T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

SÜRÜ ZEKÂSINA DAYALI METASEZGİSEL ALGORİTMALAR İLE RETİNAL DAMAR SEGMENTASYONU

Hazırlayan
Hakan DURAN

Danışman
Doç. Dr. Mehmet Bahadır ÇETİNKAYA

Doktora Tezi

Ocak 2025
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

SÜRÜ ZEKÂSINA DAYALI METASEZGİSEL ALGORİTMALAR İLE RETİNAL DAMAR SEGMENTASYONU

(Doktora Tezi)

Hazırlayan
Hakan DURAN

Danışman
Doç. Dr. Mehmet Bahadır ÇETİNKAYA

Ocak 2025
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Hakan DURAN

“Sürü Zekâsına Dayalı Metasezgisel Algoritmalar İle Retinal Damar Segmentasyonu” adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan	**Danışman**
Hakan DURAN	Doç. Dr. Mehmet Bahadır ÇETİNKAYA

Mekatronik Mühendisliği ABD Başkanı

Prof. Dr. Şahin YILDIRIM

TEŞEKKÜR

Çalışmalarım süresince bilgi ve tecrübeleriyle rehberlik eden, her türlü destek ve fedakârlığıyla yanımda olan, zorlukların üstesinden gelmemde yol gösteren ve bu çalışmanın başarıyla tamamlanmasında büyük katkı sağlayan kıymetli hocam sayın Doç. Dr. Mehmet Bahadır ÇETİNKAYA'ya teşekkürlerimi sunarım.

Tez çalışmam süresince bana her daim sabır, anlayış ve fedakârlıkla destek olan sevgili eşim Pelin DURAN'a ve hayatım boyunca maddi ve manevi desteklerini hiçbir zaman esirgemeyen, her koşulda yanımda olan değerli annem ve babam başta olmak üzere tüm aileme teşekkürlerimi iletmek isterim. Ayrıca, bu süreçte çalışmalarımı geliştirmeme katkı sağlayan, değerli destek ve önerileriyle bana yol gösteren Arş. Gör. Dr. Burak ULU'ya da içtenlikle teşekkür ederim.

Hakan DURAN

Ocak 2025, KAYSERİ

SÜRÜ ZEKÂSINA DAYALI METASEZGİSEL ALGORİTMALAR İLE RETİNAL DAMAR SEGMENTASYONU

Hakan DURAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Ocak 2025
Danışman: Doç. Dr. Mehmet Bahadır ÇETİNKAYA

ÖZET

Metasezgisel algoritmalara dayalı biyomedikal görüntü analizi son yıllarda önemli bir araştırma alanı haline gelmiştir. Özellikle, retinal görüntülerdeki retinal kan damarların yüksek doğrulukta sınıflandırılması retinal hastalıkların teşhis ve tedavisi açısından önem arz etmektedir. Sürü zekâsına dayalı metasezgisel algoritmalar küresel araştırma yetenekleri, yüksek yakınsama oranları ve az sayıda kontrol parametresine sahip olmaları gibi önemli avantajları sayesinde biyomedikal görüntülerin analizinde etkin çözümler sunmaktadırlar. Bu tez çalışmasında Denizanası Araştırma (JS) algoritması, Deniz Yırtıcıları algoritması (MPA), Tunicate Sürü algoritması (TSA), Açlık Oyunları Araştırma (HGS) algoritması, Tuna Balığı Sürü Optimizasyon (TSO) algoritması, Ateşböceği Sürü Optimizasyon (FSO) algoritması, Sürüngen Araştırma algoritması (RSA) ve Bal Porsuğu algoritması (HBA) gibi güncel metasezgisel algoritmalar retinal damar segmentasyonuna yönelik kümeleme tabanlı geliştirilmişlerdir. Daha sonra, bu algoritmaların performansları Parçacık Sürüsü Optimizasyon (PSO) algoritması, Yapay Arı Koloni (ABC) algoritması ve Gri Kurt Optimizasyon (GWO) algoritması ile mukayese edilmiştir. Her bir algoritmanın performansı duyarlılık (SE), özgüllük (SP), doğruluk (ACC), F-skor, yakınsama hızı, ortalama karesel hata (MSE), CPU süresi, standart sapma ve Wilcoxon rank-sum test metrikleri açısından ayrı ayrı analiz edilmiştir. DRIVE ve STARE veri tabanlarında gerçekleştirilmiş olan simülasyonlar sonucunda JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının retinal damar segmentasyonunda başarılı bir şekilde kullanılabileceği görülmüştür.

Anahtar Kelimeler: Metasezgisel Algoritmalar, Sürü Zekâsı, Retinal Damar Segmentasyonu, Kümeleme.

RETINAL VESSEL SEGMENTATION BY USING SWARM INTELLIGENCE BASED METAHEURISTIC ALGORITHMS

Hakan DURAN

Erciyes University, Graduate School of Natural and Applied Sciences
PhD Thesis, January 2025
Supervisor: Associate Professor Mehmet Bahadır ÇETİNKAYA

ABSTRACT

Biomedical image analysis based on metaheuristic algorithms has become a significant area of research in recent years. In particular, classification of retinal blood vessels in retinal images with high accuracy is crucial for the diagnosis and treatment of retinal diseases. Swarm intelligence based metaheuristic algorithms provide effective solutions in the analysis of biomedical images due to their advantages such as global search abilities, higher convergence rates and having only few control parameters. In this thesis, novel metaheuristic algorithms of Jellyfish Search (JS) algorithm, Marine Predator algorithm (MPA), Tunicate Swarm algorithm (TSA), Hunger Games Search (HGS) algorithm, Tuna Swarm Optimization (TSO) algorithm, Firebug Swarm Optimization (FSO) algorithm, Reptile Search algorithm (RSA) and Honey Badger algorithm (HBA) have been improved as clustering based for retinal vessel segmentation. Afterwards, the performances of these algorithms have been compared to that of Particle Swarm Optimization (PSO) algorithm, Artificial Bee Colony (ABC) algorithm and the Gray Wolf Optimization (GWO) algorithm. The performances of each algorithm have been analyzed separately in terms of sensitivity (SE), specifity (SP), accuracy (ACC), F-score, convergence rate, mean squared error (MSE), CPU time, standard deviation and Wilcoxon rank-sum test metrics. As a result of the simulations carried out on DRIVE and STARE databases, it can be concluded that JS, MPA, TSA, HGS, TSO, FSO, RSA and HBA algorithms can successfully be used in retinal vessel segmentation.

Keywords: Metaheuristic Algorithms, Swarm Intelligence, Retinal Vessel Segmentation, Clustering.

İÇİNDEKİLER

SÜRÜ ZEKÂSINA DAYALI METASEZGİSEL ALGORİTMALAR İLE RETİNAL DAMAR SEGMENTASYONU

BİLİMSEL ETİĞE UYGUNLUK ii

YÖNERGEYE UYGUNLUK ii

KABUL VE ONAY iv

TEŞEKKÜR vi

ÖZET vi

ABSTRACT vii

İÇİNDEKİLER viii

KISALTMALAR xii

TABLOLAR LİSTESİ xiii

ŞEKİLLER LİSTESİ xiv

GİRİŞ 1

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Problem Durumu 4

1.2. Araştırmanın Amacı 4

1.3. Araştırmanın Önemi 6

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1. İnsan Gözünün Yapısı 8

2.2. Retinal Görüntü ve Yapıları 9

2.3. Retinal Kan Damarları ve Özellikleri 10

2.4. Diyabetik Retinopati 10

2.4.1 Diyabetik Retinopati Kaynaklı Retinal Hastalıklar 11

2.5. Renk Uzayları .....…….. 11

2.5.1 RGB Renk Uzayı ve RGB Görüntüler 13

2.5.2 Gri Renk Uzayı ve Gri Seviyeli Görüntüler 14

2.5.3 İkili Renk Uzayı ve İkili Görüntüler 15

2.6. Görüntü Ön İşleme Yöntemleri …… 16

2.6.1 Kontrast Zenginleştirme 16

2.6.2 Top-hat Dönüşümü 17

2.6.3 Morfolojik İşlemler 17

2.7. Veri Tabanları .....…….. 18

2.7.1 DRIVE Veri Tabanı 18

2.7.2 STARE Veri Tabanı 18

3. BÖLÜM

SÜRÜ ZEKÂSINA DAYALI ALGORİTMALAR

3.1. Denizanası Araştırma (JS) Algoritması ....…….. 19

3.2. Deniz Yırtıcıları Algoritması (MPA) ……..... 21

3.3. Tunicate Sürü Algoritması (TSA) .....…….. 23

3.4. Açlık Oyunları Araştırma (HGS) Algoritması ……... 24

3.5. Tuna Balığı Sürü Optimizasyon (TSO) Algoritması ...……..... 26

3.6. Ateşböceği Sürü Optimizasyon (FSO) Algoritması ..…….. 28

3.7. Sürüngen Araştırma Algoritması (RSA) ..…….. 29

3.8. Bal Porsuğu Algoritması (HBA) ....…….. 30

3.9. Parçacık Sürüsü Optimizasyon (PSO) Algoritması ....…….. 32

3.10. Yapay Arı Koloni (ABC) Algoritması ……... 33

3.11. Gri Kurt Optimizasyon (GWO) Algoritması …… 34

4. BÖLÜM

BULGULAR

4.1. Performans Ölçümleri 47

4.2. DRIVE ve STARE Veri Tabanları Performans Analizleri 47

5. BÖLÜM
SONUÇ VE ÖNERİLER

KAYNAKÇA..... 66

ÖZGEÇMİŞ..... 75

KISALTMALAR

ABC	: Artificial Bee Colony Algorithm, yapay arı koloni algoritması
ACC	: Accuracy, doğruluk
CIE	: Commission Internationale de L'Eclairage, Uluslararası Aydınlatma Komisyonu
CMYK	: Cyan, Magenta, Yellow, Key/Black
CPU	: Central Processing Unit
DR	: Diyabetik Retinopati
DRIVE	: Digital Retinal Images for Vessel Extraction
ES	: Evolutionary Sense
FADs	: Fish Aggregating Devices
FSO	: Firebug Swarm Optimization, ateşböceği sürü optimizasyon
GWO	: Grey Wolf Optimization, gri kurt optimizasyon algoritması
HBA	: Honey Badger Algorithm, bal porsuğu algoritması
HGS	: Hunger Games Search Algorithm, açlık oyunları araştırma algoritması
HSV	: Hue, Saturation, Value
JS	: Jellyfish Search, denizanası araştırma algoritması
MPA	: Marine Predators Algorithm, deniz yırtıcıları algoritması
MSE	: Mean-Squared Error, ortalama karesel hata
NFEs	: Fonksiyon Değerlendirme Sayısı
PSO	: Particle Swarm Optimization, parçacık sürüsü optimizasyon algoritması
RGB	: Red, Green, Blue, Kırmızı-Yeşil-Mavi
RSA	: Reptile Search Algorithm, sürüngen araştırma algoritması
SE	: Sensitivity, duyarlılık
SP	: Specificity, özgüllük
STARE	: Structured Analysis of the Retina
TSA	: Tunicate Swarm Algorithm, tunicate sürü algoritması
TSO	: Tuna Swarm Optimization, tuna balığı sürü optimizasyon algoritması

TABLOLAR LİSTESİ

Tablo 4.1. Simülasyonlarda kullanılan kontrol parametre değerleri 41
Tablo 4.2. DRIVE veri tabanı 20 farklı retinal görüntü için JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri 48
Tablo 4.3. STARE veri tabanı 20 farklı retinal görüntü için JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri 49
Tablo 4.4. DRIVE veri tabanı 20 farklı retinal görüntü için PSO, ABC ve GWO algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri 50
Tablo 4.5. STARE veri tabanı 20 farklı retinal görüntü için PSO, ABC ve GWO algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri 51
Tablo 4.6. DRIVE veri tabanı 20 farklı retinal görüntü için F-skor performans analizi 52
Tablo 4.7. STARE veri tabanı 20 farklı retinal görüntü için F-skor performans analizi 53
Tablo 4.8. DRIVE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının ortalama performans mukayesesi .. 54
Tablo 4.9. STARE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının ortalama performans mukayesesi .. 55
Tablo 4.10. DRIVE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının türeve dayalı algoritmalar ile performans mukayesesi 56
Tablo 4.11. STARE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının türeve dayalı algoritmalar ile performans mukayesesi 57
Tablo 4.12. Fonksiyon değerlendirme sayısına (NFEs) dayalı yakınsama analizleri .. 60
Tablo 4.13. DRIVE ve STARE veri tabanları için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının performans

karşılaştırması 61
Tablo 4.14. Wilcoxon rank-sum test sonuçları ($p < 0.05$) 62

ŞEKİLLER LİSTESİ

Şekil 2.1. İnsan gözünün yapısı 9
Şekil 2.2. RGB retinal görüntü ve etiketlenmiş formu 9
Şekil 2.3. Renk uzayları 12
Şekil 2.4. RGB renk uzayı 13
Şekil 2.5. RGB görüntü 14
Şekil 2.6. Gri renk uzayı 14
Şekil 2.7. Gri seviyeli görüntü 15
Şekil 2.8. (0.8) eşik değeri için elde edilen ikili görüntü 16
Şekil 4.1. DRIVE veri tabanından alınan sağlıklı ve hastalıklı retinal görüntüler. (a) Sağlıklı görüntü. (b) Kanama içeren hastalıklı görüntü 37
Şekil 4.2. STARE veri tabanından alınan sağlıklı ve hastalıklı retinal görüntüler. (a) Sağlıklı görüntü. (b) Kanama içeren hastalıklı görüntü 37
Şekil 4.3. Önişleme aşamaları sonucunda elde edilen DRIVE veri tabanı görüntüleri (a, c) Bant seçimi önişlemi uygulandıktan sonra elde edilen G katmanları. (b, d) Bottom-hat filtreleme ve parlaklık düzeltme önişlemleri uygulandıktan sonra elde edilen görüntüler 38
Şekil 4.4. Önişleme aşamaları sonucunda elde edilen STARE veri tabanı görüntüleri. (a, c) Bant seçimi önişlemi uygulandıktan sonra elde edilen G katmanları. (b, d) Bottom-hat filtreleme ve parlaklık düzeltme önişlemleri uygulandıktan sonra elde edilen görüntüler 39
Şekil 4.5. Retinal damar segmentasyonu akış diyagramı. 40
Şekil 4.6. Kümeleme tabanlı JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları kullanılarak Şekil 4.1 (a) ve 4.1 (b) için elde edilen segmentasyon sonuçları. (a) JS. (b) MPA. (c) TSA. (ç) HGS. (d) TSO. (e) FSO. (f) RSA. (g) HBA. (ğ) JS. (h) MPA. (ı) TSA. (i) HGS. (j) TSO. (k) FSO. (l) RSA. (m) HBA. 43
Şekil 4.7. Kümeleme tabanlı PSO, ABC ve GWO algoritmaları kullanılarak Şekil 4.1 (a) ve 4.1 (b) için elde edilen segmentasyon sonuçları. (a) PSO. (b) ABC. (c) GWO. (ç) PSO. (d) ABC. (e) GWO. 44

Şekil 4.8. Kümeleme tabanlı JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları kullanılarak Şekil 4.2 (a) ve 4.2 (b) için elde edilen segmentasyon sonuçları. (a) JS. (b) MPA. (c) TSA. (ç) HGS. (d) TSO. (e) FSO. (f) RSA. (g) HBA. (ğ) JS. (h) MPA. (ı) TSA. (i) HGS. (j) TSO. (k) FSO. (l) RSA. (m) HBA. 45

Şekil 4.9. Kümeleme tabanlı PSO, ABC ve GWO algoritmaları kullanılarak Şekil 4.2 (a) ve 4.2 (b) için elde edilen segmentasyon sonuçları. (a) PSO. (b) ABC. (c) GWO. (ç) PSO. (d) ABC. (e) GWO. 46

Şekil 4.10. Algoritmaların yakınsama hızları. (a) Şekil 4.1 (b) için algoritmaların yakınsama hızları. (b) Şekil 4. 2 (a) için algoritmaların yakınsama hızları. 59

GİRİŞ

Görüntü işleme, görüntü işaretlerinin bilgisayar veya gömülü sistemler üzerinde yüksek seviyeli programlama dilleri kullanılarak analiz edilmesini ve düzenlenmesini kapsayan önemli bir araştırma alanıdır. Görüntü işleme yöntemleri kullanılarak görüntü üzerinde algılama, sınıflandırma, nesne ayırt etme ve nesne izleme gibi analizler yapılabilmektedir [1]. Son yıllarda, sağlık alanında gözlemlenen teknolojik gelişmelere bağlı olarak biyomedikal görüntü işleme en güncel araştırma alanlarından birisi haline gelmiştir.

Biyomedikal görüntülerin yüksek doğrulukta analiz edilmesi hastalıkların teşhis ve tedavi süreçleri açısından son derece önemlidir. Biyomedikal görüntü analizinde en yaygın olarak kullanılan algoritmalar türeve dayalı algoritmalar ve metasezgisel algoritmalar olmak üzere iki ana grupta incelenebilir. Türeve dayalı optimizasyon algoritmaları küresel araştırma yeteneklerinin zayıf olması, başlangıç çözümlerine yüksek oranda bağımlı olmaları ve yakınsama hızlarının düşük olması gibi önemli dezavantajlara sahiptirler. Bu algoritmalar, özellikle, optimum çözümü temsil eden Küresel Minimum (Global Minimum) ile birlikte daha kötü çözümleri temsil eden Yerel Minimumları (Local Minimum) da içeren çok-modlu hata yüzeyine sahip problemlerin optimizasyonunda yetersiz kalmaktadır. Metasezgisel algoritmalar ise rasgele tabanlı küresel araştırma yetenekleri sayesinde çok-modlu hata yüzeylerinde küresel minimumlara yüksek yakınsama hızlarında ulaşarak optimum çözümleri üretebilmektedir [2].

Metasezgisel algoritmalar kendi içerisinde dört ana grupta incelenebilir: evrimsel tabanlı algoritmalar, sürü zekâsına dayalı algoritmalar, fizik tabanlı algoritmalar ve insan davranışlarından ilham alan algoritmalar. Sürü zekâsına dayalı algoritmalar tabiatta var olan canlıların veya canlı gruplarının zeki davranışlarını matematiksel olarak modelleyen ve mühendislik problemlerine etkin çözümler sunan metasezgisel algoritmalardır. Yüksek yakınsama hızları ile küresel çözümlere ulaşabilmeleri, rasgele araştırma

yetenekleri, hata ve istatistiksel performansları ve esnek yapıları sürü zekâsına dayalı algoritmaların en önemli özellikleridir.

Bu tez çalışmasında, en güncel sürü zekâsına dayalı algoritmalar arasında olan Denizanası Araştırma (Jellyfish Search, JS) algoritması [3], Deniz Yırtıcıları algoritması (Marine Predators Algorithm, MPA) [4], Tunicate Sürü algoritması (Tunicate Swarm Algorithm, TSA) [5], Açlık Oyunları Araştırma (Hunger Games Search Algorithm, HGS) algoritması [6], Tuna Balığı Sürü Optimizasyon algoritması (Tuna Swarm Optimization Algorithm, TSO) [7], Ateşböceği Sürü Optimizasyon (Firebug Swarm Optimization, FSO) algoritması [8], Sürüngen Araştırma algoritması (Reptile Search Algorithm, RSA) [9] ve Bal Porsuğu algoritması (Honey Badger Algorithm, HBA) [10] retinal damar segmentasyonuna yönelik kümeleme tabanlı geliştirilmiş ve detaylı performans analizleri gerçekleştirilmiştir. Ayrıca, bu güncel algoritmaların performansları literatürde yaygın olarak kullanılan ve en güçlü sürü zekâsına dayalı algoritmalar arasında olan Parçacık Sürüsü Optimizasyon (Particle Swarm Optimization, PSO) algoritması [11], Yapay Arı Koloni (Artificial Bee Colony Algorithm, ABC) algoritması [12] ve Gri Kurt Optimizasyon (Grey Wolf Optimization, GWO) algoritması [13] ile mukayese edilmiştir.

Benzetim çalışmaları literatürde yaygın olarak kullanılan Digital Retinal Images for Vessel Extraction (DRIVE) [14] ve Structured Analysis of the Retina (STARE) [15] veri tabanlarından alınan görüntüler üzerinde gerçekleştirilmiştir.

Bu tez çalışmasının literatüre başlıca katkıları aşağıdaki şekilde özetlenebilir:

- JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları kümeleme tabanlı olarak geliştirilmiş ve retinal damar segmentasyonu işlemine uygulanmıştır.
- Geliştirilmiş olan algoritmaların performansları, literatürde yaygın olarak kullanılan ve en etkili algoritmalar arasında olan PSO, ABC ve GWO algoritmaları ile mukayese edilmiştir.
- Elde edilen sonuçlar bu algoritmaların PSO, ABC ve GWO algoritmalarına benzer veya biraz daha iyi sonuçlar ürettiğini ve retinal görüntülerin yüksek

doğrulukla segmentasyonunda başarılı bir şekilde kullanılabileceklerini gösterilmiştir.

- İstatistiksel analiz sonuçları bu algoritmaların karmaşık olmayan algoritma yapılarına rağmen istikrarlı ve güvenilir sonuçlar üretebildiğini göstermiştir.

Tez çalışmasının ilk bölümünde retinal kan damarlarında meydana gelen anomalilerin Diyabetik Retinopati (DR) hastalığına yönelik etkileri üzerinde durulmuş ve ayrıca retinal damar segmentasyonu üzerine geniş bir literatür taraması yapılmıştır. Sürü zekâsına dayalı algoritmalar ile retinal damar segmentasyonunun amacı ve önemi ifade edilmiştir.

Tez çalışmasının ikinci bölümünde insan gözünün genel yapısı, retinal görüntüler ve özellikleri, diyabetik retinopati, renk uzayları ve görüntü işleme yöntemleri açıklanmıştır. Ayrıca retinal görüntülerde damar segmentasyonuna yönelik kullanılacak retinal görüntülerin ve DR hastalıklı retinal görüntülerin elde edildiği veri tabanları anlatılmıştır.

Tez çalışmasının üçüncü bölümünde retinal damar segmentasyonuna yönelik geliştirilmiş olan sürü zekâsına dayalı JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmaları detaylıca anlatılmıştır. Daha sonra bu tez çalışmasında kullanılmış olan sürü zekâsına dayalı algoritmaların genel yapısı ve kaba kodları açıklanmıştır.

Son olarak, dördüncü bölümde retinal damar segmentasyonuna yönelik kullanılan sürü zekâsına dayalı algoritmaların performansları ve mukayeseleri üzerine tartışma-sonuç ve önerilere yer verilmiştir.

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Problem Durumu

Retinal kan damarlarında meydana gelen yapısal değişiklikler retinal hastalıklara yönelik önemli bilgiler sağlamaktadır. Bu nedenle, bilgisayar destekli retinal damar segmantasyonu uygulamalarının önemi son yıllarda giderek artmakta ve bu alandaki araştırmalar önem kazanmaktadır. Retinal kan damarları ile retina görüntüsü art alanı arasındaki kontrast farklarının çok düşük olması ve gürültü etkileri bu alanda kullanılacak algoritma seçimlerini önemli hale getirmektedir. Bu kapsamda, retinal kan damarlarının yüksek doğrulukta tespit edilmesine yönelik güçlü algoritmalara ihtiyaç duyulmaktadır.

