



T.C.  
NECMETTİN ERBAKAN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



**Çok Seviyeli Görüntü Eşikleme  
İçin Web Tabanlı Bir Aracın  
Geliştirilmesi**

Alper ÜNLÜ

YÜKSEK LİSANS TEZİ

Mekatronik Mühendisliği Anabilim Dalı

Haziran-2024  
KONYA  
Her Hakkı Saklıdır

**TEZ KABUL VE ONAYI**

Alper ÜNLÜ tarafından hazırlanan “Çok Seviyeli Görüntü Eşikleme İçin Web Tabanlı Bir Aracın Geliştirilmesi” adlı tez çalışması 27/06/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

<b>Jüri Üyeleri</b>	<b>İmza</b>
<b>Danışman</b> Prof. Dr. İlhan İLHAN	.....
<b>Üye</b> Prof. Dr. Hasan Erdiñç KOÇER	.....
<b>Üye</b> Dr. Öğr. Üyesi Murat KARAKOYUN	.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun ..... gün ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Havvanur UÇBEYİAY  
FBE Müdürü

## TEZ BİLDİRİMİ

Bu projedeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

## DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Alper ÜNLÜ

Tarih:.....

**ÖZET****YÜKSEK LİSANS TEZİ****ÇOK SEVİYELİ GÖRÜNTÜ EŞİKLEME İÇİN WEB TABANLI BİR ARACIN  
GELİŞTİRİLMESİ****Alper ÜNLÜ****Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü  
Mekatronik Mühendisliği Anabilim Dalı****Danışman: Prof. Dr. İlhan İLHAN****2024, 82 Sayfa****Jüri****Prof. Dr. İlhan İLHAN****Prof. Dr. Hasan Erdiñç KOÇER****Dr. Öğr. Üyesi Murat KARAKOYUN**

Çok seviyeli görüntü eşikleme, görüntüyü ileri düzeyde anlamlı özelliklere ayırmak için kullanılan önemli bir görüntü işleme tekniğidir. Bu çalışmada, çok seviyeli görüntü eşikleme için web tabanlı bir görüntü işleme aracı geliştirilmiştir. Bu araç yeni önerilen GWO-HS algoritmasının yanında dokuz farklı optimizasyon algoritmasını da bünyesinde barındırmaktadır. Kullanıcılar bu algoritmalarından herhangi birini kullanarak istedikleri görüntü ve eşik seviyesi için sonuçlar elde edebilmektedirler. Bu sonuçları karşılaştırabilmekte ve çıktılarını alabilmektedirler.

Önerilen GWO-HS algoritması gri kurt optimizasyon (GWO) ve harmoni arama (HS) algoritmaları hibritlenerek elde edilmiştir. Bu algoritmanın performansı dokuz diğer algoritmanın performansları ile karşılaştırılmıştır. Karşılaştırmalarda Otsu, Kapur ve Tsallis entropi tabanlı eşikleme yöntemleri kullanılmıştır. Deneylerde görüntü işleme çalışmalarında iyi bilinen ve yaygın olarak kullanılan altı görüntü tercih edilmiştir. Her bir görüntü üzerinde 2'den 10'a kadar değişen seviyeler için eşikleme işlemi uygulanmıştır. Sonuçlar, önerilen GWO-HS algoritmasının, diğer algoritmalara kıyasla özellikle yüksek eşik seviyeleri için daha üstün bir performansa sahip olduğunu göstermiştir.

**Anahtar Kelimeler:** Çok Seviyeli Görüntü Eşikleme, Kapur, Otsu, Tsallis, Gri Kurt Optimizasyon, Harmoni Arama

**ABSTRACT****MS THESIS****DEVELOPMENT OF A WEB-BASED TOOL FOR MULTI-LEVEL IMAGE THRESHOLDING****Alper ÜNLÜ****THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF  
NECMETTİN ERBAKAN UNIVERSITY  
THE DEGREE OF MASTER OF SCIENCE IN MECHATRONICS ENGINEERING****Advisor: Prof. Dr. İlhan İLHAN****2024, 82 Pages****Jury****Prof. Dr. İlhan İLHAN****Prof. Dr. Hasan Erdinç KOÇER****Asst. Prof. Dr. Murat KARAKOYUN**

Multilevel image thresholding is an important image processing technique used to segment an image into meaningful features at an advanced level. In this study, a web-based image processing tool has been developed for multilevel image thresholding. This tool includes the newly proposed GWO-HS algorithm alongside nine different optimization algorithms. Users can utilize any of these algorithms to obtain results for their desired image and threshold levels. They can compare these results and export their outputs.

The proposed GWO-HS algorithm is a hybridization of Grey Wolf Optimization (GWO) and Harmony Search (HS) algorithms. The performance of this algorithm has been compared with the performances of nine other algorithms. Otsu, Kapur, and Tsallis entropy-based thresholding methods were used for the comparisons. In the experiments, six well-known and widely used images in image processing were preferred. Thresholding was applied to each image for levels ranging from 2 to 10. The results demonstrated that the proposed GWO-HS algorithm has superior performance compared to other algorithms, especially for higher threshold levels.

**Keywords:** Multi-Level Image Thresholding, Kapur, Otsu, Tsallis, Gray Wolf Optimization, Harmony Search

## ÖNSÖZ

Bu çalışmanın gerçekleşmesine büyük katkıları olan, benimle bu alandaki bilgi ve birikimlerini her zaman paylaşan ve desteğini benden esirgemeyen danışmanım Prof. Dr. İlhan İLHAN'a teşekkürlerimi bir borç bilirim. Aynı zamanda benden kıymetli yardımlarını esirgemeyen ailem ve arkadaşlarıma teşekkür ederim

Alper ÜNLÜ  
KONYA-2024



# İÇİNDEKİLER

<b>ÖZET .....</b>	<b>iii</b>
<b>ÖNSÖZ.....</b>	<b>v</b>
<b>İÇİNDEKİLER .....</b>	<b>vi</b>
<b>ŞEKİLLER DİZİNİ .....</b>	<b>viii</b>
<b>SİMGELER VE KISALTMALAR .....</b>	<b>ix</b>
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KAYNAK ARAŞTIRMASI .....</b>	<b>4</b>
<b>3. GÖRÜNTÜ EŞİKLEME.....</b>	<b>7</b>
3.1. Tek Seviyeli Görüntü Eşikleme.....	8
3.2. Çok Seviyeli Görüntü Eşikleme.....	8
3.2.1. Kapur entropi tabanlı eşikleme yöntemi.....	9
3.2.2. Otsu eşikleme yöntemi.....	10
3.2.3. Tsallis entropi tabanlı eşikleme yöntemi .....	11
<b>4. MATERYAL VE YÖNTEM.....</b>	<b>14</b>
4.1. Web Tabanlı Görüntü Eşikleme Aracı.....	14
4.1.1. Web arayüzün geliştirilmesi .....	14
4.1.2. Docker container ile web mimari geliştirilmesi.....	16
4.1.3. API kullanımı.....	16
4.2. Optimizasyon Algoritmaları .....	17
4.2.1. Ateş Böceği Algoritması.....	18
4.2.2. Benzetilmiş Tavlama Algoritması .....	19
4.2.3. Diferansiyel Gelişim Algoritması.....	21
4.2.4. Genetik Algoritma .....	23

4.2.5. Gri Kurt Optimizasyon Algoritması .....	25
4.2.6. Harmoni Arama Algoritması .....	26
4.2.7. Klonal Seçim Algoritması.....	29
4.2.8. Parçacık Sürü Optimizasyon Algoritması.....	31
4.2.9. Yapay Arı Kolonisi Algoritması.....	32
4.3. Önerilen Hibrit Optimizasyon Algoritması .....	35
4.4 Performans Karşılaştırma Metrikleri .....	37
4.4.1. Amaç Fonksiyon Değeri .....	37
4.4.2. PSNR Değeri.....	37
4.4.3. SSIM Değeri .....	38
4.4.4. Standart Sapma ve Ortalama Değer .....	38
<b>5. ARAŞTIRMA BULGULARI VE TARTIŞMA.....</b>	<b>39</b>
5.1. Web Tabanlı Aracın Kullanımı.....	39
5.2. Optimizasyon Algoritmalarının Performanslarının Değerlendirilmesi .....	43
<b>6. SONUÇLAR VE ÖNERİLER.....</b>	<b>66</b>
6.1. Sonuçlar .....	66
6.2. Öneriler .....	66
<b>KAYNAKLAR .....</b>	<b>67</b>

## ŞEKİLLER DİZİNİ

Şekil 1. (a) Tek eşik ile bölütlenecek görüntü histogramı (b) İki eşik ile bölütlenecek görüntü histogramı .....	7
Şekil 2. (a) Madeni paralar (b) Histogram (c) Eşiklenmiş sonuç görüntüsü .....	8
Şekil 3. Web tabanlı çok seviyeli görüntü eşikleme aracının arayüzü .....	15
Şekil 4. Docker mimarisi .....	16
Şekil 5. API kullanan bir client-server mimarisinin basitleştirilmiş diyagramı .....	17
Şekil 6. FA'nın sözde kodu .....	19
Şekil 7. SA'nın sözde kodu .....	20
Şekil 8. DE'nin sözde kodu .....	22
Şekil 9. GA'nın sözde kodu .....	24
Şekil 10. GWO'nun sözde kodu .....	26
Şekil 11. HS'nin sözde kodu .....	28
Şekil 12. CLONALG'nun sözde kodu .....	30
Şekil 13. PSO'nun sözde kodu .....	32
Şekil 14. ABC'nin sözde kodu .....	34
Şekil 15. GWO-HS algoritmasının sözde kodu .....	36
Şekil 16. Geliştirilen arayüzün işlem basamakları (a, b, c, d, e, f) .....	41
Şekil 17. Tüm eşikleme algoritmaları ile yapılan bir dizi deney .....	42
Şekil 18. (a) Barbara, (b) Living room, (c) Boats, (d) Goldhill, (e) Lake, ve (f) Aerial. Görüntüleri ve histogramaları .....	44
Şekil 19. GWO-HS algoritmasının Kapur eşikleme yöntemi için yakınsama eğrileri .....	63
Şekil 20. GWO-HS algoritmasının Otsu eşikleme yöntemi için yakınsama eğrileri .....	64
Şekil 21. GWO-HS algoritmasının Tsallis eşikleme yöntemi için yakınsama eğrileri .....	65

**SİMGELER VE KISALTMALAR**

ABC	Yapay Arı Koloni Algoritması
CLONALG	Klonal Seçim Algoritması
DE	Diferansiyel Gelişim Algoritması
GA	Genetik Algoritma
GWO	Gri Kurt Optimizasyon Algoritması
GWO-HS	Geliştirilen Hibrit Algoritma
FA	Ateş Böceği Algoritması
HS	Harmoni Arama Algoritması
PSO	Parçacık Sürü Optimizasyon Algoritması
SA	Benzetilmiş Tavlama Algoritması
YZ	Yapay Zekâ
API	Uygulama Programlama Arayüzleri
PSNR	Tepe Sinyal-Gürültü Oranı
SSIM	Yapısal Benzerlik İndeksi

## 1. GİRİŞ

Görüntü işleme, dijital görüntüleri iyileştirmeyi, analiz etmeyi ve yorumlamayı amaçlayan çeşitli teknikleri ve metodolojileri kapsayan çok disiplinli bir alandır. Bilgisayarlı görme, desen tanıma, uzaktan algılama, tıbbi görüntüleme ve bilimsel görselleştirme gibi birçok alanda geniş uygulamalar bulmuştur (Gonzalez ve ark., 2002). Görüntü işleme alanı, hesaplama gücündeki ilerlemeler, algoritma geliştirme ve otomatik görüntü analizi ve anlama talebinin artmasıyla önemli ölçüde gelişmiştir.

Görüntü işleme, anlamlı bilgilerin çıkarılması veya belirli özelliklerin iyileştirilmesi amacıyla dijital görüntülerin manipüle edilmesini ve dönüştürülmesini içerir. Bu süreç genellikle görüntü edinme, ön işleme, bölütleme, öznitelik çıkarma ve nesne tanıma veya sınıflandırma gibi birkaç aşamayı içerir. Her aşama, uygulama alanının belirli gereksinimlerine uygun olarak çeşitli algoritmalar ve teknikler kullanır.

Görüntü işleme sürecinin en önemli aşamalarından biri de segmentasyon (bölütleme) aşamasıdır. Segmentasyon, görüntüyü benzer özelliklere veya özniteliklere sahip anlamlı alanlara ayırmak için kullanılır. Buradaki amaç görüntüyü daha basit parçalara ayırmaktır. Bu işlem genellikle görüntü işleme çalışmalarının ilk aşamalarında kullanılır. Segmentasyon tekniklerini genel olarak üç kısma ayırmak mümkündür. Bunlar yapısal, olasılıksal ve hibrit segmentasyon teknikleri olarak ayrılabilir. En çok kullanılan segmentasyon yöntemleri ise, eşikleme yöntemi, köşe tabanlı metot, alan tabanlı metot, kümeleme tabanlı metot, havza tabanlı metot, kısmi diferansiyel denklem tabanlı metot ve yapay sinir ağları tabanlı metot şeklindedir.

Görüntü segmentasyon yöntemlerinden en fazla kullanılan tekniklerden biri, piksel yoğunluk değerlerine dayanarak bir görüntüyü farklı bölgelere ayırma işlemi olan görüntü eşiklemedir. Bu işlem, nesne tespiti, arka plan çıkarma ve öznitelik çıkarma gibi çeşitli görevler için hayati öneme sahiptir. Geleneksel görüntü eşikleme yöntemleri, pikselleri ön plan veya arka plan olarak sınıflandırmak için tek bir eşik değeri kullanır. Ancak, değişken aydınlatma koşullarına veya birden fazla ilgi alanına sahip daha karmaşık görüntüler için, çok seviyeli eşikleme teknikleri vazgeçilmez hale gelir. Bu yöntemler, birden fazla eşik değerinin seçilmesini içerir ve böylece görüntüyü birden çok yoğunluk aralığına veya sınıfa böler. Böylelikle, çok seviyeli eşikleme,

görüntü içindeki karmaşık detayları ve nüansları daha doğru ve kapsamlı bir şekilde analiz etmeyi sağlayabilir.

Eşik değerlerinin optimal olarak seçilmesi, görüntü eşiklemede kritik bir adımdır ve aktif bir araştırma alanı olmuştur. Yapay zekâ (YZ) optimizasyon algoritmaları, özellikle evrimsel hesaplama ve sürü zekasına dayalı olanlar, bu zorluğun üstesinden gelmek için güçlü araçlar olarak ortaya çıkmıştır. Genetik, parçacık sürü optimizasyon ve gri kurt optimizasyonu gibi algoritmalar, biyolojiden ilham alan stratejiler kullanarak geniş bir çözüm alanında optimal çözümleri ararlar. Birden fazla ajanın veya bireyin kolektif zekasını kullanarak, bu algoritmalar etkili bir şekilde parametre uzayını keşfedebilir ve belirli amaç fonksiyonlarını veya performans metriklerini maksimize eden eşik değerlerine yaklaşabilir.

YZ optimizasyon algoritmalarının görüntü eşiklemeyle entegrasyonu, segmentasyon süreçlerinin doğruluğunu ve dayanıklılığını artırarak umut verici sonuçlar vermiştir. Bu teknikler, tıbbi görüntüleme alanında hastalık teşhisi ve tedavi planlaması, uzaktan algılama alanında arazi örtüsü sınıflandırması ve çevresel izleme, endüstriyel denetimde kalite kontrol ve kusur tespiti gibi çeşitli alanlarda uygulama bulmuştur.

Yazılım geliştirme alanında, Web API'ler (Uygulama Programlama Arayüzleri), farklı sistemler ve uygulamalar arasındaki entegrasyonu ve iletişimi kolaylaştırmada belirleyici bir rol oynamıştır. Web API'ler, internet üzerinden veri veya işlevsellik erişimini ve manipülasyonunu sağlayan standartlaştırılmış ve iyi tanımlanmış bir arayüz sunar. Geliştiricilere mevcut hizmetlerden ve kaynaklardan yararlanma imkânı tanır ve uyumluluğu teşvik eder, daha karmaşık ve sofistike uygulamaların geliştirilmesine katkı sağlar.

Web API'ler, uzaktan görüntü verilerine erişim ve işleme için arabirimler sağlayarak görüntü işleme alanında önemli bir rol oynayabilir. Örneğin, bulut tabanlı görüntü işleme hizmetleri, geliştiricilere görüntüler yüklemelerine, çeşitli işleme teknikleri (örneğin, eşikleme, filtreleme veya nesne tespiti) uygulamalarına ve işlenmiş sonuçları alabilmelerine olanak tanıyan Web API'ler sunabilir. Bu yaklaşım, karmaşık görüntü işleme yazılımlarının yerel kurulum ve bakım ihtiyacını ortadan kaldırarak, verimli ve ölçeklenebilir çözümler sunar.

Ayrıca, Web API'ler, görüntü işleme yeteneklerini web uygulamaları, mobil uygulamalar veya diğer yazılım sistemlerine entegre etmek için kullanılabilir ve böylece görsel veri analizi ve

manipülasyon işlevlerinin sorunsuz bir şekilde entegrasyonunu sağlar. Görüntü işleme algoritmaları ve tekniklerinin temel karmaşıklıklarını soyutlayarak, Web API'leri geliştirme sürecini basitleştirir ve kodun yeniden kullanılabilirliğini teşvik eder, sonuç olarak yenilikçi ve görsel olarak yönlendirilen uygulamaların teslimatını hızlandırır.

Bu tezin geri kalan kısmı şu şekilde organize edilmiştir: Bölüm 2' de literatürde çok seviyeli eşikleme ve görüntü işleme alanlarında geliştirilmiş arayüzler tanıtılmıştır. Bölüm 3'te görüntü eşikleme için yaygın kullanılan algoritmalarından Kapur, Otsu ve Tsallis algoritmalarının tek ve çok seviyeli eşikleme yöntemleri anlatılmıştır. Bölüm 4'te geliştirilen web tabanlı araç anlatılmış ve araçta kullanılan 9 farklı optimizasyon algoritması tanıtılmıştır. Bu algoritmalara ek olarak geliştirdiğimiz GWO-HS algoritmasından bahsedilmiştir. Bölüm 5'te geliştirilen web tabanlı arayüzün kullanımı geniş bir şekilde anlatılmıştır. Kullanılan tüm optimizasyon algoritmalarının test görselleri üzerinde yapılan deneylerin sonuçları verilmiştir. Geliştirdiğimiz GWO-HS algoritmasının diğer algoritmalar ile sonuçları kıyaslanmıştır. Son olarak Bölüm 6'da sonuçlar ve öneriler paylaşılmıştır.

## 2. KAYNAK ARAŞTIRMASI

Görüntü eşikleme, pikselleri yoğunluk değerlerine göre kategorize ederek bir görüntüyü belirgin bölgelere bölen görüntü işlemede temel bir tekniktir. Temel eşikleme yöntemleri tek bir eşik değeri kullanırken, çok seviyeli eşikleme çok sayıda eşik değeri kullanarak görüntüyü birden çok yoğunluk sınıfına böler. Bu yaklaşım, değişen aydınlatma, gürültü seviyeleri ve birden fazla ilgi alanına sahip karmaşık görüntüler için özellikle kullanışlıdır (Sezgin ve Sankur, 2004).

En eski ve en yaygın olarak kullanılan çok seviyeli eşikleme yöntemlerinden biri, Nobuyuki Otsu tarafından 1979 yılında önerilen Otsu'nun algoritmasıdır (Otsu, 1996). Otsu'nun yöntemi, eşiklenmiş görüntünün sınıflar arası varyansı maksimize etme veya sınıflar içi varyansı minimize etme prensibine dayanır. Bu, iki seviyeli eşikleme için etkili ve verimli bir tekniktir ve aynı zamanda çok seviyeli eşiklemeye de genişletilmiştir.

Bir diğer popüler yaklaşım ise 1985 yılında J. N. Kapur ve arkadaşları tarafından tanıtılan entropi tabanlı Kapur yöntemi (Kapur ve ark., 1985). Bu yöntem, ön plan ve arka plan bölgelerinin entropisini maksimize ederek optimum eşikleri seçer ve eşiklenmiş görüntünün bilgi potansiyelini etkin bir şekilde maksimize eder.

Constantino Tsallis tarafından 1988 yılında önerilen Tsallis entropi tabanlı eşikleme yöntemi (Tsallis, 1988), Shannon entropisinin bir uzantısıdır ve çok seviyeli eşikleme için kullanılmıştır. Bu yöntem, entropinin eklemeli olmayan doğasını dikkate alır ve belirli senaryolarda geleneksel Shannon entropi tabanlı yaklaşımlardan daha iyi performans gösterdiği gösterilmiştir (Portes de Albuquerque ve ark., 2004).

Tek seviyeli eşiklemede kapsamlı arama ile sonuç hızlı bir şekilde bulunabilmektedir. Çok seviyeli eşikleme için kapsamlı arama, incelenmesi gereken eşik sayısı istenen eşik değerlerinin sayısı ile üstel bir şekilde arttığı için hesaplama maliyeti oldukça yüksektir. Barbara test görselinin iki eşik değeri ile hesaplama süresi 25 ms olarak hesaplanmış, eşik sayısı beş olduğunda ise bu süre 11 dakika olarak hesaplanmıştır (Brajevic ve Tuba, 2014). Metasezgisel algoritmalar aracılığıyla bu hesaplama süreleri düşürülmeye çalışılmıştır.

Sathya ve arkadaşları, çok seviyeli eşikleme için entropi tabanlı eşikleme yöntemi olan Tsallis algoritmasını bakteriyel arama (BF) algoritması ile kullanmışlardır (Sathya ve Kayalvizhi, 2010). Bu çalışmada çok seviyeli eşiklemede BF algoritmasının GA algoritmasından daha iyi

performans sağladığını göstermişlerdir. Önerdikleri yöntem hesaplama açısından verimli olduğu için karakter tanımlama ve tıbbi görüntülerin eşiklemede kullanılabileceği sonucuna varmışlardır.

Akay, iki başarılı sürü optimizasyon algoritmasından parçacık sürü optimizasyon (PSO) ve yapay arı kolonisini (ABC) kullanarak kapsamlı bir çalışma yapmıştır (Akay, 2013). Bu çalışmada test görselleri için entropi tabanlı Kapur ve sınıflar arası varyans temelli Otsu algoritmaları çok seviyeli eşikleri bulmak için kullanmıştır. İki seviyeli eşiklemede PSO ve ABC benzer değerlerini hesaplarken eşik sayısı arttıkça ABC'nin performansının da arttığını tespit etmiştir. Ek olarak kapsamlı aramada eşik sayısının artması ile üstel bir şekilde artan hesaplama süresinin metasezgisel yöntemler ile lineer bir şekilde arttığını göstermiştir.

Alihodzic ve Tuba, yarası algoritmasının (BA) GA, PSO, ABC gibi yaygın kullanılan algoritmalar ile kıyaslanabilir bir ölçüde olduğunu tespit etmişlerdir (Alihodzic ve Tuba, 2014). Diferansiyel gelişim (DE) ve ABC algoritmalarından bazı unsurları BA algoritmasına ekleyerek geliştirilmiş yarası algoritmasını (IBA) ortaya çıkarmışlardır.

Manic ve arkadaşları ateş böceği algoritmasını (FA) kullanarak entropi tabanlı Kapur ve Tsallis algoritmalarını test görselleri ile karşılaştırmıştır (Manic ve ark., 2016). Bu çalışma sonucunda Tsallis fonksiyonunun daha iyi performans değerlerine ulaştığını, Kapur'un ise nispeten daha düşük işlemci süresi ile yakınsama sağladığını göstermişlerdir.

Koç ve arkadaşları Otsu çok seviyeli eşikleme yöntemini Gri kurt optimizasyon algoritması (GWO) ile optimize etmeyi başarmışlardır (Koç ve ark., 2018). GWO algoritmasının benzer çalışmalardaki diferansiyel gelişim (DE), ateş böceği algoritması (FA), geliştirilmiş yarası algoritması (IBA) gibi algoritmalarından çok daha iyi sonuçlar ortaya çıkardığını tespit etmişlerdir.

Abualigah ve arkadaşları literatürde yaygın kullanılan 9 farklı optimizasyon algoritması ile Kapur, Otsu ve Tsallis'in içinde bulunduğu 5 farklı yöntemi ele alan kapsamlı bir araştırma gerçekleştirmişlerdir (Abualigah ve ark., 2023). Sonuçları karşılaştırmak için uygunluk değeri, PSNR (Tepe Sinyal-Gürültü Oranı) ve SSIM (Yapısal Benzerlik İndeksi) değerlerine göre karşılaştırma yapmışlardır.

Görüntü işleme alanında arayüz geliştirmesi yapılan çalışmalar incelenmiştir. Aşağıda kısaca bahsedilen bu çalışmalar görüntü işlemenin çeşitli süreçlerini kolaylaştırmak amacıyla ticari

veya eğitim amaçlı olarak geliştirilmiştir. Görüntü işlemedeki süreçleri farklı optimizasyon algoritmaları ile karşılaştırabilen kapsamlı bir çalışma bulunamamıştır.

Mahmoudi ve arkadaşları, internetin olduğu her yerde tıbbi görüntü işleme hizmeti sunmayı amaçlamaktadırlar (Mahmoudi ve ark., 2010). Bu hizmet ile istemci tarafında kurulum gerektirmeden tamamen web tabanlı, etkileşimli, genişletilebilir, görsel ve sonuçları 2D ve 3D tıbbi görüntü işleme ve görselleştirme araçlarıyla beraber tanıtılmaktadır. Çok çeşitli tıbbi görüntü ön işleme, kayıt ve segmentasyon yöntemleri açık kaynaklı kütüphaneler kullanılarak uygulanmaktadır. Bu teknolojilerin entegrasyonu, istemci tarafında güçlü hesaplama kaynakları gerektirmeden yüksek işlevselliğe sahip tamamen web tabanlı bir yazılımın uygulanmasına olanak tanır. Kullanıcı arayüzü, kullanıcıların pratik araştırma ve klinik çalışmalar için uygun parametreleri seçmeleri için tasarlanmıştır.

Markiewicz ve arkadaşları, patolojik tanıyı desteklemek için mikroskopik görüntülerin bilgisayarla analizi için ücretsiz bir web tabanlı araç sunmaktadırlar (Markiewicz ve ark., 2016). Tıbbi görüntü analizi, depolama ve erişimde web tabanlı hizmetler için pratik çözümler de sunulmakta ve tartışılmaktadır.

Yuan ve arkadaşları, tıbbi görüntüleri sunmak ve işlemek için web tabanlı bir platform önermektedir (Yuan ve ark., 2017). İnternet ve yeni veri tabanı teknolojilerini kullanarak, yetkili kullanıcılar tıbbi görüntülere kolayca erişebilir ve herhangi bir kurulum yapmadan sunucu tarafı güçlü bilgi işlem performansı ile iş akışlarını kolaylaştırabilirler. Yazarlar Rayplus'ın ilk versiyonunda bir dizi görüntü işleme ve görselleştirme algoritması uygulamaktadır. Sistemin entegrasyonu, hem araştırma hem de klinik topluluklar için çok fazla esneklik ve kolaylık sağlar.

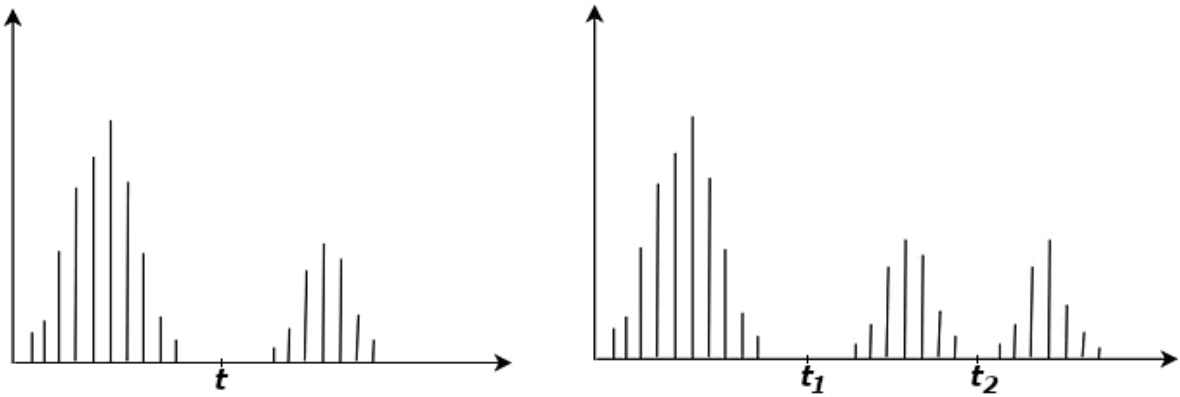
Milošević ve Vujičić, Matlab web hizmetini kullanarak mamografik görüntülerin uzaktan işlenmesi için web tabanlı bir araç sunmaktadır (Milošević ve Vujičić, 2023). Bu araç Matlab'ın hesaplama ve grafik yeteneklerini internet üzerinden uzaktan erişime açmaktadır. Mamogram verileri ve parametreleri bir web tarayıcı aracılığıyla yüklenebilir. Araç eşikleme gibi çeşitli görüntü işleme yöntemlerini kullanarak meme kanseri hakkında sonuçlar üretebilir.

