,” *Expert Systems with Applications*, vol. 38, no. 12, pp. 15103–15109, 2011.
 - [83] X. Zhang, Q. Kang, J. Cheng, and X. Wang, “A novel hybrid algorithm based on Biogeography-Based Optimization and Grey Wolf Optimizer,” *Applied Soft Computing*, vol. 67, pp. 197–214, 2018.
 - [84] R. Adalarasan and R. Malathi, “Automatic detection of blood vessels in digital retinal images using soft computing technique,” *Materials Today: Proceedings*,

- vol. 5, no. 1, pp. 1950–1959, 2018.
- [85] T. X. Pham, P. Siarry, and H. Oulhadj, “Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation,” *Applied Soft Computing*, vol. 65, pp. 230–242, 2018.
 - [86] S. Kumar, M. Pant, M. Kumar, and A. Dutt, “Colour image segmentation with histogram and homogeneity histogram difference using evolutionary algorithms,” *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 1, pp. 163–183, 2018.
 - [87] L. Na, J. Yan, and L. Shu, “Application of PSO algorithm with dynamic inertia weight in medical image thresholding segmentation,” in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Dalian, 2017, pp. 1–4.
 - [88] J. Tillett, T. Rao, F. Sahin, and R. Rao, Darwinian particle swarm optimization, *Present. other Scholarsh*, Rochester Institute of Technology, 2005.
 - [89] M. Couceiro, P. Ghamisi, *Fractional Order Darwinian Particle Swarm Optimization: Applications and Evaluation of an Evolutionary Algorithm*, London:Springer, 2015.
 - [90] E. J. Solteiro Pires, J. A. Tenreiro Machado, P. B. de Moura Oliveira, J. Boaventura Cunha, and L. Mendes, “Particle swarm optimization with fractional-order velocity,” *Nonlinear Dynamics*, vol. 61, no. 1–2, pp. 295–301, 2010.
 - [91] K. Hammouche and M. Diaf, “A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 676–688, 2010.
 - [92] P. D. Sathya and R. Kayalvizhi, “PSO-Based tsallis thresholding selection procedure for image segmentation,” *International Journal of Computer Applications*, vol. 5, no. 4, pp. 39–46, 2010.
 - [93] M. Jiang, Y. P. Luo, and S. Y. Yang, “Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm,” *Information Processing Letters*, vol. 102, no. 1, pp. 8–16, 2007.
 - [94] R. V. Kulkarni and G. K. Venayagamoorthy, “Bio-inspired algorithms for autonomous deployment and localization of sensor nodes,” *IEEE Transactions on Systems*, vol. 40, no. 6, pp. 663–675, 2010.
 - [95] M. Venkata dasu, P. VeeraNarayana Reddy, and S. Chandra Mohan Reddy, *A Proposal on Application of Nature Inspired Optimization Techniques on Hyper Spectral Images*, Singapore: Springer, 2018, pp. 309–318.
 - [96] M. M, S. Winiarti, A. Prahara, and E. Febri, “Evolutionary optimization methods for animal leather image segmentation,” *International Journal of Computer Applications*, vol. 179, no. 17, pp. 31–35, 2018.
 - [97] M. S. Couceiro, N. M. Fonseca Ferreira, and J. A. Tenreiro Machado, “Application of fractional algorithms in the control of a robotic bird,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 4, pp. 895–910, 2010.
 - [98] S. Akbar, F. Zaman, M. Asif, A. U. Rehman, and M. A. Z. Raja, “Novel application of FO-DPSO for 2-D parameter estimation of electromagnetic plane

- waves,” *Neural Computing and Applications*, pp. 1–10, 2018.
- [99] S. Lahmiri, “Glioma detection based on multi-fractal features of segmented brain MRI by particle swarm optimization techniques,” *Biomedical Signal Processing and Control*, vol. 31, pp. 148–155, 2017.
 - [100] Y. Fu, C. Su, J. Liu, Qing Gao, T. Zhang, and Tongtong Tian, “Vision-based autonomous docking for self-reconfigurable CubeSats,” in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2016, pp. 780–787.
 - [101] M. Sandeli and M. Batouche, “Multilevel thresholding for image segmentation based on parallel distributed optimization,” in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, 2014, pp. 134–139.
 - [102] E. Küçükkülahlı, P. Erdoğan, and K. Polat, “Brain MRI segmentation based on different clustering algorithms,” *International Journal of Computer Applications*, vol. 155, no. 3, pp. 37–40, 2016.
 - [103] D. Gutman *et al.*, “Skin lesion analysis toward melanoma detection: a challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC),” *ArXiv e-prints*, 2016.
 - [104] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels,” *Technical report*, 2010.
 - [105] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
 - [106] C. Tepe ve H. Sezgin, “EKG sinyallerinde gürültü gidermede ayrık dalgacık dönüşümünde farklı ana dalgacıkların ve ayrıştırma seviyelerinin karşılaştırılması,” *12. Elektrik-Elektronik-Bilgisayar Mühendisliği Ulusal Kongresi*, Eskişehir, 2007, pp. 383–387.
 - [107] I. B. Toprak, M. F. Caglar, and M. Merdan, “Automatic recognition of epilepsy from EEG using artificial neural network and discrete wavelet transform,” in *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*, 2007, pp. 1–4.
 - [108] J. W. Tukey, *Exploratory data analysis*, vol. 2., MA, Reading: Addison-Wesley, 1977.
 - [109] W. K. Pratt, *Digital image processing : PIKS Scientific inside*, NY, USA: Wiley-Interscience, 2007.
 - [110] V. Kumar, J. K. Chhabra, and D. Kumar, “Automatic cluster evolution using gravitational search algorithm and its application on image segmentation,” *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 93–103, 2014.
 - [111] E. Küçükkülahlı, P. Erdoğan, and K. Polat, “Histogram-based automatic segmentation of images,” *Neural Computing and Applications*, vol. 27, no. 5, pp. 1445–1450, 2016.
 - [112] A. Ajala Funmilola, O. A. Oke, T. O. Adediji, O. M. Alade, and E. A. Adewusi, “Fuzzy kc-means clustering algorithm for medical image segmentation,” *Journal of Information Engineering and Applications*, vol. 22245782, pp. 506–2225, 2012.