1.2. Araştırmanın Amacı

Retinal kan damarlarının yüksek doğrulukta tespiti amacıyla literatürde birçok çalışma mevcuttur. [16] numaralı çalışmada, nesnel akıl yürütme yerine yapay sinir ağları kullanılarak istatiksel sonuçlar üzerinden retinal özellikleri bölümlere ayırma işlemi gerçekleştirilmiştir. [17] numaralı çalışmada, retinal damar segmentasyonu üzerine literatürde ilk kez detaylı analizler gerçekleştirilmiştir. Kan damarlarının saptanması amacıyla geri yayılım ağı kullanan bir yaklaşım [18] numaralı çalışmada, retina görüntülerinde anatomik yapıların lokalizasyonu [19] numaralı çalışmada, RGB görüntüsünün yeşil düzlemi üzerinde Gauss uyumlu filtrenin yanıtlarından öznitelik vektörü çıkarımı [20] numaralı çalışmada, *k*-en yakın komşu (*k*-NN) algoritması [21] numaralı çalışmada, retinal damarların doğal göstergeleri ve damar merkez çizgileriyle çakışan görüntü çıkıntılarının çıkarılması [22] numaralı çalışmada, retinal damar segmentasyonu için 2-D Gabor dalgacığı ve denetimli sınıflandırma tabanlı analizler [23] numaralı çalışmada, öznitelik vektörü ve destek vektör makinelerini temel alan piksel sınıflandırma uygulamaları [24] numaralı çalışmada, damar tespiti için çok ölçekli Gabor

filtrelerinin kullanılması ve öznitelik çıkartılması [25] numaralı çalışmada, görüntü işleme teknikleri ve destek vektör makineleri ile retinal damar segmentasyonu [26] numaralı çalışmada, AdaBoost (FABC) olarak bilinen denetimli sınıflandırma yöntemini temel alan analizler [27] numaralı çalışmada, radyal projeksiyon ve destek vektör makinelerini temel alan yarı denetimli yöntem analizleri [28] numaralı çalışmada, retinal damarların segmentasyonu için sinir ağı tabanlı denetimli sınıflandırmalar [29] numaralı çalışmada, fuzzy C-means (FCM) kümeleme algoritması [30] numaralı çalışmada, retinal görüntülerde atardamarların, damarların ve fovea tabakasının Bayes görüntü analizi ile segmentasyonu [31] numaralı çalışmada, görüntü piksellerinin dağılımlarını haritalamada Radius tabanlı kümeleme algoritmasının (RACAL) geliştirilmesi [32] numaralı çalışmada, renkli fundus görüntülerinde düzgün olmayan parlaklıkları düzeltmede denetimsiz bulanık tabanlı damar segmentasyonu [33] numaralı çalışmada, retinal damar parametrelerinin optimum şekilde tahmin edilmesine yönelik analizler [34] numaralı çalışmada, retinal görüntü analizine yönelik self-organizing map (SOM) geliştirilmesi ve ayrıca *k*-Means / modified FCM kümeleme yöntemlerinin uygulaması [35-36] numaralı çalışmada, multi-scale line detection ve k-means yaklaşımlarını temel alan retinal görüntü analizleri [37] numaralı çalışmada, modifiye FCM ve inertia weight particle swarm optimization (IWPSO) algoritmaları kullanılarak retinal fundus görüntülerinde DR tespiti [38] numaralı çalışmada, ARIA ve B-COSFIRE filtreleri / KSVD ve BM3D yöntemleri / CLAHE yöntemi / PSO tabanlı damar segmentasyonu yaklaşımları ile yüksek doğrulukta otomatik hastalık tespiti ve sınıflandırması [39] numaralı çalışmada, FA algoritmasına dayalı diyabetik retinopati teşhisi [40] numaralı çalışmada, hibrit karınca koloni optimizasyon algoritması kullanarak fundus görüntülerinde retinal bozuklukların tespiti [41] numaralı çalışmada, örümcek maymun optimizasyon algoritması (SMOA) ile retinal anatomik yapıların incelenmesi [42] numaralı çalışmada, entropi tabanlı optimizasyon algoritması ile retinal damar segmentasyonu [43] numaralı çalışmada, kuş sürü algoritması (BSA) ve nehir oluşum dinamikleri (RFD) algoritması ile retinal damar segmentasyonu [44] numaralı çalışmada, retinal görüntülerde PSO tabanlı aydınlatma ve kontrast optimizasyonu [45] numaralı çalışmada, ikili-PSO (binary-PSO) algoritması ile retinal damar segmentasyonu [46] numaralı çalışmada, retinal görüntülerde mikroanevrizmaların tespiti [47] numaralı çalışmada, retinal görüntülerde parlak lezyon tanıma için hibrit svm naïve-bayes sınıflandırıcı yaklaşımının geliştirilmesi [48] numaralı çalışmada, optimize edilmiş top-hat ve homomorfik filtreleme ile retinal damar

segmentasyonu [49] numaralı çalışmada, retinal damar segmentasyonu için slime mould optimization (SMO) algotitmasını temel alan çok ölçekli uyumlu filtre tasarımı [50] numaralı çalışmada, retinal görüntü işleme için otomatik tanı ve sınıflandırma yaklaşımlarının geliştirilmesi [51] numaralı çalışmada, kaotik yapay goril birlikleri optimizasyonu (CGTO) ile retinal görüntü analizi [52] numaralı çalışmada, retinal görüntülerde glokomun teşhisi için PSO, ikili-guguk kuşu arama (BCS) ve yarasa (BA) algoritmalarını temel alan iki katmanlı yaklaşımların (BA-BCS, BCS-PSO) geliştirilmesi [53] numaralı çalışmada, geliştirilmiş bir ayrık-PSO algoritması ile retinal görüntülerde otomatik mikroanevrizma tespiti [54] numaralı çalışmada, PSO ve salp sürüsü (SSA) algoritmaları ile diyabetik retinal damar segmentasyonu [55] numaralı çalışmada, derin öğrenme (DL) ve çakal optimizasyon algoritması (COA) ile diyabetik retinopati tespiti [56] numaralı çalışmada, gabor filtrelerine ve PSO optimizasyonuna dayalı retinal kan damarı segmentasyonunu ise [57] numaralı çalışmada gerçekleştirilmiştir.

1.3. Araştırmanın Önemi

Retinal kan damarlarında meydana gelen yapısal bozukluklar ilerleyen dönemlerde birçok hastalığa sebep olmaktadır. Bu hastalıklardan biri olan DR hastalığının erken teşhisi ve buna bağlı olarak uygun tedavi yöntemlerinin belirlenmesi önem arz etmektedir. Bu amaçla, damarlarda meydana gelen anomalilerin tespiti ve hastalıkların erken teşhisi sürü zekâsına dayalı yaklaşımlar ile ele alınarak algoritmaların performansları detaylı bir şekilde mukayese edilmiştir. Literatürde, sürü zekâsına dayalı yaklaşımların retinal damar segmentasyonunda kullanılmasına yönelik az sayıda çalışma bulunmaktadır. Bu tez çalışmasında, en güncel sürü zekâsına dayalı algoritmalar arasında bulunan Denizanası Araştırma (JS) algoritması, Deniz Yırtıcıları algoritması (MPA), Tunicate Sürü algoritması (TSA), Açlık Oyunları Araştırma (HGS) algoritması, Tuna Balığı Sürü Optimizasyon (TSO) algoritması, Ateşböceği Sürü Optimizasyon (FSO) algoritması, Sürüngen Araştırma algoritması (RSA) ve Bal Porsuğu algoritması (HBA) retinal damar segmentasyonuna yönelik geliştirilmiş ve uygulanmıştır. Elde edilen sonuçlar, literatürde yaygın olarak kullanılmakta olan ve bu tez çalışmasında probleme yönelik geliştirilmiş olan Parçacık Sürüsü Optimizasyon (PSO) algoritması, Yapay Arı Koloni (ABC) algoritması ve Gri Kurt Optimizasyon (GWO) algoritması ile elde edilen sonuçlarla detaylı bir şekilde mukayese edilmiştir. Sonuç olarak, JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının retinal damar segmantasyonu ve biyomedikal

görüntü analizinde başarılı bir şekilde kullanılabileceği çok boyutlu ve detaylı analizler ile ortaya koyulmuştur.

2. BÖLÜM

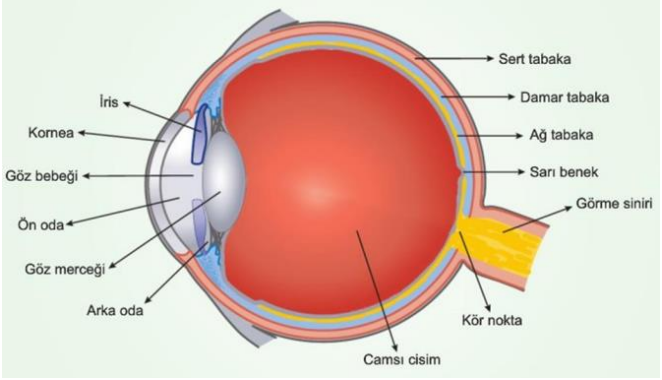
YÖNTEM VE MATERYAL

Bu bölümde, öncelikle, insan gözünün anatomisi ve retinal görüntülerin genel yapısı açıklanmıştır. Daha sonra, retinal kan damarları ve damar yapısındaki anomalilerden dolayı meydana gelen DR hastalığından bahsedilmiştir. Sonraki aşamada, bu tez çalışması kapsamında kullanılmış olan Kırmızı-Yeşil-Mavi (RGB), Gri ve İkili renk uzayları ve farklı görüntü tipleri açıklanmıştır. Daha sonra, görüntüleri yüksek doğrulukta analiz edebilmek için optimizasyon sürecinden hemen önce görüntülere uygulanmış olan görüntü önişleme yöntemleri anlatılmıştır. Son olarak, retinal damar segmentasyonu için kullanılan DRIVE ve STARE veri tabanları özetlenmiştir.

2.1. İnsan Gözünün Yapısı

İnsan gözü, üzerine gelen ışık enerjisini elektro-kimyasal bir enerjiye çevirir. Bu enerjiyi göz yapısındaki sinirler aracılığı ile beyne iletir ve görme olayı gerçekleşir. Göze gelen ışık öncelikle kornea, göz bebeği (pupil) ve göz merceğinden (lens) geçer sonrasında gözün arka kısmında bulunan retina tabakasında (ağ tabakada) toplanır. Retinada oluşan bu görüntü göz siniri (optik sinir) yardımıyla beyine taşınarak insanda görme algısı oluşur. Şekil 2.1'de insan gözünün yapısı ve ana bileşenleri gösterilmektedir.

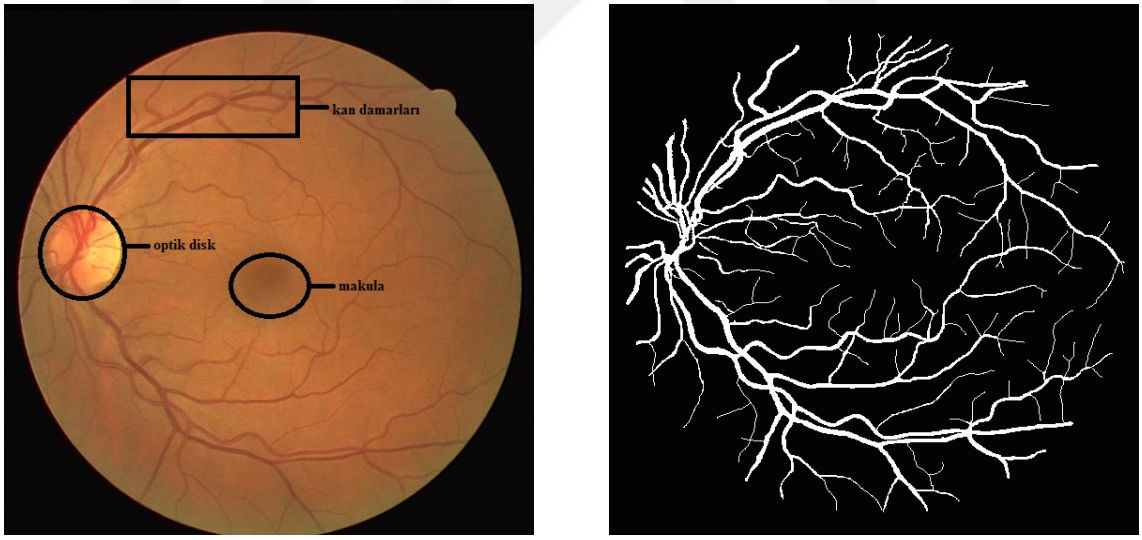
İnsan gözünün yapısı sert tabaka, damar tabaka ve ağ tabaka olmak üzere üç temel tabakadan oluşur. Sert tabaka, gözde bulunan en sert doku olmasının yanı sıra gözü dış tehlikelere karşı korur. Damar tabaka, göze rengi verir ve yapısındaki kasılıp gevşeyen kaslar yardımıyla ışığın beyne zarar vermesini önler. Retina tabakası ise renkleri algılar, ayırt eder ve görme olayını gerçekleştirir.



Şekil 2.1. İnsan gözünün yapısı

2.2. Retinal Görüntü ve Yapıları

Görme olayının gerçekleştiği retina tabakasında optik disk ve makula tabakalarına ek olarak retinayı besleyen kan damarları da bulunmaktadır. Şekil 2.2 (a)'da örnek bir RGB retinal görüntü üzerindeki temel retinal yapılar ve Şekil 2.2 (b)'de ise retinal kan damarlarının etiketlenmiş hali gösterilmiştir.



(a) RGB retinal görüntü (b) Etiketlenmiş retinal görüntü

Şekil 2.2. RGB retinal görüntü ve etiketlenmiş formu

Optik disk, retinal damarların ve görmeyi sağlayan sinirlerin çıktığı bölgedir. Optik disk aynı zamanda kan damarlarının retinaya dağıldığı bölgedir ve diğer bölgelere göre daha parlak bir görünüme sahiptir.

Makula, görmeyi sağlayan sinir uçlarının yoğunlaştığı bölgedir ve sarımsı bir pigment içermektedir. Bu bölge, keskin ve renkli görme işlevinden sorumludur ve ayrıca optik diskten yaklaşık olarak 3-4 mm uzakta bulunmaktadır.

Kan damarları, retina tabakasını besleyen damarlardır ve iki farklı besleme süreci bulunmaktadır. Koroidal damarlar dış kısmı beslerken, atardamarlar iç kısmı beslemektedir. Retinal damarlar, optik disk ve retinaya göre daha koyu renklidir. Atardamarlar, daha kalın duvarlıdır ve renk olarak daha solgundur ve ayrıca ışığı yansıtma özellikleri nedeniyle parlak bir şerit şeklinde yansıma gösterirler.

2.3. Retinal Kan Damarları ve Özellikleri

Retina kan damarları, retinanın iç kısmındaki kılcal damarlar, atardamarlar ve toplardamarlardan meydana gelmektedir ve bu damarlar retina dokusunun ihtiyaç duyduğu oksijen ve besin maddelerinin taşınmasına yardımcı olmaktadır. Retinal hücreler görevlerini yerine getirirken metabolik atıklar oluştururlar ve bu atıkların birikmesi retinada hasara ve işlev bozukluklarına yol açabilir. Diğer taraftan, atardamarlar, hücrelerin faaliyetlerinden kaynaklanan atık ürünleri toplayarak bu zararlı maddelerin birikmesini önlerler. Atardamarlar, bu atık ürünleri topladıktan sonra toplardamarlar aracılığıyla vücuda geri gönderirler. Bu süreç, retinanın sağlıklı bir şekilde çalışabilmesi için son derece önemlidir.

Retina kan damarları, yoğun ağ yapısı sayesinde retinanın her bölgesine ulaşarak ihtiyaç duyulan oksijen ve besin maddelerinin hızlı bir şekilde taşınmasını sağlarlar. Retinanın gözdeki ışığı algılayan ve görme işlevini yerine getiren hücrelerden oluşması ve bu hücrelerin retina kan damarları aracılığı ile beslenmesi göz sağlığı için hayati öneme sahiptir. Retina kan damarları, diğer kan damarlarına göre daha hassas yapıya sahiptir ve daha kolay hasar görebilirler. Kan damarlarında oluşan deformasyonlar sonucunda diyabetik retinopati ve damar tıkanıklığı gibi hastalıklar ortaya çıkabilmekte ve görme bozuklukları ve görme kayıpları riski oluşmaktadır.

2.4. Diyabetik Retinopati

Diyabetik retinopati, diyabet hastalığına bağlı olarak retina tabakasındaki kılcal damarların hasar görmesi sonucu ortaya çıkan ve ileri seviyelerinde körlüğe neden olabilecek bir hastalıktır. Kan şekerinin uzun süre yüksek seyretmesi kılcal damarlara

zarar verir ve yeterince beslenemeyen retina hücreleri retinada hasarlar meydana getirir. DR hastalığının en yaygın belirtileri arasında bulanık veya perdeli görme, çift görme, gece iyi görememe, yanıp sönen ışıklar ve renksiz noktalar, parlak ya da loş ışığa uyum sağlayamama, koyu renkli uçuşan noktalar, dalgalı görme, renkli görmede bozukluklar, gözlerin dış kenarlarında kalan yerleri görmede zorlanma ve görme kaybı yer alır. Bu belirtiler, hastalığın ilerleme seviyesine göre değişkenlik gösterebilir ve tedavi edilmediğinde ciddi görme kaybına neden olabilir. Bu nedenle, DR hastalığının erken teşhisi ve tedavisi oldukça önemlidir.

2.4.1. Diyabetik Retinopati Kaynaklı Retinal Hastalıklar

Mikroanevrizmalar, kan damarı duvarlarında oluşan deformasyonlardır ve genellikle kılcal damarlarda meydana gelen küçük şişlikler veya çıkıntılar şeklinde ortaya çıkarlar. Mikroanevrizmalar, retinada gelişen vasküler anormalliklerin bir tezahürüdür ve genellikle görme kaybına neden olabilirler.

Kanamalar, hasar görmüş kılcal damarlardan sızan kanın nokta, alev veya leke şeklinde belirginleşmesidir. Kanamalar, özellikle koyu renkli izler bırakarak retina tabakasında deformasyonlara yol açarlar.

Eksüdalar, kanın kılcal damarlardan aşırı sızması durumunda oluşan sarı renkli sıvı birikimleridir. Eksüdalar, genellikle retinanın hassas dokularının zarar gördüğü ve normalde dolaşan sıvıların damar dışına çıktığı durumlarda meydana gelir.

Neovaskülarizasyon, vitröz içinde anormal şekilde büyüyen yeni damarların oluşumunu ifade eder. Bu damarlar retina için uygun olmayan bir yapıda olabilir ve görme kaybına yol açabilir.

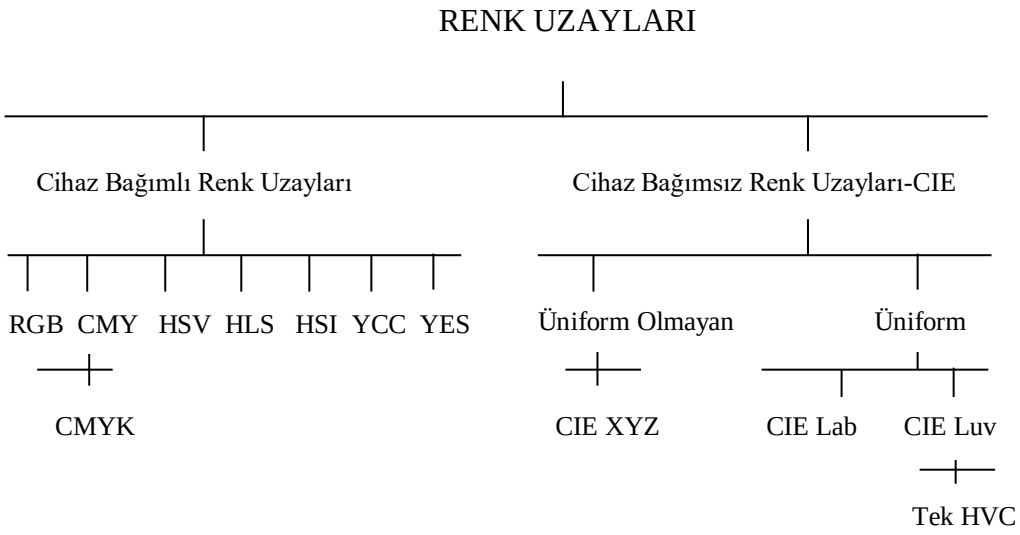
Makula ödemi, makula bölgesindeki damarlarda ortaya çıkan sıvı sızıntıları nedeniyle meydana gelir. Makula, gözdeki merkezi görüşten sorumlu bir bölgedir ve burada oluşan ödem görme keskinliğinde önemli oranda azalmalarına neden olabilir.

2.5. Renk Uzayları

Renk uzayları, renkleri tanımlamak, temsil etmek ve analiz etmek amacıyla kullanılan matematiksel modellerdir. Renk uzayları, renklerin konumlarını belirleme, renkler

arasındaki ilişkileri analiz etme ve renklerin görsel algıdaki değişimlerini modelleme yetenekleri sunar.

Renk uzayları genellikle üç boyutlu (3D) olarak tasarlanır ve Grassmann'ın birinci kanunu'na dayanarak, bir rengi tam olarak belirlemek için üç bağımsız değişkenin gerekliliğine odaklanır. Grassmann'ın birinci kanunu'na göre, üç bağımsız değişken genellikle kırmızı, yeşil ve mavi renk bileşenleridir. Bu temel renk bileşenleri, birbirleriyle kombinasyon yaparak geniş bir renk gamını oluştururlar. Renk uzayları, farklı cihazlar arasında renk tutarlılığını sağlamak, renk dönüşümlerini gerçekleştirmek ve renk analizi yapmak gibi uygulamalarda yaygın olarak kullanılır. CIE (Commission Internationale de L'Eclairage: Uluslararası Aydınlatma Komisyonu) tarafından geliştirilen renk uzayları, özellikle cihaz bağımsız standartlar sunarak renkmetri bilimine önemli katkılar sağlamıştır [58]. Şekil 2.3'te yaygın olarak kullanılan renk uzayları görülmektedir. Bu renk uzayları arasında en yaygın kullanılanlar RGB (Red, Green, Blue), CMYK (Cyan, Magenta, Yellow, Key/Black), HSV (Hue, Saturation, Value) ve CIE Lab renk uzaylarıdır.



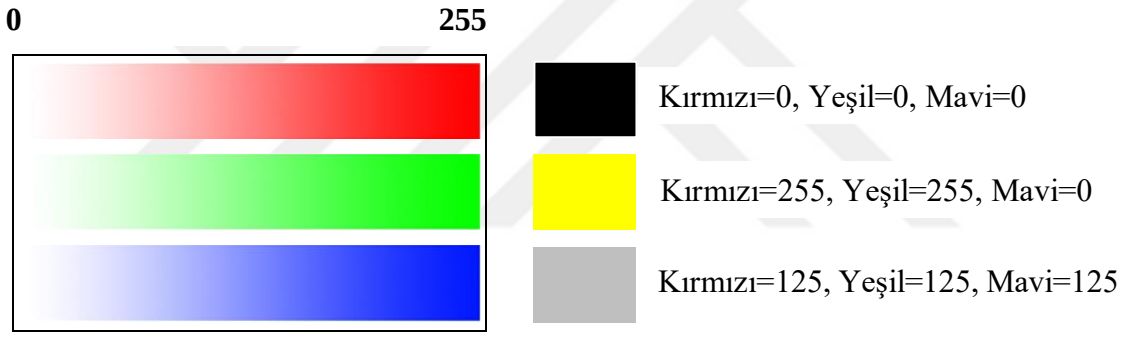
Şekil 2.3. Renk uzayları

Her renk uzayının kendine özgü bir biçimi ve standartları vardır. Örneğin, RGB renk uzayı ekran ve dijital görüntüleme sistemlerinde yaygın olarak kullanılırken, CMYK renk uzayı genellikle baskı işlemlerinde tercih edilmektedir. Her bir renk uzayı, belirli bir uygulama veya endüstri ihtiyacına uygun olarak seçilir. Renk uzayları arasında doğrusal

veya doğrusal olmayan yöntemler kullanılarak dönüşümler yapılabilmektedir. Bu tez çalışmasında, literatürde en yaygın olarak kullanılan RGB renk uzayı ve RGB görüntüler, gri seviyeli görüntüler ve ikili görüntüler kullanılarak analizler gerçekleştirilmiştir.

2.5.1. RGB Renk Uzayı ve RGB Görüntüler

RGB renk uzayı, Kırmızı-Yeşil-Mavi dalga boylarının farklı oranlarda kullanılmasıyla elde edilen 3-katmanlı (3-boyutlu) bir renk uzayıdır. Her bir katman için renk skalası, Şekil 2.4'te gösterildiği gibi, en açık tondan en koyu tona doğru [0,255] aralığında piksel değerleri almaktadır. Bir diğer ifadeyle, her bir ana renk 8-bit ile kodlanmaktadır. Diğer sonsuz adet renk bu üç ana rengin belirli piksel değerlerinin karıştırılması ile elde edilmektedir. Siyah, Sarı ve Gri tonlar için örnek değerler Şekil 2.4'te gösterilmiştir.



Şekil 2.4. RGB renk uzayı

(384x512x3) boyutunda örnek bir RGB görüntü Şekil 2.5 ile gösterilmiştir. Burada, 384 satır sayısını, 512 sütun sayısını ve 3 ise görüntünün 3-boyutlu bir görüntü olduğunu belirtmektedir. Bir diğer ifadeyle, bu görüntü (384x512) adet piksel içermektedir.



Şekil 2.5. RGB görüntü

2.5.2. Gri Renk Uzayı ve Gri Seviyeli Görüntüler

Gri renk uzayı, gri renk tonlarından oluşan tek-katmanlı (1-boyutlu) bir renk uzayıdır. Gri renk skalası, Şekil 2.6'da gösterildiği gibi, en koyu tondan en açık tona doğru [0,255] aralığında piksel değerleri almaktadır. 0 değeri siyahı, 255 değeri ise beyazı temsil ederken diğer ara piksel değerler farklı gri tonlarına karşılık gelmektedir. RGB görüntülere benzer olarak gri seviyeli görüntüler de 8-bit ile kodlanmaktadır.