### 3. GÖRÜNTÜ EŞİKLEME

Görüntü işleme alanında kullanılan önemli tekniklerden biri olan görüntü eşikleme, görüntünün segmentasyonu için basit ve etkili bir yöntem sunmaktadır. Bu işlem, görüntüdeki her bir pikselin yoğunluk seviyesine göre ayrıştırılmasını sağlar. Eşikleme yöntemi sayesinde, görüntünün ön plan ve arka plan bölgelerinin ayrılması kolaylaşır. Bu ayırım, piksellerin yoğunluk ve renk değerlerine dayanarak gerçekleştirilir. Ayrıca, belirli bir yoğunluk aralığına veya şeridine denk gelen piksel bölgelerinin tespiti de eşikleme tekniği ile mümkün hale gelir. Görüntü segmentasyonu açısından oldukça önemli olan bu yöntem, akademik çalışmalarda ve uygulamalarda sıklıkla tercih edilmektedir.

Görüntü eşikleme kolay ve etkili bir çözüm olduğu için geniş bir kullanım alanına sahiptir. Bu çalışmalar, medikal görüntüleme (Rajinikanth ve ark., 2017; Kökver ve ark., 2018), optik karakter algılama (Lázaro ve ark., 2010), nesne takibi (Parekh ve ark., 2007), otomatik hedef algılama (Anagnostopoulos, 2009) gibi birçok alanda kullanıma sahiptir.

Görüntü eşikleme işlemi, görüntünün histogramındaki gri seviye değerine bağlı olarak görüntüyü doğrudan bölgelere ayırır. Eşikleme görüntüler üzerinde iki seviyeli ve çok seviyeli olarak uygulamak mümkündür. Şekil 1a'da histogramı verilen görüntü, siyah arka plan üzerine parlak nesnelere sahip bir görüntüdür. Görüntüde siyah arka plan üzerinde parlak nesnelere baskın bir grup oluşturacaklardır. Bu histogram değerlerinde nesnelere arka plandan ayırmak için  $t$  eşik değerini seçmek yeterli olacaktır. Şekil 1b'de iki tip parlaklığı olan nesnelere görselinin grafiği verilmiştir. Nesnelere ve arka plana ait gri seviyeleri üç baskın grup oluşturmuştur. Bu görseli çok seviyeli eşikleme ile  $t_1$  ve  $t_2$  olarak iki seviyeli eşikleme yapmak daha uygun olacaktır.



Şekil 1. (a) Tek eşik ile bölümlenecek görüntü histogramı (b) İki eşik ile bölümlenecek görüntü histogramı

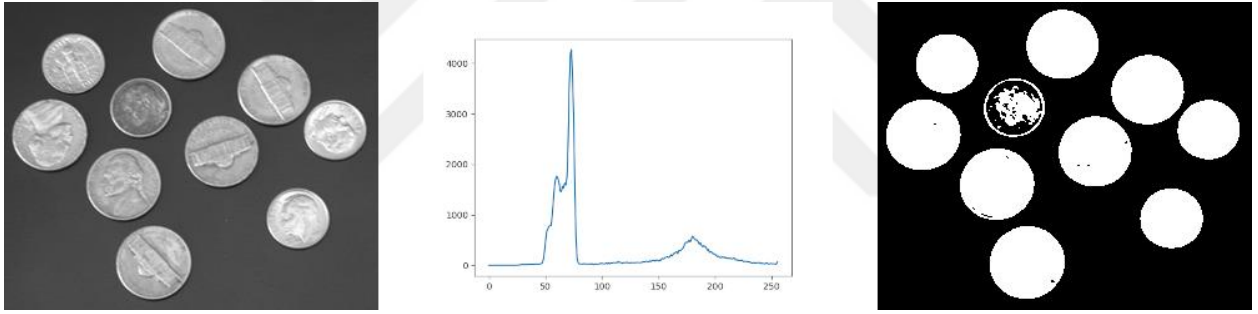
### 3.1. Tek Seviyeli Görüntü Eşikleme

Tek seviyeli eşikleme, bir görüntünün tek bir eşik değerine göre eşiklemesidir. Eşik değeri  $T$  değeri olarak belirlenir ve bu değere göre pikseller ikili bir görüntüye dönüştürülür. Bu yöntem ile görüntüdeki bir nesne diğerlerinden ayırt edilebilir hale getirilir.

Şekil 2’de de görüldüğü gibi tek bir eşik değerinin belirlenmesi ile görseldeki tek tip tüm nesnelerin arka plan ile ayrımı sağlanmıştır. Bu yöntem en basit ve hızlı eşikleme yöntemidir. Hesaplama maliyetleri oldukça düşüktür. Basit görsellerde kullanıma uygundur.

$$s = M(r) = \begin{cases} 0 & \text{eger } r < T \text{ ise} \\ 1 & \text{eger } r \geq T \text{ ise} \end{cases} \quad (3.1)$$

Bu formülde,  $s$  çıkış görüntüsündeki piksel değerini,  $r$  ise giriş görüntüsündeki işlem yapılan piksel değerini ve  $T$  ise uygulanan eşik değerini gösterir.



Şekil 2. (a) Madeni paralar (b) Histogram (c) Eşiklenmiş sonuç görüntüsü

### 3.2. Çok Seviyeli Görüntü Eşikleme

Çok seviyeli eşikleme, bir görüntüyü birden fazla eşik değeri kullanarak eşikler. Bu yöntem, tek seviyeli eşiklemenin yetersiz kaldığı durumlarda daha ayrıntılı bilgi sağlamak için kullanılır.

Eşikleme metotları, çoğunlukla verilen resmin histogramına bağlı olarak çalışmaktadır. İşlenecek bir görüntünün  $L$  gri seviyeleri,  $0$  ve  $L-1$  arasında  $t$  eşik değeri,  $I$  verilen bir resmi ifade ettiği varsayımı üzerinden, bir görüntü için iki seviyeli eşikleme aşağıdaki denklem ile tanımlanır.

$$\begin{aligned} P_F &= \{ I(x, y) \in I \mid 0 \leq I(x, y) \leq t - 1 \} \\ P_B &= \{ I(x, y) \in I \mid t \leq I(x, y) \leq L - 1 \} \end{aligned} \quad (3.2)$$

Eşikleme için segment sayısının artmasıyla, iki seviyeli eşikleme çok seviyeli eşiklemeye dönüştürülebilir.

$$\begin{aligned}
 P_0 &= \{ I(x, y) \in I \mid 0 \leq I(x, y) \leq t_0 - 1 \} \\
 P_1 &= \{ I(x, y) \in I \mid 0 \leq I(x, y) \leq t_1 - 1 \} \\
 &\dots \\
 P_n &= \{ I(x, y) \in I \mid 0 \leq I(x, y) \leq L - 1 \}
 \end{aligned} \tag{3.3}$$

Burada  $I(x, y)$  gri seviyeli piksel değerini ifade etmektedir.  $I(x, y)$   $[0, 255]$  arasındaki değere sahiptir. Çok seviyeli eşiklemenin amacı optimal eşik değerlerini ( $t_0, t_1, \dots, t_n$ ) hesaplamaktır. ( $P_0, P_1, \dots, P_n$ ) verilen resmin farklı kısımlarını temsil etmektedir.  $A \cap \tilde{A} = \emptyset$  koşulunun sağlanması gereklidir ve  $P$  değerlerinin hepsinin birleşimi  $I$  resmini oluşturmaktadır (Koç ve ark., 2018).

### 3.2.1. Kapur entropi tabanlı eşikleme yöntemi

Kapur'un entropi tabanlı fonksiyonu ilk olarak 1985'te, gri seviye görüntü histogramının entropisini maksimize ederek bölütleme yapmak için önerilmiştir (Kapur ve ark., 1985). Bu yöntem entropiyi görüntüyü bölütleme işlemlerinde belirlenecek olan optimum eşik değerini bulmak için amaç fonksiyonu olarak kullanır. Kapur entropi metodunda optimum eşik değerini bulmak için her bölgenin entropisi tek tek hesaplanmakta ve maksimum olanı eşik değeri olarak belirlenmektedir (Doğan ve ark., 2017). Kapur Entropi tabanlı eşikleme yöntemi tek seviyeli eşikleme için geliştirilmiş olsa da çok seviyeli eşikleme için de genişletilebilir (Lang ve Jia, 2019).

$N$  tane piksel değerine sahip herhangi bir görüntünün  $L$  gri seviyeli  $0, 1, 2, \dots, L-1$  olduğu düşünüldüğünde görüntüdeki gri seviye değerlerinin olasılıkları  $P_i$  ve  $h(i)$  imgedeki  $i$  gri seviyesine sahip piksel sayısını göstermektedir. Belirlenen eşik değeri ise  $t$  şeklinde ifade edilmektedir.

$$P_i = h(i)/N \tag{3.4}$$

$$\begin{aligned}
 \omega_0 &= \sum_{i=0}^t P_i & H_0 &= - \sum_{i=0}^t \frac{P_i}{\omega_0} \ln \frac{P_i}{\omega_0} \\
 \omega_1 &= \sum_{i=t}^{L-1} P_i & H_1 &= - \sum_{i=t}^{L-1} \frac{P_i}{\omega_1} \ln \frac{P_i}{\omega_1}
 \end{aligned} \tag{3.5}$$

$$f(t) = \arg \max (H_0 + H_1) \tag{3.6}$$

Eşitlik 3.6'da sınıf entropileri toplamı maksimum yapıldığında elde edilen eşik seviye değeri optimum değer olacaktır.

Çok seviyeli eşiklemede eşik değerleri  $[th_1, th_2, \dots, th_n]$  şeklinde olacaktır.

$$H(th_1, th_2, \dots, th_n) = H_0 + H_1 + \dots + H_n \quad (3.7)$$

$$H_0 = - \sum_{i=0}^{th_1-1} \frac{P_i}{\omega_0} \ln \frac{P_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{th_1-1} P_i$$

$$H_1 = - \sum_{i=th_1}^{th_2-1} \frac{P_i}{\omega_1} \ln \frac{P_i}{\omega_1}, \quad \omega_1 = \sum_{i=th_1}^{th_2-1} P_i$$

$$\dots$$

$$H_n = - \sum_{i=th_n}^{L-1} \frac{P_i}{\omega_n} \ln \frac{P_i}{\omega_n}, \quad \omega_n = \sum_{i=th_n}^{L-1} P_i \quad (3.8)$$

$H_0, H_1, \dots, H_n$  farklı sınıfların entropilerini,  $\omega_0, \omega_1, \dots, \omega_n$  her sınıfın olasılığını gösterir. En uygun eşik değerini elde etmek için aşağıdaki eşitlik kullanılır.

$$f_{Kapur}(th_1, th_2, \dots, th_n) = \arg \max \{ H(th_1, th_2, \dots, th_n) \} \quad (3.9)$$

### 3.2.2. Otsu eşikleme yöntemi

Otsu yöntemi, 1979 yılında Nobuyuki Otsu tarafından önerilmiş, gri tonlamalı görüntüler üzerinde kullanılabilen bir eşik tespit yöntemidir (Otsu, 1996). Bu yöntem görüntünün arka plan ve ön plan olmak üzere iki renk sınıfından oluştuğunu kabul eder. Daha sonra tüm eşik değerleri için bu iki renk sınıfının sınıf içi varyans değeri hesaplanır. Bu değer minimum olmasını sağlayan eşik değeri optimum eşik değeri olarak belirlenir.

Sınıf içi varyans değeri minimum değerinde iken sınıflar arası varyans değeri maksimum değerinde olur. Sınıflar arası varyans değerinin hesaplanması daha az işlem gerektirir. Bu nedenle yöntem koda dökülürken genellikle sınıflar arası varyans değeri hesaplanır.

Histogramı hesaplanan görüntünün  $t$  eşik değeri için amaç fonksiyonunun formülü 3.14 de verilmiştir.

$$p_i = \frac{x_i}{X} \quad (3.10)$$

$$\omega_0 = \sum_{i=0}^{t-1} p_i, \omega_1 = \sum_{i=t}^{L-1} p_i \quad (3.11)$$

$$\mu_0 = \frac{1}{\omega_0} \sum_{i=0}^{t-1} i \cdot p_i, \mu_1 = \frac{1}{\omega_1} \sum_{i=t}^{L-1} i \cdot p_i \quad (3.12)$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_t)^2, \sigma_1 = \omega_1(\mu_1 - \mu_t)^2 \quad (3.13)$$

$$f(t) = \sigma_0 + \sigma_1 \quad (3.14)$$

Burada  $x_i$   $i$ . seviyedeki toplam piksel sayısını,  $X$  görüntüdeki toplam piksel sayısını temsil etmektedir.  $p_i$   $i$ . gri seviyedeki olasılık değerini göstermektedir.  $\omega_0, \omega_1$  segmentlerin oluşmasının tahmin edilme olasılık değerini gösterir.  $\mu_0, \mu_1$  sınıfların ortalama yoğunluğunu ifade etmektedir.  $\sigma_0, \sigma_1$  ise sınıfların varyansını ifade etmektedir.

Birden fazla eşik değeri için, eşik değerleri  $t_0, t_1, \dots, t_n$  şeklinde belirlenir. Sınıflar arası varyans (3.17) eşitliği kullanılarak Otsu yöntemi çok seviyeli eşikleme için genişletilebilir. Eşitlik (3.15) değerine göre maksimum eşik değerleri bulunabilir.

$$t^* = \arg \max [f_m(t)] \quad (3.15)$$

$$f_b(t) = \sum_{i=0}^n \sigma_i \quad (3.16)$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_t)^2, \sigma_1 = \omega_1(\mu_1 - \mu_t)^2, \dots, \sigma_n = \omega_n(\mu_n - \mu_t)^2 \quad (3.17)$$

$$\mu_0 = \frac{1}{\omega_0} \sum_{i=0}^{t_0-1} i \cdot p_i, \mu_1 = \frac{1}{\omega_1} \sum_{i=t_0}^{t_1-1} i \cdot p_i, \dots, \mu_n = \frac{1}{\omega_n} \sum_{i=t_n}^{L-1} i \cdot p_i \quad (3.18)$$

$$\omega_0 = \sum_{i=0}^{t_0-1} p_i, \omega_1 = \sum_{i=t_0}^{t_1-1} p_i, \dots, \omega_n = \sum_{i=t_n}^{L-1} p_i \quad (3.19)$$

### 3.2.3. Tsallis entropi tabanlı eşikleme yöntemi

Tsallis entropisi, bir olasılık dağılımının düzensizliğini ölçmek için kullanılan bir entropi türüdür. Tsallis entropisi,  $q$  parametresi ile klasik Shannon entropisinin bir genellemesi olarak tanımlanır (Kapur ve ark., 1985). Entropi tabanlı eşikleme, görüntünün histogramı kullanılarak

optimal eşik değerinin bulunmasını içerir. Bu yöntem, histogramı iki sınıfa (arka plan ve ön plan) ayırarak her sınıfın Tsallis entropisini hesaplar ve toplam entropiyi maksimize eden eşik değerini belirler (Agrawal ve ark., 2013). Tsallis entropisinin genel tanımlaması (3.20) eşitliği şeklinde tanımlanabilir.

$$S_q = \frac{1 - \sum_{i=1}^t (p_i)^q}{q - 1} \quad (3.20)$$

Eşitliğe göre  $t$  eşik değerini,  $p_i$  olasılık değerini ve  $q$  entropinin non-lineerlik derecesini belirleyen parametredir. Herhangi bir görüntünün  $L$  gri seviyeli  $0, 1, 2, \dots, L-1$  olduğu düşünüldüğünde görüntüdeki gri seviye değerlerinin olasılıkları ( $P_i = p_0, p_1, p_2, p_{L-1}$ ) değerleri aşağıdaki eşitlik kullanılarak hesaplanmaktadır. Eşitlik (3.24) göre maksimum değere çıkaran en uygun  $T$  değeri bulunur.

$$P_i = h(i)/X \quad (3.21)$$

$$[S_q^A(T) + S_q^B(T) + \dots + S_q^K(T) + (1 - q) \cdot S_q^A(T) \cdot S_q^B(T)] \quad (3.22)$$

$$S_q^A(T) = \frac{1 - \sum_{j=0}^{t_1-1} (\frac{P_j}{P^A})^q}{q - 1}, P^A = \sum_{j=0}^{t_1-1} P_j \quad (3.23)$$

$$S_q^B(T) = \frac{1 - \sum_{j=t_1}^{L-1} (\frac{P_j}{P^B})^q}{q - 1}, P^B = \sum_{j=t_1}^{L-1} P_j \quad (3.24)$$

$$f(T) = [T] = \arg \max [S_q^A(T) + S_q^B(T) + (1 - q) \cdot S_q^A(T) \cdot S_q^B(T)] \quad (3.25)$$

Birden fazla eşik değerini hesaplamak için aşağıdaki eşitlik kullanılarak Tsallis entropi yöntemi genişletilebilir.

$$S_q^A(T) = \frac{1 - \sum_{j=0}^{t_1-1} (\frac{P_j}{P^A})^q}{q - 1}, P^A = \sum_{j=0}^{t_1-1} P_j \quad (3.26)$$

$$S_q^B(T) = \frac{1 - \sum_{j=t_1}^{t_2-1} (\frac{P_j}{P^B})^q}{q - 1}, P^B = \sum_{j=t_1}^{t_2-1} P_j \dots$$

$$S_q^K(T) = \frac{1 - \sum_{j=t_k}^{L-1} \left(\frac{P_j}{P^K}\right)^q}{q - 1}, P^K = \sum_{j=t_k}^{L-1} P_j$$

Aşağıdaki kısıtlamalara tabidir:

$$\begin{aligned} |P^A + P^B| - 1 < S < 1 - |P^A - P^B| \\ |P^B + P^C| - 1 < S < 1 - |P^B - P^C| \\ |P^K + P^{L-1}| - 1 < S < 1 - |P^K - P^{L-1}| \end{aligned} \quad (3.27)$$

$$\begin{aligned} f(T) = [T_1, T_2, \dots, T_k] = \arg \max \\ [S_q^A(T) + S_q^B(T) + \dots + S_q^K(T) + (1 - q) \cdot S_q^A(T) \cdot S_q^B(T) \dots S_q^K(T)] \end{aligned} \quad (3.28)$$

Bu işlemlerde amaç  $f(T)$  fonksiyonunu maksimum değere çıkaran en uygun  $T$  eşik değerini bulmaktır.

## 4. MATERYAL VE YÖNTEM

### 4.1. Web Tabanlı Görüntü Eşikleme Aracı

Görüntü işleme alanında, görüntü eşikleme önemli bir adımdır. Bu adımda, görüntüdeki pikseller belirli eşik değerlerine göre ayrılarak görüntü segmentasyonu gerçekleştirilir. Geliştirdiğimiz web tabanlı görüntü eşikleme aracı, kullanıcıların farklı optimizasyon algoritmalarını ve görüntü eşikleme fonksiyonlarını kullanarak eşikleme işlemini gerçekleştirmelerini sağlar. Bu bölümde, aracın web arayüzünden Docker Container ile mimari geliştirilmesine ve API kullanımına kadar tüm detayları ele alınmıştır.

#### 4.1.1. Web arayüzün geliştirilmesi

Web tabanlı çok seviyeli görüntü eşikleme aracının arayüzü, kullanıcı dostu ve işlevsel bir yapıda tasarlanmıştır. Arayüz, kullanıcıların kolayca görsellerini yüklemelerini ve eşik değerlerine hemen ulaşabilmelerini sağlar. Hızlı sonuçlara ulaşabilmek adına arayüz üzerinde önceden seçilmiş ayarlar ile anlaşılması kolay bir yapıdadır. Arayüz üzerinde literatürde en çok kullanılan görseller ile kullanıcıların farklı optimizasyon algoritmalarını test etme imkânı sunulmaktadır.

Arayüz üzerindeki işlemlerin çalışma sırası şu şekildedir:

1. Kullanıcı, eşik değerine ayırmak istediği görseli yükler veya önceden yüklenmiş hazır test görsellerinden birini seçer.
2. Optimizasyon algoritmasını seçer ve gerekliyse algoritmanın parametrelerini günceller.
3. Uygun çok seviyeli eşikleme yöntemi seçilir.
4. Görüntünün kaç eşik değerine ayrılacağı (minimum 2, maksimum 10) belirlenir.
5. Tüm bu parametre seçimlerinden sonra hesaplama işlemi başlatılır.
6. Hesaplama işlemi tamamlandığında, sonuçlar ekranın alt kısmında gösterilir.

Arayüzde kullanıcıların istedikleri optimizasyon algoritmasını seçmelerine olanak tanıyan bir açılır kutu bulunmaktadır. Kullanıcı, ateş böceği algoritması, benzetilmiş tavlama algoritması, diferansiyel gelişim algoritması, genetik algoritma, gri kurt optimizasyon algoritması, harmoni arama algoritması, klonal seçim algoritması, parçacık sürü optimizasyon algoritması, yapay arı koloni algoritması ve hibrit optimizasyon algoritmalarından birini seçebilir. Seçilen her bir optimizasyon algoritması için gerekli parametre ayar ekranı açılmaktadır. Bu ekranda önceden test edilmiş en iyi parametre değerleri otomatik olarak gelmektedir. Bu değerler ile oynayıp kendi görselleri için de en iyi sonucu elde eden parametreler bulunabilir.

## A Web Based Tool For Multilevel Image Thresholding

Upload or Select an Image

#	Name	Fit.	Th. Count	Th. Values	Fit. Value	Time	Image
---	------	------	-----------	------------	------------	------	-------

**Şekil 3.** Web tabanlı çok seviyeli görüntü eşikleme aracının arayüzü

Kullanıcılar, arayüz aracılığıyla yükledikleri görsellerin eşik değerlerini, literatürde görüntü eşikleme için en çok kullanılan Kapur, Otsu ve Tsallis eşikleme algoritmalarını kullanarak kolayca elde edebilirler. Bu sayede, çeşitli eşikleme yöntemlerinin görsel üzerindeki etkilerini karşılaştırarak en uygun eşikleme algoritmasını belirleyebilirler.

Sistem, server-client şeklinde bir mimariye sahiptir. Tüm hesaplama işlemleri server tarafında yapılmaktadır. Server, tamamen ölçeklenebilir bir yapıda Docker mimarisini kullanmaktadır. Arayüz, tamamen kullanıcının tarayıcısında çalışan bir uygulamadır. İki farklı programlama dili ile tasarlanmış iki farklı yazılım birbirleri ile uyum içinde çalışmaktadır.

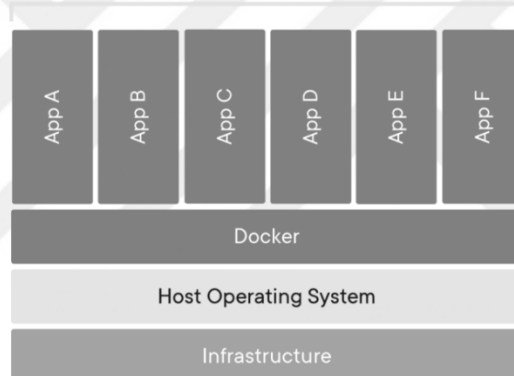
Geliştirdiğimiz arayüz uygulamasının en temel özelliklerinden biri, işlemlerin tamamen sunucu tarafında çalışmasıdır. Kullanıcılar herhangi bir web görüntüleme aracı kullanarak istedikleri platformdan arayüze erişip hesaplama yapabilirler. Arayüz bir API sayesinde çalışmaktadır. Bu API kullanılarak farklı platformlar üzerinden sunucunun çalıştığı adrese HTTP istekleri gönderilip sonuçlara saniyeler içinde ulaşılabilmektedir. Aynı anda birden fazla istek gönderilse bile, sunucu yapısı sayesinde kullanıcı tarayıcısında bekleme, donma gibi kullanıcı deneyimini etkileyen durumlar yaşanmamaktadır.

Veri güvenliği açısından sistem oldukça güvenli bir mimariye sahiptir. Haberleşme sırasında şifreli bir protokol ile veri transferleri gerçekleşmektedir. Server tarafında hiçbir veri depolanmadan sadece kullanıcının tarayıcısında geçici bellek tarafında veriler kaydedilmektedir.

Kullanıcı dilerse tarayıcısındaki verileri arayüz üzerinden temizleyebilir. Hesaplanan veriler otomatik olarak tarayıcının belleğinden 30 gün sonra silinmektedir. Kullanıcı bu verileri arayüz üzerinden ".csv" formatında kendi bilgisayarına indirip Excel gibi uygulamalarda rahatlıkla inceleyebilir.

#### 4.1.2. Docker container ile web mimari geliştirilmesi

Docker Container, modern yazılım geliştirme ve dağıtım süreçlerinde kullanılan, uygulamaların ve tüm bağımlılıklarının tek bir yerde paketlenildiği hafif taşınabilir yazılım birimleridir. Docker teknolojisi, yazılımların farklı ortamlarda tutarlı ve sorunsuz çalışmasını sağlamak amacıyla geliştirilmiştir. Her bir Docker container, küçük birer işletim sistemi barındıran yazılımlardır ve bu sayede uygulamanın tamamen izole bir ortamda çalışmasını sağlar.

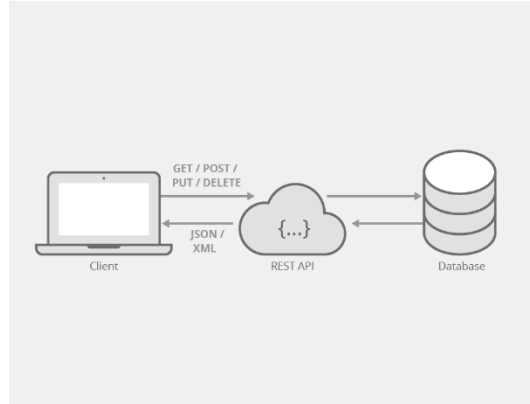


Şekil 4. Docker mimarisi

Web tabanlı aracın geliştirilmesinde Docker Container teknolojisi kullanılmıştır. Bu teknoloji sayesinde server, farklı cihazlarda sorunsuz bir şekilde çalışabilir. Docker container yapısı kullanılarak aynı cihazda birden fazla bağımsız kopya oluşturularak istemcilerin bekleme süreleri azaltılabilir.

#### 4.1.3. API kullanımı

API (Application Programming Interface), iki yazılım bileşeninin belirli tanımlar ve protokoller aracılığıyla birbiriyle iletişim kurmasını sağlar. API'ler, uygulamalar arası iletişim sağladığı için bir Kullanıcı Arayüzü (UI) yoktur. API'ler, client ve server arasındaki iletişimi sağladığı için mobil ve web uygulamalarının birçoğunda kullanılır.



**Şekil 5.** API kullanan bir client-server mimarisinin basitleştirilmiş diyagramı

Web tabanlı görüntü eşikleme aracının temel işlevselliği, server tarafında geliştirilen API'ler aracılığıyla sağlanır. Bu API'ler, kullanıcıların arayüz üzerinden yaptıkları işlemleri sunucu tarafında gerçekleştiren ve sonuçları döndüren çeşitli fonksiyonlar içerir. Veriler, şifrelenmiş bir şekilde JSON formatında uçlar arasında iletilir. API, web arayüzü haricinde HTTP isteği şeklinde de kullanılabilir. Bu sayede internet erişimi olan her cihaz API'yi kullanarak sunucunun hesaplama kabiliyetlerinden faydalanabilir.

#### 4.2. Optimizasyon Algoritmaları

Optimizasyon problemleri mühendislik, bilim, ekonomi gibi birçok alanda bulunmaktadır. Bu problemler genellikle belirli kısıtlamalara tabi olan bir dizi çözüm uzayında en iyi çözümü bulmayı içerir. Bu problemleri çözmek için lineer ve dinamik programlama gibi klasik yöntemler yaygın olarak kullanılmaktadır. Ancak bazı problemlerin çoğu karmaşık doğrusal olmayan ve yüksek boyutlu olabildiği için geleneksel yöntemlerle çözülmesi maliyetli olabilmektedir. Metasezgisel algoritmalar geleneksel yöntemlerle çözülmeyen problemleri doğadan ilham alarak çözmeye çalışmaktadır.

Metasezgisel algoritmalar biyolojik evrim, hayvan davranışları ve fiziksel süreçler gibi çeşitli doğal olaylardan ilham alınarak geliştirilmiştir. Arama alanını verimli ve etkili bir şekilde keşfetmek ve genellikle karmaşık sorunlara makul süre içinde optimum çözüme yakın sonuçları bulmak için tasarlanmıştır (Talbi, 2009).

Metasezgisel kavramı, Yunanca "bulmak" anlamına gelen "heuriskein" ve 'üst seviye yöntembilim' anlamına gelen "meta" kelimelerinin birleşiminden türetilmiştir. Metasezgisel yöntemler, çeşitli sezgisel yöntemleri üst düzeyde bir araya getirerek, arama alanını etkin ve

verimli bir şekilde incelemeyi hedefler. Bu yöntemler, her zaman küresel en iyi çözümü garanti edemese de büyük ölçekli ve karmaşık problemlere etkili çözümler üretebildikleri için oldukça yararlıdır (Onan, 2013).

#### 4.2.1. Ateş Böceği Algoritması

Ateş böceği algoritması (FA), 2009 yılında Xin-She Yang tarafından geliştirilen doğadan ilham alan bir optimizasyon algoritmasıdır. Ateş böceklerinin davranışlarını, özellikle biyolümin esanslarını taklit ederek karmaşık optimizasyon problemlerini çözmeyi amaçlar.

Ateş böcekleri, iletişim kurmak, eş çekmek ve yırtıcılardan korunmak için kullandıkları yanıp sönen ışıklarıyla tanınırlar. Işığın yoğunluğu, atmosfer tarafından emilme ve saçılma nedeniyle mesafe ile azalır. Bu olgu, ateş böceği algoritmasının ana mekanizmasının ilham kaynağı olmuştur: Ateş böcekleri, yanıp sönen ışıklarının parlaklığına göre birbirlerine çekilirler ve bu FA'nın optimizasyon problemindeki aday çözümler arasındaki çekimi modellemesiyle benzerlik gösterir (Yang, 2009).