- [113] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 3rd ed., CA, San Francisco: Morgan Kaufmann, 2011.
- [114] L. Miao, "Comparative analysis of two clustering algorithms: K-means and FSDP (Fast Search and Find of Density Peaks)," M.S. thesis, Department of Computer Science, San Jose State University, CA, USA, 2015.
- [115] D. Simon, "Biogeography-Based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [116] A. Chatterjee, P. Siarry, A. Nakib, and R. Blanc, "An improved biogeography based optimization approach for segmentation of human head CT-scan images employing fuzzy entropy," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1698–1709, 2012.
- [117] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium*, Nagoya, 1995, pp. 39–43.
- [118] Y. Shi and others, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, Seoul, 2001, vol. 1, pp. 81–86.
- [119] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and N. M. F. Ferreira, "An efficient method for segmentation of images based on fractional calculus and natural selection," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12407–12417, 2012.
- [120] M. S. Couceiro, J. M. A. Luz, C. M. Figueiredo, and N. M. F. Ferreira, "Modeling and control of biologically inspired flying robots," *Robotica*, vol. 30, no. 1, pp. 107–121, 2012.
- [121] E. J. S. Pires, Pb. Oliveira, J. A. T. Machado, and J. B. Cunha, "Particle swarm optimization versus genetic algorithm in manipulator trajectory planning," in *7th Portuguese conference on automatic control*, 2006.
- [122] J. Tang, J. Zhu, and Z. Sun, "A novel path planning approach based on AppART and particle swarm optimization," in *International Symposium on Neural Networks*, 2005, pp. 253–258.
- [123] M. R. Alrashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in power system operations," *Electric Power components and systems*, vol. 34, no. 12, pp. 1349–1357, 2006.
- [124] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.
- [125] J. Sabatier, O. P. Agrawal, and J. A. T. Machado, *Advances in Fractional Calculus*, NY, USA: Springer, 2007.
- [126] J. A. Tenreiro Machado *et al.*, "Some applications of fractional calculus in engineering," *Mathematical Problems in Engineering*, vol. 2010, no. 639801, pp. 1–34, 2010.
- [127] M. Couceiro and P. Ghamisi, "Fractional-order Darwinian PSO," in *Fractional Order Darwinian Particle Swarm Optimization*, Cham: Springer, 2016, pp. 11–

20.

- [128] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
- [129] K. H. Zou *et al.*, “Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports,” *Academic Radiology*, vol. 11, no. 2, pp. 178–189, 2004.
- [130] P. Jaccard, “The distribution of the flora in the alpine zone.,” *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.
- [131] A. A. Taha and A. Hanbury, “Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool,” *BMC Medical Imaging*, vol. 15, no. 1, p. 29, 2015.
- [132] M. Zhang, W. Jiang, X. Zhou, Y. Xue, and S. Chen, “A hybrid biogeography-based optimization and fuzzy C-means algorithm for image segmentation,” *Soft Computing*, pp. 1–14, 2017.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Enver KÜÇÜKKÜLAHLI
Doğum Tarihi ve Yeri : 1983 /TOSYA
Yabancı Dili : İngilizce
E-posta : enverkucukkulahli@duzce.edu.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Elektrik	Düzce Üniversitesi	2011
Lisans	Bilgisayar ve Öğretim Teknolojileri Eğitimi	Ondokuz Mayıs Üniversitesi	2006
Lise		Tosya Anadolu Lisesi	2001

Tez Çalışması Kapsamında Gerçekleştirilen Akademik Çalışmalar

1. E. Küçükkülahlı, P. Erdoğan, and K. Polat, “Histogram-based automatic segmentation of images,” *Neural Computing and Applications*, vol. 27, no. 5, pp. 1445–1450, 2016.
2. E. Küçükkülahlı, P. Erdoğan, and K. Polat, “Brain MRI Segmentation based on Different Clustering Algorithms,” *International Journal of Computer Applications*, vol. 155, no. 3, pp. 37–40, 2016.
3. E. Küçükkülahlı, P. Erdoğan, and K. Polat, “Design of a Graphical User Interface for Data Preprocessing and Image Segmentation Process in 2D MRI Images,” in *ICIVR 2017: International Conference on Image and Video Retrieval*, Rome, Italy, 2017, p. 2549.

4. E. Kucukkulahli, P. Erdogmus, and K. Polat, "A hybrid approach to image segmentation: Combination of BBO (Biogeography based optimization) and Histogram Based Cluster Estimation," in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, 2017, pp. 1–4.
5. S. Ashour, Y. Guo, E. Kucukkulahli, P. Erdogmus, and K. Polat, "A hybrid dermoscopy images segmentation approach based on neutrosophic clustering and histogram estimation," *Applied Soft Computing*, vol. 69, pp. 426–434, 2018.