Şekil 2.6. Gri renk uzayı

Şekil 2.5 ile gösterilmiş olan RGB görüntünün gri skaladaki karşılığı aşağıda Şekil 2.7 ile gösterilmiştir. (384x512) boyutundaki bu gri seviyeli görüntüde 384 satır sayısını ve 512 ise sütun sayısını temsil etmektedir. 3D boyutlu RGB görüntünün gri seviyeli görüntüye dönüştürülmesi ile toplam piksel sayısının değişmediği ancak gri skalada tek boyutlu bir görüntünün elde edildiği görülmektedir.

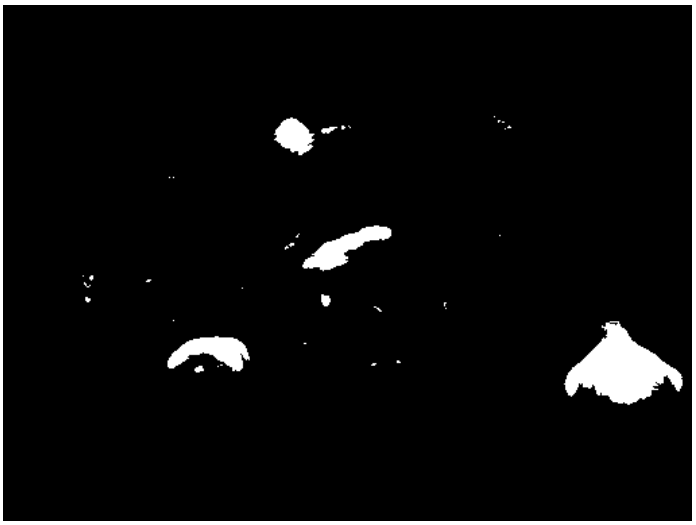


Şekil 2.7. Gri seviyeli görüntü

2.5.3. İkili Renk Uzayı ve İkili Görüntüler

İkili görüntülerde, her bir piksel siyah veya beyaz olmak üzere iki farklı değerden sadece birisini alabilir. Siyah renk 0 değeri ile temsil edilirken, beyaz renk 1 değeri ile temsil edilmektedir. RGB görüntünün [0,255] aralığındaki piksel değerleri [0,1] aralığına normalize edilerek ikili görüntüler elde edilmektedir. Bu normalizasyon işlemi yine [0,1] aralığında belirlenen bir eşik değerine göre gerçekleştirilir ve eşik değerinin üzerindeki bütün piksellere 1 (Beyaz) değeri atanırken, eşik değeri ve eşik değerinin altındaki pikseller ise 0 (Siyah) değerine atanırlar.

Şekil 2.5 ile gösterilmiş olan RGB görüntünün (0.8) eşik değeri için ikili skaladaki karşılığı Şekil 2.8 ile gösterilmiştir. RGB bir görüntü ikili görüntüye dönüştürüldüğünde ikili skalada tek boyutlu bir görüntü ortaya çıkmakta fakat toplam piksel sayısı değişmemektedir.



Şekil 2.8. (0.8) eşik değeri için elde edilen ikili görüntü

2.6. Görüntü Ön İşleme Yöntemleri

2.6.1. Kontrast Zenginleştirme

Kontrast zenginleştirme, düşük kontrastlı görüntülerin insan gözüne uygun hale getirilmesinde kullanılır. Bu işlemde, görüntüdeki piksel değerleri belirli oranda artırılır veya azaltılır. Böylece, gri tonlar arasındaki farklılıklar belirginleşir ve görüntü daha net hale gelir. Görüntünün kontrastı, o görüntüdeki en parlak ve en karanlık bölgeler arasındaki farka bağlıdır. Eğer görüntüdeki beyaz bölgeler daha parlak hale getirilirse, görüntü beyaz seviyeye yaklaştırılır ve kontrast artar. Diğer durumda, siyah bölgeler daha karanlık hale getirilirse, görüntü siyah seviyeye yaklaştırılır ve kontrast azalır. Kontrast zenginleştirme işlemi Eşitlik 2.1'deki gibi gerçekleştirilir,

$$g(x,y) = a * f(x,y) \tag{2.1}$$

burada, a katsayısı orijinal görüntüdeki her bir piksel değerine uygulanır. Eğer, a>1 olarak seçilirse, her bir pikselin değeri artar ve görüntü beyaz seviyeye yaklaşır. Bu şekilde, görüntüdeki parlak bölgeler daha belirgin hale gelir ve kontrast artar. Eğer a<1 olarak seçilirse, her bir pikselin değeri azalır ve görüntü siyah seviyeye yaklaşır. Bu şekilde, görüntüdeki karanlık bölgeler daha belirgin hale gelir ve kontrast azalır.

2.6.2. Top-hat Dönüşümü

Top-hat dönüşümü nesne tespiti, görüntü segmentasyonu ve arka plan gürültüsünün azaltılması gibi birçok uygulamada kullanılmaktadır. Top-hat dönüşümü, Beyaz (White) ve Siyah (Black) top-hat dönüşümleri olmak üzere iki farklı şekilde uygulanabilir.

Beyaz top-hat dönüşümü, görüntünün arka planındaki parlak nesneleri belirginleştirmek için kullanılır. Bu dönüşüm, orijinal görüntüye morfolojik açma işlemi uygulandıktan sonra elde edilen görüntünün orijinal görüntüden çıkarılması ile elde edilir. Bu işlem sonucunda, küçük beyaz nesneler ve ayrıntılar çıkarılmış olur ve böylece daha büyük nesnelerin sınırları belirginleştirilir.

Siyah top-hat dönüşümü ise görüntünün arka planındaki koyu nesneleri belirginleştirmek için kullanılır. Bu dönüşüm, orijinal görüntüye morfolojik kapatma işlemi uygulandıktan sonra elde edilen görüntünün orijinal görüntüden çıkarılması ile elde edilir. Bu işlem sonucunda, küçük siyah nesneler ve ayrıntılar çıkarılmış olur ve böylece daha büyük nesnelerin sınırları belirginleştirilir.

Top-hat dönüşümü, genellikle gürültülü görüntülerin işlenmesinde kullanılır. Bu işlem, gürültüyü azaltmak için kullanılan bir yöntemdir. Böylece, nesnelerin tespiti ve segmentasyonu daha doğru bir şekilde gerçekleştirilebilir.

2.6.3. Morfolojik İşlemler

Morfolojik işlemler, görüntü içerisinde nesne tanıma, kenar tespiti, gürültü azaltma ve nesne çıkarma gibi işlemlerde kullanılır. Bu işlemler, yapı elemanı olarak adlandırılan bir matris kullanılarak gerçekleştirilir. Yapı elemanı, genellikle dikdörtgen, kare veya daire şeklinde olabilir. Morfolojik işlemler, iki temel işlem olan erozyon ve genişleme işlemlerini içerir.

Erozyon işlemi, yapı elemanının görüntü üzerinde gezdirilmesiyle gerçekleştirilir. Yapı elemanının merkez noktası, görüntü üzerindeki bir pikselin değerini temsil eder. Yapı elemanının tüm piksel değerleri görüntüdeki piksel değerleri ile karşılaştırılır ve eğer yapı elemanının tüm piksel değerleri, görüntüdeki piksel değerlerine uyuyorsa o piksel değeri

korunur, diğer durumda piksel değeri silinir. Bu işlem, görüntüdeki küçük nesneleri silmek veya kenarları yumuşatmak için kullanılır.

Genişleme işlemi, erozyon işlemine benzer bir şekilde gerçekleştirilir. Bu işlemdeki amaç görüntüdeki nesneleri büyütmek veya kenarlarını belirginleştirmektir. Bu süreçte yapı elemanı görüntü üzerinde gezdirilir ve yapı elemanının her bir piksel değeri, görüntüdeki karşılık gelen piksel değerleri ile mukayese edilir. Eğer yapı elemanının herhangi bir piksel değeri, görüntüdeki bir piksel değeri ile örtüşüyorsa bu piksel değeri korunur. Aksi durumda yapı elemanının kapsadığı alandaki pikseller genişleme kuralları çerçevesinde eklenir. Bu yöntem, morfolojik görüntü işleme süreçlerinde nesnelerin sınırlarının belirginleştirilmesi veya nesnelerin büyütülmesi amacıyla kullanılır.

2.7. Veri Tabanları

Farklı algoritmaların retinal damar segmentasyonuna yönelik geliştirilmesi amacıyla literatürde en yaygın kullanılan iki veri tabanı DRIVE ve STARE veri tabanlarıdır. Bu tez çalışması kapsamında, bu iki veri tabanından alınan görüntüler üzerinde analizler gerçekleştirilmiştir.

2.7.1. DRIVE Veri Tabanı

DRIVE veri tabanı [14], retinal görüntülerdeki damar yapılarının otomatik olarak tespitinde kullanılan bir veri tabanıdır. DRIVE veri tabanında, 20 görüntü eğitim ve 20 görüntü test için kullanılmaktadır. Bu görüntüler, (565x584) piksel boyutunda olup CanonCR5 3CCD kamera kullanılarak 45° FOV açısıyla elde edilmiştir.

2.7.2. STARE Veri Tabanı

STARE veri tabanı [15], retinal görüntülerdeki kan damarlarının performansını test etmek için kullanılan veri setidir. STARE veri tabanında bulunan 20 görüntüden, 10 tanesi çeşitli patolojilere sahipken geri kalan 10 görüntü sağlıklı retina görüntüleridir. 35° FOV açısına sahip TopCon TRV-50 kamerası ile elde edilen görüntüler 3 renk kanalına ve (700x605) piksel çözünürlükle elde edilmiştir.

3. BÖLÜM

SÜRÜ ZEKÂSINA DAYALI ALGORİTMALAR

Bu bölümde, retinal damar segmentasyonuna yönelik geliştirilmiş olan sürü zekâsına dayalı algoritmalar detaylı bir şekilde açıklanacaktır.

3.1. Denizanası Araştırma (JS) Algoritması

Chou ve Truong tarafından 2020'de önerilen JS algoritması [3], denizanalarının doğal ortamlarındaki avlanma davranışından ilham alınarak geliştirilmiş bir algoritmadır. JS algoritmasının en önemli avantajlarından birisi popülasyon büyüklüğü ve döngü sayısı olmak üzere sadece iki kontrol parametresini içermesidir. Bu algoritma üç ana kuraldan esinlenmiştir,

i.) denizanası, "*okyanus akıntısında hareket etmek*" veya "*sürü ile birlikte hareket etmek*" olarak tanımlanabilecek iki tür hareket gerçekleştirir ve bu hareketler arasındaki geçiş süreci bir zaman kontrol mekanizması tarafından koordine edilir,

ii.) denizanası yiyecek ararken daha fazla yiyecek bulunan yerlere doğru hareket eder,

iii.) besin kaynağının kalitesi, ilgili kaynağın konumu ve buna karşılık gelen amaç fonksiyonu tarafından belirlenir.

Algoritmanın amacı, bir besin kaynağına ulaşarak onun kalitesini belirlemek ve daha yüksek kaliteli kaynaklara doğru hareket ederek en iyi çözümü bulmaktır. JS algoritması, karmaşık problemlerin çözümünde etkili bir araç olarak kullanılabilir.

Sürü zekâsı tabanlı bir JS algoritmasının detaylı kodu aşağıda verilmiştir.

$x_i \in \{1,2,\dots,N_{POP}\}$ çözümlerinden oluşan rasgele bir başlangıç popülasyonu oluştur,
Her x_i çözümü için popülasyondaki uygunluk değerini hesapla,
Çevrim=1

Repeat

Aşağıdaki denklemi kullanarak zaman kontrol parametresinin değerini hesapla,

$$c(t) = \left| \left(1 - \frac{1}{Max\ Çevrim\ Sayısı} \right) \ (2.rand(0,1) - 1 \right|$$

If $c(t) \geq 0{,}5 : okyanus\ akıntısını\ takip\ et$

i.) okyanus akıntısı : $\overrightarrow{trend} = X^* - \beta\ .rand(0,1)\ .\ \mu$

burada; X^* sürüde şu anda en iyi konuma sahip olan denizanasını temsil eder, μ tüm denizanalarının ortalama konumu and $\beta > 0$ ise $\overrightarrow{trend}$ uzunluğuna ilişkin dağılım katsayısı olarak tanımlanır.

ii.) $X_i\,(t+1) = X_i\,(t) + rand(0,1)\ .\ \overrightarrow{trend}$ ile yeni bir konum tanımla,

Else : $sürünün\ içinde\ hareket\ et$

If $rand(0,1) > (\,1 - c(t))$ ise yeni bir konum tanımla,

$$X_i\,(t+1) = X_i\,(t) + \gamma\ .rand(0,1)\ .(\,U_b - L_b\,)$$

burada; U_b ve L_b arama uzayının alt ve üst sınırları temsil etmektedir, $\gamma > 0$ ise hareket katsayısıdır.

else : i^{th} ve j^{th} denizanalarının yönünü ve yeni konumlarını tanımla,

$$\overrightarrow{direction} = \begin{cases} X_j(t) - X_i(t) & if \quad f\,(X_i) \geq f\big(X_j\big) \\ X_i(t) - X_j(t) & if \quad f\,(X_i) < f\big(X_j\big) \end{cases}$$

$$X_i\,(t+1) = X_i\,(t) + \ rand(0,1)\ .\ \overrightarrow{direction}$$

end if

End if

Sınır koşullarını kontrol et ve yeni konumdaki besin kaynağının uygunluk değerini hesapla,

X_i and X^* değerlerini güncelle,

Çevrim= Çevrim+1

Until *(sonlandırma kriteri karşılanıncaya kadar)*

3.2. Deniz Yırtıcıları Algoritması (MPA)

MPA algoritması, Faramarzi ve arkadaşları tarafından 2020'de önerilen [4], av ve avcı hayvanlar arasındaki ilişkiyi dikkate alarak okyanus yırtıcılarının yiyecek arama davranışlarını modelleyen sürü zekâsı tabanlı bir algoritmadır. Bu algoritmanın en önemli ölçütü, rastgele yürüyüş stratejisi olarak da adlandırılan yırtıcıların kullandığı arama stratejilerini optimize etmektir. MPA algoritması, Brownian hareketi ve Lévy hareketi olmak üzere iki ana rastgele yürüyüş stratejisine odaklanır.

MPA algoritmasının detaylı akışı aşağıda verilmiştir.

Rastgele bir av popülasyonu oluştur, $i = 1, 2, 3, \dots, n$

Çevrim = 1

While $Çevrim \leq Maksimum\, Çevrim$

Uygunluk değerlerini hesapla ve en iyi çözümleri (avların en iyi konumlarını) içeren bir Elite matrisi oluştur,

$$Elite = \begin{bmatrix} X_{1,1}^I & \cdots & X_{1,d}^I \\ \vdots & \ddots & \vdots \\ X_{n,1}^I & \cdots & X_{n,d}^I \end{bmatrix}$$

If $Çevrim < \frac{(Maksimum\ Çevrim\ Sayısı)}{3}$: $yüksek\ hız\ oranı$

Avı güncelle: $\overrightarrow{Prey_\iota} = \overrightarrow{Prey_\iota} + P \,.\, \vec{R} \otimes \overrightarrow{stepsize_\iota}$

burada; $\overrightarrow{stepsize_\iota} = \overrightarrow{R_B} \otimes (\overrightarrow{Elite_\iota} - \overrightarrow{R_B} \otimes \overrightarrow{Prey_\iota})$ avcı hayvanın avdan daha hızlı hareket ettiği aşama 1 için hız oranıdır; $\otimes$ gösterimi, giriş bazında çarpmaları temsil eder; R_B 'nin avla çarpımı avın

hareketini modeller; P = 0.5 sabit bir sayıdır; R, [0,1] aralığında değerler alan rasgele sayı vektörüdür.

Else if $\frac{(Maksimum\ Çevrim\ Sayısı)}{3} < Çevrim < 2.\ \frac{(Maksimum\ Çevrim\ Sayısı)}{3}$: $birim\ hız\ oranı$

Nüfusun ilk yarısı için avı güncelle, $i = 1, 2, 3, \ldots, n/2$

$$\overrightarrow{Prey_{\iota}} = \overrightarrow{Prey_{\iota}} + P\,.\vec{R} \otimes \overrightarrow{stepsıze_{\iota}}$$

burada; $\overrightarrow{stepsıze_{\iota}} = \overrightarrow{R_L} \otimes (\overrightarrow{Elite_{\iota}} - \overrightarrow{R_L} \otimes \overrightarrow{Prey_{\iota}})$ hem avcının hem de avın aynı hızda hareket ettiği aşama 2'nin hız oranıdır; $\overrightarrow{R_L}$ Lévy dağılımından üretilen rasgele sayı vektörüdür.

Nüfusun ikinci yarısı için avı güncelle, $i = n/2\ ,\ \ldots, n$

$$\overrightarrow{Prey_{\iota}} = \overrightarrow{Elite_{\iota}} + P\,.CF \otimes \overrightarrow{stepsıze_{\iota}}$$

burada; $\overrightarrow{stepsıze_{\iota}} = \overrightarrow{R_B} \otimes (\overrightarrow{R_B} \otimes \overrightarrow{Elite_{\iota}} - \overrightarrow{Prey_{\iota}})$ hem avcının hem de avın aynı hızda hareket ettiği aşama 2 için hız oranıdır; $\overrightarrow{R_B}$ ve *Elite* çarpımı yırtıcı hayvanın hareketini Brown hareketi ile modeller; $CF = \left(1 - \frac{Cycle}{Max\ Cycle}\right)^{\left(2 \cdot \frac{Cycle}{Max\ Cycle}\right)}$ ise adaptif bir kontrol parametresidir.

Else if $Cycle > 2.\frac{(Maximum\ Cycle\ Number)}{3}$: $düşük\ hız\ oranı$

Avı güncelle: $\overrightarrow{Prey_{\iota}} = \overrightarrow{Elite_{\iota}} + P\,.CF \otimes \overrightarrow{stepsıze_{\iota}}$

burada; $\overrightarrow{stepsıze_{\iota}} = \overrightarrow{R_L} \otimes (\overrightarrow{R_L} \otimes \overrightarrow{Elite_{\iota}} - \overrightarrow{Prey_{\iota}})$ avcının avdan daha hızlı hareket ettiği 3. aşama için hız oranıdır; çarpımı; $\overrightarrow{R_L}$ ve *Elite*, yırtıcı hayvanın hareketini Brownian tarzında simüle eder.

End if

Şu ana kadarki en iyi çözümleri tut ve *Elite* matrisini güncelle,

Algoritmanın yerel minimumlara takılıp kalmasını önlemek için aşağıda verilen Apply Fish Aggregating Devices (FADs) yaklaşımını uygula,

$$\overrightarrow{Prey_{\iota}} = \begin{cases} \overrightarrow{Prey_{\iota}} + CF.\left[\overrightarrow{X_{min}} + \overrightarrow{R} \otimes \left(\overrightarrow{X_{max}} - \overrightarrow{X_{min}}\right)\right] \otimes \overrightarrow{U} & if\ r \leq FADs \\ \overrightarrow{Prey_{\iota}} + \left[FADs\,(1-r) + r\right]\left(\overrightarrow{Prey_{r1}} - \overrightarrow{Prey_{r2}}\right)\right] & if\ r > FADs \end{cases}$$

burada; FADs = 0.2 değeri FAD'lerin optimizasyon süreci üzerindeki etkisinin olasılığıdır; $\overrightarrow{U}$ vektörü [0,1] aralığında rasgele üretilen ikili bir vektör; $\overrightarrow{X_{min}}$ ve $\overrightarrow{X_{max}}$ boyutların alt ve üst sınır vektörleri; r_1 ve r_2 değerleri ise av matrisinin rastgele indekslerini temsil etmektedir.

End While

3.3. Tunicate Sürü Algoritması (TSA)

Kaur ve arkadaşları tarafından 2020'de önerilen TSA algoritması [5], tulumluların (tunicates) yiyecek arama sürecindeki sürü davranışlarından ilham alan bir optimizasyon algoritmasıdır. TSA algoritması matematiksel olarak tulumlu sürülerinin, jet tahriki davranışı ve sürü davranışı olmak üzere iki ana davranışı üzerine inşa edilmiştir.

TSA algoritması için kod akışı aşağıda verilmiştir.

Rasgele bir tulumlu popülasyonu üret, $\overrightarrow{P_p}$

Başlangıç popülasyonundaki her arama uzayının uygunluk değerini hesapla,

Çevrim=1

Repeat

Tulumlular için jet tahriki ve sürü davranışlarını aşağıdaki denklemleri kullanarak tanımla,

i.) Jet Tahriki Davranışı

$$\overrightarrow{PD} = \left| \overrightarrow{FS} - rand[0,1] \,.\, \overrightarrow{P_p(x)} \right|$$

burada; $\overrightarrow{FS}$ arama uzayının uygunluk değer vektörünü temsil eder.

ii.) Sürü Davranışı

$$\overrightarrow{P_p(x)} = \begin{cases} \overrightarrow{FS} + \overrightarrow{A} \,.\, \overrightarrow{PD} & if \quad rand[0,1] \geq 0.5 \\ \overrightarrow{FS} - \overrightarrow{A} \,.\, \overrightarrow{PD} & if \quad rand[0,1] < 0.5 \end{cases}$$

burada; $\overrightarrow{A}$, arama uzayındaki çatışmaları önlemek için farklı varyasyonları içeren bir vektördür.

Arama uzayı konumlarını güncelle,

$$\overrightarrow{P_p(x+1)} = \frac{\overrightarrow{P_p(x)} + \overrightarrow{P_p(x+1)}}{2 + c_1}$$

burada; c_1 , [0,1] aralığında rastgele üretilmiş bir sayıdır.

Güncellenen arama uzayı konumları için uygunluk değerlerini hesapla,
Arama uzayının dışına çıkan arama ajanlarını ortadan kaldırın.
Çevrim = Çevrim + 1

Until *(sonlandırma kriteri karşılanıncaya kadar)*

3.4. Açlık Oyunları Araştırma (HGS) Algoritması

HGS algoritması [6], Yang ve arkadaşları tarafından 2021 yılında önerilmiş olan ana keşif ve kullanım davranışlarını zenginleştiren stokastik geçiş öğelerine sahip bir algoritmadır. HGS algoritmasının adaptif ve zamanla değişen mekanizmaları, bu algoritmanın hem tek modlu hem de çok modlu optimizasyon problemlerinde etkin bir şekilde kullanılmasına olanak tanımaktadır. Açlık oranı ve açlığın aktivite aralığı üzerindeki etkisi gibi metrikler HGS algoritmasının daha esnek ve yüksek performanslı olmasını sağlamaktadır.

HGS algoritmasının ayrıntılı kodu aşağıdaki gibidir,

Parametre değerlerini tanımla;

N : popülasyondaki birey sayısı,

T : maksimum çevrim sayısı,

l ve E : başlangıç pozisyonlarını ve araştırma modunu optimize eden kontrol parametreleri,

SHungry : her bir bireyin açlık oranı

Rasgele başlangıç popülasyonu oluştur, $x_i \{1,2,\dots,N_{POP}\}$

While (t ≤ T)

Popülasyondaki her bir bireyin kalite değerini hesapla,

BF, *W*, X_b değerlerini güncelle,

BF : mevcut çevrimdeki en yüksek kalite değeri,

W : yerel minimuma takılmayı önleyen ağırlık parametre değerleri,

X_b : mevcut çevrimdeki en iyi çözüm,

Açlık metriğini tanımla,

$$hungry(i) = \begin{cases} 0, & AllFitness(i) == BF \\ hungry(i) + H, & AllFitnes(i) = BF \end{cases}$$

W_1 metriğini hesapla,

$$\overrightarrow{W_1(\imath)} = \begin{cases} hungry(i).\dfrac{N}{SHungry} \times r_4\,, & r_3 < l \\ 1\,, & r_3 > 1 \end{cases}$$

W_2 metriğini hesapla,

$$\overrightarrow{W_2(\imath)} = (1 - exp(-|hungry(i) - SHungry|)) \times r_5 \times 2$$

For Popülasyondaki her bir birey için,

E metriğini hesapla,

$$E = \text{sech}(|F(i) - BF|)$$

R metriğini güncelle,

$$\vec{R} = 2 \times a \times rand - a$$

$$a = 2 \times \left(1 - \frac{t}{T}\right)$$

Konum bilgilerini güncelle,

$$\overrightarrow{X(t+1)} = \begin{cases} Game_1: \overrightarrow{X(t)}.\left(1 + randn(l)\right), \ r_1 < l \\ Game_2: \overrightarrow{W_1}.\overrightarrow{X_b} + \ \vec{R}.\overrightarrow{W_2}.\left|\overrightarrow{X_b} - \overrightarrow{X(t)}\right|, r_1 < l, r_2 > E \\ Game_3: \overrightarrow{W_1}.\overrightarrow{X_b} - \ \vec{R}.\overrightarrow{W_2}.\left|\overrightarrow{X_b} - \overrightarrow{X(t)}\right|, r_1 < l, r_2 > E \end{cases}$$

End for

t = t + 1

End While

Return *BF*, X_b

3.5. Tuna Balığı Sürü Optimizasyon (TSO) Algoritması

Xie ve arkadaşları tarafından 2021 yılında önerilen TSO algoritması [7], ton balıklarının zeki avlanma davranışlarını modellemektedir. İlk avlanma stratejisi, *spiral avlanma* olarak adlandırılır. Spiral avlanma stratejisinde, ton balıkları avlarını daha kolay saldırılabilecekleri sığ sulara sürüklemek için bir spiral formasyon oluşturarak yüzerler. *Parabolik avlanma* olarak adlandırılan ikinci stratejide ise her bir ton balığı önceki bireyin izinden giderek avlarını sarmak için bir parabolik şekil oluşturur. TSO algoritması ton balıkları tarafından uygulanan bu iki başarılı avlanma stratejisinden esinlenerek geliştirilmiştir.