Ateş böceği algoritması üç kabule dayanmaktadır. Bu kurallar:

- Tüm ateş böcekleri tek cinsiyetlidir ve cinsiyetlerine bakılmaksızın birbirlerine ilgi duyarlar.
- Ateş böceklerinin çekiciliği parlaklığıyla orantılıdır. Herhangi iki yanıp sönen ateş böceği için, daha az parlak olan kendisinden daha parlak olana doğru hareket edecektir. Ancak aralarındaki mesafe arttıkça parlaklık azalır. Bir ateş böceğinden daha parlak olan başka bir ateş böceği yoksa hareketine rastgele olarak devam eder.
- Bir ateş böceğinin parlaklığı, çözülmesi istenen optimizasyon probleminin amaç fonksiyonu olarak belirlenir.

Bu kuralları yinelemeli bir optimizasyon sürecine dahil eden ateş böceği algoritması, ateş böceklerini (aday çözümler) en parlak bireylere (en iyi çözümler) doğru hareket ettirerek küresel olarak en uygun çözümü bulmaya çalışır. Bu üç kurala dayanarak FA'nın temel adımları Şekil 6'da gösterilen sahte kod ile özetlenebilir.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
Ateş böceklerinin başlangıç popülasyonunu üret  $x_i$  ( $i = 1, 2, 3, \dots, n$ )
 $x_i$ 'deki  $I_i$  ışık yoğunluğunu  $f(x_i)$  ile belirle
 $\gamma$  ışık emilme katsayısını tanımla
while ( $t < \text{İterasyon sayısı}$ )
  for  $i = 1 : n$  (Bütün ateş böcekleri için)
    for  $j = 1 : n$  (Bütün ateş böcekleri için)
      if ( $I_i < I_j$ )
        i ateş böceğini j'ye doğru taşı
      end if
       $\exp[-\gamma r]$  aracılığıyla  $r$  mesafesi ile çekicilikleri değiştir
      Yeni çözümleri değerlendir ve ışık yoğunluklarını güncelle
    end for  $j$ 
  end for  $i$ 
  Ateş böceklerini sırala ve global en iyi çözümü bul
   $t = t + 1$ 
end while
Sonuçları döndür

```

Şekil 6. FA'nın sözde kodu

#### 4.2.2. Benzetilmiş Tavlama Algoritması

Benzetimli Tavlama (SA), metalürjideki tavlama işleminden esinlenen çok boyutlu bir optimizasyon metodudur. Tavlama, bir metalin yüksek bir sıcaklığa kadar ısıtılmasını ve ardından moleküler yapısını optimize etmek için yavaşça soğutulmasını içerir. Böylece metallerin kararlı ve yapısal olarak güçlü bir seviyeye ulaşmaları için zaman tanınmış olur. Benzer şekilde, benzetilmiş tavlama algoritması bir optimizasyon probleminin çözüm uzayını yinelemeli olarak araştırır, zaman zaman yerel optimumlardan kaçmak için daha kötü çözümleri kabul eder ve "soğudukça" kademeli olarak en iyi çözümlere odaklanır (Kirkpatrick ve ark., 1983).

Benzetilmiş Tavlama Algoritması, Kirkpatrick ve arkadaşları tarafından 1983 yılında doğrusal olmayan problemlerin çözümüne yönelik olarak geliştirilmiştir. Bu algoritma, sıcaklık değeri yüksekken, sisteme yerel minimumun dışına çıkma olasılığı tanıyarak çalışır. Algoritmanın hem basit uygulama yöntemi hem de geniş bir uygulama alanı olması birçok araştırmacının ilgisini çekmiştir (İlhan, 2021).

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
Başlangıç çözümünü ( $x_0$ ) ve başlangıç sıcaklığını ( $T_0$ ) belirle
soğutma_fonksiyonu ve max_iterasyon sayısını tanımla
t = 0
while (t < max_iterasyon)
     $x' = \text{Komşu\_Çözüm\_Üret}(x)$ 
     $\Delta f = f(x') - f(x)$ 
    if ( $\Delta f < 0$ )
         $x = x'$ 
    else if ( $\text{rand}(0, 1) < \exp(-\Delta f / T)$ )
         $x = x'$ 
    end if
     $T = \text{soğutma\_fonksiyonu}(T)$ 
     $t = t + 1$ 
end while
Sonuçları döndür

```

Şekil 7. SA'nın sözde kodu

Benzetilmiş tavlama algoritması, probleme has bir başlangıç çözümü ve yüksek bir başlangıç sıcaklığı ile başlar. Sıcaklık parametresi, tavlama işlemindeki sıcaklığa benzer ve optimizasyon işlemi sırasında daha kötü çözümleri kabul etme olasılığını kontrol eder. Her iterasyonda, algoritma mevcut çözümü biraz değiştirerek yeni bir çözüm üretir. Bu yeni çözüm “komşu” çözüm olarak adlandırılır. Komşu çözümler üretme süreci probleme özgüdür ve öğeleri değiştirme, çevirme veya kaydırma gibi çeşitli teknikler içerebilir. Algoritma hem mevcut çözüm hem de komşu çözüm için amaç fonksiyonunu değerlendirir. Komşu çözüm daha iyiyse, yeni mevcut çözüm olarak kabul edilir. Bununla birlikte, komşu çözüm daha kötüyse, mevcut sıcaklığa

ve çözümler arasındaki maliyet farkına bağlı olan belirli bir olasılıkla yine de kabul edilebilir. Sıcaklık, bir soğutma programına göre zaman içinde kademeli olarak azaltılır. Sıcaklık düştükçe, daha kötü çözümleri kabul etme olasılığı azalır ve algoritmanın küresel optimuma doğru yakınsamasına izin verir. İterasyon, önceden tanımlanmış bir iterasyon sayısına ulaşma, bir yakınsama kriterini karşılama veya yeterince düşük bir sıcaklığa ulaşma gibi bir sonlandırma koşulu karşılanana kadar devam eder.

Benzetilmiş tavlama algoritması, kısıtlamalara kolayca uyum sağlayabilme yeteneğine sahiptir. Çünkü her adımda değerlendirilen değerlere kısıtlar rahatlıkla eklenebilir. Ancak, algoritmanın en büyük dezavantajı, yerel minimuma takılma olasılığının her zaman var olmasıdır. Bu durum, uygun bir tavlama süreci ve başlangıç noktası kullanılarak yapılan bazı iyileştirmelerle önlenabilir.

#### 4.2.3. Diferansiyel Gelişim Algoritması

Diferansiyel gelişim algoritması (DE), gerçek parametreleri ve değerli fonksiyonları optimize etmek için 1997 yılında Storn ve Price tarafından tanıtılan stokastik, popülasyon tabanlı bir optimizasyon algoritmasıdır (Storn ve Price, 1997). DE, sürekli verilerin yoğunlukta olduğu problemlerde iyi sonuçlar verebilen, çalışma ve operatörler bakımından evrimsel algoritmalara dayanmaktadır (Raj ve ark., 2019).

DE, doğal evrim ilkelerinden esinlenen popülasyon tabanlı bir meta sezgisel optimizasyon algoritmasıdır. Mutasyon, çaprazlama ve seçim süreci aracılığıyla bir aday çözüm popülasyonunu iteratif olarak geliştirerek çalışır. Algoritma, doğrusal olmayan, farklılaştırılmayan ve çok modlu amaç fonksiyonlarının varlığında bile belirli bir arama uzayı içinde küresel optimum veya optimuma yakın bir çözüm bulmayı amaçlamaktadır.

Algoritma, arama uzayı içinde rastgele bir başlangıç aday çözüm popülasyonu oluşturarak başlar. Her aday çözüm, karar değişkenlerinin bir vektörü olarak temsil edilir. Her aday çözüm için (vektör), algoritma mevcut popülasyondan rastgele seçilen üç vektörü birleştirerek bir mutant vektör oluşturur. Bu süreç yeni genetik materyaller ortaya çıkarır ve arama uzayının farklı bölgelerini keşfeder. Mutant vektör daha sonra bir çaprazlama işlemi ile hedef vektör ile birleştirilir. Bu adım, mutant ve hedef vektörlerin genetik materyalini karıştırarak her iki ebeveyninden de özellikler alan bir deneme vektörü oluşturur. Deneme vektörü amaç fonksiyonu kullanılarak değerlendirilir ve uygunluğu hedef vektörün uygunluğu ile karşılaştırılır. Deneme

vektörü daha iyi bir uygunluk değerine sahipse, bir sonraki nesilde hedef vektörün yerini alır; aksi takdirde, hedef vektör korunur. Popülasyondaki her aday çözüm için mutasyon çaprazlama ve seçim operatörleri tekrarlanarak yeni bir nesil oluşturulur. Süreç, maksimum nesil sayısı veya tatmin edici bir uygunluk değeri gibi bir durdurma kriteri karşılanana kadar devam eder.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
xi (i := 1, 2, 3, \dots, n) (Her bireyin başlangıç değerlerini belirle)
F: fark ağırlığı faktörünü tanımla
CR: çaprazlama oranını tanımla
t = 0
while (t < İterasyon sayısı)
    for i = 1 to n (Tüm bireyler için)
        Rastgele ve farklı üç birey seç: xr1, xr2, xr3 (r1, r2, r3 ≠ i)
         $V = xr1 + F * (xr2 - xr3)$  (Mutasyon vektörünü oluştur)
         $U = xi$  (Deneme vektörünü başlat)
        for j := 1 to d (Her boyut için)
            if (rand(0, 1) < CR or j = jrand)
                 $U_j = V_j$ 
            else
                 $U_j = x_{ij}$ 
            end if
        end for
        if (f(U) < f(xi))
             $xi := U$  (Yeni çözümü kabul et)
        end if
    t = t + 1
end while
Sonuçları döndür

```

Şekil 8. DE'nin sözde kodu

Diferansiyel gelişim algoritması, karmaşık sayısal optimizasyon problemlerinin çözümünde etkinliğini kanıtlamış güçlü ve çok yönlü bir optimizasyon tekniğidir. Basitliği, sağlamlığı ve uyarlanabilirliği, onu optimizasyon algoritmaları içerisinde değerli bir araç haline getirmekte ve çok sayıda alan ve endüstriyi kapsayan uygulamalar sunmaktadır.

#### 4.2.4. Genetik Algoritma

Genetik algoritma (GA), doğal seçim ve evrim sürecinden esinlenen güçlü bir optimizasyon algoritmasıdır. John Holland tarafından 1975’de geliştirilen bu algoritma, mühendislik, bilgisayar bilimi ve yapay zeka dahil olmak üzere çeşitli alanlarda geniş uygulama alanı bulmuştur. Özünde genetik algoritma, en uygun olanın hayatta kalması ve genetik kalıtım ilkelerini taklit ederek bir dizi çözümü optimal veya optimele yakın bir çözüme doğru iteratif olarak iyileştirme işlemleridir (Holland, 1975).

GA, kromozomlar veya bireyler olarak bilinen aday çözümlerden oluşan bir başlangıç popülasyonunu rastgele oluşturarak başlar. Bu çözümler tipik olarak problemin değişkenlerini temsil eden dizeler veya değer dizileri olarak kodlanır. Popülasyondaki her birey, çözümün eldeki sorunu ne kadar iyi çözdüğünü ölçen önceden tanımlanmış bir uygunluk fonksiyonuna göre değerlendirilir. Uygunluk fonksiyonu, her bireye kalitesini veya uygunluğunu yansıtan bir puan veya “uygunluk” değeri atar. Her bireyin uygunluğunu değerlendirdikten sonra, algoritma mevcut popülasyondan en iyi bireylerin bir alt kümesini seçer. Bu seçim süreci, evrimdeki doğal seçim sürecini taklit ederek daha yüksek uygunluk puanlarına sahip bireylere karşı önyargılıdır. Yaygın olarak kullanılan seçim yöntemleri arasında rulet tekerleği seçimi, turnuva seçimi ve sıralamaya dayalı seçim yer alır. Seçilen bireyler, yeni nesil çözümler oluşturmak için genetik işlemlere tabi tutulur. Bu işlemler çaprazlama ve mutasyondur. İki veya daha fazla ebeveyn birey, bir veya daha fazla yavru birey üretmek için birleştirilir ve bu işlem çaprazlama olarak adlandırılır. Çaprazlama, ebeveynler arasında genetik materyal alışverişi yaparak biyolojik üremeyi taklit eder. Rastgele değişiklikler veya mutasyonlar, popülasyonda çeşitliliği teşvik ederek ve erken yakınsamayı önleyerek yavru bireylere eklenir. Genetik operatörler tarafından üretilen yavru bireyler, mevcut popülasyondaki bireylerin bir kısmının veya tamamının yerini alarak bir sonraki nesli oluşturur. Seçim ve genetik işlemler döngüsü, tatmin edici bir çözüme veya önceden tanımlanmış bir nesil sayısına ulaşmak gibi bir sonlandırma koşulu karşılanana kadar tekrarlanır.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
xi ( $i := 1, 2, 3, \dots, n$ ) (Her bireyin başlangıç değerlerini belirle)
MO: mutasyon olasılığını tanımla
CO: çaprazlama olasılığını tanımla
t = 0
while (t < İterasyon sayısı)
  Yeni_popülasyon = []
  for i = 1 to  $n/2$ 
    x_ebeveyn1 := Seçim(Popülasyon)
    x_ebeveyn2 := Seçim(Popülasyon)
    if (rand(0, 1) < CO)
      [çocuk1, çocuk2] := Çaprazlama(x_ebeveyn1, x_ebeveyn2)
    else
      [çocuk1, çocuk2] := [x_ebeveyn1, x_ebeveyn2]
    end if
    if (rand(0, 1) < MO)
      çocuk1 := Mutasyon(çocuk1)
    end if
    if (rand(0, 1) < mutasyon_olasiligi)
      çocuk2 := Mutasyon(çocuk2)
    end if
    Yeni_popülasyon = Yeni_popülasyon + [çocuk1, çocuk2]
  end for
  Popülasyon = Yeni_popülasyon
  t = t + 1
end while
Sonuçları döndür

```

Şekil 9. GA'nın sözde kodu

Genetik algoritma karmaşık optimizasyon problemlerinin çözümünde kayda değer bir başarı göstermiş olsa da performansı, problem gösterimi, seçilen genetik operatörler ve popülasyon

büyüklüğü, mutasyon oranı ve seçim yöntemleri gibi parametrelerin ayarlanması gibi faktörlere bağlıdır. Ayrıca, büyük arama uzayları veya pahalı uygunluk değerlendirmeleri olan problemler için hesaplama maliyeti yüksek olabilir.

#### 4.2.5. Gri Kurt Optimizasyon Algoritması

Gri Kurt Optimizasyon Algoritması (GWO), kurtların sosyal liderlik ve avlanma davranışlarından esinlenen popülasyon temelli metasezgisel bir algoritmadır. Mirjali ve arkadaşları tarafından 2014 yılında geliştirilen bu algoritma, gri kurt sürülerinin avlanma stratejilerini ve liderlik hiyerarşisini taklit ederek karmaşık optimizasyon problemlerine optimal veya optimale yakın çözümler bulmayı amaçlamaktadır (Mirjalili ve ark., 2014).

En uygun çözümlerin *alfa*, *beta* ve *delta* kurtları tarafından temsil edildiği ve geri kalan aday çözümlerin *omega* kurtları olarak kabul edildiği bir sürü kavramı üzerine geliştirilmiştir. Algoritma, gri kurtların avlanma davranışının matematiksel modellenmesi rehberliğinde bu konumları yinelemeli olarak günceller. Grup avlanma işlemi, kurtların toplumsal etkileşimlerinin yanı sıra gri kurtların ilginç bir sosyal davranışıdır. GWO'nun ana bölümleri avı çevreleme, avlanma ve avına saldırma aşamaları oluşturmaktadır (Ünlü ve İlhan, 2023).

Başlangıçta *alfa*, *beta* ve *delta* kurtları kendi konumlarına göre avı (potansiyel çözümleri) çevreler. Bu aşama, aday çözümlerin konumlarının en uygun çözümlere doğru ayarlanmasıyla matematiksel olarak temsil edilir. Avlanma aşamasında *alfa*, *beta* ve *delta* kurtları avın yakınlığına göre konumlarını günceller. *Omega* kurtları daha sonra konumlarını *alfa*, *beta* ve *delta* kurtlarına göre günceller, keşif ve sömürü arasındaki dengeyi korurken arama uzayını etkin bir şekilde keşfeder. Son aşamada, *alfa* kurdu avına saldırır ve optimum çözüme doğru yakınsamayı temsil eder. Bu aşama, keşif yarıçapının azaltılmasıyla modellenir ve algoritmanın bulunan en iyi çözüme yakınsamasını sağlar.

Gri kurt optimizasyon algoritmasının en büyük avantajlarından biri, basitliği ve kolay uygulanabilirliğidir. Diğer pek çok meta sezgisel algoritmanın aksine, minimum parametre ayarı gerektirir, bu da onu geniş bir optimizasyon problemleri yelpazesi için uygun kılar. Ayrıca, algoritmanın erken yakınsamayı önleme yeteneği ve arama uzayını verimli bir şekilde keşfetmesi, küresel optimumu bulmadaki başarısını artırır.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
xi ( $i := 1, 2, 3, \dots, n$ ) (gri kurt popülasyonunu oluştur)
a, A ve C değerlerini belirle
Her bir arama ajanının uygunluğunu hesapla
 $x\alpha =$  En iyi arama ajanı
 $x\beta =$  İkinci en iyi arama ajanı
 $x\delta =$  Üçüncü en iyi arama ajanı
t = 0
while (t < İterasyon sayısı)
  for i = 1 to n
     $D\alpha = |C1 * x\alpha - xi|$ 
     $D\beta = |C2 * x\beta - xi|$ 
     $D\delta = |C3 * x\delta - xi|$ 
     $X1 = x\alpha - A1 * D\alpha$ 
     $X2 = x\beta - A2 * D\beta$ 
     $X3 = x\delta - A3 * D\delta$ 
     $xi = (X1 + X2 + X3) / 3$  (Yeni pozisyonu güncelle)
  end for
  En iyi üç kurdu ( $\alpha, \beta, \delta$ ) popülasyondan güncelle
  t = t + 1
end while
Sonuçları döndür

```

Şekil 10. GWO'nun sözde kodu

#### 4.2.6. Harmoni Arama Algoritması

Harmoni Arama Algoritması (HS), müzikte hoş bir armoninin bir optimizasyon problemine en uygun çözümü bulmaya benzediği gözlemine dayanmaktadır (Geem ve ark., 2001). Tıpkı müzisyenlerin daha iyi bir uyum elde etmek için doğaçlama yapmaları ve ses perdelerini ayarlamaları gibi HS algoritması da arama uzayını yinelemeli olarak araştırır ve optimum çözümü

bulmak için çözüm vektörünü ayarlar. Geem ve arkadaşları tarafından 2001 yılında geliştirilen HS algoritması, çok çeşitli gerçek dünya optimizasyon problemlerine başarıyla uygulanmış bir metasezgisel optimizasyon tekniğidir (Oliva ve ark., 2013).

Algoritma, belirli sayıda çözüm vektörünü (harmoni) depolayan bir matris olan Harmoni Belleği (HM) ile başlar. Bu çözüm vektörleri optimizasyon probleminin arama uzayı içinde rastgele oluşturulur. Her iterasyonda, harmoni bellek değerlendirmesi, ton ayarı ve rastgele seçim olmak üzere üç operatör uygulanarak yeni bir armoni (çözüm vektörü) oluşturulur. Bu operatörler arama uzayını keşfetmekten ve çözüm vektörlerini ayarlamaktan sorumludur. Harmoni Belleği Dikkate Alma Oranı (Harmony Memory Considering Rate - HMCR) adı verilen bir olasılığa dayalı olarak HM'den bir veya daha fazla çözüm vektörü seçer. Seçilen çözüm vektörleri yeni armoniye katkıda bulunur. Arama uzayını daha verimli bir şekilde keşfetmek için, algoritma önceki adımda seçilen çözüm vektörlerinin tonunu ayarlayabilir. Bu, Ton Ayarlama Oranı (Pitch Adjusting Rate - PAR) adı verilen bir olasılığa ve bir bant genişliği parametresine dayalı olarak değerleri hafifçe değiştirerek yapılır. HM'deki çözüm vektörlerinden hiçbiri seçilmezse, algoritma yeni armoninin ilgili elemanı için rastgele yeni bir değer üretir. Yeni uyum oluşturulduktan sonra, uygunluğu (amaç fonksiyonu değeri) değerlendirilir. Yeni uyum HM'deki en kötü uyumdan daha iyi bir uygunluğa sahipse HM'deki en kötü uyumun yerini alır. Bu süreç, HM'nin sürekli olarak daha iyi çözümlere doğru evrimleşmesini sağlar. Algoritma, maksimum iterasyon sayısı veya istenen doğruluk seviyesi gibi bir sonlandırma kriteri karşılanana kadar iterasyon tekrarlanır.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
HM: Harmoni Belleği (başlangıç çözümlerini içerir)
HMS: Harmoni Belleği Boyutu
HMCR: Harmoni Belleği Dikkate Alma Oranı
PAR: Ton Ayarlama Oranı
bw: Bant Genişliği
Harmoni Belleği (HM) içindeki başlangıç çözümlerini rastgele üret ve değerlendir
t = 0
while (t < İterasyon sayısı)
  Yeni Çözüm (x_new) = []
  for i = 1 to d (Her karar değişkeni için)
    if (rand(0, 1) < HMCR)
      x_new[i] = HM içinden rastgele bir çözümün i. bileşeni
      if (rand(0, 1) < PAR)
        x_new[i] = x_new[i] + rand(-1, 1) * bw (Adım ayarlaması)
      end if
    else
      x_new[i] = Rastgele belirlenen çözüm (karar değişkeni aralığında)
    end if
  end for
  if (f(x_new) < en kötü çözüm HM'de)
    En kötü çözümü x_new ile değiştir (HM'yi güncelle)
  end if
  t = t + 1
end while
Sonuçları döndür

```

Şekil 11. HS'nin sözde kodu

#### 4.2.7. Klonal Seçim Algoritması

Klonal seçim algoritması (CLONal selection ALGORITHM – CLONALG), insan bağışıklık sisteminin biyolojik ilkelerinden esinlenen güçlü bir hesaplama tekniğidir. De Castro ve Von Zuben tarafından 2002 yılında geliştirilen bu algoritma, örüntü tanıma, optimizasyon ve makine öğrenimi dahil olmak üzere çeşitli alanlarda çok sayıda uygulama bulmuştur (De Castro ve Von Zuben, 2002).

Bağışıklık sistemi, vücudumuzu virüsler, bakteriler ve diğer yabancı maddeler gibi zararlı antijenik uyarılara karşı koruyan bir savunma mekanizmasıdır. Bir antijen vücudu istila ettiğinde, bağışıklık sistemi bu yabancı maddeleri tanıyabilen ve etkisiz hale getirebilen antikorlar üretir. Klonal seçim teorisinde, bir antikor bir antijenle karşılaştığında, antikorun birden fazla kopyasının üretildiği klonal genişlemeye uğrar. Bu kopyalar daha sonra antikorun genetik koduna rastgele değişikliklerin eklendiği bir süreç olan somatik hipermutasyona uğrar. Sonuç olarak, bu mutasyona uğramış antikorların bazıları antijen için daha yüksek bir afiniteye (bağlanma gücü) sahip olabilir ve bu da onları tehdidi etkisiz hale getirmede daha etkili hale getirir.

CLONALG, biyolojik süreci taklit eden bir hesaplama algoritmasıdır. Bağışıklık sistemindeki antikorlara benzeyen aday çözümlerden oluşan bir popülasyon üzerinde çalışır. Bu aday çözümler, verilen problemi çözümedeki etkinliklerini ölçen bir uygunluk fonksiyonu kullanılarak değerlendirilir.

CLONALG, rastgele aday çözümlerden oluşan bir başlangıç popülasyonu oluşturulması ile başlar. Verilen uygunluk fonksiyonunu kullanarak her aday çözümün uygunluğunu değerlendirir. Uygunluk değerlerine göre en iyi adaylar (antikorları) seçilir. Seçilen bu adaylar klonal genişlemeye tabi tutulacaktır. Seçilen adayların uygunluk değerleriyle orantılı olarak birden fazla kopyası (klon) oluşturulur. Daha yüksek uygunluk değerlerine sahip adayların daha fazla klonu olacaktır. Uygunluklarını iyileştirmek amacıyla klonlara rastgele değişiklikler (mutasyonlar) eklenir. Mutasyon oranı tipik olarak uygunluk değeriyle ters orantılıdır, yani daha uygun klonlar daha az mutasyona uğrayacaktır. Mutasyona uğramış klonların uygunluğu değerlendirilir ve yeni popülasyonu oluşturmak için en iyileri seçilir. Sonlandırma kriterlerinin karşılanıp karşılanmadığı kontrol edilerek iterasyona devam edilir. Klonal genişleme, somatik hipermutasyon ve seçim ilkelerini birleştirerek aday çözüm popülasyonunu iteratif olarak iyileştirir ve sonuçta optimal veya optimale yakın bir çözüme yakınsar.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
xi ( $i := 1, 2, 3, \dots, n$ ) (Her bireyin başlangıç değerlerini belirle)
MO: mutasyon oranı
CO: klonlama oranı
r: en iyi antikor sayısı
t = 0
while (t < İterasyon sayısı)
    Değerlendir: Her bireyin uygunluğunu hesapla
    Sırala: Bireyleri uygunluk değerlerine göre sırala
    for i = 1 to r
        Klonla: Seçilen bireyi CO kadar klonla
        Klonların Uygunluğunu Değerlendir: Klonları MO kadar mutasyona uğrat
        for each klon
            if (rand(0, 1) < MO)
                klon = Mutasyon(klon)
            end if
        end for
        Yeni Bireyler Ekle: Klonları popülasyona ekle
    end for
    Yeni Popülasyon Oluştur: En iyi bireyleri seçerek yeni popülasyonu oluştur
    Çeşitlilik Sağla: Rastgele yeni bireyler ekle (çeşitliliği sağlamak için)
    for i = 1 to n - r
        yeni_birey = Rastgele yeni bir birey oluştur
        Popülasyona yeni_birey ekle
    end for
    t = t + 1
end while
Sonuçları döndür

```

Şekil 12. CLONALG'nun sözde kodu

#### 4.2.8. Parçacık Sürü Optimizasyon Algoritması

Parçacık Sürü Optimizasyon Algoritması (PSO), ilk olarak Kennedy ve Eberhart (1995) tarafından geliştirilmiş popülasyon temelli sezgisel bir optimizasyon tekniğidir (Kennedy ve Eberhart, 1995). Kuş veya balık sürülerinin kolektif davranışlarından esinlenilerek geliştirilmiştir. Temel olarak sürüde bulunan bireylerin pozisyonunun, sürünün en iyi pozisyona sahip olan bireyine yaklaştırılması ilkesine dayanır. Çok boyutlu arama uzaylarında optimum çözümler bulmak için sürü zekasının gücünden yararlanır (Mozaffari ve Lee, 2017).

PSO algoritması temelinde, her biri amaç problemine potansiyel bir çözümü temsil eden bir parçacık sürüsünün hareketini simüle eder. Bu parçacıklar, kendi deneyimlerine ve sürüdeki komşularının deneyimlerine dayanarak konumlarını ve hızlarını ayarlayarak arama uzayında dolaşırlar.

PSO, arama uzayı içinde rastgele konumlara ve hızlara sahip bir parçacık sürüsünü başlatarak başlar. Her parçacık, o ana kadar en iyi çözümü bulduğu konum olan kişisel en iyi konumunu takip eder. Ek olarak, sürü bir bütün olarak sürüdeki herhangi bir parçacık tarafından keşfedilen genel en iyi çözümü temsil eden küresel bir en iyi konumu korur. Yinelemeli optimizasyon süreci sırasında, parçacıklar hızlarını ve konumlarını iki temel faktöre göre günceller. Bunlar bireysel deneyimleri (kişisel en iyi) ve sürünün kolektif bilgisi (küresel en iyi). Hız güncelleme denklemi üç bileşen içerir: atalet, bilişsel etki ve sosyal etki. Atalet, parçacıkların mevcut hızlarının bir kısmını korumalarını sağlayarak, önceki hareketlerinden bir miktar momentumu korurken yeni bölgeleri keşfetmelerine olanak tanır. Bilişsel etki, parçacıkları bireysel deneyimlerinden ve yerel olarak optimize edilmiş çözümlerden yararlanarak kişisel en iyi konumlarına doğru hareket etmeye teşvik eder. Sosyal etki, parçacıkları küresel en iyi konuma doğru çekerek sürü üyeleri arasında bilgi paylaşımını ve iş birliğini kolaylaştırır. Bu bileşenleri dikkatlice dengeleyerek, parçacıklar hem bireysel hem de kolektif bilgiden yararlanarak umut verici bölgelere doğru yaklaşırken arama uzayını etkili bir şekilde keşfedebilir.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
Popülasyonun başlangıç parçacıklarını üret
 $p_i$  ( $i := 1, 2, 3, \dots, n$ ) (Her parçacığın başlangıç pozisyonlarını belirle)
 $v_i$  ( $i := 1, 2, 3, \dots, n$ ) (Her parçacığın başlangıç hızlarını belirle)
 $p_{best} = p_i$  (Her parçacığın kendi en iyi pozisyonu)
 $g_{best} =$  En iyi  $p_{best}$  (Tüm parçacıklar arasındaki en iyi pozisyonu)
 $w$ : atalet ağırlığı
 $c1$ : bilişsel (bireysel) katsayı
 $c2$ : sosyal katsayı
 $t = 0$ 
while ( $t <$  İterasyon sayısı)
  for  $i = 1$  to  $n$ 
     $v[i] = w * v[i] + c1 * rand(0, 1) * (p_{best}[i] - p[i]) + c2 * rand(0, 1) * (g_{best} - p[i])$ 
     $p[i] = p[i] + v[i]$  (Parçacığın pozisyonunu güncelle)
    if ( $f(p[i]) < f(p_{best}[i])$ )
       $p_{best}[i] = p[i]$  (Kendi en iyi pozisyonunu güncelle)
    end if
    if ( $f(p_{best}[i]) < f(g_{best})$ )
       $g_{best} = p_{best}[i]$  (Global en iyi pozisyonu güncelle)
    end if
  end for
   $t = t + 1$ 
end while
Sonuçları döndür

```

Şekil 13. PSO'nun sözde kodu

#### 4.2.9. Yapay Arı Kolonisi Algoritması

Yapay Arı Kolonisi (ABC) algoritması, bal arılarının yiyecek arama davranışını taklit eden sürü tabanlı bir metasezgisel algoritmadır. Derviş Karaboğa tarafından 2005 yılında önerilen bu

algoritma, çeşitli alanlardaki karmaşık optimizasyon problemlerini çözümedeki basitliği ve verimliliği nedeniyle büyük ilgi görmüştür (Karaboga, 2005).