TSO algoritmasının detaylı akışı aşağıda verilmiştir.

Parametre değerlerini tanımla;

NP : popülasyon büyüklüğü,

t_{max} : maksimum çevrim sayısı,

Rasgele başlangıç popülasyonu üret, $X_i^{int} \quad (i = 1, 2 \ldots, n)$

a ve *z* parametre değerlerini tanımla,

While ($t < t_{max}$)

Ton balıklarının uygunluk değerlerini hesapla,

Güncelle: X_{best}^{t}

For (her ton balığı) **do**

Güncelle: α_1, α_2, p

If (rand < z) **then** Pozisyonu güncelle, X_i^{t+1}

$$X_i^{int} = rand.(ub - lb) + lb, \quad i = 1, 2 \ldots, NP$$

Else if (rand ≥ z) **then**

If (rand < 0.5) **then**

If (t/t_{max} < rand) **then** Pozisyonu güncelle, X_i^{t+1}

$$X_i^{t+1} = \begin{cases} \alpha_1.(X_{rand}^{t} + \beta.|X_{rand}^{t} - X_i^{t}|) + \alpha_2.X_i^{t}, & i = 1 \\ \alpha_1.(X_{rand}^{t} + \beta.|X_{rand}^{t} - X_i^{t}|) + \alpha_2.X_{i-1}^{t}, & i = 2, 3 \ldots, NP \end{cases}$$

Else if (t/t_{max} ≥ rand) **then** Pozisyonu güncelle, X_i^{t+1}

$$X_i^{t+1} = \begin{cases} \alpha_1.(X_{best}^{t} + \beta.|X_{best}^{t} - X_i^{t}|) + \alpha_2.X_i^{t}, & i = 1 \\ \alpha_1.(X_{best}^{t} + \beta.|X_{best}^{t} - X_i^{t}|) + \alpha_2.X_{i-1}^{t}, & i = 2, 3 \ldots, NP \end{cases}$$

Else if (rand ≥ 0.5) **then** Pozisyonu güncelle, X_i^{t+1}

$$X_i^{t+1} = \begin{cases} X_{best}^{t} + rand.(X_{best}^{t} - X_i^{t}) + TF.p^2.(X_{best}^{t} - X_i^{t}), & if rand < 0.5 \\ TF.p^2.X_i^{t}, & if\ rand \geq 0.5 \end{cases}$$

End for

t = t + 1

End While

Güncelle: X_{best} (en iyi çözüm) ve $F(X_{best})$ (en iyi uygunluk değeri)

3.6. Ateşböceği Sürü Optimizasyon (FSO) Algoritması

FSO algoritması, Noel ve arkadaşları tarafından 2021 yılında önerilmiştir ve ateş böcekleri tarafından sergilenen zeki davranışları modellemektedir [8]. Ateş böcekleri, temel olarak iki tür davranış sergilerler. *Bireysel Keşif Davranış* modelinde, ateş böcekleri tek başlarına dolaşabilir ve yeni alanları keşfetme eğilimindedirler. Bu davranış, genellikle kaynak arayışı veya genel keşif amaçlıdır. *Toplu Davranış ve Agregasyonlar* modelinde ise ateş böcekleri sürü halinde hareket etme ve topluluklar oluşturma eğilimindedirler. Algoritma özellikle erkek ve dişi ateş böcekleri arasındaki iletişim ve etkileşimleri modellemektedir.

FSO algoritmasının ayrıntılı kodu aşağıdaki gibi verilebilir.

Parametre değerlerini tanımla;

D: arama uzayı boyutu,

C_i : ivme katsayısı,

S_{max}: toplam fonksiyon değerlendirme sayısı ($10000D$),

$m.F$: erkek böcek (m) ile ilişkili dişi böcek (f) pozisyonlarını içeren matris,

g : optimal çözümün konum bilgisi,

$\odot$: hadamard çarpımı,

Rasgele popülasyon oluştur, (N_m : erkek ve N_f dişi ateşböceği sayısı),
Erkek ateş böceğine karşılık gelen en iyi dişiyi ayarla,

While $(s > s1_{max})$

Dişi ateş böceklerinin konumunu güncelle,

$$m\,(m).F \leftarrow m\,(m).F + Col_{\,1} \odot (M_x - m\,(m).F) + Col_{\,2} \odot (M_y - m\,(m).F)$$

If $(I > l1)$

Yeni konumu güncelle,

$$m(m).x \leftarrow m\,(m).x + Col_{\,3} \odot (g - m(m).x)$$

Else if

Sonraki konumu güncelle,

Else if

Sonraki dişi ateş böceği pozisyonunu güncelle,

$$m(m).x \leftarrow m(m).x + Col_4 \odot (g - m(b).x)$$

End While

3.7. Sürüngen Araştırma Algoritması (RSA)

RSA algoritması, Abualigah ve arkadaşları tarafından 2022'de önerilmiştir [9]. Bu algoritma, timsahların doğadaki çembere alma, avlanma ve ayrıca aktif işbirliğini teşvik etmek ve genel dayanıklılığı maksimize etmek amacıyla bağlı gruplar oluşturma gibi davranışlarından esinlenilerek geliştirilmiştir.

RSA algoritmasının akışı aşağıda detaylı bir şekilde gösterilmiştir.

RSA parametrelerini başlat : α ve β
Çözüm pozisyonlarını rastgele başlat, *X: 1, 2, ..., N.*
While ($t < T$) do

Her bir X aday çözümü için kalite değerini hesapla,
Şu ana kadarki en iyi çözümü bul,
Evolutionary Sense (*ES*) olasılık değerini güncelle,

$$ES(t) = 2 \times r_3 \times \left(1 - \frac{1}{T}\right)$$

For (i=1, ..., N) **do**

For (j=1, ..., N) **do**

η, R, P parametre değerlerini güncelle,

$$n_{(i,j)} = Best_j(t) \times P_{(i,j)}$$

$$R_{(i,j)} = \frac{Best_j(t) - x_{(r_2,j)}}{Best_j(t) + \epsilon}$$

$$P_{(i,j)} = \alpha + \frac{x_{(i,j)} - M(x_i)}{Best_j(t) \times (UB_{(j)} - LB_{(j)}) + \epsilon}$$

If $\left(t \leq \frac{T}{4}\right)$ **then** *High walking* fazını uygula,

$$x_{(i,j)}(t+1) = Best_j(t) \times -n_{(i,j)}(t) \times \beta - R_{(i,j)}(t) \times rand \ ,$$

Else if $\left(t \leq 2\frac{T}{4} ve\ t > \frac{T}{4}\right)$ **then** *Belly walking* fazını uygula,

$$x_{(i,j)}(t+1) = Best_j(t) \times x_{(r_1,j)} \times ES(t) \times rand \ ,$$

Else if $\left(t \leq 3\frac{T}{4} ve\ t > 2\frac{T}{4}\right)$ **then** *Hunting coordination* fazını uygula,

$$x_{(i,j)}(t+1) = Best_j(t) \times P_{(i,j)}(t) \times rand \ ,$$

Else

$$x_{(i,j)}(t+1) = Best_j(t) \times -n_{(i,j)}(t) \times \varepsilon - R_{(i,j)}(t) \times rand,$$

Hunting cooperation fazını uygula,

End if

End for

End for

t=t+1

End While

En iyi çözümü döndür, ($Best(X)$).

3.8. Bal Porsuğu Algoritması (HBA)

Hashim ve arkadaşları tarafından 2022 yılında önerilen HBA [10], bal porsuklarının avlanma davranışını modelleyen bir optimizasyon algoritmasıdır. Bu algoritmanın temel amacı, bir hedefe yönelik en iyi stratejiyi bulmaktır. Bu amaca ulaşmak için bal porsuğunun *Kazma Modu* (*Digging Mode*) ve *Bal Modu* (*Honey Mode*) olarak adlandırılan iki avlanma stratejisinden ilham alınmaktadır.

HBA algoritmasının detaylı akışı aşağıda sunulmuştur.

Parametre değerlerini ayarla: t_{max}, N, β, C

Popülasyonu rasgele konumlar ile başlat,

Amaç fonksiyonunu kullanarak her bir bal porsuğu pozisyonu için x_i uygunluk değerini hesapla ve f_i , $i \in [1, 2, \ldots, N]$ olarak ata,

En iyi x_{prey} pozisyonunu kaydet ve f_{prey} uygunluk değerine ata,

While $t \leq t_{max}$ **do** (α) azalma faktörü değerini güncelle,

$$\alpha = C \times exp\left(\frac{-t}{t_{max}}\right)$$

For (i=1 to N) **do** (I_i) yoğunluk değerini hesapla,

$I_i = r_2 \times \frac{S}{4\pi d_i^2}$, r_2 0 ile 1 arasında rastgele bir sayıdır.

If $r < 0.5$ **then** (x_{new}) pozisyon değerini güncelle,

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_3 \times \alpha \times d_i \times |\cos(2\pi r_4) \times [1 - cos(2\pi r_5]|$$

Else (x_{new}) pozisyon değerini güncelle,

$x_{new} = x_{prey} + F \times r_7 \times \alpha \times d_i$, (r_7 0 ile 1 arasında rastgele bir sayı)

End if

Yeni pozisyonu değerlendir ve f_{new} olarak ata,

If $f_{new} \leq f_i$ **then**

$x_i = x_{new}$ ve $f_i = f_{new}$ olarak ata,

End if

If $f_{new} \leq f_{prey}$ **then**

$x_{prey} = x_{new}$ ve $f_{prey} = f_{new}$ olarak ata,

End if

End for

End While

Return x_{prey}

3.9. Parçacık Sürüsü Optimizasyon (PSO) Algoritması

Kennedy ve arkadaşları tarafından 1995 yılında önerilmiş olan PSO algoritması [11], literatürde yaygın olarak kullanılan algoritmalar arasındadır ve kuş sürülerinin yiyecek arama davranışlarından esinlenerek geliştirilmiştir. Kuş sürüsünü temsil eden rasgele oluşturulmuş bir popülasyon içerisinde her bir olası çözümün (sürü içerisindeki her bir kuşun) konumunu sürekli optimize edilerek en uygun konumların belirlenmesi hedeflenmektedir. PSO algoritmasında gerçek kuş sürülerinde kuşlar arasında gerçekleşen etkileşimler ve hafızaya alma mekanizmaları da matematiksel olarak modellenmektedir.

PSO algoritmasının ayrıntılı kodu aşağıda gösterilmiştir.

Rasgele başlangıç popülasyonu üret: $N_i,\ i = 1, \dots, N$

Popülasyondaki her bir birey için rasgele başlangıç konumu bilgisi ata, $\vec{x}(t)$

Popülasyondaki her bir birey için $[\upsilon_{min}, \upsilon_{max}]$ aralığında rasgele başlangıç hızı bilgisi ata,

Repeat

Her bir çözümüm kalite değerini hesapla,

Her bir çözümüm konum bilgisini optimize et ve [$\vec{p}(t)$] olarak güncelle,

Popülasyonda en yüksek kalite değerini üreten çözümü [$\vec{g}(t)$] olarak güncelle,

Her bir çözümün mevcut hız bilgisini [$\vec{\upsilon}(t)$] güncelle,

$$\vec{\upsilon}(t+1) = \omega\vec{\upsilon}(t) + r_1 c_1 (\vec{p}(t) - \vec{x}(t)) + r_2 c_2 (\vec{g}(t) - \vec{x}(t))$$

r_1 ve r_2 : uniform dağılımdan üretilen rasgele sayılar,

ω (*inertia weight*): $\vec{\upsilon}(t)$ hız vektörü için genlik kontrol parametresi,

c_1 ve c_2 : $\vec{p}(t)$ ve $\vec{g}(t)$ için katkı oranı sabitleri,

Her bir çözümün mevcut pozisyon bilgisini [$\vec{x}(t)$] güncelle,

$$\vec{x}(t+1) = \vec{x}(t) + \vec{\upsilon}(t+1)$$

Until *(sonlandırma kriteri karşılanıncaya kadar)*

3.10. Yapay Arı Koloni (ABC) Algoritması

ABC algoritması, Karaboğa tarafından 2005 yılında önerilmiştir ve bal arılarının akıllı yiyecek arama davranışını matematiksel olarak modellemektedir [12]. Görevli arılar, gözcü arılar ve kaşif arılar olmak üzere üç farklı arı grubundan oluşturulan popülasyonda her bir arı grubu için farklı görev tanımlamaları yapılmaktadır. Görevli arılar, potansiyel çözümleri değerlendirir ve her bir çözümün kalite değerlerini hesaplar. Her bir yiyecek kaynağının kalite değerini belirleme görevi sadece bir görevli arıya atanmıştır. Bu nedenle, görevli arı sayısı kullanılan yiyecek kaynağı sayısıyla eşleşmektedir. Gözcü arılar, görevli arıların aktardıkları bilgileri değerlendirerek yeni çözümleri temsil eden yiyecek kaynakları tespit ederler. Belirli sayıda değerlendirmeye rağmen kalite değeri geliştirilemeyen çözümlerin görevli arıları kaşif arıya dönüşürler ve rasgele yeni çözümler/kaynaklar üretirler.

ABC algoritmasının detaylı akışı aşağıda sunulmuştur.

Rasgele başlangıç popülasyonu üret, $x_i \in \{1,2,\dots,SN\}$

Her bir x_i çözümü için kalite değerini hesapla, $fit_i = \dfrac{1}{1 + f_i}$

Çevrim=1

Repeat

x_{ij} çözümlerinin komşuluğunda yeni v_{ij} çözümleri üret,

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Her bir v_{ij} çözümü için kalite değerini hesapla,

x_i ve v_i çözümleri arasında *Greedy Selection* uygula ve iyi çözümü hafızaya al,

Her bir x_i çözümü için fit_i kalite değerlerine göre p_i olasılık değerlerini hesapla,

$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_n}$, p_i değerlerini [0,1] aralığına normalize et,

x_i ve p_i üzerinden elde edilen yeni (v_i) çözümlerini üret ve kalite değerlerini hesapla,

x_i ve v_i çözümleri arasında *Greedy Selection* uygula ve iyi çözümü hafızaya al,

Kalite değeri düşük çözümleri, aşağıdaki eşitlik kullanılarak rasgele üretilen yeni çözümler ile yer değiştir,

$$x_i^j = x_{min}^j + rand[0,1](x_{max}^j - x_{min}^j)$$

Mevcut en iyi çözümü hafızaya al,

Çevrim = Çevrim +1

Until *(sonlandırma kriteri karşılanıncaya kadar)*

3.11. Gri Kurt Optimizasyon (GWO) Algoritması

Mirjalili ve arkadaşları tarafından 2014'te önerilmiş olan GWO algoritması [13], gri kurt sürülerindeki sosyal hiyerarşi ve avlanma stratejileri üzerine modellenmiştir. Hiyerarşik yapı içerisinde, en üst seviyede, bir erkek ve bir dişi alfa kurt çifti bulunmaktadır [alpha (α)]. Hiyerarşinin ikinci ve üçüncü seviyelerini alfa kurtlara yardımcı olan ve onların yerine geçebilecek adayları içeren beta ve delta kurtlar oluşturur [beta (β) ve delta (δ)]. En düşük hiyerarşiye sahip kurt gurubu ise omega kurtlardır [omega (w)]. α kurtlar popülasyondaki en iyi çözümleri temsil ederler, β ve δ kurtlar, sırasıyla, popülasyondaki en iyi ikinci ve üçüncü çözümleri temsil ederler ve son olarak popülasyonun geri kalan kısmı *w* kurtlar ile temsil edilir.

GWO algoritması için kod yapısı aşağıdaki gibi verilebilir.

Rasgele başlangıç popülasyonu üret, X_i $(i = 1, 2, \ldots, n)$

$\vec{a}$, $\vec{A}$ ve $\vec{C}$ parametreleri için başlangıç değerleri tanımla,

$\vec{a}$: [2,0] aralığında her bir çevrimde doğrusal olarak azalan katsayı,

$\vec{A}$ ve $\vec{C}$: katsayı vektörleri, $\vec{A} = 2\,\vec{a}\,\overrightarrow{r_1} - \vec{a}$ ve $\vec{C} = 2\,\overrightarrow{r_2}$

burada, $\overrightarrow{r_1}$ ve $\overrightarrow{r_2}$, [0,1] aralığında rasgele üretilen vektörler,

Her bir başlangıç çözümü için kalite değerlerini hesapla, ($\overrightarrow{X_a}$, $\overrightarrow{X_\beta}$ ve $\overrightarrow{X_\delta}$)

While *(t < Max Çevrim Sayısı)*

For (her bir araştırma ajanı için)

Mevcut en iyi pozisyon bilgilerini aşağıdaki eşitlikler ile güncelle,

$$\overrightarrow{D_\propto} = \left| \overrightarrow{C_1}\ \overrightarrow{X_\propto} - \overrightarrow{X(t)} \right|,\ \overrightarrow{D_\beta} = \left| \overrightarrow{C_2}\ \overrightarrow{X_\beta} - \overrightarrow{X(t)} \right|,\ \overrightarrow{D_\delta} = \left| \overrightarrow{C_3}\ \overrightarrow{X_\delta} - \overrightarrow{X(t)} \right|$$

$$\overrightarrow{X_1} = \left| \overrightarrow{X_\propto} - \overrightarrow{a_1}\ (\overrightarrow{D_\propto}) \right|,\ \overrightarrow{X_2} = \left| \overrightarrow{X_\beta} - \overrightarrow{a_2}(\overrightarrow{D_\beta}) \right|,\ \overrightarrow{X_3} = \left| \overrightarrow{X_\delta} - \overrightarrow{a_3}\ (\overrightarrow{D_\delta}) \right|$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3}$$

End for

$\vec{a}$, $\vec{A}$ ve $\vec{C}$ parametre değerlerini güncelle,

Her bir mevcut pozisyon için kalite değerlerini hesapla,

($\overrightarrow{X_a}$, $\overrightarrow{X_\beta}$ ve $\overrightarrow{X_\delta}$) değerlerini güncelle,

t = t+1

End

Mevcut $\overrightarrow{X_a}$ değerini optimize etmek için üçüncü adıma dön

4. BÖLÜM

BULGULAR

Bu tez çalışmasında, en güncel sürü zekâsına dayalı algoritmalar arasında olan JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları retinal damar segmentasyonuna yönelik geliştirilmiş ve performansları PSO, ABC ve GWO algoritmaları ile mukayese edilmiştir. Simülasyon çalışmaları DRIVE ve STARE veri tabanlarından alınan sağlıklı ve hastalıklı retinal görüntüler için ayrıntılı bir şekilde gerçekleştirilmiştir. DRIVE ve STARE veri tabanlarından alınmış olan sağlıklı ve hastalıklı retinal görüntüler, sırasıyla, Şekil 4.1 ve Şekil 4.2 ile gösterilmiştir.

Damar ve arka plan piksellerini yüksek doğrulukla ayırt edebilmek amacıyla kümeleme işleminden önce bant seçimi, bottom-hat dönüşümü ve parlaklık düzeltme önişleme yöntemleri uygulanarak retina görüntüsü iyileştirilmelidir.

RGB retina görüntülerinin kırmızı (R), yeşil (G) ve mavi (B) katmanları kontrast ve parlaklık düzeyleri açısından bant seçimi önişlemi uygulanarak ayrı ayrı analiz edildiğinde, G katmanının daha yüksek aydınlatma, optimum kontrast ve optimum parlaklık düzeyleri sayesinde en yüksek kümeleme performansını sağladığı görülmüştür.

DRIVE veri tabanından alınan retina görüntülerine uygulanan bant seçimi önişlemi sonucunda elde edilen yeşil katmanlar, Şekil 4.3 (a) ve 4.3 (c)'de verilmiştir. Bottom-hat dönüşümü ve parlaklık düzeltme uygulanarak kontrast farkı açısından iyileştirilen retina görüntüleri ise Şekil 4.3 (b) ve 4.3 (d)'de gösterilmiştir.

STARE veri tabanından alınan retina görüntülerinin yeşil katmanları Şekil 4.4 (a) ve 4.4 (c)'de gösterilmiştir. Ayrıca, iki ardışık önişlem sonrasında kontrast farkı açısından iyileştirilen retina görüntüleri Şekil 4.4 (b) ve 4.4 (d)'de verilmiştir.



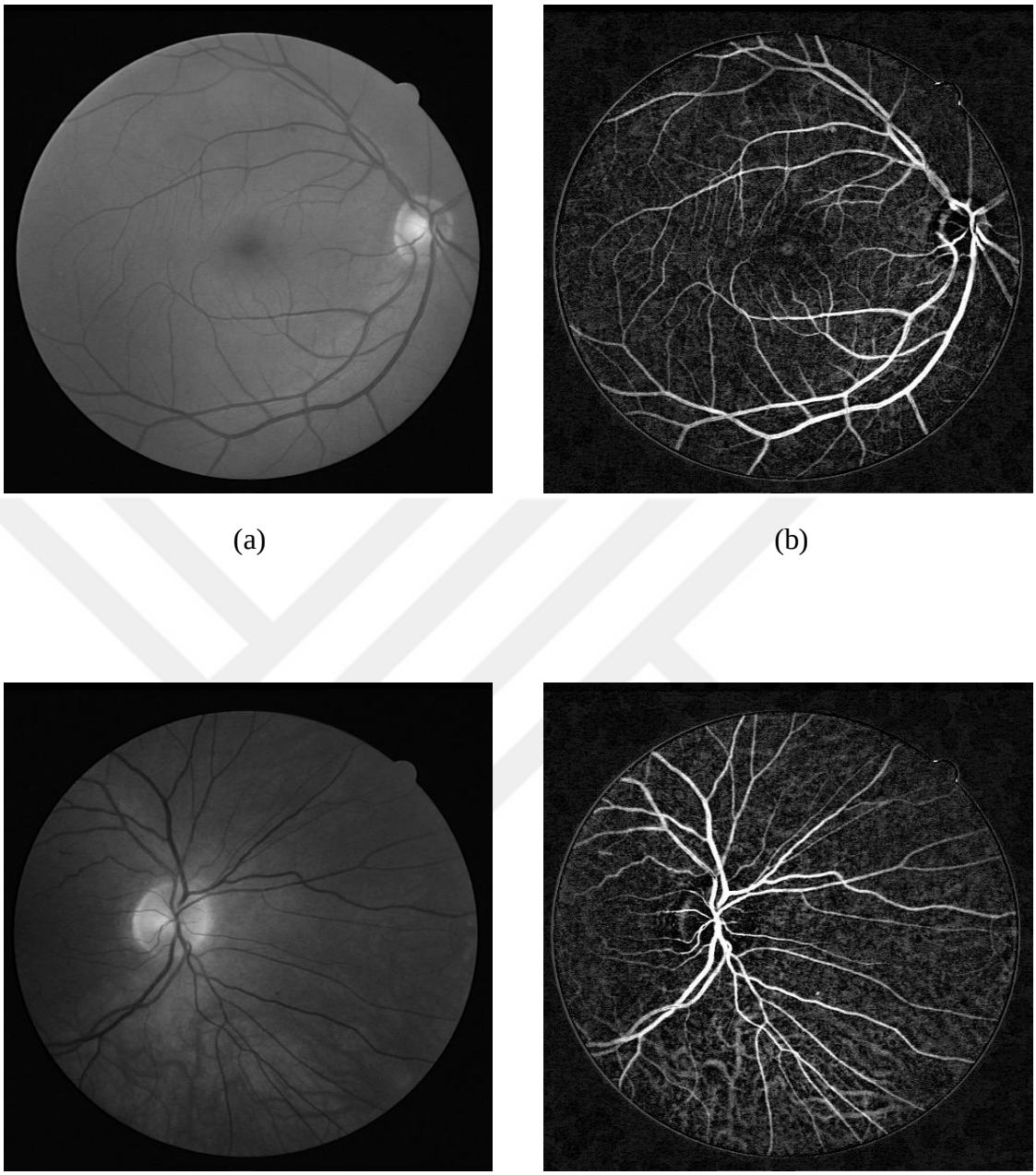
(a) (b)

Şekil 4.1. DRIVE veri tabanından alınan sağlıklı ve hastalıklı retinal görüntüler. (a) Sağlıklı görüntü. (b) Kanama içeren hastalıklı görüntü.