Gerçek bir arı kolonisinde, işçi arılar, gözcü arılar ve kâşif arılar olmak üzere üç tür arı çeşidi vardır. İşçi arılar halihazırda keşfettikleri besin kaynaklarını kovana getirmekten ve gittikleri kaynağın kalitesi ve yerleri ile ilgili bilgileri diğer arılarla paylaşmaktan sorumludur. Gözcü arılar, işçi arılar tarafından paylaşılan bilgilere dayanarak gıda kaynaklarını seçer ve böylece yeni gelecek vaat eden bölgelerin keşfini kolaylaştırır. Kâşif arılar ise mevcut gıda kaynakları terk edildiğinde rastgele yeni gıda kaynakları arar (Akay, 2013).

ABC algoritması, optimizasyon problemlerini çözmek için bal arılarının bu akıllı yiyecek arama davranışını simüle eder. Arama uzayı içerisinde rastgele oluşturulan gıda kaynaklarından (çözümler) oluşan bir başlangıç popülasyonu ile başlar. Her bir besin kaynağı problemin olası bir çözümünü temsil eder ve uygunluk fonksiyonu kullanılarak değerlendirilir. Bu değer kaynaktaki nektar karşılığına denk gelir. Her çalışan arı bir besin kaynağına atanır ve mevcut besin kaynağının komşuluğunda yeni bir aday çözüm üretir. Bu çözümün uygunluğu mevcut çözümden daha iyiyse, istihdam edilen arı besin kaynağını yeni aday çözümle günceller ve başarısızlık sayacını sıfırlar. Daha kötü olduğu durumda başarısızlık sayacını artırır ve mevcut çözümü korur. Tüm çalışan arılar aramalarını tamamladıktan sonra, besin kaynaklarının kalitesi hakkındaki bilgileri gözcü arılarla paylaşırlar. Gözcü arılar, daha iyi gıda kaynaklarının seçilme olasılığı daha yüksek olacak şekilde, gıda kaynaklarını uygunluk değerlerine göre olasılıksal olarak seçerler. Bu arılar istihdam edilen arı aşamasına benzer şekilde, seçilen gıda kaynaklarının komşuluğunda yeni aday çözümler üretir. Yeni bir aday çözüm mevcut çözümden daha iyi bir uygunluğa sahipse, besin kaynağı güncellenir ve başaramama sayacı sıfırlanır. Kötü olduğu durumda ise başaramama sayacı artırılır. Rastgele yeni bir besin kaynağı oluşturmak için bir kâşif arı gönderilir. Kâşif arı başaramama sayacı önceden belirlenen limit değerinden yüksekse rastgele yeni bir değer üretir. Bu işlem arama uzayında yeni bölgelerin keşfedilmesini kolaylaştırır. Algoritma, maksimum iterasyon sayısına ulaşmak veya istenen uygunluk seviyesini karşılayan bir çözüm bulmak gibi bir durdurma kriteri karşılanana kadar bu aşamalar boyunca iterasyon yapar.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
xi (i := 1, 2, 3, ..., n) (Her bireyin başlangıç değerlerini belirle)
sınır değerini belirle ve uygunluk_degerleri (xi) hesapla
t = 0
while (t < İterasyon sayısı)
  for i = 1 to n/2 (İşçi Arı fazı)
    Yeni Çözüm (vi) üret: vi = xi + φ * (xi - xk) (φ: rastgele sayı, k: rastgele indeks)
    if (f(vi) < f(xi))
      xi = vi (Yeni çözümü kabul et)
      sayaç_i (Başarısızlık sayaç değerini belirle)
    end if
  end for
  for i = 1 to n/2 (Gözcü Arı fazı)
    pi = Gözcü arıların olasılık değerlerini hesapla
    if (f(vi) < f(xi))
      xi = vi (Yeni çözümü kabul et)
      sayaç_i (Başarısızlık sayaç değerini belirle)
    end if
  end for
  for i = 1 to n/2 (Kaşif Arı fazı)
    if (uyum_i > sınır)
      Rastgele yeni çözüm üret ve xi ile değiştir
      sayaç_i = 0 (Başarısızlık sayacını sıfırla)
    end if
  end for
  t = t + 1
end while
Sonuçları döndür

```

Şekil 14. ABC'nın sözde kodu

### 4.3. Önerilen Hibrit Optimizasyon Algoritması

Optimizasyon algoritmaları ile yapılan deneylerde en iyi sonuçları gri kurt optimizasyon algoritmasının verdiği görülmüştür. Ancak, eşik sayısı arttıkça gri kurt optimizasyon algoritmasının yerel minimumlardan kaçmakta yetersiz kaldığı ve en iyi üç kurt üzerinden pozisyon güncelleme yapısının çözüm uzayı büyüdükçe yeterince iyi sonuç vermediği anlaşılmıştır. Gri kurt optimizasyon algoritmasındaki bu problemleri en aza indirmek için harmoni arama algoritmasının ton ayarlama süreci bu algoritmaya dahil edilmiş, GWO-HS olarak isimlendirilen hibrit bir algoritma önerilmiştir.

GWO-HS'nin başlangıcında, verilen popülasyon büyüklüğü değeri kadar (görüntü eşikleme problemi için konum değerleri) sınırları 0 ile 255 arasında değerler olacak şekilde rastgele konum matrisi üretilir. Bu matrisin satır sayısı popülasyon miktarına, sütunları ise eşik sayısına eşit olacaktır. Bu matriste her satır bizim için bir arama ajanı olarak kabul edilmektedir. Her bir arama ajanı için verilmiş olan amaç fonksiyon değeri hesaplanır. En iyi  $k$  uygunluk değerine sahip arama ajanı  $(x_1, x_2, \dots, x_k)$  tanımlanır. Her bir arama ajanının konumları  $x_1, x_2, \dots, x_k$  ajanlarına göre güncellenir. Bu sayede diğer arama ajanları, çözüm uzayını etkin bir şekilde keşfeder. Tüm arama ajanları için uygunluk değeri hesaplanır ve en iyi  $k$  tane sonuç güncellenir. HS'de ton ayarlama sürecinden farklı olarak sadece  $x_1, x_2, \dots, x_k$  ajanlarına ton ayarlama işlemi uygulanır. Böylece en iyi sonucun yakın çevrelerinde olabilecek daha iyi çözümler aranır ve daha iyi çözüm varsa mevcut çözüm güncellenir. Her iterasyonun sonunda bant genişliği değeri bant genişlik faktörü ile çarpılarak azaltılır. Bu sayede iterasyonun başında daha uzak noktadaki komşular aranırken iterasyon sonuna doğru daha yakın komşu çözümler aranmaktadır.

GWO-HS algoritmasında kullanılan 3 parametre vardır. Bunlar  $a$ ,  $k$  ve  $bw$  parametreleridir.  $a$  parametresi en iyi çözüm havuzunun mesafesini kontrol eden bir parametredir. Bu parametre değeri iterasyon boyunca lineer olarak sıfıra doğru azalır.  $k$  parametresi en iyi çözümlerin tutulduğu havuzun boyutunu belirtmektedir.  $bw$  parametresi ile bant genişlik faktörü  $(2/bw)^{1/(IT-1)}$  eşitliği kullanılarak hesaplanır. İterasyon boyunca  $bw$  değeri bant genişliği faktörü değeri ile çarpılarak azaltılır. Bu değer ile en iyi çözüm havuzundaki değerlerin lokal minimumdan kurtulmaları sağlanır.

Eşik sayısı arttıkça ton ayarlama işleminin ve sürüdeki tüm ajanların konumlarını etkileyecek olan  $k$  tane en iyi sonuca sahip ajanların etkisi daha çok ortaya çıkmaktadır. Düşük

eşik sayılarında GWO ile hibrit algoritma benzer sonuçlar verirken eşik sayısı arttıkça hibrit algoritma GWO ve diğer optimizasyon algoritmalarından daha iyi sonuçlar üretir.