(a) (b)

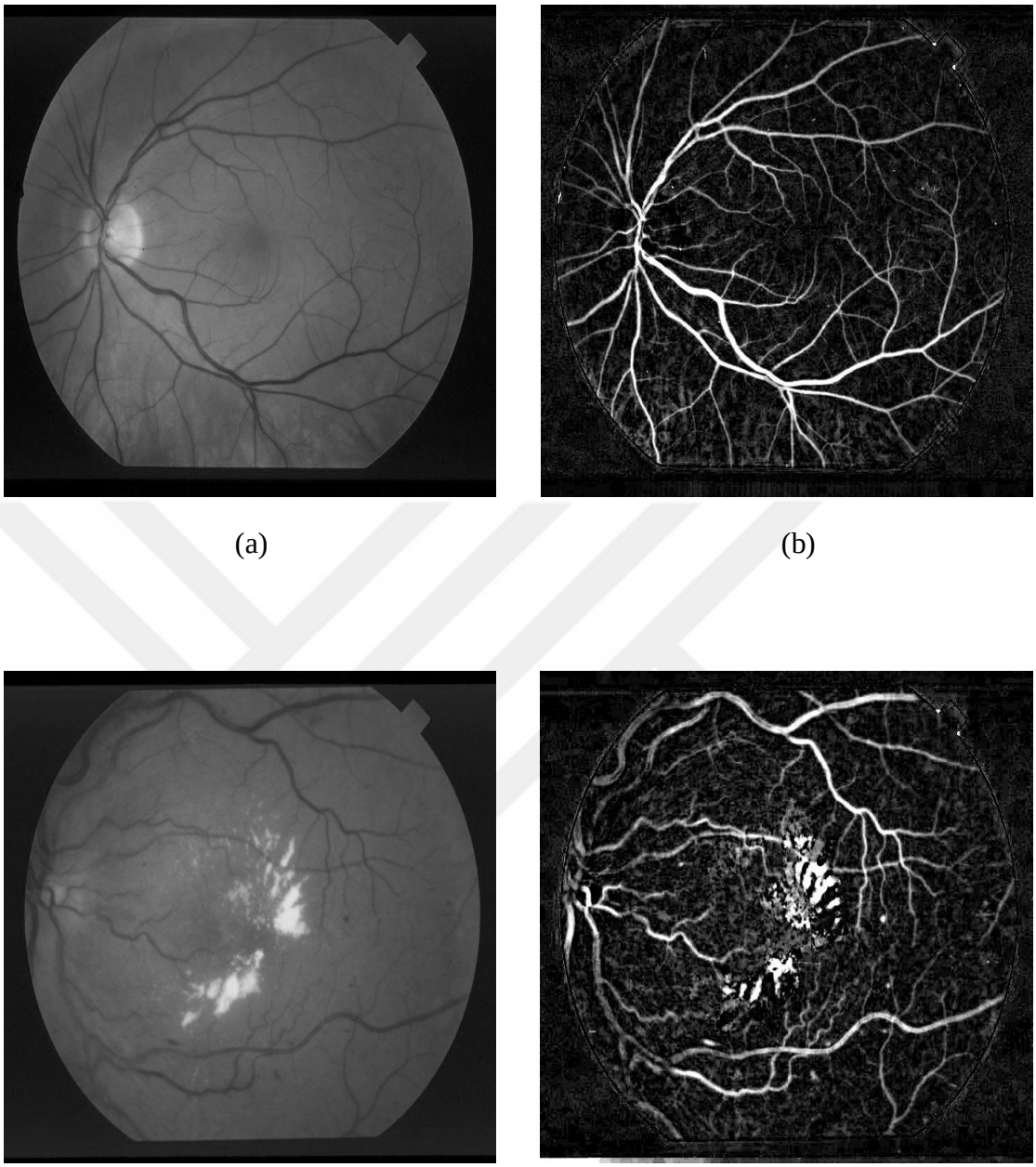
Şekil 4.2. STARE veri tabanından alınan sağlıklı ve hastalıklı retinal görüntüler. (a) Sağlıklı görüntü. (b) Kanama içeren hastalıklı görüntü.



(a) (b)

(c) (d)

Şekil 4.3. Önişleme aşamaları sonucunda elde edilen DRIVE veri tabanı görüntüleri.
(a, c) Bant seçimi önişlemi uygulandıktan sonra elde edilen G katmanları.
(b, d) Bottom-hat filtreleme ve parlaklık düzeltme önişlemleri uygulandıktan sonra elde edilen görüntüler.

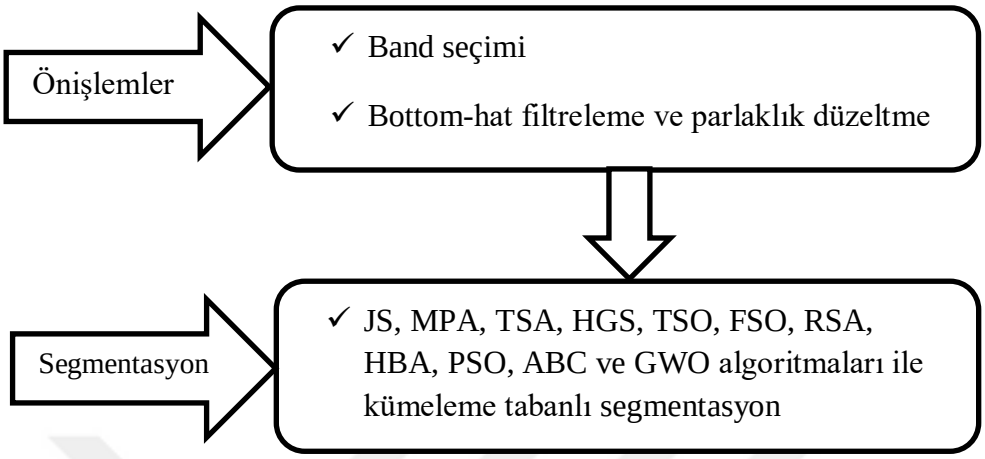


(a) (b)

(c) (d)

Şekil 4.4. Önişleme aşamaları sonucunda elde edilen STARE veri tabanı görüntüleri.
(a, c) Bant seçimi önişlemi uygulandıktan sonra elde edilen G katmanları.
(b, d) Bottom-hat filtreleme ve parlaklık düzeltme önişlemleri uygulandıktan sonra elde edilen görüntüler.

Retina damar segmentasyonu prosedürünün akış diyagramı Şekil 4.5'te gösterilmiştir.



Şekil 4.5. Retinal damar segmentasyonu akış diyagramı.

Simülasyonlarda, görüntü içerisindeki pikseller, damar pikselleri ve damar olmayan pikseller olarak kümelenmiştir. Hem damar pikselleri hem de damar olmayan pikseller için piksel aralıkları oluşturmak amacıyla küme merkezi sayısı $k = 2$ olarak seçilmiştir. JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmaları kullanılarak her birisi optimal bir küme merkezi değerini temsil eden optimal piksel değerleri elde edilmiştir.

Optimizasyon sürecinin ilk aşamasında, her bir piksel değeri elde edilen uygunluk değerlerine göre en yakın küme merkezlerine atanır. Her bir olası küme merkezinin uygunluk değeri, aşağıda verilen ortalama karesel hata (Mean-Squared Error, MSE) fonksiyonu kullanılarak hesaplanabilir.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2 \tag{4.1}$$

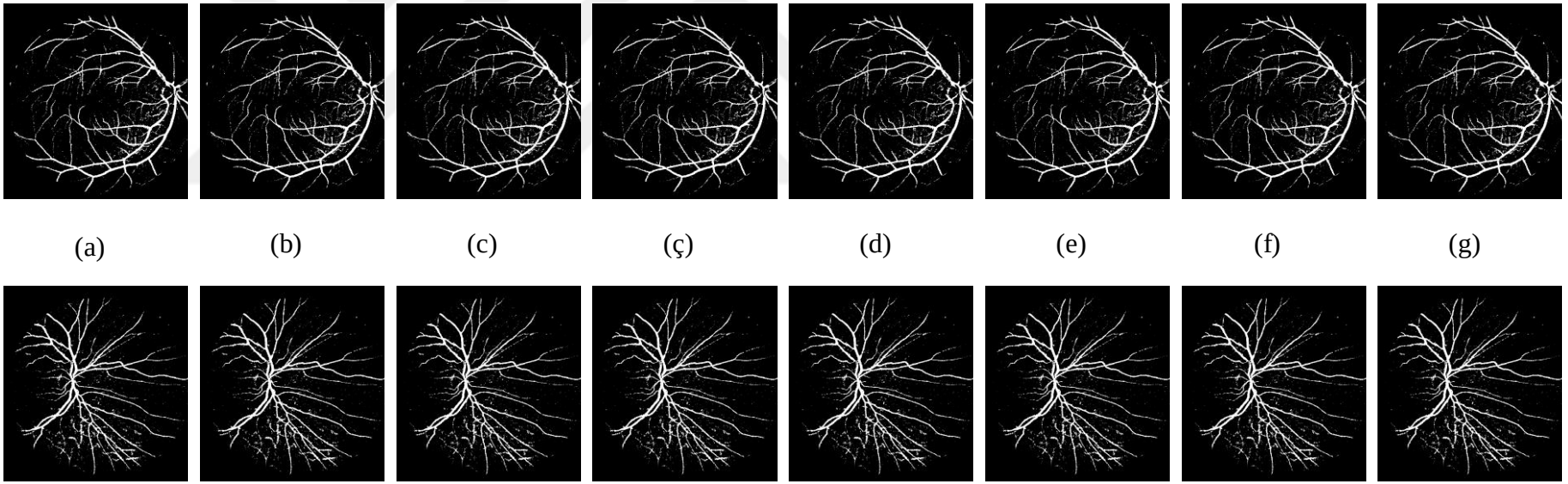
MSE hata fonksiyonu, her bir piksel ile ilgili küme merkezi arasındaki mesafeye dayalı bir kalite değeri hesaplama prosedürü yürütmektedir. Bu denklemde, N retina görüntüsündeki toplam piksel sayısını temsil etmektedir. f_i, i'nci piksele en yakın küme merkezi değerini ve y_i ise i'nci pikselin piksel değerini temsil eder. MSE hata fonksiyonu kullanılarak piksellerin uygun küme merkezleri etrafında minimum hata ile gruplanması hedeflenmektedir.

Algoritmaların tamamında kullanılmış olan dört adet ortak kontrol parametresi bulunmaktadır. Bunlar; popülasyon boyutu, maksimum çevrim sayısı, X_{max} ve X_{min} parametreleridir. Simülasyonlarda, her bir algoritma için popülasyon boyutu 10 olarak, maksimum çevrim sayısı ise 100 olarak belirlenmiştir. Ayrıca, X_{max} ve X_{min} sırasıyla ilgili görüntünün en yüksek ve en düşük piksel değerlerini temsil etmektedir. Ortak kontrol parametreleri haricinde algoritmaların diğer kontrol parametreleri ve ilgili değerler aşağıda Tablo 4.1'de verilmiştir.

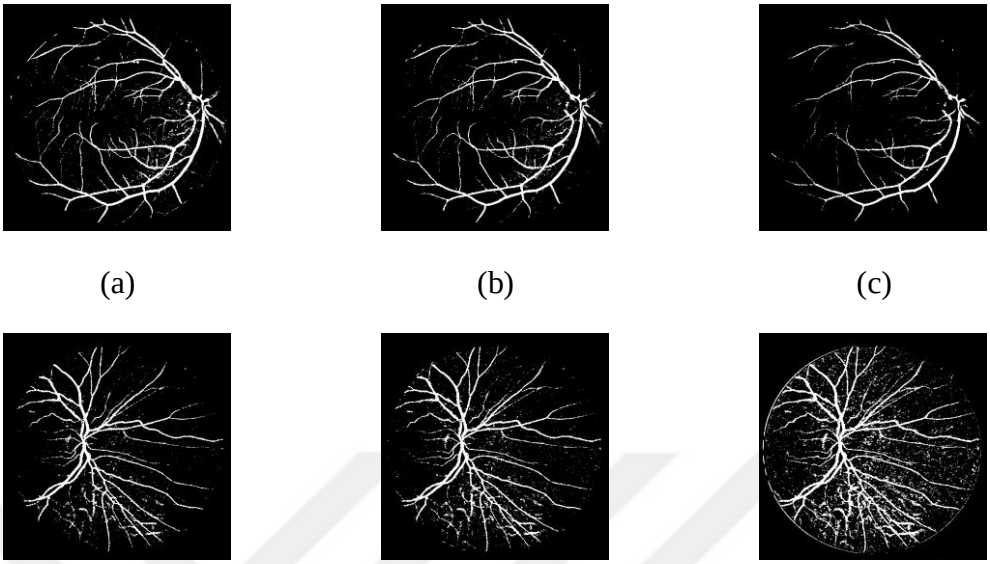
Tablo 4.1. Simülasyonlarda kullanılan kontrol parametre değerleri.

Algoritma	Kontrol Parametre Değerleri
JS	-
MPA	$FADs = 0.2$ ve $P = 0.5$
TSA	$P_{min} = 1$ ve $P_{max} = 4$
HGS	$E = 0.03$ ve $SHungry = 0$
TSO	$a = 0.7$ ve $z = 0.05$
FSO	-
RSA	$\alpha = 0.1$ ve $\beta = 0.005$
HBA	$\beta = 6$ ve $C=2$
PSO	$w = 0.72$, $c_1 = 1.49$ ve $c_2 = 1{,}49$
ABC	-
GWO	$\vec{a}$ = [2,0] aralığında her bir çevrimde doğrusal olarak azalan katsayı, $\vec{A}$ ve $\vec{C}$ = rasgele üretilen katsayılara göre oluşan katsayı vektörleri

JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları kullanılarak, Şekil 4.1 (a) ve 4.1 (b)'deki görüntülere uygulanan segmentasyon işlemi sonrasında elde edilen retina görüntüleri Şekil 4.6'da verilmiştir. Benzer şekilde, PSO, ABC ve GWO tabanlı segmentasyon sonrası elde edilen görüntüler ise Şekil 4.7'de verilmiştir. Görüldüğü gibi arka plana ait bazı pikseller, damar piksellerine yakın piksel değerine sahip oldukları için damar pikseli olarak hatalı bir şekilde tanımlanabilmektedir. Segmentasyon sürecinde karşılaşılan bu zorlukların üstesinden gelmek için literatürde bazı son işlem yöntemleri önerilmiştir ancak bu çalışmada algoritmaların gerçek performanslarını ortaya koyabilmek amacıyla bu son işlem yöntemleri kullanılmamıştır. Simülasyon sonuçları incelendiğinde JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının PSO, ABC ve GWO algoritmalarına benzer şekilde, yakın piksel değerlerini yüksek doğrulukla ayırt edebildiğini göstermiştir. Şekil 4.8'de STARE veri tabanından alınan Şekil 4.2 (a) ve 4.2 (b) görüntüleri için JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının kümeleme performansları gösterilmektedir. Benzer şekilde, Şekil 4.9'da STARE veri tabanı için PSO, ABC ve GWO algoritmaları tarafından üretilen retinal görüntüler gösterilmiştir. Elde edilen sonuçlardan, algoritmaların genel olarak yüksek segmentasyon performansları ürettikleri, bununla birlikte, STARE veri tabanında DRIVE veri tabanına kıyasla daha düşük performansların ortaya çıktığı görülmüştür. Son olarak, simülasyon sonuçlarından, metasezgisel algoritmaların literatürdeki geleneksel algoritmalara benzer fakat biraz daha iyi performanslar sergiledikleri gözlemlenmiştir.



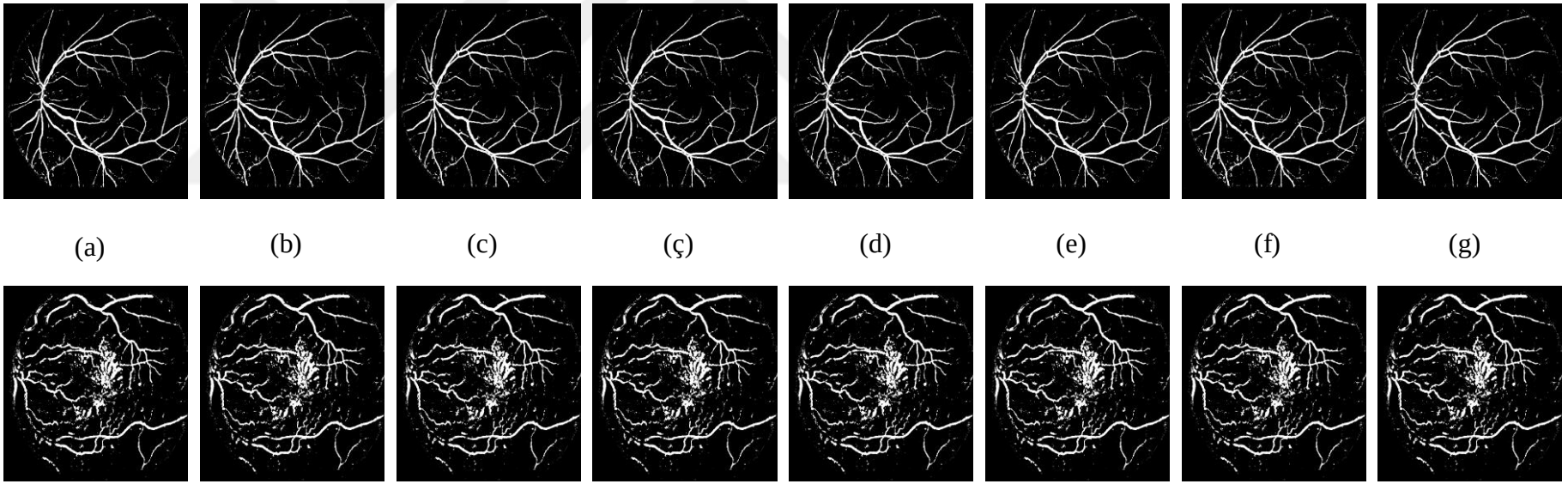
Şekil 4.6. Kümeleme tabanlı JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları kullanılarak Şekil 4.1 (a) ve 4.1 (b) için elde edilen segmentasyon sonuçları. (a) JS. (b) MPA. (c) TSA. (ç) HGS. (d) TSO. (e) FSO. (f) RSA. (g) HBA. (ğ) JS. (h) MPA. (ı) TSA. (i) HGS. (j) TSO. (k) FSO. (l) RSA. (m) HBA.



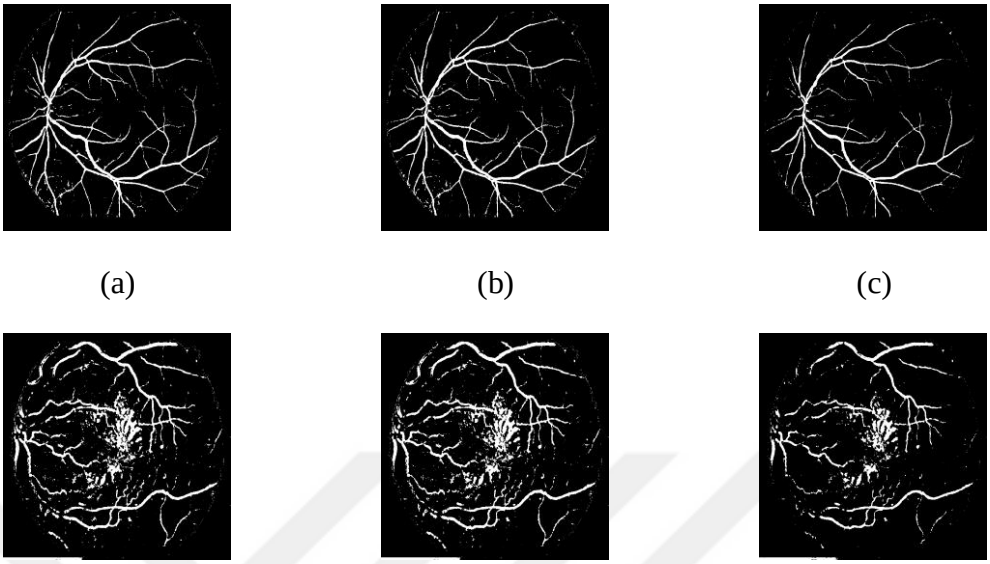
(a) (b) (c)

(ç) (d) (e)

Şekil 4.7. Kümeleme tabanlı PSO, ABC ve GWO algoritmaları kullanılarak Şekil 4.1 (a) ve 4.1 (b) için elde edilen segmentasyon sonuçları. (a) PSO. (b) ABC. (c) GWO. (ç) PSO. (d) ABC. (e) GWO.



Şekil 4.8. Kümeleme tabanlı JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları kullanılarak Şekil 4.2 (a) ve 4.2 (b) için elde edilen segmentasyon sonuçları. (a) JS. (b) MPA. (c) TSA. (ç) HGS. (d) TSO. (e) FSO. (f) RSA. (g) HBA. (ğ) JS. (h) MPA. (ı) TSA. (i) HGS. (j) TSO. (k) FSO. (l) RSA. (m) HBA.



(a) (b) (c)

(ç) (d) (e)

Şekil 4.9. Kümeleme tabanlı PSO, ABC ve GWO algoritmaları kullanılarak Şekil 4.2 (a) ve 4.2 (b) için elde edilen segmentasyon sonuçları. (a) PSO. (b) ABC. (c) GWO. (ç) PSO. (d) ABC. (e) GWO.

4.1. Performans Ölçümleri

Geliştirilen algoritmaların matematiksel tabanlı performansları Duyarlılık (*SE*), Özgüllük (*SP*), Doğruluk (*ACC*) ve F-skor metrikleri açısından da detaylı bir şekilde mukayese edilmiştir. Ayrıca, algoritmaların yakınsama hızı, MSE, CPU süresi ve standart sapma açısından performansları da birbirleriyle karşılaştırılmıştır.

SE, SP, ACC ve F-skor parametrelerinin matematiksel ifadeleri aşağıdaki gibi tanımlanmaktadır.

$$SE = \frac{TP}{(TP+FN)} \tag{4.2}$$

$$SP = \frac{TN}{(TN+FP)} \tag{4.3}$$

$$ACC = \frac{(TP+TN)}{(TP+FN+TN+FP)} \tag{4.4}$$

$$F - skor = \frac{TP}{[TP+1/2(FP+FN)]} \tag{4.5}$$

burada; TP doğru tahmin edilen damar piksel sayısını, FN damar pikseli olup yanlış bir şekilde arka plan pikseli olarak tanımlanan piksel sayısını, TN doğru tahmin edilen arka plan piksel sayısını ve son olarak FP arka plan pikseli olup yanlış bir şekilde damar pikseli olarak tanımlanan piksel sayısını ifade etmektedir.

4.2. DRIVE ve STARE Veri Tabanları Performans Analizleri

Tablo 4.2 ve Tablo 4.3'te, DRIVE ve STARE veri tabanları için 20 farklı görüntü üzerinde algoritmalar tarafından elde edilen SE, SP ve ACC sonuçları verilmiştir. Benzer şekilde PSO, ABC ve GWO algoritmaları için elde edilen sonuçlar, DRIVE veri tabanı için Tablo 4.4'te ve STARE veri tabanı için ise Tablo 4.5'te gösterilmiştir. SE, SP ve ACC parametrelerine ek olarak her bir algoritma için DRIVE ve STARE veri tabanlarında elde edilen F-skor değerleri, sırasıyla, Tablo 4.6 ve Tablo 4.7'de sunulmuştur.

Tablo 4.2. DRIVE veri tabanı 20 farklı retinal görüntü için JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri.