```

f(x) amaç fonksiyonunu belirle  $x = (x_1, x_2, \dots, x_d)^T$ 
xi (i := 1, 2, 3, \dots, n) (gri kurt popülasyonunu oluştur)
a, k ve bw değerini belirle
bwf = (2/bw)1/(IN-1) bant genişliği faktörünü tanımla
x1, x2, \dots, xk En iyi değerleri büyükten küçüğe olacak şekilde ata
t = 0
while (t < İterasyon sayısı)
    foreach xi in x
        Arama ajanlarının pozisyonlarını güncelle
    end for
    a, A ve C değerlerini güncelle
    Arama ajanlarının amaç fonksiyon değerlerini hesapla
    x1, x2, \dots, xk en iyi ilk k değer olacak şekilde güncelle
    for i to 10
         $x_{1,2,\dots,k}^{new} = x_{1,2,\dots,k}^{new} \pm rand(0, 1) \cdot bw$ 
        if  $x_{1,2,\dots,k}^{new} \cdot IsBetter(x_{1,2,\dots,k})$ 
             $x_{1,2,\dots,k} = x_{1,2,\dots,k}^{new}$ 
        end if
    end for
     $bw = bw * bwf$ 
     $t = t + 1$ 
end while
Sonuçları döndür

```

Şekil 15. GWO-HS algoritmasının sözde kodu

#### 4.4 Performans Karşılaştırma Metrikleri

Bu çalışmada elde edilen sonuçları karşılaştırabilmek için farklı metrikler kullanılmıştır. Bu metrikler amaç fonksiyon değeri, PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Index), ortalama değer ve standart sapmadır.

##### 4.4.1. Amaç Fonksiyon Değeri

Görüntü eşikleme probleminde eşik değerini bulmak için geliştirilen yöntemler bir amaç fonksiyonunu maksimize ya da minimize ederek en optimum çözümü bulmayı amaçlarlar. Bu çalışmada Kapur, Otsu ve Tsallis yöntemlerinin amaç fonksiyon değerleri maksimize edilerek optimum eşik değerleri bulunmaya çalışılmıştır. Her bir yöntemin maksimum değeri o algoritma için en iyi sonuç olarak kabul edilmektedir. Optimizasyon algoritmaları eşik değeri bulma sürecinin maliyetini azaltmak için kullanılmıştır. Amaç fonksiyonları olarak Kapur, Otsu ve Tsallis algoritmalarının çok seviyeye göre genişletilmiş fonksiyonları verilmiştir.

##### 4.4.2. PSNR Değeri

PSNR, dijital görüntü ve videoların kalitesini değerlendirmek için literatürde yaygın olarak kullanılan bir ölçüttür. PSNR, iki görüntü arasındaki farkı (genellikle orijinal görüntü ile sıkıştırılmış görüntü arasındaki farkı) nicel olarak ifade eder. Bu hesaplama, bir görüntünün en yüksek piksel değeri ile iki görüntü arasındaki MSE (Mean Squared Error) değerinin oranını dB (desibel) cinsinden ifade eder.

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - k(i, j)]^2 \quad (4.1)$$

MSE değeri eşitlik 4.1 ile tanımlanır. Burada  $m$  ve  $n$  görüntünün boyutlarını (genişlik ve yükseklik)  $I$  orijinal görüntüyü,  $K$  bozulmuş veya eşiklenmiş görüntüyü temsil eder.

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX^2}{MSE} \right) \quad (4.2)$$

PSNR, eşitlik 4.2 kullanılarak hesaplanır. MAX görüntüdeki maksimum piksel değeri ifade eder. 8 bit görüntüler için bu değer 255'tir.

PSNR değerinin yüksek olması eşiklenmiş görüntünün performansının yüksek olduğunu gösterir.

#### 4.4.3. SSIM Değeri

SSIM, iki görüntü arasındaki yapısal benzerliği ölçmek ve kalitesini tahmin etmek için geliştirilmiş bir yöntemdir. SSIM, insan görsel algısının dikkate aldığı yapısal bilgiyi ölçer ve bu nedenle iki görüntü arasındaki yapısal benzerliği değerlendirir. SSIM değeri 0 ile 1 arasında değişir. 1 değeri iki görüntünün tamamen aynı olduğunu, 0 ise tamamen farklı olduğunu gösterir.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.3)$$

SSIM ölçütü eşitlik 4.3 kullanılarak hesaplanır.  $\mu_x$  ve  $\mu_y$  sırasıyla  $x$  ve  $y$  görüntülerinin ortalama gri seviye değerleridir.  $\sigma_x$  ve  $\sigma_y$   $x$  ve  $y$  görüntülerinin standart sapmalarıdır.  $\sigma_{xy}$ ,  $x$  ve  $y$  arasındaki yerel korelasyon katsayısı,  $c_1$  ve  $c_2$  değerleri  $c_1 = c_2 = 0.065$  olan sabitlerdir.

SSIM değerinin yüksek olması eşiklenmiş görüntünün yapısal olarak daha kaliteli olduğunu gösterir.

#### 4.4.4. Standart Sapma ve Ortalama Değer

Optimizasyon algoritmaları rastgelelik üzerine çalıştığı için her çalıştırmada aynı sonucu verememektedir. Sonuçların hesaplamasında bir dizi deney yapılmıştır. Algoritmaların çözüme ulaşmada doğruluğunu ve kararlılığını ölçmek için deney sonuçlarının ortalama değerleri ve standart sapması alınmıştır. Ortalama değer 4.4 eşitliği kullanılarak standart sapma ise 4.5 eşitliği kullanılarak hesaplanır.

$$\mu = \frac{\sum_{i=1}^k S_i}{k} \quad (4.4)$$

$$std = \sqrt{\frac{1}{k} \sum_{i=1}^k (S_i - \mu)^2} \quad (4.5)$$

## 5. ARAŞTIRMA BULGULARI VE TARTIŞMA

### 5.1. Web Tabanlı Aracın Kullanımı

Tek bir optimizasyon algoritması, her problemde en iyi sonucu verememektedir. Bu yüzden, her problemin literatürde olan birçok optimizasyon algoritması ile test edilmesi gerekmektedir. Bu çalışma, araştırmacıların kendi görsel kalıplarını eşikleme işlemlerinde en uygun optimizasyon algoritmasını bulmalarına kolaylık sağlaması amacıyla geliştirilmiştir. Araştırmacılar, kendi görsellerini Kapur, Otsu ve Tsallis eşikleme algoritmalarıyla da test edebilirler. Geliştirdiğimiz arayüzün araştırmacılar için bir diğer önemli faydası, görsellerini iki ile on eşik arasında eşikleyebilmeleridir. Yapılan eşikleme sonuçlarını karşılaştırarak kendi çalışmalarını için optimum sonucu kısa sürede bulabilirler.

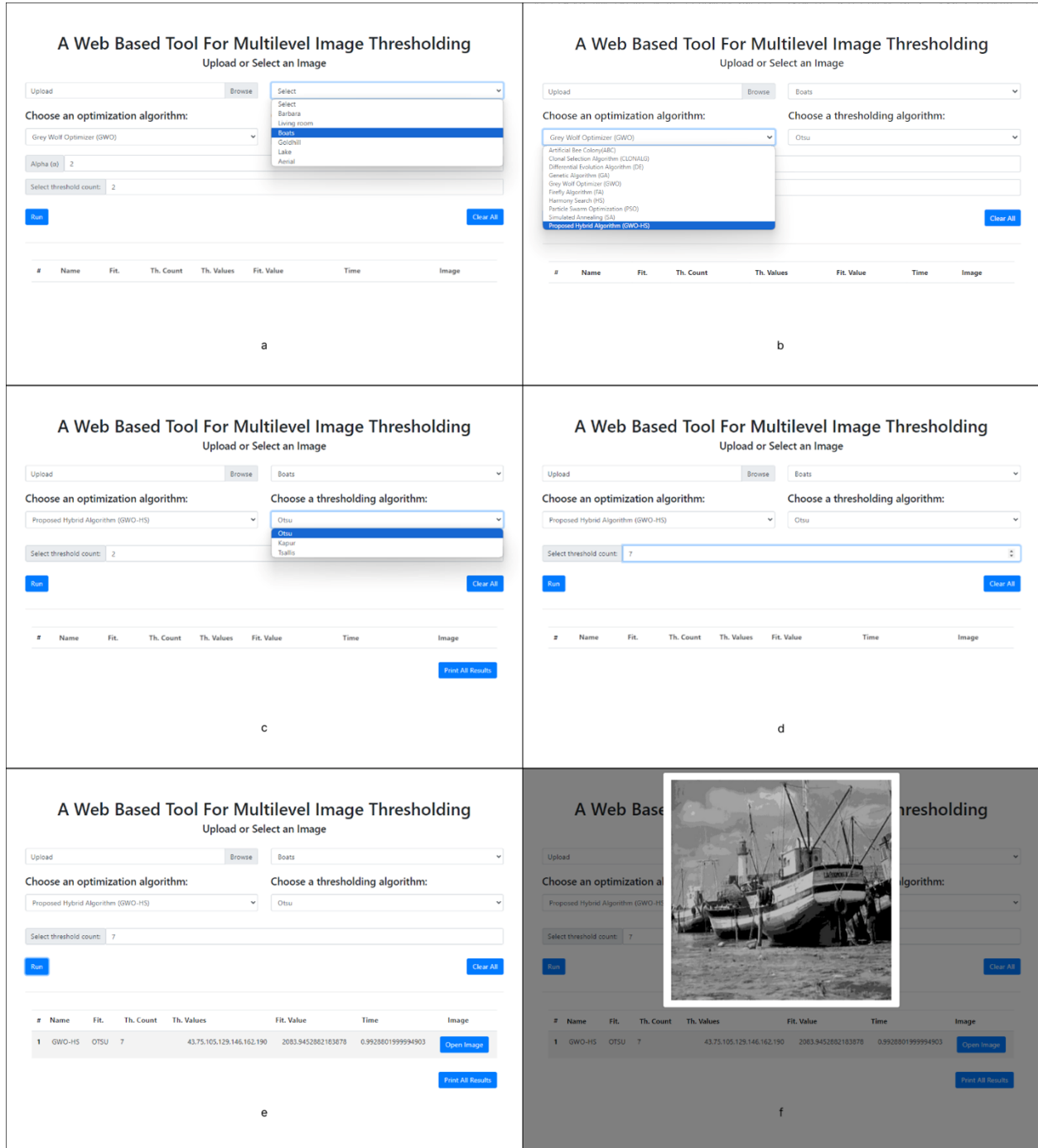
Web tabanlı arayüzün kullanımı oldukça basit tutulmuştur. Arayüzde her işlem için bir işlem sırası gerekmektedir. Bu işlem sırasındaki hatalar için kullanıcı uyarılarak olası sorunların önlenmesi amaçlanmıştır.

Geliştirilen arayüzün kullanım aşamaları Şekil 16'da gösterilmiştir. Öncelikle arayüze test edilmesi gereken görselin yüklenmesi gerekmektedir. "Upload" butonuna tıklayarak görsel yükleme penceresi açılmaktadır. Bu pencereden yerel bilgisayarınızda olan herhangi bir görsel seçilerek sisteme yüklenmiş olur. Yerel bilgisayardan görsel yüklemek istemeyen araştırmacılar için literatürde yaygın olarak kullanılan test görsellerinin isimlerinden herhangi birini seçerek devam edilebilir. Şekil 16b'de gösterilen algoritma seçim ekranında çalıştırılması istenilen optimizasyon algoritması seçilebilmektedir. Burada on farklı algoritma mevcuttur. Bu algoritmalar sırasıyla Yapay Arı Koloni Algoritması (ABC), Klonal Seçim Algoritması (CLONALG), Diferansiyel Gelişim Algoritması (DE), Genetik Algoritma (GA), Gri Kurt Optimizasyon Algoritması (GWO), Ateş Böceği Algoritması (FA), Harmoni Arama Algoritması (HS), Parçacık Sürü Optimizasyon Algoritması (PSO), Benzetilmiş Tavlama Algoritması (SA) ve önerilen hibrit algoritma (GWO-HS) şeklindedir.

Optimizasyon algoritmasının seçiminin ardından Şekil 16c'de gösterilen amaç fonksiyonu olarak kullanılan görüntü eşikleme algoritması seçilir. Eşikleme algoritması olarak Kapur, Otsu ve Tsallis algoritmalarından herhangi biri seçilebilmektedir. Son aşama olarak da istenen eşik sayısı Şekil 16d'de görüldüğü gibi seçilebilir. Eşik sayısı minimum iki, maksimum on olacak şekilde sınırlandırılmıştır. Tüm seçimler bittikten sonra "Run" butonuna basıldıktan sonra hesaplamalar

server tarafına iletilerek yapılır. Şekil 16e’de görüldüğü gibi sonuçlar ekranın altında tablo şeklinde sıralanmaktadır. Tabloda sırasıyla kullanılan algoritmanın ismi, amaç fonksiyonu, eşik sayısı, eşik değerleri, amaç fonksiyonunun sonucu, hesaplama süresi ve görselin eşiklenmiş hali bulunmaktadır. “Open Image” butonu ile eşiklenmiş görüntü açılıp Şekil 16f’deki gibi incelenebilmektedir.

Arayüz üzerinde optimizasyon algoritmalarının parametre ayar kısımları bulunmaktadır. Her optimizasyon algoritması seçiminden sonra ilgili algoritmanın parametre ekranı açılmaktadır. Burada yapılan değişiklikler sonuçların tamamen değişmesine ve algoritma başarımlarında farklılıklara neden olabilmektedir. Buradaki veriler parametre optimizasyonu yapılarak her bir optimizasyon algoritmasının test görseli üzerinde verdiği en iyi sonuçlar ele alınarak belirlenmiştir.



**Şekil 16.** Geliştirilen arayüzün işlem basamakları (a, b, c, d, e, f)

Web arayüz her yenilendiğinde önceki yapılan parametre ayarlarının hepsi sıfırlanmaktadır. Hesaplanan veriler 30 gün boyunca tarayıcının yerel belleğinde kaydedilmektedir. Yapılan hesapların silinmesi için “Clear All” butonuna tıklanması yeterli olmaktadır.

Şekil 17’de, arayüz üzerinde yapılan bir dizi deney sonuçları görülmektedir. Bu deney “Boats” isimli test görseli kullanılarak yapılmıştır. Görüntü eşikleme algoritması olarak Otsu algoritması seçilmiş ve 7 eşikleme için tüm optimizasyon algoritmaları tek tek çalıştırılmıştır. Yeni hesaplama yapmak için bir önceki hesaplamanın bitmesini beklemeden sürekli bir hesaplama gönderilebilmektedir. Hesaplanan veriler arayüzde tablo şeklinde sıralanmaktadır.

## A Web Based Tool For Multilevel Image Thresholding

Upload or Select an Image

Upload

Browse

Boats

Choose an optimization algorithm:

Proposed Hybrid Algorithm (GWO-HS)

Choose a thresholding algorithm:

Otsu

Select threshold count:

Run

Clear All

#	Name	Fit.	Th. Count	Th. Values	Fit. Value	Time	Image
1	FA	OTSU	7	52,75,88,92,133,155,181	2055.4056640078807	0.6386925999995583	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
2	ABC	OTSU	7	8,34,61,114,142,159,207	2048.5286916061173	0.6651274999985617	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
3	CLONALG	OTSU	7	58,87,112,130,151,175,205	2074.107529595514	0.9589425000012852	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
4	GA	OTSU	7	43,63,89,113,171,176,202	2066.0191326261174	0.720518699999569	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
5	DE	OTSU	7	68,81,105,136,154,170,216	2053.3258544427044	0.6598595999985264	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
6	GWO	OTSU	7	40,57,72,103,132,153,192	2069.0001742920313	0.8041614000012487	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
7	HS	OTSU	7	48,91,109,123,152,186,196	2059.188074509387	0.0139870000002702	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
8	PSO	OTSU	7	108,114,128,130,133,135,154	1839.5788079540628	0.36650120000012976	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
9	SA	OTSU	7	39,84,100,138,161,168,203	2059.371745109735	0.7797929999996995	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>
10	GWO-HS	OTSU	7	40,71,103,128,147,164,192	2084.2458378916745	0.9924887999986822	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">Open Image</a>

[Print All Results](#)

Şekil 17. Tüm eşikleme algoritmaları ile yapılan bir dizi deney

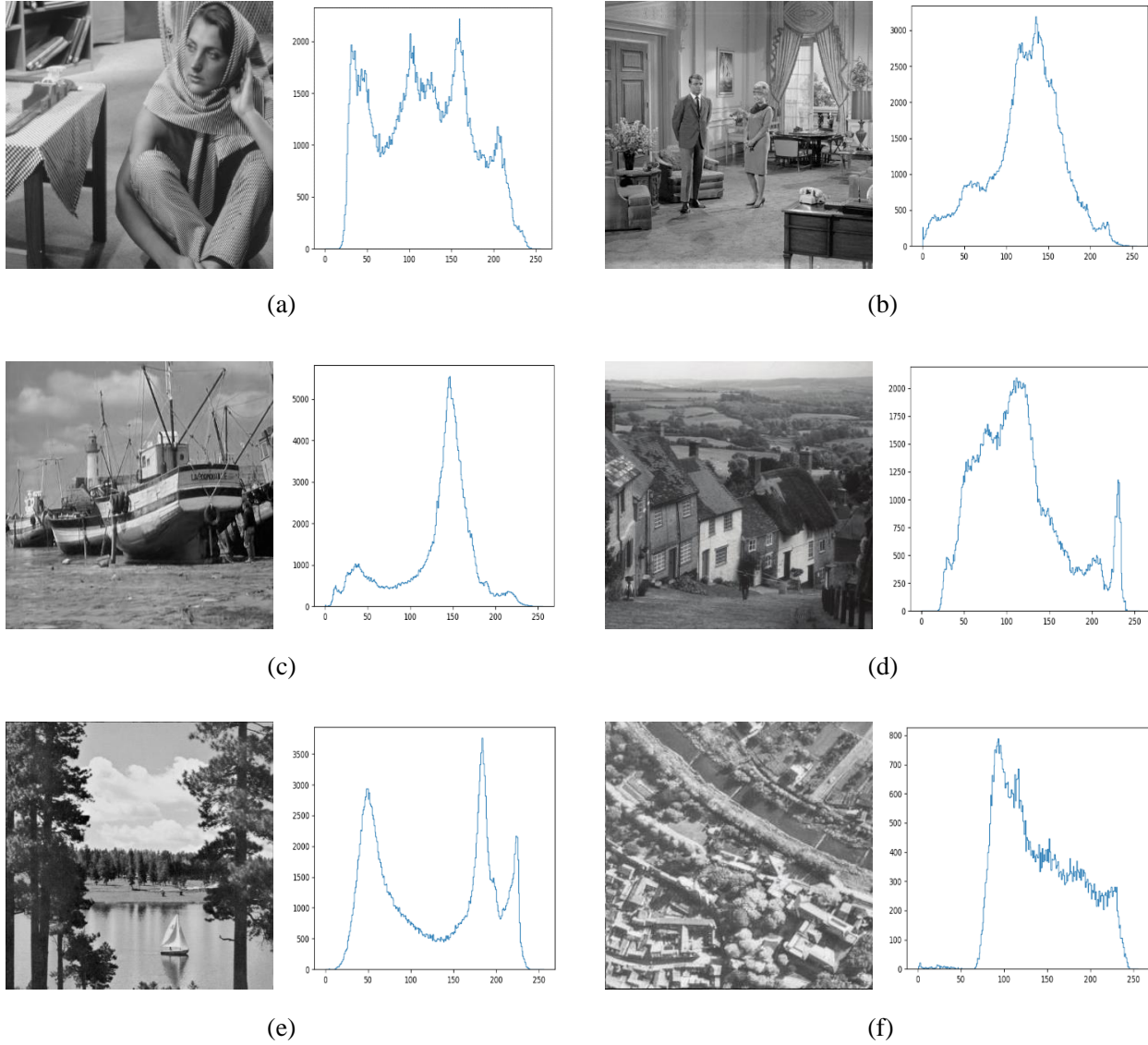
Arayüz üzerinde “Print All Results” butonu ile yapılan deneylerin sonuçları yerel bilgisayara indirilebilmektedir. Şekil 17’de hesaplanan verilerin çıktısı arayüz aracılığıyla indirilmiştir. Çizelge 1’de Excel ile açılmış gösterimi bulunmaktadır.

**Çizelge 1.** Arayüz tarafından hesaplanan kaydedilmiş veriler

Sıra	İsim	Eşikleme Yöntemi	Eşik Sayısı	Eşik Değerleri	Uygunluk Değeri	Süre
1	FA	Otsu	7	52-75-88-92-133-155-181	2055.405664	0.63826
2	ABC	Otsu	7	8-34-61-114-142-159-207	2048.528692	0.651275
3	CLONALG	Otsu	7	58-87-112-130-151-175-205	2074.107532	0.958425
4	GA	Otsu	7	43-68-89-113-171-176-202	2066.019133	0.7205187
5	DE	Otsu	7	68-81-105-136-154-170-216	2053.325854	0.6598596
6	GWO	Otsu	7	40-57-72-103-132-153-192	2069.000174	0.041614
7	HS	Otsu	7	48-91-109-123-152-186-196	2059.188075	0.0137
8	PSO	Otsu	7	108-114-128-130-133-135-154	1839.578808	0.565012
9	SA	Otsu	7	39-84-100-138-161-168-203	2059.371745	0.779793
10	GWO-HS	Otsu	7	40-71-103-128-147-164-192	2084.245838	0.9248

## 5.2. Optimizasyon Algoritmalarının Performanslarının Değerlendirilmesi

Bu çalışmada ABC, CLONALG, DE, GA, GWO, FA, HS, PSO, SA ve geliştirdiğimiz hibrit algoritmanın performansları, görüntü eşikleme için kullanılan Kapur, Otsu ve Tsallis algoritmaları kullanılarak test edilmiştir. Deneylerde görüntü işleme çalışmalarında yaygın olarak tercih edilen ve Şekil 18’de verilen “Barbara”, “Living room”, “Boats”, “Goldhill”, “Lake” ve “Aerial” görüntüleri kullanılmıştır. Bu görüntülerden “Aerial” 256x256 boyutunda iken diğerleri 512x512 boyutundadır.



Şekil 18. (a) Barbara, (b) Living room, (c) Boats, (d) Goldhill, (e) Lake, ve (f) Aerial. Görüntüleri ve histogramları

Adil bir hesaplama olması için tüm algoritmaların durdurma kriteri olarak 100 iterasyon sayısı belirlenmiştir. Her bir arama ajanı sayısı da 100 olarak alınmıştır. Deneylerde kullanılan algoritmaların parametreleri Çizelge 2’de verilmiştir. Bu değerler bir dizi ön deney yapılarak deneme yanılma yöntemi ile elde edilmiştir

Çizelge 2. Optimizasyon algoritmalarının parametre optimizasyonu

Algoritma	Parametreler
ABC	İterasyon sayısı: 100, Arı sayısı: 100, Deneme sayısı: 0.80, Yiyecek kaynağı sayısı: 0.30, Parametre sayısı:
CLONALG	İterasyon sayısı: 100, Antikor havuzu: 100, Klonlama Faktörü: 0.9, En iyi antikor sayısı: 0.7

DE	Iterasyon sayısı: 100, Antikor havuzu: 100, Çaprazlama oranı:0.7, Ölçekleme faktörü:0.5
GA	Iterasyon sayısı: 100, Popülasyon sayısı: 100, Çaprazlama oranı: 0.80, Mutasyon oranı: 0.30
GWO	Iterasyon sayısı: 100, Gri kurt Sayısı: 100, Mesafe kontrol parametresi a: Lineer azalma 2 den 0
FA	Iterasyon sayısı: 100, Popülasyon sayısı: 100, Alfa:0.4, Gama:0.4, Beta:0.2
HS	Iterasyon sayısı: 100, Harmoni bellek boyutu: 100, Harmoni belleği dikkate alma oranı: 0.95, Ton ayarlama oranı: 0.2, Bant genişliği: 4.5, Bant genişliği faktörü: 0.99
PSO	Iterasyon sayısı: 100, Parçacık sayısı: 100, Bilişsel katsayı: 2, Sosyal katsayı: 2, Atalet ağırlığı: 0.7,
SA	Iterasyon sayısı: 100, Deneme sayısı: 100, Başlangıç sıcaklığı: 10, Bitirme sıcaklığı: 2
GWO-HS	Iterasyon sayısı: 100, Gri kurt sayısı: 100, Mesafe kontrol parametresi a: Lineer azalma 3 den 0, En iyi çözüm sayısı k: 5 Bant genişliği: 4

GWO-HS algoritmasında üç farklı parametre bulunmaktadır. Her bir parametre için minimum ve maksimum değerler arasındaki tüm değerler için 100 bağımsız çalıştırma gerçekleştirilmiştir. Bu çalışmaların sonucunda ortalama en iyi değeri veren parametreler büyükten küçüğe doğru sıralanarak en iyi 10 sonucu veren parametre değerleri Çizelge 3'te paylaşılmıştır. Çizelgeden de görülebildiği gibi a, k ve bw parametrelerinin sırasıyla 3, 5 ve 4 değerleri için en iyi sonuçlar elde edilmiştir. Bu değerler GWO-HS algoritmasının parametre değerleri olarak kullanılmıştır.

**Çizelge 3.** GWO-HS algoritmasının parametre optimizasyon sonuçları

<b>a</b>	<b>k</b>	<b>bw</b>	<b>Ort</b>	<b>StdSap</b>
<b>3</b>	<b>5</b>	<b>4</b>	<b>32.55319598</b>	<b>0.051473096</b>
2	5	4	32.5499892	0.061194782
4	5	5	32.54974309	0.054859407
3	5	6	32.54660263	0.061762323
3	5	5	32.54484136	0.059192579
2	5	5	32.54458983	0.060308581
4	5	6	32.54437952	0.055298569
2	5	7	32.54386659	0.058936584
2	5	3	32.5427763	0.062633084
4	5	4	32.54247415	0.062617273

Tüm test görüntüleri için 2 ile 10 arasında değişen eşik seviyeleri için 50 bağımsız çalıştırma gerçekleştirilmiştir. Kapur eşikleme yöntemi için elde edilen sonuçlar Çizelge 4'te, Otsu

eşikleme yöntemi için elde edilen sonuçlar Çizelge 5’de ve Tsallis eşikleme yöntemi için elde edilen sonuçlar Çizelge 6’da verilmiştir. Bu çizelgelerde “K” eşik sayısı, “Ort” 50 sonucun ortalaması, “StdSap” ise 50 sonucun standart sapmasıdır. En iyi değerler kalın yazı karakteri ile ifade edilmiştir.

Çizelge 4’den de görüldüğü gibi GWO-HS algoritmasının Kapur entropi tabanlı görüntü eşikleme algoritması ile test görselleri üzerinde gerçekleştirilen deneylerde, ABC, CLONALG, DE, GA, FA, HS, PSO ve SA gibi çeşitli optimizasyon algoritmalarına karşı belirgin bir üstünlük sağladığı gözlemlenmiştir. İki eşik seviyesinde, GWO algoritması ile benzer ortalama ve standart sapma değerleri elde edilirken, eşik sayısı ikiden fazla olduğunda GWO-HS algoritmasının performans üstünlüğü daha da belirgin hale gelmektedir.

**Çizelge 4.** Kapur Entropi Tabanlı Eşikleme Yöntemi İçin Algoritmalarından Elde Edilen Sonuçların Karşılaştırılması

Alg	K	Barbara		Living Room		Boats		Goldhill		Lake		Aerial	
		Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap
ABC		12.64	0.031	12.58	0.029	12.65	0.022	12.54	0.02	12.5	0.03	12.49	0.033
CLONALG		12.67	8E-04	12.62	8E-04	12.7	6E-04	12.56	4E-04	12.53	4E-04	12.54	9E-04
DE		12.64	0.021	12.56	0.054	12.67	0.026	12.54	0.022	12.5	0.034	12.47	0.033
GA		12.65	0.019	12.58	0.041	12.68	0.026	12.54	0.023	12.51	0.018	12.49	0.028
GWO	2	<b>12.67</b>	<b>0</b>	<b>12.62</b>	<b>2E-15</b>	<b>12.7</b>	<b>2E-15</b>	<b>12.56</b>	<b>0</b>	<b>12.53</b>	<b>2E-15</b>	<b>12.54</b>	<b>2E-15</b>
GWO-HS		<b>12.67</b>	<b>0</b>	<b>12.62</b>	<b>2E-15</b>	<b>12.7</b>	<b>2E-15</b>	<b>12.56</b>	<b>0</b>	<b>12.53</b>	<b>2E-15</b>	<b>12.54</b>	<b>2E-15</b>
FA		12.65	0.02	12.59	0.036	12.68	0.022	12.54	0.018	12.5	0.026	12.5	0.032
HS		12.66	0.01	12.6	0.022	12.69	0.014	12.55	0.012	12.52	0.01	12.51	0.024
PSO		12.66	0.016	12.61	0.012	12.69	0.01	12.55	0.01	12.52	0.011	11.61	0.284
SA		12.64	0.03	12.56	0.065	12.67	0.027	12.53	0.045	12.49	0.041	12.46	0.104
ABC		15.65	0.083	15.71	0.086	15.84	0.06	15.51	0.063	15.47	0.044	15.63	0.087
CA		15.74	0.003	15.87	0.008	15.93	0.005	15.64	0.003	15.56	0.005	15.74	0.009
DE		15.64	0.09	15.71	0.096	15.83	0.065	15.55	0.061	15.42	0.082	15.59	0.09
GA		15.66	0.057	15.7	0.091	15.86	0.05	15.56	0.05	15.44	0.07	15.64	0.065
GWO	3	15.75	4E-05	15.88	2E-04	15.94	2E-04	15.64	2E-04	15.57	6E-05	15.75	1E-04
GWO-HS		<b>15.75</b>	<b>2E-15</b>	<b>15.88</b>	<b>0</b>	<b>15.94</b>	<b>2E-15</b>	<b>15.64</b>	<b>2E-15</b>	<b>15.57</b>	<b>2E-15</b>	<b>15.75</b>	<b>0</b>
FA		15.66	0.069	15.74	0.093	15.83	0.049	15.55	0.059	15.43	0.077	15.63	0.085
HS		15.7	0.041	15.78	0.049	15.88	0.037	15.59	0.03	15.5	0.042	15.68	0.024
PSO		15.71	0.028	15.81	0.06	15.9	0.041	15.6	0.051	15.5	0.063	13.68	0.562
SA		15.6	0.101	15.7	0.127	15.85	0.058	15.53	0.084	15.45	0.083	15.55	0.107
ABC		18.37	0.117	18.43	0.133	18.81	0.056	18.18	0.146	18.1	0.119	18.28	0.164
CLONALG	4	18.53	0.014	18.69	0.024	18.92	0.014	18.43	0.01	18.34	0.017	18.56	0.026
DE		18.37	0.103	18.48	0.12	18.7	0.112	18.26	0.112	18.11	0.136	18.32	0.167

GA		18.4	0.085	18.49	0.105	18.79	0.07	18.26	0.089	18.18	0.104	18.38	0.141
GWO		18.56	6E-04	18.74	0.002	18.95	0.001	18.45	5E-04	18.38	8E-04	18.61	0.004
GWO-HS		<b>18.56</b>	<b>0</b>	<b>18.74</b>	<b>0</b>	<b>18.95</b>	<b>4E-15</b>	<b>18.45</b>	<b>0</b>	<b>18.38</b>	<b>4E-15</b>	<b>18.62</b>	<b>4E-15</b>
FA		18.36	0.105	18.53	0.097	18.74	0.115	18.29	0.095	18.14	0.098	18.35	0.128
HS		18.46	0.055	18.58	0.071	18.82	0.054	18.36	0.053	18.22	0.082	18.42	0.086
PSO		18.47	0.068	18.59	0.09	18.84	0.081	18.38	0.063	18.23	0.107	15.31	0.665
SA		18.3	0.166	18.49	0.119	18.72	0.127	18.23	0.126	18.09	0.175	18.29	0.22
ABC		20.83	0.166	21.01	0.132	21.31	0.161	20.5	0.232	20.6	0.192	20.62	0.216
CLONALG		21.18	0.028	21.36	0.03	21.65	0.029	21.07	0.037	20.94	0.041	21.11	0.028
DE		20.82	0.177	21.05	0.14	21.36	0.13	20.72	0.199	20.56	0.142	20.67	0.186
GA		20.93	0.137	21.16	0.118	21.44	0.095	20.8	0.136	20.67	0.15	20.74	0.169
GWO	5	21.24	0.003	21.43	0.015	21.71	0.01	21.14	0.008	20.99	0.041	21.07	0.069
GWO-HS		<b>21.25</b>	<b>8E-05</b>	<b>21.44</b>	<b>3E-05</b>	<b>21.73</b>	<b>3E-04</b>	<b>21.15</b>	<b>7E-05</b>	<b>21.03</b>	<b>2E-04</b>	<b>21.21</b>	<b>1E-04</b>
FA		20.92	0.171	21.07	0.154	21.42	0.137	20.82	0.178	20.64	0.157	20.75	0.178
HS		21.04	0.098	21.21	0.079	21.51	0.071	20.91	0.092	20.77	0.108	20.94	0.083
PSO		21.05	0.115	21.22	0.129	21.52	0.104	20.94	0.159	20.79	0.14	16.7	0.822
SA		20.81	0.195	21.06	0.2	21.38	0.16	20.67	0.222	20.6	0.173	20.62	0.296
ABC		22.96	0.246	23.52	0.206	23.86	0.206	22.93	0.27	22.83	0.179	22.93	0.231
CLONALG		23.64	0.049	23.89	0.04	24.26	0.04	23.5	0.044	23.38	0.052	23.45	0.043
DE		23.11	0.285	23.52	0.194	23.87	0.154	23	0.24	22.93	0.219	22.93	0.164
GA		23.27	0.19	23.62	0.147	23.97	0.135	23.09	0.247	23	0.196	22.98	0.178
GWO	6	23.77	0.012	23.95	0.018	24.3	0.029	23.66	0.009	23.35	0.081	23.54	0.034
GWO-HS		<b>23.79</b>	<b>5E-04</b>	<b>24</b>	<b>0.009</b>	<b>24.37</b>	<b>0.006</b>	<b>23.67</b>	<b>2E-04</b>	<b>23.52</b>	<b>2E-04</b>	<b>23.61</b>	<b>0.01</b>
FA		23.25	0.253	23.61	0.179	23.97	0.166	23.06	0.301	23.01	0.225	23.09	0.266
HS		23.44	0.138	23.74	0.106	24.12	0.102	23.24	0.176	23.15	0.146	23.23	0.109
PSO		23.43	0.224	23.74	0.139	24.11	0.125	23.2	0.232	23.18	0.177	17.93	1.333
SA		23.1	0.343	23.48	0.225	23.87	0.219	22.84	0.336	22.93	0.243	22.97	0.279
ABC		25.27	0.384	25.7	0.204	26.24	0.306	24.97	0.445	25.22	0.275	24.95	0.339
CLONALG		25.94	0.082	26.31	0.051	26.75	0.069	25.77	0.076	25.66	0.066	25.72	0.08
DE		25.27	0.305	25.78	0.267	26.23	0.266	25	0.368	25.05	0.256	25.14	0.226
GA		25.42	0.302	25.97	0.216	26.39	0.172	25.21	0.305	25.21	0.268	25.29	0.276
GWO	7	26.09	0.051	26.36	0.084	26.69	0.099	25.96	0.032	25.71	0.046	25.64	0.188