Görüntü	Duyarlılık								Özgüllük								Doğruluk							
	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA
1	0,8913	0,8913	0,8913	0,8375	0,8375	0,8375	0,7869	0,8375	0,9884	0,9884	0,9884	0,9749	0,9749	0,9749	0,9793	0,9749	0,9791	0,9791	0,9791	0,9644	0,9644	0,9644	0,9625	0,9644
2	0,7857	0,7857	0,7857	0,9387	0,9387	0,9387	0,9102	0,9387	0,9703	0,9703	0,9703	0,9655	0,9655	0,9655	0,9698	0,9655	0,9479	0,9479	0,9479	0,9635	0,9635	0,9635	0,965	0,9635
3	0,8787	0,8787	0,8787	0,6616	0,6616	0,6616	0,573	0,6616	0,9867	0,9867	0,9867	0,9755	0,9755	0,9755	0,9807	0,9755	0,976	0,976	0,976	0,9437	0,9437	0,9437	0,9294	0,9437
4	0,723	0,723	0,723	0,9464	0,9464	0,9464	0,9278	0,9464	0,9645	0,9645	0,9645	0,9615	0,9615	0,9615	0,9648	0,9615	0,9371	0,9371	0,9371	0,9606	0,9606	0,9606	0,9626	0,9606
5	0,9514	0,9514	0,9514	0,7867	0,7867	0,7867	0,5487	0,7867	0,9955	0,9955	0,9955	0,9777	0,9777	0,9777	0,9865	0,9777	0,9918	0,9918	0,9918	0,9629	0,9629	0,9629	0,931	0,9629
6	0,9572	0,9572	0,9572	0,6958	0,6958	0,6958	0,5801	0,6958	0,9954	0,9954	0,9954	0,9718	0,9718	0,9718	0,9772	0,9718	0,9917	0,9917	0,9917	0,945	0,945	0,945	0,9271	0,945
7	0,8874	0,8874	0,8874	0,669	0,669	0,669	0,3282	0,669	0,9903	0,9903	0,9903	0,9808	0,9808	0,9808	0,9922	0,9808	0,9822	0,9822	0,9822	0,9557	0,9557	0,9557	0,86	0,9557
8	0,5506	0,5506	0,5506	0,4551	0,4551	0,4551	0,3523	0,4551	0,9449	0,9449	0,9449	0,9869	0,9869	0,9869	0,99	0,9869	0,9019	0,9019	0,9019	0,9225	0,9225	0,9225	0,8845	0,9225
9	0,7818	0,7818	0,7818	0,5847	0,5847	0,5847	0,4565	0,5847	0,9761	0,9761	0,9761	0,9773	0,9773	0,9773	0,9819	0,9773	0,9568	0,9568	0,9568	0,9367	0,9367	0,9367	0,9068	0,9367
10	0,8932	0,8932	0,8932	0,8089	0,8039	0,8039	0,5872	0,8089	0,9908	0,9908	0,9908	0,978	0,978	0,978	0,9855	0,978	0,9831	0,9831	0,9831	0,9673	0,9673	0,9673	0,9457	0,9673
11	0,787	0,787	0,787	0,8795	0,8795	0,8795	0,8474	0,8795	0,976	0,976	0,976	0,9682	0,9682	0,9682	0,9716	0,9682	0,9569	0,9569	0,9569	0,9629	0,9629	0,9629	0,9633	0,9629
12	0,6624	0,6624	0,6624	0,5479	0,5479	0,5479	0,4398	0,5479	0,9574	0,9574	0,9574	0,9842	0,9842	0,9842	0,9879	0,9842	0,9243	0,9243	0,9243	0,9313	0,9313	0,9313	0,9005	0,9313
13	0,7992	0,7992	0,7992	0,8461	0,8461	0,8461	0,8461	0,8461	0,9725	0,9725	0,9725	0,9599	0,9599	0,9599	0,9599	0,9599	0,9516	0,9516	0,9516	0,9513	0,9513	0,9513	0,9513	0,9513
14	0,7439	0,7439	0,7439	0,6156	0,6156	0,6156	0,4179	0,6156	0,973	0,973	0,973	0,9857	0,9857	0,9857	0,9914	0,9857	0,9511	0,9511	0,9511	0,9485	0,9485	0,9485	0,8988	0,9485
15	0,9444	0,9444	0,9444	0,7257	0,7257	0,7257	0,7257	0,7257	0,9953	0,9953	0,9953	0,9746	0,9746	0,9746	0,9746	0,9746	0,9913	0,9913	0,9913	0,9571	0,9571	0,9571	0,9571	0,9571
16	0,8387	0,8387	0,8387	0,8776	0,8776	0,8776	0,7099	0,8776	0,9824	0,9824	0,9824	0,9734	0,9734	0,9734	0,9813	0,9734	0,9683	0,9683	0,9683	0,9668	0,9668	0,9668	0,9552	0,9668
17	0,7017	0,7017	0,7017	0,5701	0,5701	0,5701	0,4695	0,5701	0,9672	0,9672	0,9672	0,9845	0,9845	0,9845	0,9878	0,9845	0,9409	0,9409	0,9409	0,9409	0,9409	0,9409	0,9178	0,9409
18	0,8972	0,8972	0,8972	0,8025	0,8025	0,8025	0,8025	0,8025	0,9885	0,9885	0,9885	0,971	0,971	0,971	0,971	0,971	0,9794	0,9794	0,9794	0,9573	0,9573	0,9573	0,9573	0,9573
19	0,8161	0,8161	0,8161	0,881	0,881	0,881	0,7966	0,881	0,9758	0,9758	0,9758	0,9659	0,9659	0,9659	0,9713	0,9659	0,9572	0,9572	0,9572	0,9593	0,9593	0,9593	0,9553	0,9593
20	0,7477	0,7477	0,7477	0,5852	0,6958	0,5852	0,4693	0,5852	0,9667	0,9667	0,9667	0,977	0,9714	0,977	0,9821	0,977	0,9412	0,9412	0,9412	0,9285	0,945	0,9285	0,8972	0,9285
Ortalama	**0,81193**	**0,81193**	**0,81193**	**0,73578**	**0,74106**	**0,73553**	**0,62878**	**0,73578**	**0,977885**	**0,977885**	**0,977885**	**0,974715**	**0,974435**	**0,974715**	**0,97934**	**0,974715**	**0,96049**	**0,96049**	**0,96049**	**0,95131**	**0,952135**	**0,95131**	**0,93142**	**0,95131**

Tablo 4.3. STARE veri tabanı 20 farklı retinal görüntü için JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri.

Görüntü	Duyarlılık								Özgüllük								Doğruluk							
	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA
1	0,7498	0,7498	0,7498	0,5163	0,5163	0,5163	0,5163	0,5163	0,9742	0,9742	0,9742	0,9747	0,9747	0,9747	0,9747	0,9747	0,9533	0,9533	0,9533	0,9298	0,9298	0,9298	0,9298	0,9298
2	0,6876	0,6876	0,6876	0,4574	0,4574	0,4574	0,3998	0,4574	0,9701	0,9701	0,9701	0,9766	0,9766	0,9766	0,9806	0,9766	0,9454	0,9454	0,9454	0,9286	0,9286	0,9286	0,9132	0,9286
3	0,6423	0,6423	0,6423	0,4552	0,4552	0,4552	0,4054	0,4552	0,9568	0,9568	0,9568	0,9753	0,9753	0,9753	0,9787	0,9753	0,9229	0,9229	0,9229	0,9146	0,9146	0,9146	0,8986	0,9146
4	0,3513	0,3513	0,3513	0,27	0,27	0,27	0,27	0,27	0,8508	0,8508	0,8508	0,9733	0,9733	0,9733	0,9733	0,9733	0,7575	0,7575	0,7575	0,7997	0,7997	0,7997	0,7997	0,7997
5	0,8407	0,8407	0,8407	0,7198	0,7198	0,7198	0,6738	0,7198	0,9825	0,9825	0,9825	0,9616	0,9616	0,9616	0,9682	0,9616	0,9685	0,9685	0,9685	0,9448	0,9448	0,9448	0,9435	0,9448
6	0,8211	0,8211	0,8211	0,7411	0,7411	0,7411	0,6469	0,7411	0,981	0,981	0,981	0,9633	0,9633	0,9633	0,9716	0,9633	0,9656	0,9656	0,9656	0,949	0,949	0,949	0,9436	0,949
7	0,9324	0,9945	0,9945	0,7076	0,7076	0,7076	0,6521	0,7076	0,9937	0,9995	0,9995	0,974	0,974	0,974	0,9785	0,974	0,9884	0,9991	0,9991	0,9526	0,9526	0,9526	0,9479	0,9526
8	0,9987	0,8955	0,8955	0,6654	0,6654	0,6654	0,6228	0,6654	0,999	0,9906	0,9906	0,979	0,979	0,979	0,9821	0,979	0,9998	0,9828	0,9828	0,9527	0,9527	0,9527	0,9482	0,9527
9	0,9281	0,9281	0,9281	0,7604	0,7604	0,7604	0,7328	0,7604	0,9932	0,9932	0,9932	0,9738	0,9738	0,9738	0,9762	0,9738	0,9876	0,9876	0,9876	0,9578	0,9578	0,9578	0,9565	0,9578
10	0,8143	0,8143	0,8143	0,5664	0,5664	0,5664	0,5195	0,5664	0,9796	0,9796	0,9796	0,9716	0,9716	0,9716	0,9758	0,9716	0,9633	0,9633	0,9633	0,93	0,93	0,93	0,921	0,93
11	0,9377	0,9377	0,9377	0,7396	0,7396	0,7396	0,7005	0,7396	0,995	0,995	0,995	0,9765	0,9765	0,9765	0,9796	0,9765	0,9907	0,9907	0,9907	0,9609	0,9609	0,9609	0,9589	0,9609
12	0,8163	0,8163	0,8163	0,7899	0,7899	0,7899	0,7899	0,7899	0,9785	0,9785	0,9785	0,963	0,963	0,963	0,963	0,963	0,9615	0,9615	0,9615	0,9511	0,9511	0,9511	0,9511	0,9511
13	0,8813	0,8813	0,8813	0,6632	0,6632	0,6632	0,5681	0,6632	0,9883	0,9883	0,9883	0,963	0,963	0,963	0,9759	0,963	0,9787	0,9787	0,9787	0,9421	0,9421	0,9421	0,9334	0,9421
14	0,9822	0,9822	0,9822	0,6804	0,6804	0,6504	0,6504	0,6804	0,9986	0,9986	0,9986	0,9756	0,9756	0,9783	0,9783	0,9756	0,9973	0,9973	0,9973	0,9535	0,9535	0,9513	0,9513	0,9535
15	0,7968	0,7968	0,7968	0,5866	0,5866	0,7968	0,3504	0,5866	0,9794	0,9794	0,9794	0,9777	0,9777	0,9794	0,9868	0,9777	0,9625	0,9625	0,9625	0,9403	0,9403	0,9625	0,8677	0,9403
16	0,9095	0,9095	0,9095	0,6488	0,6488	0,6488	0,6488	0,6488	0,9913	0,9913	0,9913	0,9745	0,9745	0,9745	0,9745	0,9745	0,9841	0,9841	0,9841	0,9454	0,9454	0,9454	0,9454	0,9454
17	0,9449	0,9449	0,9449	0,728	0,728	0,728	0,6545	0,728	0,9955	0,9955	0,9955	0,9755	0,9755	0,9755	0,9811	0,9755	0,9916	0,9916	0,9916	0,9586	0,9586	0,9586	0,9535	0,9586
18	0,7593	0,7593	0,7593	0,6584	0,5577	0,6584	0,5577	0,6584	0,9748	0,9748	0,9748	0,9563	0,9619	0,9563	0,9619	0,9563	0,9544	0,9544	0,9544	0,9409	0,9329	0,9409	0,9329	0,9409
19	0,9055	0,9055	0,9055	0,4203	0,4203	0,4203	0,3042	0,4203	0,9896	0,9896	0,9896	0,9455	0,9455	0,9455	0,9516	0,9455	0,9813	0,9813	0,9813	0,8924	0,8924	0,8924	0,8425	0,8924
20	0,8817	0,8817	0,8817	0,5321	0,5321	0,5823	0,4534	0,5823	0,9869	0,9869	0,9869	0,9591	0,9591	0,9517	0,9672	0,9517	0,9764	0,9764	0,9764	0,9165	0,9165	0,9229	0,8974	0,9229
Ortalama	**0,829075**	**0,82702**	**0,82702**	**0,615345**	**0,61031**	**0,626865**	**0,555865**	**0,617855**	**0,97794**	**0,97781**	**0,97781**	**0,969495**	**0,969775**	**0,969345**	**0,97398**	**0,969125**	**0,96154**	**0,961225**	**0,961225**	**0,933065**	**0,932665**	**0,934385**	**0,921805**	**0,933385**

Tablo 4.4. DRIVE veri tabanı 20 farklı retinal görüntü için PSO, ABC ve GWO algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri.

Görüntü	Duyarlılık			Özgüllük			Doğruluk		
	PSO	ABC	GWO	PSO	ABC	GWO	PSO	ABC	GWO
1	0,8913	0,8913	0,8606	0,9884	0,9884	0,9847	0,9791	0,9791	0,9724
2	0,7857	0,8302	0,8302	0,9703	0,9776	0,9776	0,9479	0,9604	0,9604
3	0,8787	0,9408	0,5392	0,9867	0,9939	0,923	0,976	0,9889	0,868
4	0,723	0,723	0,7793	0,9645	0,9645	0,9735	0,9371	0,9371	0,9527
5	0,9514	0,9514	0,9514	0,9955	0,9955	0,9955	0,9918	0,9918	0,9918
6	0,9572	0,9572	0,8605	0,9954	0,9954	0,9836	0,9917	0,9917	0,9707
7	0,8874	0,9593	0,7331	0,9903	0,9967	0,9727	0,9822	0,994	0,9504
8	0,5506	0,5506	0,8672	0,9449	0,9449	0,9892	0,9019	0,9019	0,98
9	0,7818	0,7818	0,8251	0,9761	0,9761	0,9817	0,9568	0,9568	0,9669
10	0,8932	0,8932	0,8932	0,9908	0,9908	0,9908	0,9831	0,9831	0,9831
11	0,787	0,787	0,9879	0,976	0,976	0,9989	0,9569	0,9569	0,998
12	0,6624	0,6624	0,6824	0,9574	0,9574	0,9609	0,9243	0,9243	0,9305
13	0,7992	0,7407	0,6489	0,9725	0,9621	0,9426	0,9516	0,9338	0,9013
14	0,7439	0,7439	0,9292	0,973	0,973	0,9939	0,9511	0,9511	0,9887
15	0,9444	0,8966	0,5551	0,9953	0,9908	0,9393	0,9913	0,9831	0,8932
16	0,8387	0,7784	0,7084	0,9824	0,9742	0,9631	0,9683	0,9538	0,9345
17	0,7017	0,7017	0,7017	0,9672	0,9672	0,9672	0,9409	0,9409	0,9409
18	0,8972	0,8972	0,7438	0,9885	0,9885	0,9663	0,9794	0,9794	0,9405
19	0,8161	0,8161	0,5084	0,9758	0,9758	0,9037	0,9572	0,9572	0,839
20	0,7477	0,9646	0,7464	0,9967	0,9963	0,9665	0,9412	0,9933	0,9408
Ortalama	**0,81193**	**0,823237**	**0,7676**	**0,979385**	**0,979255**	**0,968735**	**0,96049**	**0,96293**	**0,94519**

Tablo 4.5. STARE veri tabanı 20 farklı retinal görüntü için PSO, ABC ve GWO algoritmalarının duyarlılık, özgüllük ve doğruluk açısından performans ölçümleri.

Görüntü	Duyarlılık			Özgüllük			Doğruluk		
	PSO	ABC	GWO	PSO	ABC	GWO	PSO	ABC	GWO
1	0,7205	0,7205	0,9611	0,9702	0,9702	0,9968	0,9461	0,9461	0,9941
2	0,6876	0,6876	0,8461	0,9701	0,9701	0,9878	0,9454	0,9454	0,9774
3	0,6423	0,6423	0,9761	0,9568	0,9568	0,998	0,9229	0,9229	0,9963
4	0,6634	0,3513	0,6634	0,954	0,8508	0,954	0,9191	0,7575	0,9191
5	0,8407	0,8407	0,7572	0,9825	0,9825	0,9708	0,9685	0,9685	0,9479
6	0,8211	0,8211	0,7104	0,981	0,981	0,9649	0,9656	0,9656	0,9374
7	0,9945	0,9324	0,7999	0,9995	0,9937	0,9785	0,9991	0,9884	0,9612
8	0,8955	0,8955	0,7947	0,9906	0,9906	0,9795	0,9828	0,9828	0,9627
9	0,9281	0,9962	0,9281	0,9932	0,9997	0,9932	0,9876	0,9994	0,9876
10	0,8143	0,8143	0,925	0,9796	0,9796	0,9927	0,9633	0,9633	0,9866
11	0,9377	0,9377	0,7654	0,995	0,995	0,9772	0,9907	0,9907	0,9584
12	0,8163	0,8163	0,8163	0,9785	0,9785	0,9785	0,9615	0,9615	0,9615
13	0,8813	0,8813	0,7034	0,9883	0,9883	0,9643	0,9787	0,9787	0,9362
14	0,9822	0,9822	0,7102	0,9986	0,9986	0,9687	0,9973	0,9973	0,9434
15	0,7968	0,7968	0,9393	0,9794	0,9794	0,9947	0,9625	0,9625	0,9902
16	0,9905	0,9095	0,9095	0,9913	0,9913	0,9913	0,9841	0,9841	0,9841
17	0,9449	0,9612	0,9612	0,9955	0,9969	0,9969	0,9916	0,9942	0,9942
18	0,7593	0,7593	0,7593	0,9748	0,9748	0,9748	0,9544	0,9544	0,9544
19	0,9095	0,9055	0,9055	0,9896	0,9896	0,9896	0,9813	0,9813	0,9813
20	0,8817	0,8817	0,6438	0,9869	0,9869	0,9482	0,9764	0,9764	0,9095
Ortalama	**0,84541**	**0,82667**	**0,823795**	**0,98277**	**0,977715**	**0,98002**	**0,968945**	**0,96105**	**0,964175**

Tablo 4.6. DRIVE veri tabanı 20 farklı retinal görüntü için F-skor performans analizi.

Görüntü	F-skor										
	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA	PSO	ABC	GWO
1	0,7831	0,7831	0,7831	0,7831	0,7831	0,7831	0,7853	0,7831	0,7831	0,7831	0,7638
2	0,7904	0,7904	0,7904	0,7904	0,7904	0,7904	0,8065	0,7904	0,7904	0,8065	0,8065
3	0,7043	0,7043	0,7043	0,7043	0,7043	0,7043	0,6715	0,7043	0,7043	0,7064	0,6055
4	0,7221	0,7221	0,7221	0,7221	0,7221	0,7221	0,7447	0,7221	0,7221	0,7221	0,7162
5	0,766	0,766	0,766	0,766	0,766	0,766	0,6685	0,766	0,766	0,766	0,766
6	0,711	0,711	0,711	0,711	0,711	0,711	0,6678	0,711	0,711	0,711	0,723
7	0,7089	0,7089	0,7089	0,7089	0,7089	0,7089	0,4828	0,7089	0,7089	0,722	0,6757
8	0,587	0,587	0,587	0,587	0,587	0,587	0,5024	0,587	0,587	0,587	0,7029
9	0,6563	0,6563	0,6563	0,6563	0,6563	0,6563	0,5835	0,6563	0,6563	0,6563	0,6912
10	0,7574	0,7574	0,7574	0,7574	0,7574	0,7574	0,6835	0,7574	0,7574	0,7574	0,7574
11	0,7418	0,7418	0,7418	0,7418	0,7418	0,7418	0,7555	0,7418	0,7418	0,7418	0,7546
12	0,6591	0,6591	0,6591	0,6591	0,6591	0,6591	0,5849	0,6591	0,6591	0,6591	0,6575
13	0,7246	0,7246	0,7246	0,7246	0,7246	0,7246	0,7246	0,7246	0,7246	0,7055	0,6058
14	0,706	0,706	0,706	0,706	0,706	0,706	0,5714	0,706	0,706	0,706	0,7623
15	0,7037	0,7037	0,7037	0,7037	0,7037	0,7037	0,7037	0,7037	0,7037	0,6886	0,5825
16	0,7843	0,7843	0,7843	0,7843	0,7843	0,7843	0,7529	0,7843	0,7843	0,7717	0,7046
17	0,6701	0,6701	0,6701	0,6701	0,6701	0,6701	0,6067	0,6701	0,6701	0,6701	0,6701
18	0,7538	0,7538	0,7538	0,7538	0,7538	0,7538	0,7538	0,7538	0,7538	0,7538	0,6981
19	0,7691	0,7691	0,7691	0,7691	0,7691	0,7691	0,7661	0,7691	0,7691	0,7691	0,5967
20	0,6696	0,6696	0,6696	0,6696	0,7083	0,6696	0,6018	0,6696	0,6696	0,7083	0,7055
Ortalama	**0,71843**	**0,71843**	**0,71843**	**0,71843**	**0,720365**	**0,71843**	**0,670895**	**0,71843**	**0,71843**	**0,71959**	**0,697295**

Tablo 4.7. STARE veri tabanı 20 farklı retinal görüntü için F-skor performans analizi.

Görüntü	F-skor										
	JS	MPA	TSA	HGS	TSO	FSO	RSA	HBA	PSO	ABC	GWO
1	0,601	0,601	0,601	0,5902	0,5902	0,5902	0,5902	0,5902	0,601	0,5768	0,5897
2	0,5594	0,5594	0,5594	0,542	0,542	0,542	0,5166	0,542	0,5594	0,542	0,5708
3	0,568	0,568	0,568	0,5544	0,5544	0,5544	0,5277	0,5544	0,568	0,5544	0,6102
4	0,4385	0,4385	0,4385	0,3996	0,3996	0,3996	0,3996	0,3996	0,5233	0,3996	0,5233
5	0,6418	0,6418	0,6418	0,6444	0,6444	0,6444	0,667	0,6444	0,6418	0,6444	0,6137
6	0,6522	0,6522	0,6522	0,6505	0,6505	0,6505	0,6645	0,6505	0,6522	0,6505	0,5906
7	0,7231	0,7248	0,7248	0,7057	0,7057	0,7057	0,7015	0,7057	0,7248	0,7025	0,7015
8	0,7123	0,7098	0,7098	0,7021	0,7021	0,7021	0,6941	0,7021	0,7098	0,7021	0,6986
9	0,7271	0,7271	0,7271	0,7297	0,7297	0,7297	0,7314	0,7297	0,7271	0,7314	0,7271
10	0,6191	0,6191	0,6191	0,6243	0,6243	0,6243	0,6121	0,6243	0,6191	0,6243	0,5992
11	0,7131	0,7131	0,7131	0,7142	0,7142	0,7142	0,7166	0,7142	0,7131	0,7142	0,6682
12	0,6996	0,6996	0,6996	0,6898	0,6898	0,6898	0,6898	0,6898	0,6996	0,6898	0,6996
13	0,6367	0,6367	0,6367	0,6153	0,6153	0,6153	0,6401	0,6153	0,6367	0,6153	0,5642
14	0,6969	0,6969	0,6969	0,6865	0,6865	0,6876	0,6876	0,6865	0,6969	0,6865	0,6783
15	0,6686	0,6686	0,6686	0,6529	0,6529	0,6686	0,4979	0,6529	0,6686	0,6529	0,6865
16	0,6938	0,6938	0,6938	0,6796	0,6796	0,6796	0,6796	0,6796	0,6938	0,6796	0,6938
17	0,7082	0,7082	0,7082	0,7061	0,7061	0,7061	0,7037	0,7061	0,7082	0,7081	0,7097
18	0,5525	0,5525	0,5525	0,5344	0,5442	0,5344	0,5442	0,5344	0,5525	0,5344	0,5525
19	0,4678	0,4678	0,4678	0,4411	0,4411	0,4411	0,3944	0,4411	0,4678	0,4411	0,4678
20	0,5552	0,5552	0,5552	0,5597	0,5597	0,5405	0,5456	0,5405	0,5552	0,5405	0,4092
Ortalama	**0,631745**	**0,631705**	**0,631705**	**0,621125**	**0,621615**	**0,621005**	**0,61021**	**0,620165**	**0,635945**	**0,61952**	**0,617725**

SE, SP, ACC ve F-skor parametreleri için ulaşılabilecek optimum değer 1 olup yüksek metrik değerleri yüksek performanslara karşılık gelmektedir. SE için elde edilen sonuçlar tüm algoritmaların damar piksellerini başarıyla sınıflandırabildiğini kanıtlamaktadır. Ayrıca, simülasyonlarda elde edilen yüksek SP değerleri, damar pikselleri ile arka plan piksellerinin başarılı bir şekilde ayırt edilebildiğini göstermektedir. Son olarak, elde edilen yüksek ACC değerleri, her bir algoritmanın hem damar piksellerini hem de arka plan piksellerini yeterince başarılı bir şekilde sınıflandırabildiğini göstermektedir. F-skor parametresi ise duyarlılık ve kesinlik değerlerini dengeleyen bir skor sağlamaktadır. Algoritmalar tarafından elde edilen yüksek ve benzer F-skor değerleri, tüm görüntüler üzerindeki sınıflandırma başarısını kanıtlamaktadır.

Daha anlaşılır sonuçlar sunabilmek amacıyla tüm parametreler için 20 görüntüde elde edilen ortalama değerler DRIVE ve STARE veri tabanları için Tablo 4.8 ve Tablo 4.9'da birleştirilmiştir. Tablo 4.8'de görüldüğü üzere DRIVE veri tabanında duyarlılık, doğruluk ve F-skor değerleri için ABC algoritması, özgüllük değeri için ise PSO algoritması en iyi sonuçları üretmiştir. Tablo 4.9 incelendiğinde ise STARE veri tabanında duyarlılık, özgüllük, doğruluk ve F-skor değerleri için PSO algoritması en iyi sonuçları ürettiği görülmektedir. Sonuçlar genel olarak değerlendirildiğinde ise metasezgisel algoritmaların benzer kümeleme performansları sergilediği ve kümeleme tabanlı biyomedikal görüntü işleme uygulamalarında başarıyla kullanılabileceği ifade edilebilir.

Tablo 4.8. DRIVE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının ortalama performans mukayesesi.

Metot	Duyarlılık	Özgüllük	Doğruluk	F-skor
JS	0,81193	0,977885	0,96049	0.71843
MPA	0,81193	0,977885	0,96049	0.71843
TSA	0,81193	0,977885	0,96049	0.71843
HGS	0,73578	0,974715	0,95131	0,71843
TSO	0,74106	0,974435	0,952135	0,720365
FSO	0,73553	0,974715	0,95131	0,71843
RSA	0,62878	0,97934	0,93142	0,670895
HBA	0,73578	0,974715	0,95131	0,71843
PSO	0.81193	0.979385	0.96049	0.71843
ABC	0,823237	0,979255	0,96293	0.71959
GWO	0.7676	0.968735	0.94519	0.697295

Tablo 4.9. STARE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının ortalama performans mukayesesi.

Metot	Duyarlılık	Özgüllük	Doğruluk	F-skor
JS	0,829075	0,97794	0,96154	0.631745
MPA	0,82702	0,97781	0,961225	0.631705
TSA	0,82702	0,97781	0,961225	0.631705
HGS	0,615345	0,969495	0,933065	0,621125
TSO	0,61031	0,969775	0,932665	0,621615
FSO	0,626865	0,969345	0,934385	0,621005
RSA	0,555865	0,97398	0,921805	0,61021
HBA	0,617855	0,969125	0,933385	0,620165
PSO	0.84541	0.98277	0.968945	0.635945
ABC	0,82667	0,977715	0,96105	0.61952
GWO	0.823795	0.98002	0.964175	0.617725

Tablo 4.10 ve Tablo 4.11'de, bu tez çalışması kapsamında analiz edilen metasezgisel yaklaşımlar ile literatürde yayımlanan çalışmalar arasında [59-78] detaylı performans mukayeseleri gerçekleştirilmiştir.

DRIVE veri tabanı için verilen Tablo 4.10'dan, RSA algoritması dışındaki diğer tüm algoritmaların duyarlılık ve doğruluk açısından [61] ile yakın sonuçlar ürettiği görülmektedir. RSA algoritmasının duyarlılık ve doğruluk açısından performansı zayıf kalmakla birlikte ABC algoritmasının bu iki metrik açısından performansı diğer tüm algoritmalara göre nispeten daha iyi gözükmektedir. Özgüllük değeri incelendiğinde ise özellikle RSA, PSO ve ABC algoritmalarının [60] ile oldukça yakın sonuçlar ürettiği görülmektedir.

STARE veri tabanı için Tablo 4.11 incelendiğinde JS, MPA, TSA, PSO, ABC ve GWO algoritmalarının duyarlılık açısından literatürdeki optimum sonuçlara [61,76] çok yakın veya daha iyi sonuçlar ürettikleri görülmüştür. Özgüllük ve Doğruluk metrikleri açısından en iyi sonuçların ise PSO algoritması tarafından üretildiği görülmektedir.

DRIVE ve STARE veri tabanları için Tablo 4.10 ve Tablo 4.11'de verilen sonuçlar birlikte değerlendirildiğinde JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının piksel sınıflandırma açısından geleneksel türeve dayalı algoritmalara göre genellikle daha yüksek performans sergiledikleri görülmektedir.

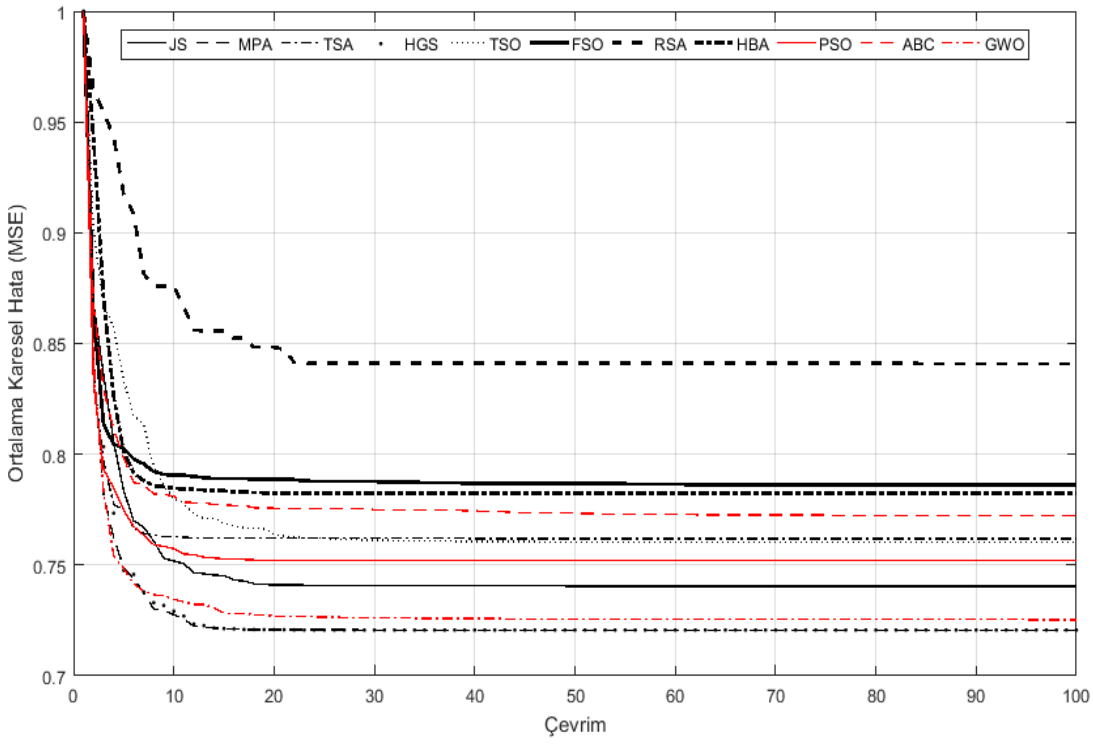
Tablo 4.10. DRIVE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının türeve dayalı algoritmalar ile performans mukayesesi.

Metot	Duyarlılık	Özgüllük	Doğruluk
You et al. [59]	0,7410	0,9751	0,9434
Roychowdhury et al. [60]	0,725	0,983	0,952
Wang et al. [61]	0,8173	0,9733	0,9767
Mendonca et al. [62]	0,7344	0,9764	0,9452
Azzopardi et al. [63]	0,765	0,970	0,944
Fraz et al. [64]	0,7406	0,9807	0,9480
Odstrcilik et al. [65]	0,706	0,9693	0,934
Marin et al. [66]	0,7068	0,9801	0,9452
Kaba et al. [67]	0,7466	0,968	0,941
Dash et al. [68]	0,719	0,976	0,955
Argüello et al. [69]	0,7209	0,9758	0,9431
Asad et al. [70]	0,7388	0,9288	0,9028
Zhao et al. [71]	0,735	0,978	0,947
Zhang et al. [72]	0,7120	0,9724	0,9382
Imani et al. [73]	0,7524	0,9753	0,9523
Metasezgisel algoritmalar			
JS	0,81193	0,977885	0,96049
MPA	0,81193	0,977885	0,96049
TSA	0,81193	0,977885	0,96049
HGS	0,73578	0,974715	0,95131
TSO	0,74106	0,974435	0,952135
FSO	0,73553	0,974715	0,95131
RSA	0,62878	0,97934	0,93142
HBA	0,73578	0,974715	0,95131
PSO	0.81193	0.979385	0.96049
ABC	0,823237	0,979255	0,96293
GWO	0.7676	0.968735	0.94519

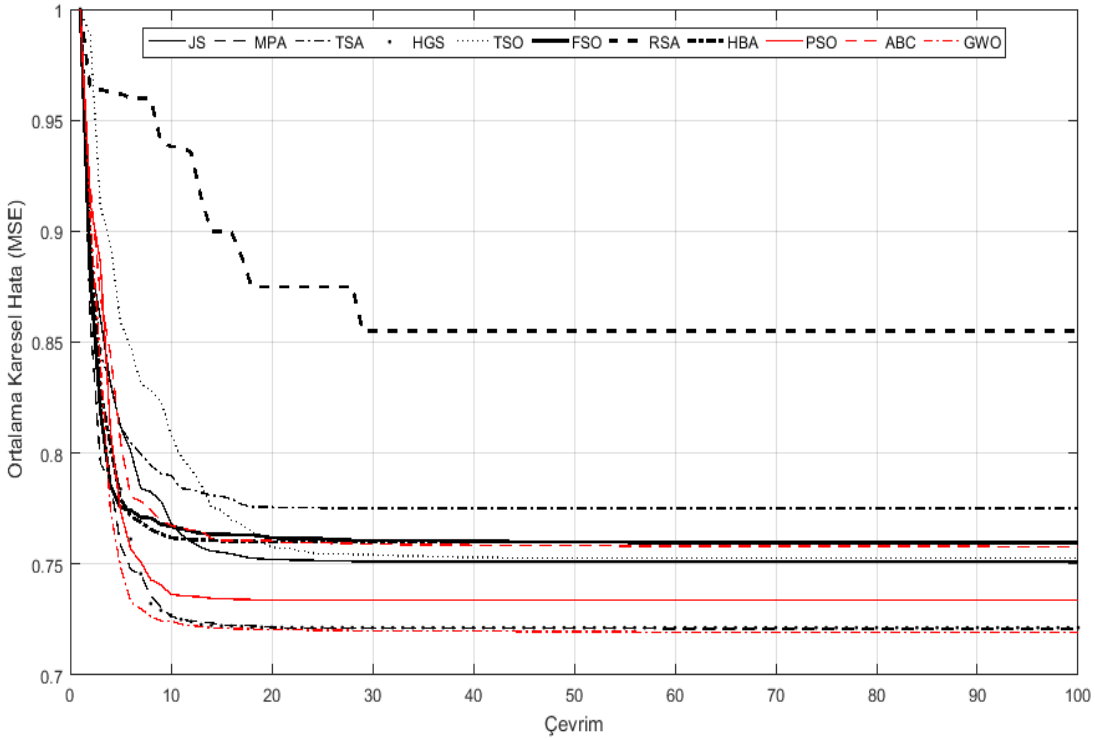
Tablo 4.11. STARE veri tabanı için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının türeve dayalı algoritmalar ile performans mukayesesi.

Metot	Duyarlılık	Özgüllük	Doğruluk
You et al. [59]	0,726	0,976	0,950
Roychowdhury et al. [60]	0,772	0,973	0,952
Wang et al. [61]	0,8104	0,9791	0,9813
Fraz et al. [64]	0,755	0,976	0,953
Odstrcilik et al. [65]	0,7847	0,9512	0,9341
Kaba et al. [67]	0,7619	0,967	0,9456
Argüello et al. [69]	0,7305	0,9688	0,9448
Zhao et al. [71]	0,719	0,977	0,951
Zhang et al. [72]	0,718	0,975	0,948
Imani et al. [73]	0,7502	0,9745	0,959
Mendonca et al. [74]	0,718	0,973	0,946
Xiao et al. [75]	0,715	0,974	0,948
Yin et al. [76]	0,854	0,942	0,933
Lázáret al. [77]	0,725	0,975	0,949
Strisciuglio et al. [78]	0,801	0,972	0,954
Metasezgisel algoritmalar			
JS	0,829075	0,97794	0,96154
MPA	0,82702	0,97781	0,961225
TSA	0,82702	0,97781	0,961225
HGS	0,615345	0,969495	0,933065
TSO	0,61031	0,969775	0,932665
FSO	0,626865	0,969345	0,934385
RSA	0,555865	0,97398	0,921805
HBA	0,617855	0,969125	0,933385
PSO	0.84541	0.98277	0.968945
ABC	0,82667	0,977715	0,96105
GWO	0.823795	0.98002	0.964175

Metasezgisel algoritmalar için bir diğer önemli performans kriteri ise algoritmanın optimal çözümlere ne kadar hızlı veya ne kadar az çevrimde ulaştığını gösteren yakınsama hızıdır. DRIVE veri tabanından alınan hastalıklı retinal görüntü ve STARE veri tabanından alınan sağlıklı retinal görüntü için elde edilmiş olan yakınsama hızları Şekil 4.10'da karşılaştırılmıştır. Şekil 4.10 (a)'dan görüldüğü gibi DRIVE veri tabanından alınan retinal görüntü için MPA algoritması minimum hata değerine en yüksek yakınsama hızı ile ulaşmıştır. Benzer şekilde, Şekil 4.10 (b)'den görüldüğü gibi STARE veri tabanından alınan retinal görüntü için GWO algoritmasının en yüksek yakınsama hızında minimum hata değerine ulaştığı görülmektedir. Genel olarak, RSA haricindeki tüm metasezgisel algoritmaların yaklaşık olarak 15 çevrimde optimal MSE değerlerine ulaştıkları gözlemlenmiştir. RSA ise yaklaşık olarak 30 çevrimde optimal MSE değerine ulaşmıştır. Ayrıca MPA, HGS ve GWO algoritmaları her iki veri tabanı için de yakın minimum MSE değerleri üretmiştir.



(a)



(b)

Şekil 4.10. Algoritmaların yakınsama hızları. (a) Şekil 4.1 (b) için algoritmaların yakınsama hızları. (b) Şekil 4. 2 (a) için algoritmaların yakınsama hızları.

Gerçek hesaplama gereksinimlerini yansıtabilmek ve daha detaylı performans analizi sunabilmek amacıyla Fonksiyon Değerlendirme Sayısı (NFEs) metriği açısından da performans mukayeseleri gerçekleştirilmiştir. NFEs, optimizasyon sürecinde hedef fonksiyonun kaç kez değerlendirildiğini tanımlar ve performans analizi açısından daha adil ve objektif sonuçlar sağlar. Bu tez çalışmasında, her bir algoritmanın NFEs değerlerine ek olarak bu değerlendirmeler için geçen CPU süreleri ve bu sürelerin toplam CPU sürelerine oranı da hesaplanmıştır. NFEs metriği için elde edilen analiz sonuçları Tablo 4.12'de sunulmuştur. Sonuçlardan, NFEs değeri ile CPU süresi arasında bir doğru orantı bulunduğu görülmektedir. Bununla birlikte, Şekil 4.10 ve Tablo 4.12'de birlikte değerlendirildiğinde NFEs değerinin ulaşılan minimum MSE değerleri üzerinde önemli bir değişikliğe yol açmadığı da görülmektedir.

Tablo 4.12. Fonksiyon değerlendirme sayısına (NFEs) dayalı yakınsama analizleri.

Algoritma		NFEs geçen süre (saniye)	
	Toplam NFEs	Şekil 4.1 (b)	Şekil 4.2 (a)
JS	1010	14.841 (Toplam CPU Süre %82)	18.698 (Toplam CPU Süre %80)
MPA	2000	29.373 (Toplam CPU Süre %87)	36.861 (Toplam CPU Süre %86)
TSA	1000	14.909 (Toplam CPU Süre %83)	19.183 (Toplam CPU Süre %80)
HGS	1037	16.922 (Toplam CPU Süre %89)	21.844 (Toplam CPU Süre %91)
TSO	1000	14.972 (Toplam CPU Süre %82)	17.928 (Toplam CPU Süre %79)
FSO	2012	33.390 (Toplam CPU Süre %78)	39.978 (Toplam CPU Süre %92)
RSA	910	13.057 (Toplam CPU Süre %73)	16.375 (Toplam CPU Süre %81)
HBA	1050	14.995 (Toplam CPU Süre %77)	18.829 (Toplam CPU Süre %83)
PSO	1010	14.582 (Toplam CPU Süre %74)	18.070 (Toplam CPU Süre %83)
ABC	2148	30.702 (Toplam CPU Süre %86)	38.236 (Toplam CPU Süre %87)
GWO	1010	16.273 (Toplam CPU Süre %86)	20.032 (Toplam CPU Süre %91)

Algoritmaların kararlılığını temsil eden standart sapma analiz edilmesi gereken bir diğer önemli performans kriterleridir. Bir algoritma her çevrimde yaklaşık olarak benzer MSE değerlerine ulaşabiliyorsa o algoritmanın düşük standart sapma değeri kararlılığının bir göstergesi olacaktır. DRIVE ve STARE veri tabanları için algoritmaların 20 rastgele koşma sonucunda ürettikleri standart sapma değerleri Tablo 4.13'te verilmiştir. Ayrıca, daha ayrıntılı bir analiz sunmak amacıyla, ulaşılan minimum MSE değerleri ve CPU süreleri de Tablo 4.13'e eklenmiştir. Algoritmalar için elde edilen yakın MSE değerleri her bir algoritmanın benzer yakınsama ve kümeleme performansına sahip olduğunun bir göstergesidir. Ayrıca, DRIVE ve STARE veri tabanlarından alınan görüntülerde daha düşük standart sapma değerlerine sahip oldukları için MPA, HGS ve HBA algoritmaları diğer algoritmalara kıyasla kümeleme açısından nispeten daha kararlı gözükmektedir CPU süresi, üzerinde çalışılan donanım ile doğrudan ilişkili bir performans metriğidir. Bu tez çalışması kapsamındaki simülasyonlar; (Intel i3-330M CPU, 2.13 GHz, 3 GB RAM ve 64 bit Windows 7) bir bilgisayar üzerinde gerçekleştirilmiştir. Her iki veri tabanı için elde edilen CPU süreleri incelendiğinde RSA algoritmasın en hızlı algoritma olduğu görülmektedir.

Tablo 4.13. DRIVE ve STARE veri tabanları için JS, MPA, TSA, HGS, TSO, FSO, RSA, HBA, PSO, ABC ve GWO algoritmalarının performans karşılaştırması.

		DRIVE	STARE
JS	Minimum MSE	0.7421	0.7507
	Standart Sapma	1.03908$e-05$	3.11027e − 07
	CPU süresi (saniye)	18.026	23.234
MPA	Minimum MSE	0.7215	0.7218
	Standart Sapma	7.72127$e-09$	4.84936e − 09
	CPU süresi (saniye)	33.942	42.634
TSA	Minimum MSE	0.7645	0.7782
	Standart Sapma	0.020777359	0.01876178
	CPU süresi (saniye)	18.023	24.085
HGS	Minimum MSE	0.7233	0.7116
	Standart Sapma	2.16438e-09	5.64313e-10
	CPU süresi (saniye)	19.001	24.131
TSO	Minimum MSE	0.7515	0.7490
	Standart Sapma	0.000526432	0.050300336
	CPU süresi (saniye)	18.297	22.592
FSO	Minimum MSE	0.7782	0.7610
	Standart Sapma	0.223240165	0.355717255
	CPU süresi (saniye)	42.764	43.554
RSA	Minimum MSE	0.8307	0.8422
	Standart Sapma	0.758405013	0.95267809
	CPU süresi (saniye)	17.937	20.265
HBA	Minimum MSE	0.7745	0.7420
	Standart Sapma	7.72069e-10	3.59347e-10
	CPU süresi (saniye)	19,453	22.671
PSO	Minimum MSE	0.752	0.7337
	Standart Sapma	2.58125e − 08	3.06854$e-08$
	CPU süresi (saniye)	19.588	21.739
ABC	Minimum MSE	0.7722	0.7578
	Standart Sapma	7.18762e-05	6.76833e-05
	CPU süresi (saniye)	35.753	43.804
GWO	Minimum MSE	0.7253	0.719
	Standart Sapma	5.45881e − 06	9.51344$e-06$
	CPU süresi (saniye)	18.959	21.950

Elde edilen sayısal sonuçları istatistiksel olarak doğrulamak için algoritmalar arasındaki anlamlı farklılıkları gösteren Wilcoxon rank-sum test istatistiksel analizi de uygulanmıştır [79]. RStudio-Software kullanılarak %95 güven aralığı ($p < 0.05$) için elde edilen test sonuçları Tablo 4.14'te verilmiştir. DRIVE veri tabanı için elde edilen sonuçlar HGS ve HBA algoritmalarının istatistiksel olarak anlamlı sonuçlar üreten diğer algoritmalardan daha iyi bir performans sergilediğini göstermektedir. STARE veri tabanı için ise JS, MPA, HGS, HBA ve PSO algoritmalarının diğer algoritmalara kıyasla istatistiksel olarak daha anlamlı sonuçlar ürettikleri gözlemlenmiştir. Son olarak her iki veri tabanı için de RSA algoritmasının diğer algoritmalara kıyasla istatistiksel olarak anlamlı sonuçlar üretmediği görülmüştür.

Tablo 4.14. Wilcoxon rank-sum test sonuçları ($p < 0.05$).

	Algoritma	Daha iyi
DRIVE veri tabanı Şekil 4.1 (b)	JS	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) RSA (6.7860e-08) TSA (6.7956e-08)
	MPA	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) RSA (6.7860e-08) TSA (6.7956e-08)
	TSA	FSO (0.0077) RSA (6.7860e-08) ABC (7.8980e-08)
	HGS	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) JS (0.0315) MPA (7.9479e-07) PSO (6.9166e-07) RSA (6.7860e-08) TSA (6.7956e-08) TSO (9.2780e-05)
	TSO	ABC (7.8980e-08) FSO (7.8980e-08) GWO (1.2009e-06) RSA (6.7860e-08) TSA (7.8980e-08)
	FSO	RSA (6.7860e-08)
	RSA	
	HBA	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) MPA (2.2149e-07) PSO (9.1159e-07) RSA (6.7765e-08) TSA (6.7860e-08) TSO (7.4006e-05)
	PSO	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) RSA (6.7860e-08) TSA (6.7956e-08)
	ABC	RSA (6.7860e-08)
	GWO	ABC (6.7956e-08) FSO (6.7956e-08) RSA (6.7860e-08) TSA (6.7956e-08)
STARE veri tabanı Şekil 4.2 (a)	JS	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) RSA (6.7956e-08) TSA (6.7956e-08) TSO (1.4438e-04)
	MPA	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) JS (3.3819e-04) RSA (6.7956e-08) TSA (6.7956e-08)
	TSA	ABC (9.1728e-08) FSO (1.8074e-05) RSA (6.7956e-08)
	HGS	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) JS (0.0098) MPA (1.0646e-07) PSO (7.8980e-08) RSA (6.7956e-08) TSA (6.7956e-08) TSO (1.4309e-07)
	TSO	ABC (1.2346e-07) FSO (3.9388e-07) RSA (6.7956e-08) TSA (1.2009e-06)
	FSO	ABC (2.7451e-04) RSA (6.7956e-08)
	RSA	
	HBA	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) MPA (9.1728e-08) PSO (6.7956e-08) RSA (6.7956e-08) TSA (6.7956e-08) TSO (6.7956e-08)
	PSO	ABC (6.7956e-08) FSO (6.7956e-08) GWO (6.7956e-08) JS (1.4438e-04) RSA (6.7956e-08) TSA (6.7956e-08)
	ABC	RSA (6.7956e-08)
	GWO	ABC (6.7956e-08) FSO (1.0473e-06) RSA (6.7956e-08) TSA (0.0294) TSO (1.2009e-06)

5. BÖLÜM

SONUÇLAR VE ÖNERİLER

Bilgisayar destekli biyomedikal görüntü analizi, hastalıkların tanı ve tedavisinde yüksek doğrulukla etkili yaklaşımlar sunmaktadır. Metasezgisel algoritmalar en önemli bilgisayar destekli yaklaşımlardan olup küresel arama yeteneği, yüksek yakınsama hızı, dayanıklılık ve esneklik gibi avantajlara sahiptir.

Bu tez çalışmasında, öncelikle, sürü zekâsına dayalı en güncel metasezgisel algoritmalar arasında olan JS, MPA, TSA, HGS, TSO, FSO, RSA ve HBA algoritmaları retinal damar segmentasyonuna yönelik kümeleme tabanlı geliştirilmiştir. Daha sonra, DRIVE veri tabanından alınan sağlıklı ve hastalıklı görüntülerden oluşan 20 adet retinal görüntü ve STARE veri tabanından alınan sağlıklı ve hastalıklı görüntülerden oluşan 20 adet retinal görüntü üzerinde ayrı ayrı retinal damar segmentasyonu gerçekleştirilmiştir.