GWO-HS		<b>26.22</b>	<b>0.001</b>	<b>26.55</b>	<b>7E-04</b>	<b>27.01</b>	<b>0.003</b>	<b>26.03</b>	<b>6E-04</b>	<b>25.92</b>	<b>8E-04</b>	<b>26.01</b>	<b>0.003</b>
FA		25.38	0.307	25.92	0.252	26.35	0.23	25.1	0.365	25.25	0.318	25.17	0.254
HS		25.63	0.178	26.14	0.114	26.56	0.15	25.46	0.187	25.39	0.147	25.41	0.172
PSO		25.64	0.261	26.1	0.2	26.52	0.201	25.47	0.338	25.34	0.239	18.88	0.999
SA		25.21	0.364	25.77	0.281	26.2	0.241	25.06	0.394	25.03	0.305	25.09	0.315
ABC		27	0.335	27.87	0.351	28.32	0.221	26.88	0.462	26.87	0.685	26.96	0.319
CLONALG		28.07	0.106	28.63	0.091	29.09	0.096	27.86	0.12	27.83	0.082	27.83	0.128
DE	8	27.18	0.386	27.96	0.291	28.42	0.281	26.85	0.502	27.04	0.34	26.94	0.317
GA		27.51	0.313	28.15	0.237	28.58	0.226	27.22	0.332	27.3	0.3	27.21	0.27
GWO		28.11	0.218	28.38	0.306	28.83	0.218	28.08	0.132	27.87	0.158	27.43	0.167
GWO-HS		<b>28.48</b>	<b>0.002</b>	<b>28.92</b>	<b>0.003</b>	<b>29.41</b>	<b>0.003</b>	<b>28.26</b>	<b>0.001</b>	<b>28.18</b>	<b>0.021</b>	<b>28.25</b>	<b>0.011</b>

FA		27.48	0.326	28.16	0.297	28.57	0.285	27.18	0.408	27.39	0.367	27.27	0.281
HS		27.71	0.242	28.43	0.186	28.85	0.181	27.46	0.193	27.44	0.185	27.57	0.21
PSO		27.65	0.325	28.33	0.271	28.74	0.243	27.41	0.393	27.56	0.282	19.54	1.443
SA		27.24	0.383	27.99	0.313	28.4	0.295	26.94	0.487	27.13	0.31	27.06	0.43
ABC		28.83	0.64	30.03	0.453	30.25	0.405	28.54	0.659	28.93	0.299	28.87	0.491
CLONALG		30.08	0.167	30.82	0.115	31.26	0.115	29.82	0.134	29.87	0.117	29.85	0.11
DE		29	0.45	29.95	0.368	30.43	0.378	28.57	0.466	28.99	0.382	28.74	0.427
GA		29.36	0.308	30.25	0.327	30.69	0.285	28.95	0.376	29.17	0.397	29.17	0.314
GWO	9	29.88	0.318	30.24	0.309	30.89	0.181	29.91	0.214	29.74	0.275	29.29	0.462
GWO-HS		<b>30.62</b>	<b>0.004</b>	<b>31.27</b>	<b>0.007</b>	<b>31.73</b>	<b>0.004</b>	<b>30.39</b>	<b>0.005</b>	<b>30.41</b>	<b>0.015</b>	<b>30.34</b>	<b>0.026</b>
FA		29.31	0.455	30.25	0.354	30.85	0.38	28.99	0.649	29.52	0.381	29.26	0.36
HS		29.67	0.291	30.58	0.217	31.04	0.185	29.38	0.304	29.46	0.205	29.47	0.237
PSO		29.58	0.367	30.35	0.323	30.85	0.294	29.33	0.475	29.56	0.301	20.28	1.497
SA		29.09	0.544	30.02	0.342	30.52	0.363	28.67	0.575	29.03	0.41	28.88	0.401
ABC		30.88	0.36	31.82	0.594	32.22	0.287	30.65	0.683	30.75	0.319	30.45	0.353
CLONALG		31.96	0.188	32.83	0.115	33.32	0.12	31.68	0.184	31.86	0.152	31.65	0.178
DE		30.76	0.549	31.9	0.295	32.34	0.331	30.29	0.583	30.82	0.488	30.5	0.517
GA		31.14	0.399	32.1	0.359	32.59	0.327	30.72	0.404	31.07	0.39	30.83	0.334
GWO	10	31.49	0.327	32.05	0.312	32.61	0.307	31.32	0.493	31.4	0.366	30.87	0.376
GWO-HS		<b>32.63</b>	<b>0.008</b>	<b>33.45</b>	<b>0.004</b>	<b>33.9</b>	<b>0.01</b>	<b>32.39</b>	<b>0.004</b>	<b>32.56</b>	<b>0.033</b>	<b>32.29</b>	<b>0.053</b>
FA		31.19	0.502	32.48	0.363	32.95	0.345	30.82	0.594	31.47	0.449	31	0.549
HS		31.49	0.264	32.52	0.288	33.05	0.22	31.07	0.332	31.38	0.256	31.25	0.196
PSO		31.4	0.442	32.44	0.314	32.91	0.345	30.97	0.549	31.43	0.421	20.52	1.545
SA		30.66	0.51	31.95	0.367	32.37	0.387	30.34	0.631	30.74	0.461	30.59	0.496

Çizelge 5'e bakıldığında, GWO-HS algoritmasının Otsu görüntü eşikleme algoritması ile yapılan testlerde de benzer şekilde, ABC, CLONALG, DE, GA, FA, HS, PSO ve SA algoritmalarına karşı belirgin bir üstünlük gösterdiği anlaşılmaktadır. İki eşik seviyesinde, GWO algoritması ile eşit değerlere ulaşılmasına rağmen, eşik sayısı iki seviyesini aştığında GWO-HS algoritmasının üstünlüğü açık bir şekilde ortaya çıkmaktadır.

**Çizelge 5.** Otsu Eşikleme Yöntemi İçin Algoritmalarından Elde Edilen Sonuçların Karşılaştırılması

Alg	K	Barbara		Living Room		Boats		Goldhill		Lake		Aerial	
		Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap
ABC		2593	15.83	1817.1	4.846	1606.8	14.03	2077	9.6758	3962	9.897	1799	11.03
CLONALG		2608	0.229	1822.9	0.247	1625.1	0.188	2086	0.15	3977	0.153	1808	0.128
DE	2	2593	16.11	1813.4	8.95	1606.7	13.53	2072	10.682	3968	8.621	1791	18.2
GA		2596	13.04	1816.1	6.065	1614.3	10.88	2074	11.647	3969	8.499	1799	7.863
GWO		<b>2609</b>	<b>5E-13</b>	<b>1823.1</b>	<b>2E-13</b>	<b>1625.3</b>	<b>2E-13</b>	<b>2086</b>	<b>0</b>	<b>3978</b>	<b>0</b>	<b>1808</b>	<b>2E-13</b>

GWO-HS	<b>2609</b>	<b>5E-13</b>	<b>1823.1</b>	<b>2E-13</b>	<b>1625.3</b>	<b>2E-13</b>	<b>2086</b>	<b>0</b>	<b>3978</b>	<b>0</b>	<b>1808</b>	<b>2E-13</b>
FA	2596	14.91	1815.6	8.002	1612.7	15.3	2075	11.349	3971	6.587	1801	8.808
HS	2601	7.365	1819	4.749	1619.3	6.326	2081	4.8264	3973	5.532	1803	3.524
PSO	2606	4.717	1818.7	8.035	1621.4	6.169	2081	6.995	3976	2.927	1672	39.62
SA	2591	22.81	1812.1	13.83	1610	20.26	2066	16.922	3967	11.36	1796	12.06
ABC	2755	21.52	1932.9	14.66	1725	20.41	2205	17.711	4091	17.5	1878	23.02
CLONALG	2783	1.301	1952.6	0.895	1755.3	0.972	2230	1.1176	4114	1.151	1903	1.271
DE	2751	21.41	1932.2	17.16	1729.7	16.01	2199	19.58	4093	14.59	1877	21.8
GA	2757	17.22	1937	11.7	1732.7	16.41	2206	15.734	4090	15.62	1883	13.85
GWO	2785	0.009	1954	0.016	1757.2	0.005	2232	0.0078	4115	0.018	1905	0.008
GWO-HS	<b>2785</b>	<b>0</b>	<b>1954</b>	<b>0</b>	<b>1757.2</b>	<b>2E-13</b>	<b>2232</b>	<b>0</b>	<b>4115</b>	<b>0</b>	<b>1905</b>	<b>0</b>
FA	2755	18.79	1931.4	17.47	1728.2	20.23	2209	17.212	4095	14.87	1881	15.97
HS	2770	10.18	1939.6	8.984	1742.2	7.994	2219	9.8248	4103	8.285	1889	10.88
PSO	2772	14.26	1946.6	6.741	1746.7	9.488	2221	9.2205	4105	10.64	1715	57.34
SA	2749	25.15	1931.8	14.16	1726.9	21.02	2197	29.88	4086	21.17	1872	21.07
ABC	2797	23.46	1975.8	14.38	1779.4	23.56	2258	14.833	4145	13.63	1920	11.24
CLONALG	2852	2.323	2010.3	2.196	1821.8	2.06	2302	2.7051	4180	1.955	1952	2.637
DE	2810	23.07	1982.9	16.22	1793.2	16.76	2271	17.642	4148	15.08	1916	15.19
GA	2816	18.73	1990.6	14.82	1797.7	12.51	2278	16.068	4152	14.02	1925	12.66
GWO	2856	0.095	2014.1	0.911	1825.8	0.186	2306	0.0652	4184	0.103	1955	4.07
GWO-HS	<b>2856</b>	<b>5E-13</b>	<b>2014.4</b>	<b>0</b>	<b>1826</b>	<b>2E-13</b>	<b>2307</b>	<b>5E-13</b>	<b>4184</b>	<b>9E-13</b>	<b>1957</b>	<b>5E-13</b>
FA	2817	18.85	1986.3	15.26	1795.1	18.82	2273	18.49	4154	16.54	1927	13.5
HS	2836	12.7	1997.9	7.893	1807.8	8.412	2283	9.1366	4169	7.844	1937	10.79
PSO	2839	14.39	2000.7	13.42	1807.5	14.01	2288	14.667	4168	15.1	1734	50.29
SA	2819	22.98	1979.2	17.41	1792.4	15.62	2264	21.951	4149	19.27	1911	16.1
ABC	2849	16.64	2016.8	15.36	1822.2	18.27	2302	28.091	4180	9.66	1948	17.78
CLONALG	2883	3.048	2043.4	2.55	1862.3	2.5	2336	2.5777	4213	2.628	1974	2.634
DE	2849	17.67	2017.9	13.78	1831.3	14.61	2308	16.31	4185	15.44	1952	12.16
GA	2858	11.94	2022.8	9.295	1839.5	11	2315	14.727	4193	11.08	1955	7.608
GWO	2891	0.38	2047.4	3.451	1865.6	6.773	2342	0.2051	4217	8.392	1972	10.4
GWO-HS	<b>2891</b>	<b>7E-04</b>	<b>2049.4</b>	<b>0.001</b>	<b>1869.1</b>	<b>0.001</b>	<b>2342</b>	<b>0.0041</b>	<b>4220</b>	<b>0.002</b>	<b>1981</b>	<b>0.004</b>
FA	2851	19.01	2018.8	12.73	1833.6	11.94	2308	16.441	4188	15.25	1952	9.58
HS	2867	11.73	2030	7.698	1846.8	8.595	2321	7.3691	4200	7.787	1963	6.887
PSO	2872	11.3	2031.7	9.707	1849.8	11.82	2321	10.743	4201	11.05	1762	60.91
SA	2852	17.22	2016.6	13.82	1831.7	17.6	2308	15.509	4185	14.97	1952	12
ABC	2874	8.086	2036.9	11.52	1862.4	14.57	2323	16.763	4198	12.54	1964	9.592
CLONALG	2905	3.565	2063.3	2.72	1886.9	3.033	2357	3.8713	4232	2.834	1992	2.638
DE	2879	11.5	2039.4	11.24	1860.6	11.77	2331	13.815	4206	11.18	1968	9.98
GA	2884	11.3	2043	9.182	1867.4	9.213	2336	10.261	4212	11.66	1973	7.777
GWO	2910	2.446	2064.2	5.169	1887.3	5.54	2364	2.3737	4230	8.052	1986	7.163
GWO-HS	<b>2914</b>	<b>0.011</b>	<b>2070.6</b>	<b>0.01</b>	<b>1895.5</b>	<b>0.006</b>	<b>2367</b>	<b>0.0117</b>	<b>4240</b>	<b>0.009</b>	<b>1997</b>	<b>0.027</b>
FA	2879	14.62	2043.8	9.995	1864.9	12.81	2336	13.779	4212	10.71	1972	12.23
HS	2889	8.311	2051.8	6.622	1873	8.551	2344	6.3413	4222	7.081	1980	6.601

PSO		2892	11.4	2050.7	10.91	1873.2	10.14	2348	10.697	4224	7.685	1796	46.12
SA		2879	11.96	2040.6	11.13	1860	11.09	2333	15.356	4205	12.43	1964	14.5
ABC		2894	11.18	2054.6	8.366	1869.5	13.07	2351	7.4453	4219	9.106	1981	12.61
CLONALG		2919	3.33	2075.9	2.59	1901.3	3.314	2373	2.8842	4244	2.376	2004	2.098
DE		2895	10.28	2053.4	8.084	1878	11.04	2345	13.21	4223	10.1	1985	8.563
GA		2903	9.346	2060.7	8.46	1883	7.429	2355	9.1278	4228	8.743	1987	8.108
GWO	7	2923	4.399	2072.3	5.772	1896.9	7.741	2379	2.4656	4238	6.403	1997	8.741
GWO-HS		<b>2929</b>	<b>0.014</b>	<b>2084.3</b>	<b>0.04</b>	<b>1911.8</b>	<b>0.019</b>	<b>2382</b>	<b>0.0216</b>	<b>4253</b>	<b>0.055</b>	<b>2012</b>	<b>0.022</b>
FA		2900	9.97	2057.8	9.265	1879.1	11.34	2354	9.9203	4227	7.353	1981	7.577
HS		2907	6.249	2067	6.386	1888.9	6.28	2360	6.559	4236	5.465	1994	5.948
PSO		2909	8.474	2065	8.489	1892.1	9.881	2363	9.3507	4236	10.01	1800	51.3
SA		2898	11.32	2055.6	8.282	1879.3	10.36	2348	13.681	4222	9.957	1981	12.35
ABC		2905	8.391	2064.6	8.524	1887.9	12.76	2351	16.716	4229	7.849	1988	7.88
CLONALG		2929	3.104	2084.6	2.542	1912.5	2.912	2383	2.8873	4253	2.578	2013	2.108
DE		2907	9.718	2068.2	7.488	1891.3	9.668	2359	10.828	4236	7.358	1994	6.369
GA		2912	7.61	2073.7	6.24	1897.1	7.645	2370	6.6053	4240	6.794	2000	5.319
GWO	8	2930	4.696	2076.4	5.337	1908	5.34	2387	3.4669	4246	6.874	1999	7.1
GWO-HS		<b>2940</b>	<b>0.025</b>	<b>2094.2</b>	<b>0.043</b>	<b>1923.1</b>	<b>0.036</b>	<b>2394</b>	<b>0.0306</b>	<b>4264</b>	<b>0.026</b>	<b>2022</b>	<b>0.036</b>
FA		2910	10.13	2070.3	9.048	1897.4	8.099	2361	12.689	4237	9.138	1995	8.37
HS		2918	6.078	2076.8	4.628	1903.2	5.856	2374	5.9137	4246	5.142	2005	4.764
PSO		2918	6.887	2078.1	6.746	1903.7	7.376	2374	9.4927	4246	8.251	1814	59.5
SA		2907	11.17	2065.8	9.306	1893.7	8.972	2362	14.696	4232	10.5	1994	9.559
ABC		2910	15.28	2075.6	9.521	1902.4	7.416	2371	8.4022	4237	14.99	1997	9.461
CLONALG		2936	2.414	2091.8	2.022	1921.5	2.872	2391	2.5351	4261	2.589	2019	2.58
DE		2916	6.301	2076.5	7.704	1903.2	9.341	2371	9.0118	4246	7.169	2004	5.854
GA		2924	6.804	2080.1	7.087	1908.7	6.815	2379	6.3672	4249	4.963	2007	5.836
GWO	9	2935	3.819	2085.7	4.206	1913.5	5.256	2391	4.8581	4251	6.667	2005	6.126
GWO-HS		<b>2947</b>	<b>0.036</b>	<b>2101.2</b>	<b>0.057</b>	<b>1930.8</b>	<b>0.121</b>	<b>2402</b>	<b>0.0363</b>	<b>4271</b>	<b>0.076</b>	<b>2028</b>	<b>0.05</b>
FA		2921	8.468	2080	6.664	1907.6	8.338	2375	9.9848	4249	7.996	2005	6.897
HS		2927	4.291	2085	5.395	1910.4	5.344	2383	5.7424	4255	4.361	2012	4.222
PSO		2927	6.676	2084	5.905	1912.9	5.788	2384	7.2169	4253	6.015	1816	52.28
SA		2917	9.136	2078.3	6.818	1902.5	8.451	2370	13.057	4245	7.719	2004	6.962
ABC		2921	11.52	2080.9	5.428	1904.2	7.992	2376	11.664	4252	7.005	2008	10.61
CLONALG		2943	2.437	2097.7	2.304	1926.3	1.786	2397	2.3541	4266	2.204	2024	1.995
DE		2925	8.354	2084.7	6.753	1909.7	6.11	2378	10.28	4251	7.299	2011	6.554
GA		2932	5.945	2087.5	5.407	1916.1	5.508	2386	6.2659	4256	5.739	2014	5.188
GWO	10	2938	5.538	2090.8	4.513	1919.4	5.143	2395	5.388	4255	5.774	2009	4.829
GWO-HS		<b>2953</b>	<b>0.083</b>	<b>2106.8</b>	<b>0.169</b>	<b>1936.9</b>	<b>0.144</b>	<b>2408</b>	<b>0.1019</b>	<b>4276</b>	<b>0.156</b>	<b>2033</b>	<b>1.284</b>
FA		2930	7.193	2087.7	6.75	1916.4	8.776	2381	9.0072	4255	6.897	2010	5.668
HS		2935	4.726	2092	3.425	1919.4	4.597	2389	4.656	4261	3.942	2017	3.485
PSO		2935	7.08	2090.8	5.298	1919.7	6.697	2390	6.6671	4260	7.068	1810	42.23
SA		2924	7.613	2081.8	5.805	1910.8	6.105	2380	8.8044	4250	7.812	2011	7.409

Çizelge 6'dan da görüldüğü üzere GWO-HS algoritması Tsallis entropi tabanlı görüntü eşikleme algoritması ile de ABC, CLONALG, DE, GA, FA, HS, PSO ve SA algoritmalarına karşı belirgin bir üstünlük sergilemektedir. İki eşik seviyesinde GWO algoritması ile benzer sonuçlar elde edilirken, eşik sayısı iki seviyesinin üzerine çıktığında GWO-HS algoritmasının üstünlüğü belirginleşmektedir.

**Çizelge 6.** Tsallis Entropi Tabanlı Eşikleme Yöntemi İçin Algoritmalarından Elde Edilen Sonuçların Karşılaştırılması

Alg	K	Barbara		Living Room		Boats		Goldhill		Lake		Aerial	
		Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap	Ort	StdSap
ABC		0.889	1E-06	0.889	9E-07	0.889	1E-06	0.889	2E-06	0.889	2E-06	0.889	5E-06
CLONALG		0.889	1E-08	0.889	7E-09	0.889	1E-08	0.889	6E-09	0.889	3E-09	0.889	8E-09
DE		0.889	4E-07	0.889	3E-07	0.889	3E-07	0.889	5E-07	0.889	4E-07	0.889	4E-07
GA		0.889	2E-07	0.889	3E-07	0.889	2E-07	0.889	4E-07	0.889	3E-07	0.889	4E-07
GWO	2	<b>0.889</b>	<b>1E-16</b>	<b>0.889</b>	<b>7E-09</b>	<b>0.889</b>	<b>1E-16</b>	<b>0.889</b>	<b>0</b>	<b>0.889</b>	<b>1E-16</b>	<b>0.889</b>	<b>0</b>
GWO-HS		<b>0.889</b>	<b>1E-16</b>	<b>0.889</b>	<b>0</b>	<b>0.889</b>	<b>1E-16</b>	<b>0.889</b>	<b>0</b>	<b>0.889</b>	<b>1E-16</b>	<b>0.889</b>	<b>0</b>
FA		0.889	2E-07	0.889	3E-07	0.889	2E-07	0.889	3E-07	0.889	2E-07	0.889	1E-06
HS		0.889	1E-07	0.889	2E-07	0.889	1E-07	0.889	2E-07	0.889	1E-07	0.889	4E-07
PSO		0.889	2E-07	0.889	1E-07	0.889	1E-07	0.889	2E-07	0.889	2E-07	0.889	4E-05
SA		0.889	5E-07	0.889	3E-07	0.889	3E-07	0.889	4E-07	0.889	3E-07	0.889	2E-06
ABC		1.296	1E-04	1.296	2E-05	1.296	5E-06	1.296	1E-05	1.296	2E-05	1.296	9E-06
CLONALG		1.296	2E-07	1.296	2E-07	1.296	2E-07	1.296	2E-07	1.296	2E-07	1.296	3E-07
DE		1.296	3E-06	1.296	5E-06	1.296	2E-06	1.296	5E-06	1.296	5E-06	1.296	3E-06
GA		1.296	2E-06	1.296	3E-06	1.296	1E-06	1.296	3E-06	1.296	5E-06	1.296	5E-06
GWO	3	1.296	7E-10	1.296	1E-09	1.296	5E-08	1.296	6E-09	1.296	2E-09	1.296	6E-07
GWO-HS		<b>1.296</b>	<b>0</b>	<b>1.296</b>	<b>0</b>	<b>1.296</b>	<b>2E-16</b>	<b>1.296</b>	<b>0</b>	<b>1.296</b>	<b>0</b>	<b>1.296</b>	<b>0</b>
FA		1.296	3E-06	1.296	4E-06	1.296	2E-06	1.296	3E-06	1.296	5E-06	1.296	3E-06
HS		1.296	2E-06	1.296	1E-06	1.296	9E-07	1.296	1E-06	1.296	3E-06	1.296	2E-06
PSO		1.296	1E-06	1.296	2E-06	1.296	9E-07	1.296	2E-06	1.296	4E-06	1.296	3E-04
SA		1.296	5E-06	1.296	7E-06	1.296	2E-06	1.296	6E-06	1.296	7E-06	1.296	4E-06
ABC		1.654	7E-05	1.654	3E-05	1.654	2E-05	1.654	2E-04	1.654	9E-05	1.654	2E-04
CLONALG		1.654	1E-06	1.654	2E-06	1.654	8E-07	1.654	1E-06	1.654	2E-06	1.654	3E-06
DE		1.654	2E-05	1.654	1E-05	1.654	7E-06	1.654	2E-05	1.654	2E-05	1.654	1E-05
GA		1.654	1E-05	1.654	1E-05	1.654	7E-06	1.654	1E-05	1.654	1E-05	1.654	1E-05
GWO	4	1.654	3E-08	1.654	8E-08	1.654	4E-08	1.654	5E-08	1.654	1E-07	1.654	7E-07
GWO-HS		<b>1.654</b>	<b>0</b>	<b>1.654</b>	<b>0</b>	<b>1.654</b>	<b>0</b>	<b>1.654</b>	<b>2E-16</b>	<b>1.654</b>	<b>2E-16</b>	<b>1.654</b>	<b>2E-16</b>
FA		1.654	2E-05	1.654	1E-05	1.654	6E-06	1.654	2E-05	1.654	2E-05	1.654	2E-05
HS		1.654	7E-06	1.654	8E-06	1.654	3E-06	1.654	8E-06	1.654	1E-05	1.654	4E-06
PSO		1.654	7E-06	1.654	1E-05	1.654	5E-06	1.654	8E-06	1.654	1E-05	1.653	7E-04
SA		1.654	3E-05	1.654	2E-05	1.654	1E-05	1.654	3E-05	1.654	2E-05	1.654	1E-05
ABC	5	1.995	9E-04	1.996	1E-04	1.995	4E-04	1.995	2E-04	1.995	2E-04	1.995	3E-04

CLONALG	1.996	4E-06	1.996	4E-06	1.996	3E-06	1.996	5E-06	1.996	6E-06	1.996	4E-06
DE	1.996	4E-05	1.996	4E-05	1.996	3E-05	1.996	6E-05	1.996	9E-05	1.996	1E-04
GA	1.996	3E-05	1.996	2E-05	1.996	2E-05	1.996	3E-05	1.996	4E-05	1.996	4E-05
GWO	1.996	3E-07	1.996	7E-07	1.996	2E-07	1.996	6E-07	1.996	4E-07	1.996	2E-06
GWO-HS	<b>1.996</b>	<b>3E-09</b>	<b>1.996</b>	<b>2E-08</b>	<b>1.996</b>	<b>3E-10</b>	<b>1.996</b>	<b>3E-08</b>	<b>1.996</b>	<b>5E-09</b>	<b>1.996</b>	<b>9E-08</b>
FA	1.996	5E-05	1.996	3E-05	1.996	3E-05	1.996	5E-05	1.996	6E-05	1.996	8E-05
HS	1.996	2E-05	1.996	1E-05	1.996	1E-05	1.996	3E-05	1.996	3E-05	1.996	2E-05
PSO	1.996	4E-05	1.996	2E-05	1.996	2E-05	1.996	2E-05	1.996	5E-05	1.99	0.005
SA	1.996	5E-05	1.996	5E-05	1.996	4E-05	1.996	7E-05	1.996	8E-05	1.996	6E-05
ABC	2.331	3E-04	2.331	2E-04	2.331	4E-04	2.331	8E-04	2.331	5E-04	2.33	0.001
CLONALG	2.332	1E-05	2.332	1E-05	2.332	8E-06	2.332	1E-05	2.332	2E-05	2.332	2E-05
DE	2.332	2E-04	2.332	8E-05	2.332	8E-05	2.332	3E-04	2.331	2E-04	2.332	1E-04
GA	2.332	7E-05	2.332	6E-05	2.332	6E-05	2.332	1E-04	2.332	1E-04	2.332	8E-05
GWO	2.332	2E-06	2.332	4E-06	2.332	4E-06	2.332	4E-06	2.332	7E-05	2.332	3E-05
GWO-HS	<b>2.332</b>	<b>6E-08</b>	<b>2.332</b>	<b>3E-08</b>	<b>2.332</b>	<b>4E-08</b>	<b>2.332</b>	<b>4E-08</b>	<b>2.332</b>	<b>7E-08</b>	<b>2.332</b>	<b>1E-06</b>
FA	2.332	1E-04	2.332	1E-04	2.332	6E-05	2.332	1E-04	2.332	2E-04	2.332	1E-04
HS	2.332	5E-05	2.332	3E-05	2.332	2E-05	2.332	5E-05	2.332	7E-05	2.332	6E-05
PSO	2.332	7E-05	2.332	4E-05	2.332	8E-05	2.332	6E-05	2.332	1E-04	2.325	0.004
SA	2.332	1E-04	2.332	1E-04	2.332	9E-05	2.332	3E-04	2.332	2E-04	2.331	4E-04
ABC	2.662	0.002	2.665	4E-04	2.665	4E-04	2.637	0.055	2.662	0.004	2.664	0.001
CLONALG	2.666	3E-05	2.666	2E-05	2.666	2E-05	2.666	3E-05	2.666	4E-05	2.666	5E-05
DE	2.665	4E-04	2.666	2E-04	2.666	1E-04	2.665	4E-04	2.665	4E-04	2.665	4E-04
GA	2.666	2E-04	2.666	1E-04	2.666	1E-04	2.666	2E-04	2.666	2E-04	2.665	4E-04
GWO	2.666	9E-06	2.666	9E-06	2.666	7E-06	2.666	3E-05	2.666	1E-04	2.666	4E-05
GWO-HS	<b>2.666</b>	<b>2E-07</b>	<b>2.666</b>	<b>2E-06</b>	<b>2.666</b>	<b>5E-07</b>	<b>2.666</b>	<b>2E-07</b>	<b>2.666</b>	<b>2E-07</b>	<b>2.666</b>	<b>2E-05</b>
FA	2.666	3E-04	2.666	2E-04	2.666	1E-04	2.665	3E-04	2.665	4E-04	2.665	3E-04
HS	2.666	8E-05	2.666	6E-05	2.666	6E-05	2.666	1E-04	2.666	1E-04	2.666	2E-04
PSO	2.666	1E-04	2.666	1E-04	2.666	1E-04	2.666	2E-04	2.666	3E-04	2.633	0.02
SA	2.665	4E-04	2.666	2E-04	2.666	1E-04	2.665	4E-04	2.665	5E-04	2.665	7E-04
ABC	2.992	0.006	2.997	0.002	2.997	0.002	2.987	0.017	2.996	0.002	2.988	0.012
CLONALG	2.999	6E-05	2.999	5E-05	3	4E-05	2.999	7E-05	2.999	8E-05	2.999	8E-05
DE	2.998	0.001	2.999	2E-04	2.999	3E-04	2.998	0.002	2.998	8E-04	2.998	5E-04
GA	2.999	4E-04	2.999	2E-04	2.999	1E-04	2.999	7E-04	2.999	5E-04	2.999	4E-04
GWO	2.999	3E-05	2.999	3E-05	3	5E-05	2.999	3E-05	2.999	1E-04	2.999	2E-04
GWO-HS	<b>3</b>	<b>2E-06</b>	<b>3</b>	<b>1E-06</b>	<b>3</b>	<b>2E-07</b>	<b>3</b>	<b>5E-07</b>	<b>2.999</b>	<b>2E-06</b>	<b>2.999</b>	<b>1E-05</b>
FA	2.999	6E-04	2.999	3E-04	2.999	2E-04	2.998	9E-04	2.998	6E-04	2.998	8E-04
HS	2.999	2E-04	2.999	1E-04	2.999	1E-04	2.999	2E-04	2.999	3E-04	2.999	2E-04
PSO	2.999	4E-04	2.999	2E-04	2.999	2E-04	2.999	6E-04	2.999	4E-04	2.934	0.033
SA	2.998	0.001	2.999	3E-04	2.999	3E-04	2.998	0.002	2.998	9E-04	2.998	8E-04
ABC	3.313	0.016	3.328	0.002	3.329	0.006	3.312	0.014	3.32	0.017	3.321	0.007
CLONALG	3.332	9E-05	3.333	7E-05	3.333	8E-05	3.332	1E-04	3.332	1E-04	3.332	2E-04
DE	3.33	0.002	3.332	5E-04	3.332	4E-04	3.33	0.002	3.33	0.001	3.33	0.002
GA	3.331	8E-04	3.332	3E-04	3.332	2E-04	3.331	0.001	3.331	7E-04	3.331	7E-04

GWO	3.333	6E-04	3.332	3E-04	3.333	1E-04	3.333	2E-04	3.332	2E-04	3.331	7E-04
GWO-HS	<b>3.333</b>	<b>7E-06</b>	<b>3.333</b>	<b>2E-06</b>	<b>3.333</b>	<b>8E-07</b>	<b>3.333</b>	<b>2E-06</b>	<b>3.333</b>	<b>5E-05</b>	<b>3.333</b>	<b>2E-05</b>
FA	3.331	0.002	3.332	5E-04	3.332	3E-04	3.33	0.002	3.331	1E-03	3.331	0.001
HS	3.332	3E-04	3.332	2E-04	3.333	1E-04	3.332	5E-04	3.332	4E-04	3.332	3E-04
PSO	3.332	8E-04	3.332	4E-04	3.332	3E-04	3.332	7E-04	3.332	7E-04	3.14	0.164
SA	3.33	0.003	3.332	6E-04	3.332	4E-04	3.329	0.004	3.33	0.001	3.329	0.001
ABC	3.578	0.127	3.658	0.007	3.663	8E-04	3.58	0.127	3.641	0.032	3.648	0.015
CLONALG	3.665	2E-04	3.666	1E-04	3.666	1E-04	3.665	2E-04	3.665	3E-04	3.665	3E-04
DE	3.662	0.003	3.664	7E-04	3.665	5E-04	3.659	0.008	3.663	0.001	3.661	0.002
GA	3.664	0.001	3.665	5E-04	3.665	4E-04	3.663	0.001	3.663	0.001	3.663	0.001
GWO	3.665	5E-04	3.664	0.001	3.665	4E-04	3.664	0.005	3.664	0.001	3.661	0.003
GWO-HS	<b>3.666</b>	<b>9E-06</b>	<b>3.666</b>	<b>6E-06</b>	<b>3.666</b>	<b>4E-06</b>	<b>3.666</b>	<b>5E-06</b>	<b>3.666</b>	<b>8E-05</b>	<b>3.666</b>	<b>8E-05</b>
FA	3.662	0.003	3.665	9E-04	3.665	5E-04	3.661	0.004	3.664	0.001	3.663	0.001
HS	3.665	6E-04	3.665	2E-04	3.665	2E-04	3.664	8E-04	3.664	7E-04	3.664	6E-04
PSO	3.664	0.002	3.665	6E-04	3.665	6E-04	3.664	0.002	3.664	8E-04	3.454	0.152
SA	3.661	0.004	3.664	0.001	3.665	7E-04	3.659	0.006	3.662	0.002	3.66	0.003

Çizelgelerden de görülebildiği gibi özellikle iki eşik seviyesinde GWO algoritması ile benzer sonuçlar elde edilirken eşik sayısı arttıkça GWO-HS algoritması belirgin bir üstünlük sağlamaktadır. Bu bulgular, GWO-HS algoritmasının çok eşikli görüntü eşikleme problemlerinde daha etkili bir seçenek olduğunu göstermektedir.

Çizelge 7’de Kapur yöntemi için hesaplanan PSNR ve SSIM değerleri verilmiştir. Düşük eşik değerlerinde sürekli en iyi değeri veren bir algoritma bulunmamaktadır. Eşik sayısı arttıkça GWO-HS algoritmasının en iyi değerlere ulaşabildiği gözlemlenmiştir.

Çizelge 7. Kapur yöntemi PSNR ve SSIM değerleri

Alg	K	Barbara		Living Room		Boats		Goldhill		Lake		Aerial	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ABC		27.749	<b>0.4572</b>	28.097	0.4663	27.824	0.5074	27.881	0.2445	27.553	0.4431	27.785	0.4511
CLONALG		27.744	0.4406	28.082	0.4757	27.843	0.5338	27.888	0.3765	27.547	0.4643	27.784	<b>0.6293</b>
DE		27.744	0.429	28.097	<b>0.4882</b>	27.836	<b>0.5438</b>	27.888	0.3769	27.56	0.434	27.785	0.6286
GA		27.769	0.3593	28.09	0.4756	27.827	0.5385	27.872	0.3986	27.558	0.4042	27.783	0.4749
GWO	2	27.744	0.4389	28.082	0.4761	27.834	0.5313	27.888	0.3765	27.556	0.4662	27.784	0.6231
GWO-HS		27.744	0.4389	28.082	0.4761	27.834	0.5313	27.888	0.3765	27.556	0.4662	27.784	0.6231
FA		27.785	0.4193	28.103	0.4723	27.824	0.5329	27.877	<b>0.4111</b>	27.561	<b>0.4666</b>	27.784	0.5991
HS		<b>27.79</b>	0.4282	<b>28.104</b>	0.4858	27.843	0.5324	27.883	0.3938	<b>27.57</b>	0.4588	27.785	0.6281
PSO		27.766	0.4209	28.076	0.3687	27.845	0.4834	<b>27.91</b>	0.3132	27.546	0.4348	<b>27.788</b>	0.2601
SA		27.773	0.4506	28.065	0.4659	<b>27.848</b>	0.5279	27.888	0.3769	<b>27.57</b>	0.4642	27.785	0.6277

ABC		27.789	0.3933	27.881	0.5563	27.854	0.5291	27.527	0.3175	<b>27.993</b>	0.5114	28.12	0.7262
CLONALG		27.957	0.5346	27.954	0.5863	27.959	0.6151	27.923	0.4493	27.801	0.5265	<b>28.293</b>	0.7407
DE		28.039	0.542	27.77	0.5782	27.693	0.592	27.583	0.3253	27.639	<b>0.5621</b>	28.237	0.7314
GA		28.004	<b>0.5788</b>	27.893	0.5272	<b>28.063</b>	0.4861	28.024	0.4406	27.619	0.4757	28.266	0.635
GWO	3	28.036	0.5427	28.003	0.5854	27.935	0.6182	<b>28.039</b>	0.461	27.852	0.5276	28.259	0.7432
GWO-HS		28.036	0.5427	28.003	0.5854	27.935	0.6182	<b>28.039</b>	0.461	27.852	0.5276	28.259	0.7432
FA		27.717	0.4785	27.969	0.5735	27.881	0.6053	27.821	0.4387	27.779	0.4898	28.241	0.6963
HS		27.976	0.5473	27.939	<b>0.5918</b>	27.934	0.6046	27.932	0.4486	27.635	0.5611	28.245	0.7326
PSO		<b>28.224</b>	0.437	<b>28.035</b>	0.4429	27.988	0.4812	27.351	0.3501	27.469	0.4231	27.694	0.3876
SA		27.714	0.5595	27.955	0.5266	27.908	<b>0.6248</b>	28.009	<b>0.4898</b>	27.843	0.5047	28.228	<b>0.7471</b>
ABC		27.861	0.5553	28.399	0.6301	27.614	0.5929	27.732	0.4809	28.062	0.6046	27.561	0.7405
CLONALG		27.889	0.5973	28.43	<b>0.6642</b>	27.472	<b>0.6537</b>	<b>28.275</b>	<b>0.5817</b>	27.978	0.5494	27.922	0.7639
DE		27.921	0.5703	28.142	0.5985	27.814	0.6485	27.747	0.4909	28.318	0.6036	27.972	0.7522
GA		27.916	0.6066	27.798	0.5273	<b>28.149</b>	0.6406	27.571	0.4312	27.632	0.5043	27.581	0.6444
GWO	4	27.954	0.6059	28.386	0.6554	27.648	0.6475	28.086	0.5571	27.992	0.5652	<b>28.06</b>	0.7727
GWO-HS		27.936	<b>0.6067</b>	28.352	0.6534	27.637	0.6493	28.086	0.5617	28.007	0.563	28.052	0.7758
FA		27.866	0.5727	28.385	0.6553	27.97	0.6086	28.032	0.5294	27.901	<b>0.6788</b>	28.053	<b>0.7806</b>
HS		<b>28.053</b>	0.5673	<b>28.451</b>	0.6489	27.641	0.6515	27.95	0.5028	28.038	0.5269	27.937	0.7637
PSO		27.906	0.1544	28.106	0.2001	27.76	0.207	27.502	0.3822	27.486	0.4168	27.796	0.3593
SA		27.612	0.5587	28.34	0.6377	27.487	0.6257	27.943	0.504	<b>28.537</b>	0.6469	27.622	0.727
ABC		27.531	0.5804	28.355	0.6711	28.807	<b>0.7282</b>	27.888	0.4233	27.475	0.5839	<b>28.539</b>	0.7104
CLONALG		28.156	0.6219	28.35	0.6841	28.682	0.6816	28.142	0.6212	28.075	0.6818	27.709	0.7938
DE		27.607	0.5367	28.131	0.6416	28.224	0.686	27.792	0.5955	27.929	0.6441	28.11	0.7869
GA		<b>28.595</b>	0.6107	28.565	0.6959	27.598	0.6265	28.415	<b>0.6637</b>	27.519	0.6706	27.596	0.7282
GWO	5	28.471	<b>0.6488</b>	28.528	<b>0.7002</b>	28.757	0.6827	28.204	0.6295	27.65	0.6097	27.796	0.8013
GWO-HS		28.45	0.6451	28.646	0.698	28.86	0.681	28.168	0.6151	27.638	0.6091	27.927	0.8083
FA		28.055	0.544	28.5	0.6846	27.419	0.6109	28.194	0.5969	<b>28.737</b>	<b>0.6996</b>	28.417	<b>0.8172</b>
HS		27.971	0.6129	<b>28.668</b>	0.6969	<b>28.908</b>	0.6819	<b>28.53</b>	0.6488	27.938	0.6511	28.273	0.8118
PSO		27.751	0.2505	27.907	0.4025	28.063	0.0732	27.626	0.3197	27.637	0.2395	27.927	0.4763
SA		27.633	0.5325	28.144	0.6529	28.717	0.6748	28.524	0.6093	28.457	0.5572	28.036	0.8018
ABC		27.845	0.5216	28.831	0.7096	28.117	0.7124	27.612	0.5035	27.824	0.6773	<b>29.124</b>	0.8606
CLONALG		28.097	0.645	28.549	0.7271	28.414	0.6925	28.045	0.6638	28.185	0.7503	28.085	0.8162
DE		28.011	0.6754	28.832	0.7405	28.449	0.7075	<b>28.8</b>	<b>0.702</b>	27.845	0.6924	28.929	<b>0.8677</b>
GA		<b>28.795</b>	<b>0.7201</b>	27.792	0.7096	28.204	0.6972	28.036	0.6632	27.807	0.6323	27.479	0.8105
GWO	6	28.395	0.6824	28.524	<b>0.7422</b>	27.915	<b>0.7447</b>	28.47	0.7002	<b>28.449</b>	<b>0.7795</b>	28.302	0.8545
GWO-HS		28.428	0.6784	28.32	0.721	28.326	0.6991	28.599	0.7003	28.351	0.7642	27.901	0.8197
FA		28.452	0.6475	<b>28.92</b>	0.7394	27.842	0.6854	28.346	0.6455	27.987	0.7124	27.892	0.8167
HS		27.871	0.6177	28.617	0.7139	28.309	0.6865	27.629	0.6104	28.024	0.5777	27.647	0.6987
PSO		27.816	0.2316	27.24	0.2916	27.853	0.4704	27.933	0.168	27.575	0.3584	27.962	0.2417
SA		27.58	0.5274	27.64	0.6996	<b>28.575</b>	0.7204	28.089	0.585	27.903	0.6106	28.327	0.8362
ABC	7	27.899	0.6499	28.143	0.7206	27.306	0.7404	<b>29.018</b>	0.724	27.624	0.7473	28	0.8064
CLONALG		27.927	0.6683	29.254	0.7662	29.197	0.7403	28.72	0.6963	27.744	0.7461	28.155	0.8386

DE	<b>28.953</b>	<b>0.7706</b>	28.982	0.7505	28.563	0.6778	28.039	0.7298	<b>28.574</b>	0.7196	<b>28.947</b>	<b>0.8894</b>
GA	28.435	0.7205	27.591	0.6208	27.829	0.6812	27.662	0.7157	28.194	0.6824	28.656	0.8323
GWO	28.254	0.713	29.083	0.7537	<b>29.804</b>	<b>0.7836</b>	28.398	<b>0.7394</b>	27.422	0.7743	28.547	0.8843
GWO-HS	28.147	0.6967	<b>29.326</b>	0.7657	29.678	0.7679	28.599	0.7253	28.103	<b>0.7774</b>	28.729	0.8886
FA	28.539	0.6456	28.688	0.7226	29.557	0.7607	27.769	0.6941	28.095	0.7347	27.534	0.856
HS	28.474	0.6952	29.237	<b>0.7722</b>	29.25	0.7378	28.637	0.7194	27.419	0.7711	28.565	0.8751
PSO	27.768	0.3903	28.126	0.1438	27.515	0.39	27.963	0.0863	27.596	0.4639	27.901	0.1773
SA	27.63	0.7169	28.057	0.7232	29.142	0.7616	27.685	0.6774	28.455	0.743	28.403	0.8428
ABC	28.737	0.7545	28.863	0.735	28.317	0.757	27.708	0.6334	<b>29.493</b>	0.7546	27.962	0.811
CLONALG	28.491	0.7116	29.267	0.788	28.731	0.7787	27.964	0.7492	28.014	0.7824	28.969	0.8929
DE	27.972	0.7253	27.378	0.7251	28.548	0.7615	27.418	0.6568	28.765	0.7591	27.351	0.8807
GA	29.028	0.7644	29.007	0.7337	28.393	0.7012	27.648	0.2921	27.504	0.7337	28.741	0.8165
GWO	<b>29.115</b>	<b>0.7729</b>	29.091	0.7916	<b>29.415</b>	<b>0.8005</b>	<b>29.077</b>	<b>0.7745</b>	27.321	0.7836	27.696	0.8591
GWO-HS	28.562	0.7225	<b>29.774</b>	<b>0.7996</b>	28.667	0.7747	28.583	0.7605	28.044	<b>0.7955</b>	28.81	0.8953
FA	27.42	0.6914	28.183	0.7759	27.344	0.7702	28.832	0.7579	28.588	0.7374	28.565	<b>0.9077</b>
HS	28.064	0.7557	29.043	0.7543	28.194	0.7765	29	0.7667	28.442	0.7489	<b>29.094</b>	0.8888
PSO	27.841	0.42	28.322	0.4407	27.795	0.3455	27.948	0.108	27.435	0.4267	27.816	0.3865
SA	28.223	0.6892	28.517	0.7729	28.517	0.7259	27.939	0.6144	28.028	0.7681	28.801	0.877
ABC	27.605	0.644	29.728	0.7837	27.712	0.7787	27.808	0.4751	27.637	0.656	28.421	0.8781
CLONALG	28.059	0.7125	27.941	0.7878	29.736	0.7916	<b>29.183</b>	0.7778	27.467	0.7801	28.241	0.8884
DE	<b>29.312</b>	<b>0.8126</b>	29.599	0.7887	28.426	0.7767	27.704	0.699	27.688	0.7472	27.957	0.8532
GA	27.684	0.6252	28.379	0.7881	28.052	0.7903	28.307	0.7154	27.461	0.7073	<b>29.064</b>	0.8879
GWO	28.426	0.7843	27.71	0.8033	29.027	0.7963	28.966	<b>0.8035</b>	27.391	0.7976	28.483	0.8627
GWO-HS	28.725	0.7596	<b>30.197</b>	<b>0.8356</b>	<b>30.082</b>	<b>0.8196</b>	28.592	0.7784	<b>28.491</b>	<b>0.8326</b>	28.954	<b>0.9083</b>
FA	28.013	0.708	27.867	0.7339	29.811	0.7889	27.669	0.5417	27.795	0.7174	27.611	0.8979
HS	28.485	0.7699	29.436	0.8276	27.757	0.7706	27.814	0.7084	27.883	0.7773	27.99	0.7795
PSO	28.146	0.4191	27.56	0.3593	27.604	0.3109	27.283	0.2787	27.679	0.3496	27.803	0.3513
SA	27.843	0.6806	29.766	0.8112	28.085	0.772	28.839	0.7373	27.77	0.79	28.551	0.8437
ABC	27.864	0.7302	28.561	0.813	27.581	0.7594	27.82	0.7436	28.685	0.7884	27.655	0.8535
CLONALG	29.023	0.7615	29.873	0.8267	29.598	0.8166	<b>29.259</b>	0.7813	28.004	0.8252	28.754	0.9192
DE	28.371	0.6947	27.286	0.6964	28.046	0.8031	27.632	0.7322	27.99	0.7855	27.997	0.8669
GA	<b>29.518</b>	0.8069	29.097	0.7746	27.723	0.7475	28.09	0.6807	28.031	0.7884	28.534	0.85
GWO	29.335	<b>0.8339</b>	27.418	0.7936	28.761	0.7639	27.82	0.774	28.507	0.809	28.42	0.888
GWO-HS	28.549	0.7643	<b>30.213</b>	<b>0.8449</b>	<b>29.816</b>	<b>0.838</b>	29.259	<b>0.8111</b>	28.65	<b>0.8517</b>	<b>28.792</b>	0.9135
FA	28.613	0.8128	28.109	0.7982	29.69	0.7986	28.321	0.7054	28.107	0.779	28.44	<b>0.9209</b>
HS	29.216	0.7934	28.787	0.7989	27.952	0.7494	29.015	0.7767	29.325	0.8263	27.912	0.8966
PSO	27.45	0.3213	27.938	0.4432	27.732	0.3451	27.642	0.1994	27.716	0.2447	28.246	0.4626
SA	27.539	0.6932	27.384	0.6854	27.818	0.7832	27.669	0.717	<b>29.766</b>	0.8149	27.732	0.847

Çizelge 8’de Otsu yöntemi için hesaplanan PSNR ve SSIM değerleri verilmiştir. Tüm eşik değerlerinde sürekli en iyi değeri veren bir algoritma bulunmamaktadır. SA’nın diğer algoritmalara göre kısmen daha başarılı sonuçlar verdiği gözlemlenmiştir.

Çizelge 8. Otsu yöntemi PSNR ve SSIM değerleri

Alg	K	Barbara		Living Room		Boats		Goldhill		Lake		Aerial	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ABC		<b>27.79</b>	0.4743	28.089	0.3864	27.839	0.4897	27.874	0.1428	27.567	0.46	27.764	0.3676
CA		<b>27.79</b>	0.4675	28.076	0.4491	27.827	0.4518	<b>27.928</b>	0.3701	27.577	0.4821	27.735	0.4528
DE		27.76	0.4687	28.093	<b>0.4611</b>	27.843	0.4239	27.888	<b>0.3769</b>	27.564	0.4805	27.734	0.4387
GA		27.785	0.3869	28.068	0.4318	27.84	<b>0.5077</b>	27.868	0.2812	27.564	0.4378	27.78	0.3863
GWO	2	27.76	0.4696	28.105	0.4447	27.843	0.4572	27.905	0.3667	27.553	0.4849	27.736	0.4555
GWO-HS		27.76	0.4696	28.105	0.4447	27.843	0.4572	27.905	0.3667	27.553	0.4849	27.736	0.4555
FA		27.779	0.4739	28.093	0.4405	<b>27.845</b>	0.4784	27.863	0.3589	<b>27.579</b>	0.4756	27.736	0.4538
HS		27.766	<b>0.4825</b>	28.078	0.4429	27.841	0.4267	27.867	0.374	27.566	0.4867	27.748	0.4693
PSO		27.771	0.3449	<b>28.105</b>	0.3835	27.834	0.4771	27.896	0.2557	27.549	0.4234	27.773	0.4109
SA		27.783	0.4727	28.078	0.4242	27.829	0.4708	27.908	0.3473	<b>27.579</b>	<b>0.4885</b>	<b>27.793</b>	<b>0.4967</b>
ABC		27.721	0.4988	<b>27.978</b>	0.5507	27.772	0.4631	27.883	<b>0.5451</b>	27.517	<b>0.5955</b>	<b>28.234</b>	0.6883
CA		27.984	0.5441	27.785	0.5255	27.579	0.576	27.815	0.4298	27.956	0.5099	27.653	0.5772
DE		27.954	0.5696	27.691	0.4761	27.919	0.5523	27.69	0.4269	<b>27.988</b>	0.5015	27.686	0.4824
GA		27.694	0.4747	27.687	0.5325	27.919	<b>0.6101</b>	27.524	0.3935	27.686	0.5606	28.216	<b>0.7219</b>
GWO	3	<b>28.027</b>	0.5426	27.682	0.5147	27.669	0.5709	<b>27.93</b>	0.4413	27.922	0.5136	27.604	0.5667
GWO-HS		<b>28.027</b>	0.5426	27.682	0.5147	27.669	0.5709	27.929	0.4417	27.922	0.5136	27.604	0.5667
FA		27.867	0.5239	27.664	0.4881	27.78	0.5753	27.542	0.4113	27.67	0.5579	27.804	0.4739
HS		28.004	0.5336	27.592	0.4934	27.784	0.5572	27.575	0.4015	27.934	0.4683	27.802	0.6098
PSO		27.802	0.3366	27.27	0.3705	27.805	0.4735	27.314	0.2256	27.436	0.4026	27.651	0.397
SA		27.908	<b>0.5742</b>	27.607	<b>0.5517</b>	<b>27.924</b>	0.6018	27.869	0.5184	27.898	0.5036	27.584	0.5824
ABC		27.741	<b>0.6886</b>	28.041	0.5374	27.727	0.5571	27.717	0.5143	28.14	0.56	<b>28.127</b>	<b>0.785</b>
CA		27.823	0.5831	27.921	0.6302	27.665	0.5656	27.875	0.5124	28.095	0.5625	27.741	0.6326
DE		27.87	0.6073	28.179	0.6052	27.979	0.5961	27.838	<b>0.6016</b>	<b>28.362</b>	<b>0.6309</b>	27.535	0.4783
GA		27.587	0.5148	27.899	0.475	27.696	<b>0.6763</b>	27.917	0.552	27.549	0.4311	27.875	0.7772
GWO	4	27.807	0.585	28.036	0.6123	27.877	0.5656	28.031	0.5352	28.078	0.58	27.752	0.6246
GWO-HS		27.807	0.585	28.128	0.6041	27.791	0.57	28.08	0.5399	28.079	0.5756	27.679	0.6179
FA		27.875	0.5397	28.226	0.6065	<b>28.027</b>	0.5896	28.012	0.4973	28.104	0.5795	27.621	0.5258
HS		27.79	0.5438	27.596	0.5349	27.973	0.6098	28.089	0.5306	28.196	0.589	27.807	0.6968
PSO		27.672	0.3422	27.591	0.3542	27.412	0.4468	27.438	0.346	27.595	0.5255	27.756	0.3304
SA		<b>27.944</b>	0.572	<b>28.354</b>	<b>0.6466</b>	27.825	0.5664	<b>28.14</b>	0.5859	28.064	0.6246	27.722	0.7574
ABC		27.878	0.5643	28.043	0.6049	27.769	0.6265	<b>28.359</b>	<b>0.6256</b>	27.647	0.5023	27.58	0.7273
CA	5	27.69	0.5899	27.843	0.6239	27.923	0.5537	28.304	0.5971	27.59	0.6034	27.961	0.638
DE		<b>28.853</b>	<b>0.7054</b>	27.916	0.575	27.453	0.5824	27.549	0.5429	27.943	0.5773	27.467	0.7153

GA	28.358	0.6232	<b>28.457</b>	0.6377	27.532	0.582	28.253	0.6105	28.009	0.6354	<b>28.473</b>	<b>0.8256</b>
GWO	27.786	0.6237	28.04	0.6445	28.157	0.6367	27.999	0.585	<b>28.128</b>	<b>0.6765</b>	28.012	0.8135
GWO-HS	27.767	0.6287	27.973	0.6617	28.191	0.6323	27.964	0.5866	27.743	0.6349	27.867	0.6487
FA	28.473	0.6214	27.934	0.6636	27.951	0.551	27.719	0.471	27.614	0.6348	27.598	0.6606
HS	28.042	0.6095	27.443	0.5899	28.544	0.5966	28.055	0.5516	27.803	0.5986	27.535	0.7578
PSO	27.445	0.387	27.544	0.3756	27.596	0.4589	27.64	0.2339	27.513	0.4181	27.512	0.4146
SA	28.085	0.6299	28.377	<b>0.669</b>	<b>28.999</b>	<b>0.6973</b>	28.036	0.6102	27.88	0.6344	28.135	0.819
ABC	28.361	<b>0.6841</b>	<b>28.952</b>	<b>0.7292</b>	28.267	0.6716	27.514	0.5692	28.329	0.7162	27.881	0.7898
CA	27.87	0.6281	28.405	0.6998	28.163	<b>0.7154</b>	<b>28.358</b>	0.6489	27.644	0.6711	27.958	<b>0.8114</b>
DE	<b>28.384</b>	0.635	27.357	0.7131	<b>28.684</b>	0.7104	27.832	0.5955	28.407	0.6343	27.412	0.7242
GA	28.227	0.5747	28.649	0.6901	28.443	0.7147	28.312	0.4781	27.844	0.5629	<b>28.765</b>	0.8068
GWO	28.034	0.6699	27.938	0.674	27.847	0.6559	28.073	0.6356	27.37	0.6153	27.812	0.7769
GWO-HS	27.905	0.6624	28.023	0.7084	27.89	0.6561	27.996	0.6295	27.845	0.691	27.915	0.8061
FA	27.799	0.6673	27.754	0.6794	27.664	0.6067	28.123	0.464	27.364	0.6679	27.906	0.7577
HS	27.563	0.6573	28.152	0.6695	27.913	0.673	27.805	<b>0.6562</b>	28.036	0.7164	27.93	0.7972
PSO	27.612	0.3367	27.536	0.4017	27.439	0.4889	27.626	0.2912	27.458	0.4766	27.691	0.3675
SA	27.935	0.6033	27.647	0.6163	28.129	0.6966	27.836	0.6098	<b>28.537</b>	<b>0.7308</b>	27.674	0.7559
ABC	<b>28.685</b>	<b>0.7819</b>	28.205	0.6956	27.543	<b>0.7501</b>	<b>29.096</b>	<b>0.7332</b>	27.839	0.7166	<b>28.637</b>	<b>0.8398</b>
CA	27.857	0.6959	28.542	<b>0.7499</b>	28.447	0.6696	28.102	0.6723	27.559	0.6765	27.794	0.8176
DE	27.474	0.5837	27.809	0.7255	27.992	0.7153	27.644	0.6084	<b>28.903</b>	0.725	28.319	0.8225
GA	28.195	0.7311	27.9	0.496	29.046	0.7203	27.671	0.6382	27.873	0.7045	27.688	0.6877
GWO	27.953	0.6742	28.374	0.698	28.268	0.6703	28.248	0.7144	28.742	<b>0.7409</b>	28.002	0.7797
GWO-HS	27.994	0.7126	28.27	0.7369	28.445	0.6839	28.06	0.6685	28.488	0.6937	27.887	0.8112
FA	27.546	0.5672	27.482	0.6968	28.19	0.6783	27.767	0.564	27.555	0.7034	27.993	0.7544
HS	28.213	0.7088	28.253	0.6906	<b>29.08</b>	0.7332	27.939	0.6065	27.435	0.6782	27.715	0.8182
PSO	27.77	0.3431	27.44	0.4037	27.281	0.4234	27.426	0.2768	27.497	0.4508	27.804	0.3898
SA	28.016	0.667	<b>28.791</b>	0.7363	28.748	0.7063	28.174	0.5267	28.176	0.6818	27.605	0.7969
ABC	28.541	0.6792	27.372	0.6672	27.814	0.7076	28.655	<b>0.756</b>	27.528	0.6065	27.359	0.86
CA	28.51	0.6925	28.238	0.7838	28.52	0.6955	27.955	0.6615	27.711	0.6725	27.863	0.7887
DE	29.157	0.817	<b>28.39</b>	0.7322	<b>29.048</b>	0.7698	28.592	0.6963	28.088	<b>0.793</b>	28.432	0.7547
GA	27.788	0.6649	27.232	0.7365	27.793	0.6608	28.643	0.7545	27.461	0.728	27.58	0.7169
GWO	27.988	<b>0.8188</b>	28.004	<b>0.7917</b>	28.008	0.7184	28.54	0.6957	<b>28.851</b>	0.7904	28.047	0.8627
GWO-HS	28.052	0.723	28.061	0.7577	28.02	0.7232	28.131	0.7157	28.28	0.7351	28.061	0.8259
FA	<b>29.324</b>	0.767	28.156	0.7772	28.189	0.7118	<b>28.736</b>	0.7538	28.354	0.7329	<b>28.616</b>	0.8792
HS	28.418	0.7828	27.999	0.7364	27.503	0.7386	28.544	0.6997	28.208	0.7379	28.196	<b>0.884</b>
PSO	27.649	0.3984	27.171	0.4289	27.731	0.4655	27.547	0.2554	27.52	0.4873	27.32	0.4531
SA	29.306	0.7968	28.313	0.7798	28.836	<b>0.7703</b>	27.643	0.7453	28.375	0.703	28.553	0.87
ABC	27.539	0.7355	28.324	0.7916	<b>29.696</b>	0.7957	27.749	0.528	28.194	0.6104	28.3	0.8525
CA	27.879	0.7144	28.557	0.8029	28.012	0.7276	28.497	0.7377	27.4	0.6775	28.727	0.8632
DE	<b>28.5</b>	<b>0.7883</b>	27.764	0.6743	28.705	0.733	<b>29.667</b>	0.7402	27.696	0.751	28.211	0.8457
GA	27.865	0.6586	<b>29.309</b>	0.7665	28.264	0.704	28.349	0.7851	28.385	0.7558	27.852	0.6209
GWO	27.762	0.7165	28.294	0.7838	27.934	0.7444	28.077	0.7838	27.631	<b>0.8068</b>	<b>28.873</b>	<b>0.883</b>

GWO-HS	28.15	0.7417	28.509	0.7762	28.391	0.7574	28.188	0.7451	28.163	0.7746	27.943	0.8244	
FA	27.719	0.6953	28.518	0.7132	28.288	0.7532	27.685	0.6956	<b>29.354</b>	0.803	28.01	0.7355	
HS	27.566	0.6409	28.375	0.7952	28.723	0.7018	28.986	<b>0.7883</b>	28.121	0.6474	27.908	0.8691	
PSO	27.436	0.3681	27.798	0.4051	27.241	0.4456	27.465	0.2732	27.462	0.4675	28.053	0.401	
SA	27.676	0.7523	27.563	<b>0.8043</b>	28.217	<b>0.8011</b>	27.667	0.7155	27.95	0.7107	27.357	0.875	
ABC	27.648	0.6197	28.584	0.7764	27.625	0.7361	27.998	0.6077	28.527	0.7232	28.12	0.7132	
CA	28.149	0.7224	27.435	0.7998	28.245	0.7749	27.691	0.7697	27.868	0.7689	28.205	0.8417	
DE	27.97	0.7275	27.669	<b>0.8116</b>	28.154	0.7801	27.842	0.7529	27.862	0.7915	27.925	0.8549	
GA	27.998	0.6969	27.993	0.7083	<b>29.556</b>	<b>0.8132</b>	27.709	0.6904	27.639	0.4892	28.186	0.9079	
GWO	10	28.569	0.7923	<b>28.669</b>	0.7797	28.427	0.7393	28.288	<b>0.8035</b>	27.833	0.786	27.505	0.8047
GWO-HS		28.381	0.7704	28.376	0.8045	28.301	0.7651	28.25	0.7582	28.325	0.7883	28.086	0.8228
FA		<b>30.07</b>	<b>0.852</b>	27.478	0.7913	28.866	0.7857	28.77	0.7637	28.386	0.6093	27.671	0.8922
HS		28.645	0.8386	28.009	0.7712	29.136	0.7718	28.117	0.7676	27.819	0.7549	28.435	0.885
PSO		27.576	0.3569	27.45	0.4395	27.893	0.4752	27.597	0.2401	27.56	0.4623	27.808	0.3433
SA		28.561	0.7748	28.495	0.7834	28.268	0.8019	<b>29.115</b>	0.7792	<b>28.662</b>	<b>0.8121</b>	<b>28.608</b>	<b>0.9115</b>

Çizelge 9’da Tsallis yöntemi için hesaplanan PSNR ve SSIM değerleri verilmiştir. Tüm eşik değerlerinde sürekli en iyi değeri veren bir algoritma bulunmamaktadır. GWO-HS algoritmasının diğer algoritmalara göre kısmen daha başarılı sonuçlar verdiği gözlemlenmiştir.

Çizelge 9. Tsallis yöntemi PSNR ve SSIM değerleri

Alg	K	Barbara		Living Room		Boats		Goldhill		Lake		Aerial	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ABC		27.758	<b>0.4761</b>	28.072	<b>0.46</b>	27.827	0.5447	27.867	0.3747	27.553	0.4403	27.78	0.3906
CA		27.774	0.4302	28.083	0.4578	27.837	0.5423	27.911	0.3796	27.553	0.4639	27.734	0.4409
DE		27.744	0.3965	28.075	0.4419	27.839	0.5011	27.915	0.402	27.567	0.4593	27.744	0.4659
GA		27.769	0.444	28.085	0.0481	27.84	0.4446	27.937	0.2984	27.54	0.4461	27.777	0.4729
GWO	2	<b>27.79</b>	0.4348	<b>28.097</b>	0.457	27.837	0.5423	27.888	0.3763	27.553	0.4639	27.735	0.4369
GWO-HS		<b>27.79</b>	0.4348	<b>28.097</b>	0.457	27.837	0.5423	27.888	0.3763	27.553	0.4639	27.735	0.4369
FA		27.771	0.4005	28.093	0.453	27.827	<b>0.5458</b>	27.927	<b>0.4065</b>	27.549	0.4334	27.734	0.4416
HS		27.759	0.4256	28.078	0.4434	27.826	0.5024	27.888	0.347	<b>27.57</b>	<b>0.4687</b>	27.735	0.4541
PSO		27.75	0.3484	28.057	0.1168	27.84	0.4633	27.87	0.1737	27.551	0.4588	27.768	0.3895
SA		27.779	0.4424	28.091	0.4242	<b>27.847</b>	0.5444	<b>27.938</b>	0.3381	27.546	0.4533	<b>27.814</b>	<b>0.4899</b>
ABC		27.665	0.2934	<b>28.192</b>	0.4701	28.012	0.494	27.327	0.3749	<b>27.952</b>	0.545	28.15	<b>0.728</b>
CA		27.967	0.5287	27.714	0.5616	27.929	0.6128	27.802	0.4377	27.705	0.5096	27.623	0.5259
DE	3	28.005	0.541	27.697	0.5504	27.945	0.6043	27.689	0.4157	27.677	0.5159	<b>28.337</b>	0.6893
GA		27.756	0.524	27.88	0.5552	<b>28.069</b>	0.4992	27.563	0.393	27.578	<b>0.572</b>	27.468	0.6455
GWO		27.964	0.5394	27.847	<b>0.5876</b>	27.916	0.6191	27.822	0.4384	27.752	0.5225	27.617	0.5228

GWO-HS	27.964	0.5394	27.847	<b>0.5876</b>	27.897	0.6169	27.822	0.4384	27.752	0.5225	27.626	0.5042	
FA	27.94	0.5319	27.684	0.5811	27.986	0.6017	27.785	0.4393	27.686	0.4794	27.702	0.4796	
HS	27.962	0.529	27.617	0.5673	27.937	<b>0.6265</b>	<b>27.974</b>	<b>0.4532</b>	27.683	0.5385	27.502	0.4873	
PSO	27.751	0.0794	28.07	0.0582	27.433	0.4502	27.344	0.2546	27.46	0.4577	27.386	0.4743	
SA	<b>28.049</b>	<b>0.5423</b>	27.623	0.5602	27.946	0.6155	27.741	0.4279	27.894	0.5334	28.142	0.7252	
ABC	28.026	<b>0.6182</b>	<b>28.385</b>	0.593	28.028	0.5976	28.033	0.4703	27.803	<b>0.6505</b>	27.734	0.674	
CA	27.877	0.5821	28.137	0.6434	27.639	<b>0.6568</b>	27.872	0.5153	28.038	0.58	28.088	0.7841	
DE	27.716	0.542	28.34	0.6487	<b>28.045</b>	0.621	28.01	0.5306	27.893	0.5402	27.745	0.7198	
GA	27.445	0.4402	27.807	0.5441	27.951	0.5468	28.075	0.4845	27.502	0.4606	27.864	0.6939	
GWO	<b>4</b>	27.932	0.6073	28.247	0.646	27.659	0.6481	28.074	0.5431	27.689	0.568	28.004	0.7697
GWO-HS		27.932	0.6073	28.292	<b>0.6489</b>	27.626	0.6505	28.074	0.5431	27.658	0.5655	28.004	0.7697
FA		27.789	0.6083	28.232	0.6416	27.885	0.6445	27.581	0.323	28.089	0.5722	<b>28.104</b>	<b>0.785</b>
HS		27.842	0.587	27.821	0.6001	27.913	0.5774	27.658	0.3607	28.066	0.5658	27.778	0.7501
PSO		27.624	0.3213	27.998	0.2142	27.556	0.4284	27.743	0.2604	27.715	0.4473	27.475	0.4282