İlk olarak, geliştirilmiş olan metasezgisel algoritmaların matematiksel performansları Duyarlılık (SE), Özgüllük (SP), Doğruluk (ACC) ve F-skor metrikleri açısından analiz edilmiştir. Retinal görüntülerin önemli bir kısmı için bütün algoritmaların birbirlerine yakın ve yüksek değerli SE, SP, ACC ve F-skor değerleri ürettikleri gözlemlenmiştir. Bu sonuçlar, damar pikselleri ile arka plan piksellerinin algoritmalar tarafından yüksek doğrulukta ayırt edilebildiğini kanıtlamıştır. Bununla birlikte, sadece bazı görüntülerde görüntü özelliklerine bağlı olarak bazı algoritmaların SE, SP, ACC ve F-skor metrikleri açısından kötü sonuçlar ürettikleri de gözlemlenmiştir. Örneğin, DRIVE veri tabanındaki 8 numaralı görüntü ve STARE veri tabanındaki 4 numaralı görüntü için duyarlılık performanslarının zayıf kaldığı görülmüştür.

Geliştirilmiş olan algoritmalar daha sonra yakınsama hızı ve MSE performansları açısından mukayese edilmiştir. HBA algoritmasının optimum çözümlere yakınsama

hızının diğer algoritmalara göre daha yavaş kaldığı gözlemlenmiştir. MPA, HGS ve GWO algoritmalarının ise her iki veri tabanı için optimum yakınsama performansı ürettikleri görülmektedir. Algoritmaların hata performansları her iki veri tabanı için analiz edildiğinde ise bütün algoritmaların yaklaşık olarak benzer MSE değerlerine yakınsadıkları görülmektedir. Bu ise algoritmaların benzer kümeleme performansı ürettiklerini kanıtlamaktadır.

Fonksiyon Değerlendirme Sayısı (NFEs) ve algoritmalar tarafından üretilen CPU süreleri birlikte değerlendirildiğinde MPA, FSO ve ABC algoritmalarının daha uzun CPU süreleri gerektirdikleri görülmüştür. DRIVE veri tabanından alınan retinal görüntülerin segmentasyon işlemlerinin ise daha düşük CPU sürelerinde gerçekleştirilebildiği gözlemlenmiştir.

Algoritmaların istatistiksel performanslarını ortaya koyabilmek amacıyla standart sapma ve Wilcoxon rank-sum test ($p < 0,05$) tabanlı istatistiksel analizler gerçekleştirilmiştir. Algoritmalar ayrı ayrı değerlendirildiğinde MPA, HGS ve HBA algoritmalarının daha düşük standart sapma değerleri nedeniyle diğer algoritmalara kıyasla kümeleme tabanlı retina damar segmentasyonu açısından daha kararlı oldukları gözlemlenmiştir. Genel olarak değerlendirildiğinde ise tüm algoritmaların düşük standart sapma değerlerine ulaştıkları yani her bir çevrimde benzer hata değerlerine yakınsadıkları ifade edilebilir. Benzer şekilde, Wilcoxon rank-sum test ($p < 0,05$) tabanlı istatistiksel analizlerden elde edilen sonuçlar metasezgisel algoritmaların doğası gereği kararlı, dayanıklı ve esnek bir davranış sergilediğini kanıtlamaktadır. DRIVE veri tabanı için HGS ve HBA algoritmalarının, STARE veri tabanı için ise JS, MPA, HGS, HBA ve PSO algoritmalarının istatistiksel olarak diğer algoritmalardan daha anlamlı sonuçlar ürettikleri görülmüştür.

Gelecekteki araştırmalarda, öncelikle bu çalışmada kullanılan yeni algoritmalar biyomedikal görüntülerin segmentasyonunda yüksek doğrulukla optimal eşik değerleri elde etmek amacıyla geliştirilebilir. Daha sonra, her bir algoritmanın matematiksel performansını artırmak amacıyla duyarlılık, özgüllük, doğruluk ve F-skor değerlerini optimize etmeye yönelik stratejiler geliştirilebilir. Benzer şekilde, algoritmaların yakınsama hızı, MSE performansı ve CPU süresi açısından performanslarını artırmak amacıyla analizler gerçekleştirilebilir ve bu analizler sonucunda iyileştirilmiş

algoritmaların istatistiksel performansları standart sapma ve Wilcoxon rank-sum test yaklaşımları ile test edilebilir. Son olarak, retina damar segmentasyonu için seçilen bir mikroişlemci üzerinde donanım uygulamaları geliştirilebilir.

KAYNAKÇA

1. Zhang, J., Xia, Y., Xie, Y., Fulham, M., Feng, D., 2017. Classification of medical images in the biomedical literature by jointly using deep and handcrafted visual features. **IEEE Journal of Biomedical and Health Informatics, vol. 2194**(2): pp. 1-10.
2. Çetinkaya, M.B., 2010. Yapay Arı Koloni Algoritmasıyla Sayısal Süzgeç Tasarımı. Erciyes Üniversitesi, Doktora Tezi, Kayseri, s.170.
3. Chou, J.S., Truong, D.N., 2021. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. **Applied Mathematics and Computation**, **vol. 389**: 125535.
4. Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H., 2020. Marine predators algorithm: A nature-inspired metaheuristic. **Expert Systems with Applications, vol. 152**: 113377.
5. Kaur, S., Awasthi, L.K., Sangal, A.L., Dhiman, G., 2020. Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. **Engineering Applications of Artificial Intelligence, vol. 90**: 103541.
6. Yang, Y., Chen, H., Heidari, A.A., Gandomi, A.H., 2021. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. **Expert Systems with Applications, vol. 177**: 114864.
7. Xie, L., Han, T., Zhou, H., Zhang, Z.R., Han, B., Tang, A., 2021. Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization. **Computational Intelligence and Neuroscience, vol. 2021**: 9210050, pp. 22.
8. Noel, M.M., Muthiah-Nakarajan, V., Amali, G.B., Trivedi, A.S., 2021. A new biologically inspired global optimization algorithm based on firebug reproductive swarming behaviour. **Expert Systems with Applications, vol. 183**: 115408.

9. Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z.W., Gandomi, A.H., 2022. Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer. **Expert Systems with Applications, vol.191**: 116158.
10. Hashim, F.A., Houssein, E.H., Hussain, K., Mabrouk, M.S., Al-Atabany, W., 2022. Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. **Mathematics and Computers in Simulation, vol.192**: pp. 84-110.
11. Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, pp. 1942-1948. IEEE International Conference on Neural Networks, 27 November-01 December, 1995, Perth.
12. Karaboga, D., 2005. Technical Report: An idea based on honey bee swarm for numerical optimization. Computer Engineering Department, Engineering Faculty, Erciyes University, Turkey. (Web sayfası **http:// mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf**), (Erişim tarihi: Aralık 2024)
13. Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. **Advances in Engineering Software, vol.69**: pp. 46-61.
14. Alonso-Montes, C., Vilarino, D.L., Dudek, P., Penedo, M.G., 2008. Fast retinal vessel tree extraction: a pixel parallel approach. **International Journal of Circuit Keory and Applications, vol.36**(5-6): pp. 641–651.
15. Hoover, A.D., Kouznetsova, V., Goldbaum, M., 2000. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. **IEEE Transactions on Medical Imaging, vol.19**(3): pp. 203–210.
16. Akita, K., Kuga, H., 1982. A computer method of understanding ocular fundus images. **Pattern Recognition, vol.15**(6): pp. 431-443.
17. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M., 1989. Detection of blood vessels in retinal images using two-dimensional matched filters. **IEEE Transactions on Medical Imaging, vol.8**(3): pp. 263-269.
18. Nekovei, R., Ying, S., 1995. Back-propagation network and its configuration for blood vessel detection in angiograms. **IEEE Transactions on Neural Networks, vol. 6**(1): pp. 64–72.
19. Sinthanayothin, C., Boyce, J.F., Cook, H.L., Williamson, T.H., 1999. Automated localisation of the optic disc, fovea, and retinal blood vessels from digital

colour fundus images. **British Journal of Ophthalmology, vol.83**(8): pp. 902–910.

20. Niemeijer, M., Staal, J., Ginneken, B.V., Loog, M., Abràmoff, M.D., 2004. Comparative study of retinal vessel segmentation methods on a new publicly available database. **SPIE Medical Imaging, vol.5370**: pp. 648–656.
21. Gregory, S., Trevor, D., Piotr, I., 2006. Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing), MIT Press, pp. 280.
22. Staal, J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B., 2004. Ridge-based vessel segmentation in color images of the retina. **IEEE Transactions on Medical Imaging, vol.23**(4): pp. 501-509.
23. Soares, J.V.B., Leandro, J.J.G., Cesar, R.M., Jelinek, H.F., Cree, M.J., 2006. Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. **IEEE Transactions on Medical Imaging, vol. 25**(9): pp. 1214-1222.
24. Ricci, E., Perfetti, R., 2007. Retinal blood vessel segmentation using line operators and support vector classification**. IEEE Transactions on Medical Imaging, vol.26**(10): pp. 1357-1365.
25. Osareh, A., Shadgar, B., 2009. Automatic blood vessel segmentation in color images of retina. **Iranian Journal Of Science And Technology Transaction B – Engineering, vol.33**(B2): pp. 191–206.
26. Xu, L., Luo, S., 2010. A novel method for blood vessel detection from retinal images. **BioMedical Engineering Online, vol.9**(1): pp. 14.
27. Lupascu, C.A., Tegolo, D., Trucco, E., 2010. FABC: Retinal vessel segmentation using AdaBoost. **IEEE Transactions on Information Technology in Biomedicine, vol.14**(5): pp. 1267-1274.
28. You, X., Peng, Q., Yuan, Y., Cheung, Y., Lei, J., 2011. Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. **Pattern Recognition, vol.44**(10-11): pp. 2314-2324.
29. Marín, D., Aquino, A., Gegundez-Arias, M.E., Bravo, J.M., 2011. A new supervised method for blood vessel segmentation in retinal images by using

gray-level and moment invariants-based features. **IEEE Transactions on Medical Imaging, vol.30**(1): pp. 146-158.

30. Tolias, Y.A., Panas, S.M., 1998. A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. **IEEE Transactions on Medical Imaging, vol.17**(2): pp. 263-273.
31. Simó, A., de Ves, E., 2001. Segmentation of macular fluorescein angiographies. A statistical approach. **Pattern Recognition, vol.34**(4): pp. 795–809.
32. Salem, S., Salem, N., Nandi, A., 2007. Segmentation of retinal blood vessels using a novel clustering algorithm (RACAL) with a partial supervision strategy. **Medical and Biological Engineering and Computing, vol.45**(3): pp. 261–273.
33. Kande, G.B., Subbaiah, P.V., Savithri, T.S., 2009. Unsupervised fuzzy based vessel segmentation in pathological digital fundus images. **Journal of Medical Systems, vol.34**(5): pp. 849–858.
34. Ng, J., Clay, S.T., Barman, S.A., Fielder, A.R., Moseley, M.J., Parker, K.H., Paterson, C., 2010. Maximum likelihood estimation of vessel parameters from scale space analysis. **Image and Vision Computing, vol.28**(1): pp. 55–63.
35. Lupascu, C.A., Tegolo, D., 2011. Automatic unsupervised segmentation of retinal vessels using self-organizing maps and k-means clustering. pp. 263–274, *In:* Computational Intelligence Methods for Bioinformatics and Biostatistics, Lecture Notes in Computer Science, vol.(6685): (Eds: R. Rizzo, P.J.G. Lisboa). Springer, Berlin.
36. Lupascu, C.A., Tegolo, D., 2011. Stable automatic unsupervised segmentation of retinal vessels using self-organizing maps and a modified fuzzy c-means clustering, pp. 244–252, *In*: Fuzzy Logic and Applications, Lecture Notes in Computer Science, vol.(6857): (Eds: A. M. Fanelli, W. Pedrycz, A. Petrosino). Springer, Berlin.
37. Saffarzadeh, V.M., Osareh, A., Shadgar, B., 2014. Vessel segmentation in retinal images using multi-scale line operator and k-means clustering. **Journal of Medical Signals and Sensors, vol.4**(2): pp. 122-129.
38. Palani, D., Venkatalakshmi, K., Jabeen A.R., Ram, V.M.A.B., 2019. Effective detection of diabetic retinopathy from human retinal fundus images using

modified FCM and IWPSO, pp. 1-5. *IEEE International Conference on System, Computation, Automation and Networking (ICSCAN),* 2019, Pondicherry, India. doi: 10.1109/ICSCAN.2019.8878786.

39. Mircea, W., 2019. Segmentation and machine learning techniques applied to automatic detection of diseases from eye-fundus retina images, pp. 1-6. *6th International Symposium on Electrical and Electronics Engineering (ISEEE),* Galati, Romania. doi: 10.1109/ISEEE48094.2019.9136125.
40. Keerthiveena, B., Veerakumar, T., Esakkirajan S., Subudhi, B.N., 2019. Computer-aided diagnosis for diabetic retinopathy based on firefly algorithm, pp. 310-315. *11th International Conference on Advanced Computing (ICoAC),* Chennai, India. doi: 10.1109/ICoAC48765.2019.246858.
41. Devarajan, D., Ramesh, S.M., Gomathy, B.A., 2020. Metaheuristic segmentation framework for detection of retinal disorders from fundus images using a hybrid ant colony optimization. **Soft Computing, vol.24**(17): pp. 13347–13356.
42. Rajinikanth, V., Lin, H., Panneerselvam, J., Sri Madhava Raja, N., 2020. Examination of retinal anatomical structures—A study with spider monkey optimization algorithm, pp. 177–197. *In*: Dey, N., Ashour, A., Bhattacharyya, S. (eds) Applied Nature-Inspired Computing: Algorithms and Case Studies. Springer Tracts in Nature-Inspired Computing. Springer, Singapore. https://doi.org/10.1007/978-981-13-9263-4_8.
43. Kaur, S., Mann, K. S., 2020. Retinal vessel segmentation using an entropy-based optimization algorithm. **International Journal of Healthcare Information Systems and Informatics (IJHISI), vol.15**(2): pp. 61-79. http://doi.org/10.4018/IJHISI.2020040105.
44. Pruthi, J., Arora, S., Khanna, K., 2020. Segmentation of blood vessels from retinal fundus images using bird swarm algorithm and river formation dynamics algorithm, pp. 995-1007. *In*: Singh Tomar, G., Chaudhari, N.S., Barbosa, J.L.V., Aghwariya, M.K. (eds) International Conference on Intelligent Computing and Smart Communication. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-0633-8_101.

45. Sathananthavathi, V., Indumathi, G., 2021. Particle swarm optimization based retinal image enhancement. **Wireless Personal Communications vol.(121)**, pp. 543–555. https://doi.org/10.1007/s11277-021-08649-z.
46. Irshad, S., Yin, X., Zhang, Y., 2021. A new approach for retinal vessel differentiation using binary particle swarm optimization. **Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, vol.9(**5): pp. 510-522. DOI: 10.1080/21681163.2020.1870001.
47. Sandhya, S.G., Suhasini, A., 2021. An efficient detection of micro aneurysms from fundus images with CDLNN algorithm. Materials Today: Proceedings. https://doi.org/10.1016/j.matpr.2021.04.010.
48. Gharaibeh, N., Al-hazaimeh, O.M., Abu-Ein, A., Nahar, K.M.O., 2021. A hybrid svm naïve-bayes classifier for bright lesions recognition in eye fundus ımages. **International Journal on Electrical Engineering and Informatics, vol.13**(3): pp. 530-545.
49. Ramos-Soto, O., Rodríguez-Esparza, E., Balderas-Mata, S.E., Oliva, D., Hassanien, A.E., Meleppat, R.K., Zawadzki, R.J., 2021. An efficient retinal blood vessel segmentation in eye fundus images by using optimized top-hat and homomorphic filtering, **Computer Methods and Programs in Biomedicine, vol.201**(1): pp. 105949.
50. Kadry, S., Rajinikanth, V., Damaševičius, R., Taniar, D., 2021. Retinal vessel segmentation with slime-mould-optimization based multi-scale-matched-filter, pp. 1-5. *In*: Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII), Chennai, India. doi:10.1109/ICBSII51839.2021.9445135.
51. Zhou, T.H., Liu, Y.M., Xie, W.C., Li, H.N., Wang, L., 2021. Automatic identification and classification method for diabetic retinopathy FFA image processing, pp. 204-210. *In*: Pan, JS., Li, J., Namsrai, OE., Meng, Z., Savić, M. (eds) Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, Springer, Singapore. https://doi.org/10.1007/978-981-33-6420-2_25.
52. Sayed, G.I., Hassanien, A.E., 2022. A novel chaotic artificial gorilla troops optimizer and its application for fundus images segmentation, pp.318-329.

In: Hassanien, A.E., Snášel, V., Chang, KC., Darwish, A., Gaber, T. (eds) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2021. AISI 2021. Lecture Notes on Data Engineering and Communications Technologies, Springer, Cham. https://doi.org/10.1007/978-3-030-89701-7_28.

53. Singh, L.K., Khanna, M., Thawkar, S., Singh, R., 2022. Collaboration of features optimization techniques for the effective diagnosis of glaucoma in retinal fundus images. **Advances in Engineering Software, vol.173**(C): pp. 103283.
54. Bhimavarapu, U., Battineni, G., 2022. Automatic microaneurysms detection for early diagnosis of diabetic retinopathy using improved discrete particle swarm optimization. **Journal of Personalized Medicine, vol.12**(2): pp. 317. doi: 10.3390/jpm12020317.
55. Deng, L., Liu, S., Wang, X., Zhao, G., Xu, J., 2022. Particle swarm optimization and salp swarm algorithm for the segmentation of diabetic retinal blood vessel images. **Computational Intelligence and Neuroscience, vol.2022**, Article ID 1936482, 14 pages. https://doi.org/10.1155/2022/1936482
56. Parthiban, K., Kamarasan, M., 2023. Diabetic retinopathy detection and grading of retinal fundus images using coyote optimization algorithm with deep learning. **Medical Applications of Multimedia, vol.82**: pp. 18947–18966. https://doi.org/10.1007/s11042-022-14234-8
57. Fauzi, A., Lubis, L.E., 2023. Optimization of retinal blood vessel segmentation based on Gabor filters and particle swarm optimization. **Indonesian Journal of Electrical Engineering and Computer Science, vol.29**(3): pp. 1590-1596.
58. Kang H.R., 1996. Color Technology for Electronic Imaging Devices. Xerox Corporation, SPIE Optical Engineering Press, Washington, pp. 369.
59. You, X., Peng, Q., Yuan, Y., Cheung, Y.-m., Lei, J., 2011. Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. **Pattern Recognition, vol.44**(10): pp. 2314–2324.
60. Roychowdhury, S., Koozekanani, D., Parhi, K., 2014. Blood vessel segmentation of fundus images by major vessel extraction and sub-image classification.

IEEE Journal of Biomedical and Health Informatics, vol.19(3): pp. 1118-1128.

61. Wang, S., Yin, Y., Cao, G., Wei, B., Zheng, Y., Yang, G., 2015. Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. **Neurocomputing, vol. 149**(B): pp. 708–717.
62. Mendonca, A.M., Campilho, A., 2006. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. **IEEE Transactions on Medical Imaging, vol.25**(9): pp. 1200–1213.
63. Azzopardi, G., Strisciuglio, N., Vento, M., Petkov, N., 2015. Trainable COSFIRE filters for vessel delineation with application to retinal images, **Medical Image Analysis, vol.19**(1): pp. 46–57.
64. Fraz, M.M., Remagnino, P., Hoppe, A., 2012. An ensemble classification-based approach applied to retinal blood vessel segmentation, **IEEE Transactions on Biomedical Engineering, vol.59**(9): pp. 2538–2548.
65. Odstrcilik, J., Kolar, R., Budai, A., 2013. Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database, **IET Image Processing, vol.7**(4): pp. 373–383.
66. Marin, D., Aquino, A., Gegundez-Arias, M.E., Bravo, J.M., 2011. A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariantsbased features, **IEEE Transactions on Medical Imaging, vol.30**(1): pp. 146–158.
67. Kaba, D., Wang, C., Li, Y., Salazar-Gonzalez, A., Liu, X., Serag, A., 2014. Retinal blood vessels extraction using probabilistic modelling, **Health Information Science and Systems, vol.2**(1): pp. 2.
68. Dash, J., Bhoi, N., 2017. A thresholding based technique to extract retinal blood vessels from fundus images, **Future Computing and Informatics Journal, vol.2**(2): pp. 103–109.
69. Arguello, F., Vilarino, D.L., Heras, D.B., Nieto, A.,2014. GPU based segmentation of retinal blood vessels, **J. Real Time Image Process, vol.14**(4): pp. 1–10.
70. Asad, A.H., Azar, A.T., Hassaanien, A.E., 2012. Ant colonybased system for retinal blood vessels segmentation, **Advances in Intelligent Systems and Computing, vol.201**: pp. 441–452.

71. Zhao, Y.Q., Hong Wang, X., Fang Wang, X., Shih, F.Y., 2014. Retinal vessels segmentation based on level set and region growing, **Pattern Recognition, vol.47**(7): pp. 2437–2446.

72. Zhang, B., Zhang, L., Karray, F., 2010. Retinal vessel extraction by matched filter with first-order derivative of Gaussian, **Computers in Biology and Medicine, vol.40**(4): pp. 438–445.

73. Imani, E., Javidi, M., Pourreza, H.R., 2015. Improvement of retinal blood vessel detection using morphological component analysis, **Computer Methods and Programs in Biomedicine, vol.118**(3): pp. 263–279.

74. Mendonca, A., Dashtbozorg, B., Campilho, A., 2014. Segmentation of the vascular network of the retina, pp. 85-109. *In*: Image Analysis and Modeling in Ophthalmology (Eds: Ng, E.Y.K., Acharya, U.R., Campilho, A., Suri, J.S), CRC Press, Florida.

75. Xiao, Z., Adel, M., Bourennane, S., 2013. Bayesian method with spatial constraint for retinal vessel segmentation, **Computational and Mathematical Methods in Medicine, vol. 2013**: 401413.

76. Yin, B., Li, H., Sheng B., 2015. Vessel extraction from nonfluorescein fundus images using orientation-aware detector, **Medical Image Analysis, vol.26**(1): pp. 232–242.

77. Lazar, I., Hajdu, A., 2015. Segmentation of retinal vessels by means of directional response vector similarity and region growing, **Computers in Biology and Medicine, vol.66**(1): pp. 209–221.

78. Strisciuglio, N., Azzopardi, G., Vento, M., Petkov, N., 2015. Multiscale blood vessel delineation using b-cosfire filters, **Computer Analysis of Images and Patterns, vol.9257**: pp. 300–312.

79. Arcuri, A., Briand, L., 2012. A Hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering, **Software Testing, Verification and Reliability, vol.24**(3): pp. 219–250.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Hakan DURAN
Uyruğu: Türkiye (T.C.)
Doğum Tarihi ve Yeri:
Medeni Durum:
e-mail:
Yazışma Adresi:

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Fen Bilimleri Enstitüsü	2018
Lisans	Erciyes Üniversitesi, Mekatronik Mühendisliği	2013
Lise	Sürmene Hasan Sadri Yetmişbir Anadolu Lisesi, Trabzon	2009

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2018-Halen	Devlet Hava Meydanları İşletmesi	7 Yıl
2016-2018	Erciyes Üniversitesi	3 Yıl

YABANCI DİL

İngilizce

YAYINLAR

1. Çetinkaya, M.B., Duran, H. (2023). Retinal vessel segmentation based on hunger games search and reptile search algorithms. **WSEAS Transactıons on Signal Processıng, vol.19**, pp.221-228.

2. Çetinkaya, M.B., Duran, H. (2022). Performance comparison of most recently proposed evolutionary, swarm intelligence, and physics-based metaheuristic algorithms for retinal vessel segmentation. **Mathematical Problems in Engineering, vol.2022.**

3. Çetinkaya, M.B., Duran, H. (2021). A detailed and comparative work for retinal vessel segmentation based on the most effective heuristic approaches, **Biomedıcal Engineering-Biomedizinische Technik, vol.66(2)** , pp.181-200.

4. Cihan, M.C., Çetinkaya, M.B., Duran, H. (2021). Performance comparison of artificial bee colony algorithm based approaches for retinal vessel segmentation, **Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi , vol.23**(2), pp.792-807.

5. Çetinkaya, M.B., Duran, H. (2017). Kümeleme tabanlı diferansiyel gelişim algoritması kullanarak retinal kılcal damarların bölütlenmesi, *Tıp Tekno 2017 Tıp Teknolojileri Kongresi*, Trabzon, Turkey, pp.109-112.

6. Çetinkaya, M.B., et al. (2016). Fundus floresan anjiografi görüntülerinde diyabetik retinopati nedeniyle oluşan kanama alanlarının belirlenmesi, *20. Biyomedikal Mühendisliği Ulusal Toplantısı (BİYOMUT 2016)*, İzmir, Turkey, pp.14.