SA		<b>28.136</b>	0.614	28.053	0.5888	27.454	0.6312	<b>28.289</b>	<b>0.5858</b>	<b>28.299</b>	0.5964	27.916	0.7508
ABC		27.313	0.4995	27.645	0.6312	28.09	0.5367	27.411	0.5919	<b>28.447</b>	<b>0.7136</b>	27.942	0.7528
CA		28.002	0.6157	27.903	0.6694	28.715	0.6758	28.04	0.5839	27.647	0.6328	27.823	0.7742
DE		28.055	0.5545	28.003	0.6438	28.23	0.6748	27.939	0.5882	28.065	0.6665	28.033	0.7374
GA		<b>28.275</b>	<b>0.7038</b>	27.751	0.5654	28.755	0.6115	27.535	0.3455	27.598	0.5093	27.925	0.6958
GWO	<b>5</b>	28.004	0.6356	27.773	<b>0.6913</b>	27.74	0.6744	28.034	0.5729	27.604	0.5982	28.027	0.7723
GWO-HS		27.976	0.6324	27.766	0.6898	28.397	0.6618	28.12	0.5779	27.604	0.5982	27.891	<b>0.8069</b>
FA		28.237	0.6396	27.767	0.686	27.868	0.6962	27.547	0.4493	27.902	0.5626	27.814	0.706
HS		27.835	0.5836	27.573	0.6771	<b>28.874</b>	<b>0.6963</b>	<b>28.43</b>	0.6157	27.786	0.536	<b>28.086</b>	0.7949
PSO		27.906	0.3847	27.867	0.3108	27.879	0.4882	27.922	0.1423	27.622	0.3599	28.026	0.2303
SA		28.265	0.6265	<b>28.151</b>	0.6164	28.711	0.6441	28.288	<b>0.6211</b>	27.679	0.5677	27.737	0.7736
ABC		27.593	0.4242	27.76	0.7349	27.953	0.6632	28.153	0.5819	<b>28.451</b>	0.677	27.567	0.7526
CA		28.088	0.6333	<b>28.838</b>	0.7127	27.86	<b>0.7238</b>	28.496	0.6633	27.799	0.6785	27.789	0.7922
DE		28.587	0.6868	27.867	0.6787	27.944	0.7001	28.433	0.6215	28.041	<b>0.7041</b>	27.733	0.7878
GA		<b>28.618</b>	<b>0.7417</b>	28.084	0.6624	27.728	0.4763	<b>28.758</b>	0.681	27.92	0.6896	27.615	0.576
GWO	<b>6</b>	28.227	0.6786	28.394	<b>0.7421</b>	27.823	0.7221	28.196	0.644	27.771	0.658	27.804	0.808
GWO-HS		28.035	0.6681	28.477	0.737	27.687	0.7079	28.081	0.6427	27.578	0.6266	27.691	0.8113
FA		27.532	0.6399	28.779	0.7335	<b>28.359</b>	0.6428	28.618	0.6111	27.803	0.6335	27.784	0.7635
HS		27.663	0.6276	28.318	0.6703	27.652	0.7219	27.788	0.6041	27.882	0.5569	27.739	0.7825
PSO		27.374	0.3152	27.758	0.2282	28.199	0.5118	27.365	0.2282	27.542	0.4403	27.848	0.1239
SA		28.421	0.6601	27.695	0.7051	27.687	0.6726	28.339	<b>0.6921</b>	28.028	0.5688	<b>28.125</b>	<b>0.8352</b>
ABC		27.945	0.6388	27.683	0.6803	27.817	0.7123	27.623	0.398	27.95	0.6855	27.501	0.6866
CA		28.172	0.6425	27.898	0.7575	28.697	0.7248	28.161	0.7028	27.515	0.7072	28.082	0.8335
DE		27.683	0.5278	27.825	0.7365	27.537	0.7349	28.173	0.6418	<b>29.043</b>	<b>0.7702</b>	28.109	0.8476
GA	<b>7</b>	28.338	<b>0.731</b>	28.279	0.643	29.192	0.7373	28.429	0.6273	28.054	0.7036	<b>28.746</b>	0.8573
GWO		28.055	0.723	27.457	0.7628	27.551	<b>0.7633</b>	28.028	0.7253	27.453	0.6469	28.6	<b>0.8829</b>
GWO-HS		28.074	0.6898	<b>28.7</b>	<b>0.7702</b>	29.04	0.733	<b>28.559</b>	<b>0.7312</b>	27.473	0.6628	28.058	0.8022
FA		<b>28.369</b>	0.7154	28.447	0.7379	<b>29.525</b>	0.7612	27.524	0.6721	28.075	0.6264	27.787	0.7568

HS	27.661	0.6385	28.565	0.7219	27.814	0.7346	27.644	0.5572	28.589	0.6345	28.348	0.8614	
PSO	27.774	0.2529	28.128	0.1467	27.918	0.4634	27.899	0.0754	27.796	0.4784	27.78	0.3909	
SA	27.602	0.6408	27.928	0.6859	28.892	0.7222	28.275	0.6412	29.007	0.7561	27.489	0.7719	
ABC	28.593	0.707	28.191	0.6944	27.895	0.7357	27.836	0.594	27.753	0.6681	28.117	0.7832	
CA	28.66	<b>0.7416</b>	28.384	0.7775	28.723	<b>0.7826</b>	27.726	0.7214	28.063	0.682	27.504	0.8005	
DE	27.835	0.6439	28.712	0.7087	28.675	0.7291	<b>29.013</b>	0.7264	<b>28.695</b>	0.6922	28.264	0.8539	
GA	27.836	0.5581	27.775	0.704	28.102	0.7298	28.335	0.6325	27.466	0.6504	<b>29.228</b>	0.8799	
GWO	8	27.984	0.7281	27.502	0.7783	27.524	0.7746	28.477	<b>0.7601</b>	27.67	<b>0.788</b>	28.179	<b>0.9041</b>
GWO-HS		28.403	0.7125	27.995	<b>0.8015</b>	28.149	0.7718	28.531	0.7578	27.447	0.6984	28.013	0.8164
FA		28.63	0.6311	28.54	0.7338	28.687	0.7536	27.805	0.7284	27.708	0.7633	28.112	0.8401
HS		28.256	0.7204	<b>29.022</b>	0.7576	<b>28.984</b>	0.7458	29.003	0.7553	28.493	0.7817	28.102	0.7812
PSO		27.779	0.393	28.24	0.4274	27.948	0.478	27.603	0.1927	27.486	0.4781	27.929	0.3266
SA		<b>28.671</b>	0.7327	28.596	0.7449	27.506	0.6021	27.964	0.6851	27.673	0.6714	27.895	0.7792
ABC		27.488	0.5428	27.499	0.7793	28.527	0.7812	27.59	0.6884	27.47	0.7031	28.655	0.8782
CA		27.646	0.6906	28.39	0.8145	27.724	0.8077	<b>28.65</b>	0.7802	27.474	0.6203	27.684	0.8664
DE		27.439	0.7223	28.812	0.7735	27.63	0.7061	27.534	0.6563	27.943	0.7282	29.377	0.8906
GA		27.607	0.542	27.884	0.6478	<b>29.412</b>	0.724	27.845	0.7181	<b>28.116</b>	0.5606	27.863	0.7469
GWO	9	27.759	0.735	27.376	0.7709	27.553	0.7812	28.226	<b>0.7835</b>	27.513	<b>0.7682</b>	<b>29.387</b>	<b>0.9191</b>
GWO-HS		<b>28.646</b>	0.7597	28.093	<b>0.8157</b>	27.713	0.7938	28.561	0.7749	27.805	0.7397	27.878	0.8177
FA		28.394	0.6576	<b>29.175</b>	0.7994	27.895	<b>0.8136</b>	27.94	0.6811	27.532	0.7516	27.928	0.8675
HS		28.283	0.7083	28.258	0.7674	28.611	0.7517	28.246	0.6928	28.047	0.7642	28.964	0.895
PSO		27.921	0.3862	27.687	0.4081	27.867	0.508	27.497	0.3735	27.427	0.443	27.406	0.5153
SA		28.267	<b>0.7789</b>	27.47	0.7625	28.368	0.7744	27.929	0.7041	27.683	0.6623	27.59	0.8182
ABC		27.922	0.5824	28.324	0.7893	27.492	0.6697	27.349	0.4043	28.346	0.6107	<b>29.474</b>	<b>0.9015</b>
CA		28.546	0.7504	27.923	0.811	27.869	0.7934	27.829	0.7751	27.468	0.7	27.868	0.8419
DE		28.514	0.7756	28.738	0.8121	27.26	0.7935	<b>29.762</b>	<b>0.7981</b>	<b>28.455</b>	0.803	28.244	0.8835
GA		27.255	0.6448	28.742	0.7305	27.695	0.6998	29.073	0.777	27.909	0.7317	27.739	0.8395
GWO	10	27.56	0.7753	27.506	0.7419	27.629	0.8141	28.731	0.7788	27.751	0.7893	27.746	0.8446
GWO-HS		<b>28.709</b>	<b>0.7822</b>	28.015	<b>0.8372</b>	29.143	<b>0.8337</b>	28.65	0.7964	28.429	0.822	28.601	0.9005
FA		27.595	0.7618	28.894	0.793	<b>29.91</b>	0.7986	27.951	0.7538	28.378	0.8024	27.603	0.8594
HS		28.387	0.7624	<b>28.952</b>	0.776	28.583	0.8098	27.861	0.7755	27.572	0.619	27.663	0.8589
PSO		27.76	0.3286	28.1	0.0032	27.921	0.4625	27.962	0.1174	27.645	0.4677	27.47	0.4348
SA		27.981	0.6639	27.86	0.8036	29.082	0.743	27.417	0.7265	28.418	<b>0.8305</b>	27.713	0.8379

PSNR ve SSIM açısından yaptığımız deneylerden elde ettiğimiz sonuçlara göre düşük eşiklerde en iyi sonuçları veren bir optimizasyon algoritması bulunmamaktadır. Entropi tabanlı eşikleme yöntemlerinden Kapur ve Tsallis yöntemlerinde eşik sayısı arttıkça GWO-HS algoritmasının performansının öne çıktığı görülmüştür. GWO-HS algoritması Kapur ile Tsallis açısından kıyaslandığında ise Kapur yöntemi Tsallis yöntemine nazaran daha stabil sonuçlara

ulaşmıştır. Otsu eşikleme yöntemi için PSNR ve SSIM açısından sürekli iyi sonuçlar veren bir optimizasyon algoritması tespit edilememiştir.

GWO-HS algoritması ile test görselleri için 2 ile 10 eşik arasında bulunan eşik değerleri tablosu Çizelge 10’da verilmiştir. Kapur ve Tsallis algoritmaları ile hesaplanan eşik değerlerinin birbirlerine daha yakın değerler olduğu görülmüştür.

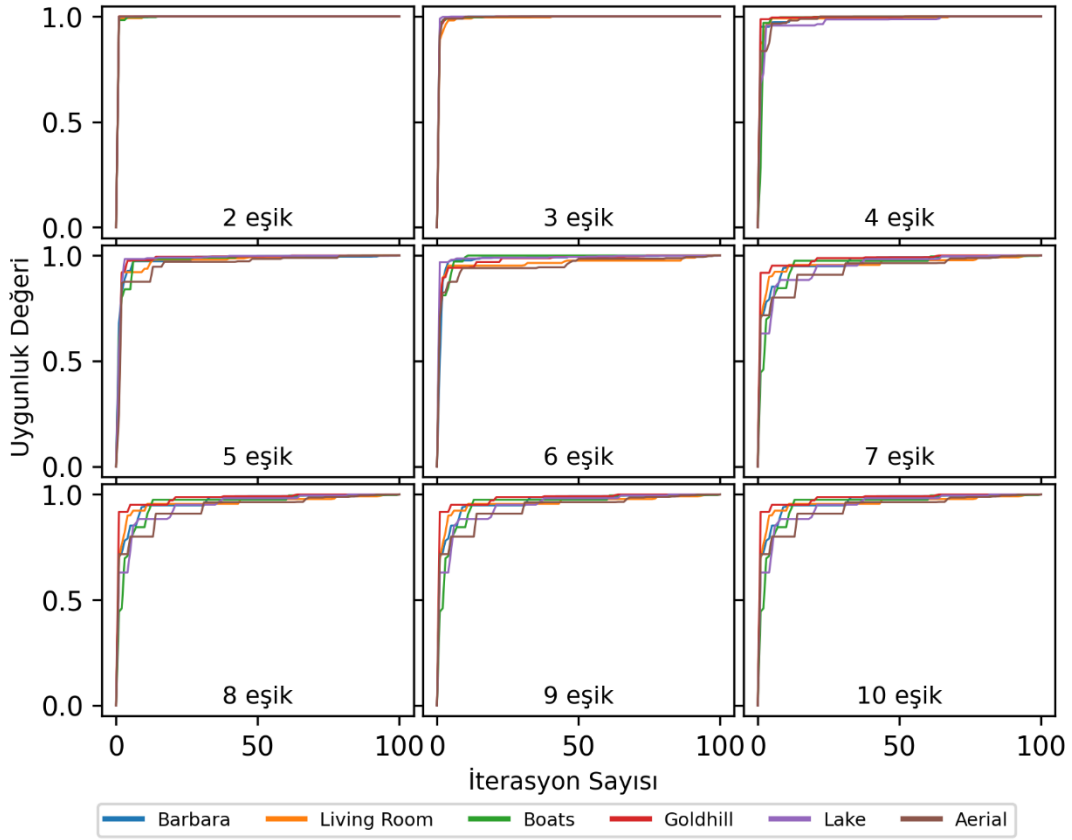
**Çizelge 10.** GWO-HS algoritması ile elde edilen eşik değerleri

	<b>K</b>	<b>Otsu</b>	<b>Kapur</b>	<b>Tsallis</b>
<b>BARBARA</b>	2	82, 147	96, 168	96, 168
	3	75, 127, 176	76, 127, 178	76, 127, 178
	4	66, 106, 142, 182	60, 99, 141, 185	60,97,137,181
	5	57, 88, 118, 148, 184	58, 95, 133, 172, 210	56,88,120,152,190
	6	54, 84, 112, 140, 167, 195	55, 87, 119, 151, 185, 221	54,83,112,140,169,197
	7	46, 70, 95, 118, 143, 169, 196	53, 82, 111, 140, 168, 194, 221	50,75,99,124,149,174,199
	8	45, 68, 91, 112, 133, 153, 174, 199,	49, 73, 97, 122, 147, 172, 196, 222	49,72,96,120,145,168,189,210
	9	42, 61, 81, 100, 118, 137, 156, 177, 201	45, 67, 88, 109, 131, 153, 176, 199, 222	45,67,88,109,130,152,175,198,222
	10	41, 59, 78, 96, 114, 132, 149, 167, 186, 207	42, 61, 80, 103, 123, 143, 162, 180, 200, 222	45,65,85,107,127,146,166,185,202,2 22
	<b>LIVING ROOM</b>	2	86, 144	91, 172
3		76, 124, 163	47, 104, 176	47,101,170
4		57, 98, 133, 169	47, 99, 150, 197	44,91,137,184
5		49, 89, 121, 147, 179	44, 88, 128, 167, 200	37,70,104,143,184
6		41, 75, 104, 128, 152, 182	41, 84, 122, 161, 198, 236	37,70,102,135,169,200
7		38, 71, 99, 122, 142, 163, 190	37, 71, 102, 137, 169, 200, 236	33,63,91,120,145,172,200
8		36, 66, 92, 112, 129, 147, 167, 193	28, 55, 83, 111, 142, 171, 200, 236	24,47,72,97,123,147,173,201
9		34, 60, 82, 103, 121, 136, 153, 172, 197	25, 49, 75, 100, 125, 150, 176, 202, 236	23,44,65,86,109,132,154,178,203
10		29, 52, 73, 93, 110, 125, 140, 156, 176, 201	22, 44, 66, 88, 110, 134, 159, 182, 204, 236	23,43,64,83,103,124,144,164,184,20 5
<b>BOATS</b>		2	93, 155	109, 179
	3	73, 126, 167	63, 120, 179	62,115,179
	4	67, 116, 149, 181	51, 91, 129, 179	52,89,128,180
	5	54, 95, 129, 154, 184	50, 90, 128, 166, 195	49,84,119,155,194
	6	46, 81, 115, 139, 158, 187	51, 90, 126, 163, 193, 227	43,72,100,129,159,196
	7	43, 74, 105, 130, 148, 165, 193	34, 65, 98, 129, 164, 193, 227	42,70,97,126,154,179,205
	8	38, 65, 94, 119, 138, 153, 169, 195	31, 61, 94, 125, 154, 178, 204, 234	35,59,82,105,129,154,179,205
	9	32, 55, 80, 106, 127, 143, 156, 173, 198	28, 55, 81, 105, 130, 155, 178, 204, 234	24,44,65,86,107,130,154,179,205
	10	31, 54, 78, 102, 123, 138, 150, 162, 178, 201	24, 45, 66, 89, 111, 131, 157, 179, 204, 234	24,44,63,86,108,130,153,176,194,21 2
	<b>GOLDHILL</b>	2	94, 161	91, 158
3		83, 126, 179	79, 134, 181	85,134,176
4		69, 102, 138, 186	64, 103, 144, 188	68,102,139,180

	5	63, 91, 117, 147, 191	59, 95, 132, 166, 200	67,100,134,165,195
	6	61, 88, 112, 138, 171, 207	45, 74, 104, 135, 167, 200	59,85,111,139,169,197
	7	56, 79, 100, 120, 144, 176, 211	45, 72, 100, 128, 153, 178, 205	44,70,96,122,149,174,200
	8	51, 71, 91, 111, 131, 153, 181, 212	43, 66, 89, 112, 136, 160, 185, 210	43,66,88,111,134,156,179,202
	9	48, 67, 85, 102, 118, 137, 159, 186, 214	43, 65, 87, 108, 130, 153, 174, 195, 217	43,63,83,103,123,144,165,186,206
	10	45, 64, 80, 96, 110, 125, 142, 163, 188, 214	41, 60, 78, 97, 116, 135, 155, 174, 195, 216	43,62,81,99,117,136,155,174,192,209
LAKE	2	85, 154	91, 163	95,157
	3	78, 140, 194	73, 120, 170	76,121,168
	4	67, 110, 158, 198	69, 112, 157, 195	66,101,135,171
	5	57, 88, 128, 167, 200	62, 96, 131, 166, 198	64,96,127,160,196
	6	51, 74, 104, 139, 171, 201	37, 69, 102, 135, 168, 199	60,87,114,141,168,198
	7	49, 70, 97, 129, 161, 184, 205	37, 65, 91, 118, 146, 172, 200	55,79,102,125,148,172,200
	8	46, 64, 86, 112, 141, 168, 189, 209	13, 37, 64, 91, 118, 145, 171, 199	49,70,91,111,132,152,173,200
	9	43, 59, 77, 99, 124, 151, 173, 191, 210	13, 35, 60, 82, 105, 128, 151, 174, 201	47,67,87,107,128,149,170,191,210
	10	39, 54, 70, 88, 110, 134, 157, 175, 191, 210	13, 34, 58, 80, 102, 124, 147, 170, 191, 211	43,61,81,98,116,134,152,172,192,211
	AERIAL	2	125, 178	68, 159
3		109, 147, 190	68, 130, 186	120,160,199
4		104, 134, 167, 202	68, 117, 159, 200	68,120,160,199
5		99, 123, 148, 175, 205	68, 108, 141, 174, 207	68,109,141,173,205
6		61, 101, 125, 150, 177, 206	68, 101, 128, 156, 184, 212	68,104,131,158,185,211
7		54, 97, 116, 137, 161, 185, 210	31, 68, 101, 128, 156, 184, 212	68,101,123,145,168,191,214
8		54, 93, 109, 127, 147, 168, 189, 212	31, 68, 98, 123, 148, 170, 194, 216	52,98,118,138,159,179,199,218
9		49, 91, 106, 122, 139, 157, 176, 195, 216	31, 68, 94, 117, 137, 158, 178, 200, 221	31,67,97,117,137,157,177,196,216
10		56, 90, 104, 119, 134, 150, 166, 183, 201, 219	27, 68, 84, 102, 119, 139, 160, 181, 201, 222	31,67,94,112,130,148,166,184,202,220

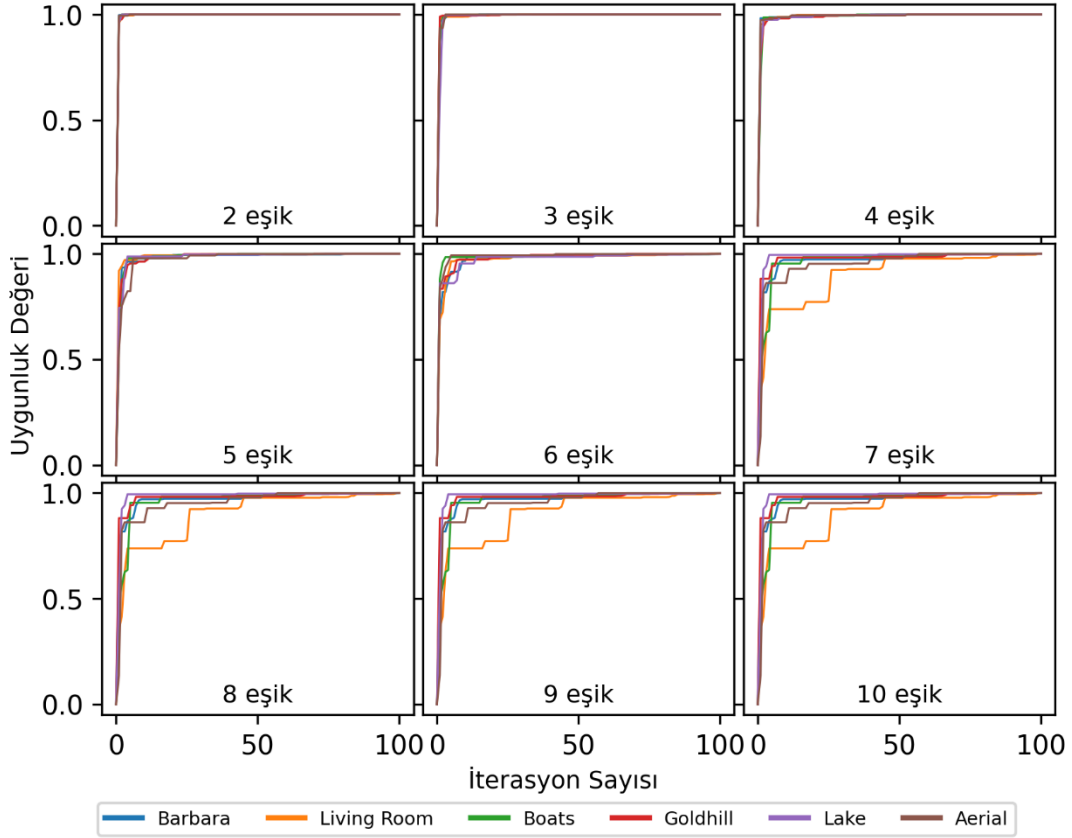
Şekil 19’da Kapur yönteminin altı test görseli için 2 ile 10 arasındaki eşik değerleri için GWO-HS algoritması kullanılarak elde edilen yakınsama eğrileri verilmiştir. Verilerin tablo gösterimini daha anlaşılabilir kılmak için değerler 0 ve 1 aralığına ölçeklenmiştir. Küçük eşik

değerlerinde algoritmanın en iyi değere çok hızlı bir şekilde ulaştığı görülmektedir. Eşik sayısı arttıkça arama uzayı genişlediği için algoritmanın en iyi değere oturma zamanı da artmaktadır.



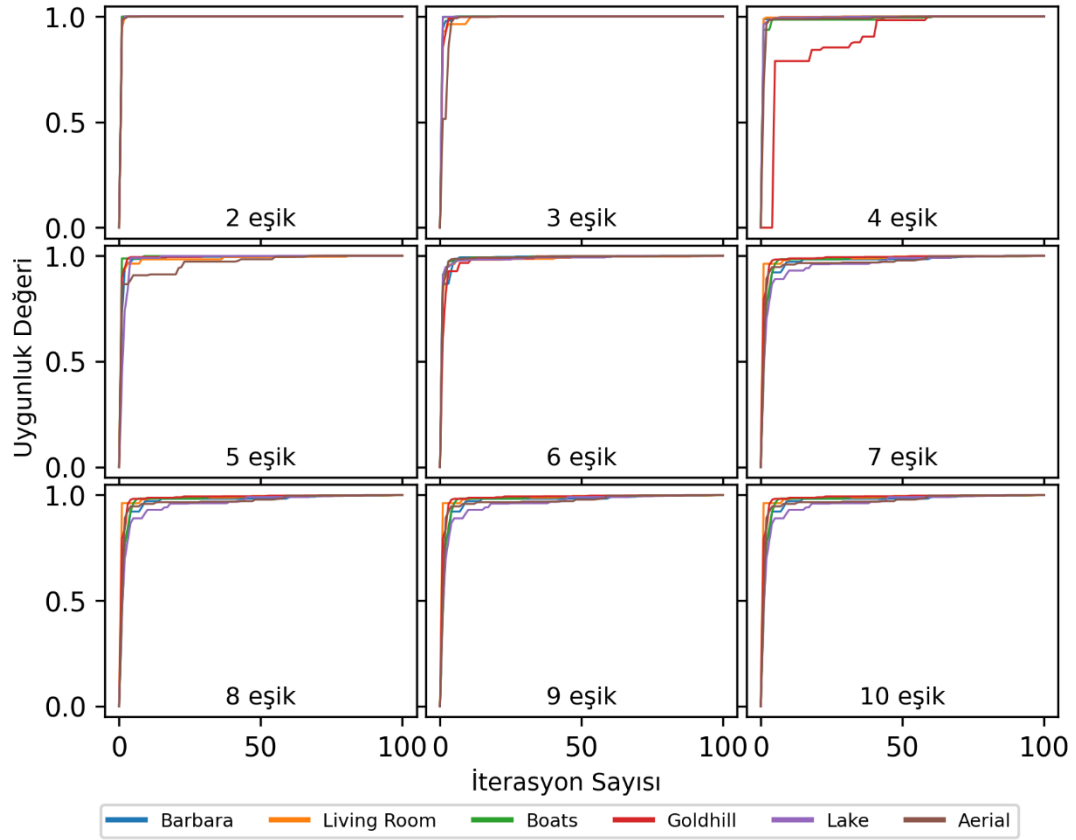
Şekil 19. GWO-HS algoritmasının Kapur eşikleme yöntemi için yakınsama eğrileri

Şekil 20’de Otsu yönteminin altı test görseli için 2 ile 10 arasındaki eşik değerleri için GWO-HS algoritması kullanılarak elde edilen yakınsama eğrileri verilmiştir. Verilerin tablo gösterimini daha anlaşılabilir kılmak için değerler 0 ve 1 aralığına ölçeklenmiştir. Kapur yönteminde olduğu gibi küçük eşik değerlerinde algoritma en iyi değere çok hızlı bir şekilde ulaşmaktadır.



**Şekil 20.** GWO-HS algoritmasının Otsu eşikleme yöntemi için yakınsama eğrileri

Şekil 21’de Tsallis yönteminin altı test görseli için 2 ile 10 arasındaki eşik değerleri için GWO-HS algoritması kullanılarak elde edilen yakınsama eğrileri verilmiştir. Verilerin tablo gösterimini daha anlaşılabilir kılmak için değerler 0 ve 1 aralığına ölçeklenmiştir. Diğer iki yöntem de olduğu gibi küçük eşik değerlerinde algoritma en iyi değere çok hızlı bir şekilde ulaşmaktadır.



Şekil 21. GWO-HS algoritmasının Tsallis eşikleme yöntemi için yakınsama eğrileri

## 6. SONUÇLAR VE ÖNERİLER

### 6.1. Sonuçlar

Bu çalışmada, çok seviyeli görüntü eşikleme için web tabanlı bir görüntü işleme aracı geliştirilmiştir. Bu araç yeni önerilen GWO-HS algoritmasının yanında dokuz farklı optimizasyon algoritmasını da bünyesinde barındırmaktadır. Kullanıcılar bu algoritmalarından herhangi birini kullanarak istedikleri görüntü ve eşik seviyesi için sonuçlar elde edebilmektedirler. Bu sonuçları karşılaştırabilmekte ve çıktılarını alabilmektedirler.

Önerilen GWO-HS algoritması gri kurt optimizasyon (GWO) ve harmoni arama (HS) algoritmaları hibritlenerek elde edilmiştir. Bu algoritmanın performansı dokuz diğer algoritmanın performansları ile karşılaştırılmıştır. Karşılaştırmalarda Otsu, Kapur ve Tsallis entropi tabanlı eşikleme yöntemleri kullanılmıştır. Deneysel görüntü işleme çalışmalarında iyi bilinen ve yaygın olarak kullanılan altı görüntü tercih edilmiştir. Her bir görüntü üzerinde 2'den 10'a kadar değişen seviyeler için eşikleme işlemi uygulanmıştır. Sonuçlar, önerilen GWO-HS algoritmasının, diğer algoritmalara kıyasla özellikle yüksek eşik seviyeleri için daha üstün bir performansa sahip olduğunu göstermiştir.

### 6.2. Öneriler

Geliştirilen hibrit algoritmanın daha geniş veri setleri üzerinde test edilmesi ve performansının iyileştirilmesi sağlanabilir. Ayrıca, algoritma farklı görüntü işleme problemlerine uyarlanabilir. Özellikle tıbbi görüntüler üzerinde (örneğin MRI, CT taramaları) önerilen algoritma test edilebilir. Algoritma üzerinde iyileştirmeler yapılarak teşhis ve tedavi süreçlerine katkı sağlanabilir.

## KAYNAKLAR

- Abualigah, L., Almotairi, K. H., ve Elaziz, M. A. (2023). Multilevel thresholding image segmentation using meta-heuristic optimization algorithms: comparative analysis, open challenges and new trends. *Applied Intelligence*, 53(10), 11654-11704. <https://doi.org/10.1007/s10489-022-04064-4>
- Agrawal, S., Panda, R., Bhuyan, S., ve Panigrahi, B. K. (2013). Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. *Swarm and Evolutionary Computation*, 11, 16-30. <https://doi.org/10.1016/j.swevo.2013.02.001>
- Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 13(6), 3066-3091. <https://doi.org/10.1016/J.ASOC.2012.03.072>
- Alihodzic, A., ve Tuba, M. (2014). Improved bat algorithm applied to multilevel image thresholding. *Scientific World Journal*, 2014. <https://doi.org/10.1155/2014/176718>
- Anagnostopoulos, G. C. (2009). SVM-based target recognition from synthetic aperture radar images using target region outline descriptors. *Nonlinear Analysis, Theory, Methods and Applications*, 71(12). <https://doi.org/10.1016/j.na.2009.07.030>
- Brajevic, I., ve Tuba, M. (2014). Cuckoo search and firefly algorithm applied to multilevel image thresholding. *Studies in Computational Intelligence*, 516, 115-139. [https://doi.org/10.1007/978-3-319-02141-6\\_6](https://doi.org/10.1007/978-3-319-02141-6_6)
- De Castro, L. N., ve Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3), 239-251. <https://doi.org/10.1109/TEVC.2002.1011539>
- Doğan, H., Ayas, S., Gedikli, E., ve Ekinci, M. (2017). Parçacık Sürü Zekası Optimizasyonu ile Mikroskobik Görüntülerin Segmentasyonunda Farklı Entropi Ölçülerinin Etkisi. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(2), 703. <https://doi.org/10.19113/sdufbed.64012>
- Geem, Z. W., Kim, J. H., ve Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, 76(2), 60-68. <https://doi.org/10.1177/003754970107600201>

- Gonzalez, R. C., Woods, R. E., ve Eddins, S. L. (2002). Digital image processing. İçinde *Pearson Education* (2nd Edition).
- Holland, John. (1975). Adaptation in Artificial and Natural Systems. *Ann Arbor: The University of Michigan Press*.  
[https://scholar.google.com/scholar?hl=tr&as\\_sdt=0%2C5&q=Holland%2C+John.+%22Adaptation+in+artificial+and+natural+systems.%22+Ann+Arbor%3A+The+University+of+Michigan+Press+%281975%29.&btnG=](https://scholar.google.com/scholar?hl=tr&as_sdt=0%2C5&q=Holland%2C+John.+%22Adaptation+in+artificial+and+natural+systems.%22+Ann+Arbor%3A+The+University+of+Michigan+Press+%281975%29.&btnG=)
- İlhan, İ. (2021). An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem. *Swarm and Evolutionary Computation*, 64, 100911.  
<https://doi.org/10.1016/J.SWEVO.2021.100911>
- Kapur, J. N., Sahoo, P. K., ve Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, & Image Processing*, 29(3), 273-285. [https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/10.1016/0734-189X(85)90125-2)
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*.
- Kennedy, J., ve Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942-1948.  
<https://doi.org/10.1109/ICNN.1995.488968>
- Kirkpatrick, S., Gelatt, C. D., ve Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680. <https://doi.org/10.1126/science.220.4598.671>
- Koç, I., Baykan, O. K., ve Babaoglu, I. (2018). Multilevel image thresholding selection based on grey wolf optimizer. *Journal of Polytechnic*. <https://doi.org/10.2339/politeknik.389613>
- Kökver, Y., ÜNVER, H. M., ve AYDOĞAN, E. (2018). Retina görüntülerinden kan damarlarının segmentasyonu. *Ortadoğu Tıp Dergisi*. <https://doi.org/10.21601/ortadogutipdergisi.352987>
- Lang, C., ve Jia, H. (2019). Kapur's entropy for color image segmentation based on a hybrid whale optimization algorithm. *Entropy*, 21(3). <https://doi.org/10.3390/e21030318>

- Lázaro, J., Martín, J. L., Arias, J., Astarloa, A., ve Cuadrado, C. (2010). Neuro semantic thresholding using OCR software for high precision OCR applications. *Image and Vision Computing*, 28(4), 571-578. <https://doi.org/10.1016/j.imavis.2009.09.011>
- Mahmoudi, S. E., Akhondi-Asl, A., Rahmani, R., Faghieh-Roohi, S., Taimouri, V., Sabouri, A., ve Soltanian-Zadeh, H. (2010). Web-based interactive 2D/3D medical image processing and visualization software. *Computer Methods and Programs in Biomedicine*, 98(2), 172-182. <https://doi.org/10.1016/J.CMPB.2009.11.012>
- Manic, K., Priya, R., ve Rajinikanth, V. (2016). Image multithresholding based on Kapur/Tsallis entropy and firefly algorithm. *Indian Journal of Science and Technology*. <http://www.academia.edu/download/45295663/89949-167174-2-PB.pdf>
- Markiewicz, T., Korzynska, A., Kowalski, A., Swiderska-Chadaj, Z., Murawski, P., Grala, B., Lorent, M., Wdowiak, M., Zak, J., Roszkowiak, L., Kozłowski, W., ve Pijanowska, D. (2016). MIAP – Web-based platform for the computer analysis of microscopic images to support the pathological diagnosis. *Biocybernetics and Biomedical Engineering*, 36(4), 597-609. <https://doi.org/10.1016/J.BBE.2016.06.006>
- Milošević, M., ve Vujičić, D. (2023). Web-Based System for Remote Mammogram Processing. *ijeec.etf.ues.rs.baM Milošević, D Vujičić, Đ Damnjanović, M VujovićInternational Journal of Electrical Engineering and Computing, 2023•ijeec.etf.ues.rs.ba, 7(1).* <https://doi.org/10.7251/IJEEC2301034M>
- Mirjalili, S., Mirjalili, S. M., ve Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46-61. <https://doi.org/10.1016/J.ADVENGSOFT.2013.12.007>
- Mozaffari, M. H., ve Lee, W. S. (2017). Convergent heterogeneous particle swarm optimisation algorithm for multilevel image thresholding segmentation. *IET Image Processing*, 11(8), 605-619. <https://doi.org/10.1049/iet-ipr.2016.0489>
- Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., ve Perez-Cisneros, M. (2013). Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*, 2013. <https://doi.org/10.1155/2013/575414>

- Onan, A. (2013). Metasezgisel Yöntemler ve Uygulama Alanları. *Çukurova Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 17(2), 113-128.
- Otsu Nobuyuki. (1996). A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*, 9(1), 62-66.  
[https://cw.fel.cvut.cz/b201/\\_media/courses/a6m33bio/otsu.pdf](https://cw.fel.cvut.cz/b201/_media/courses/a6m33bio/otsu.pdf)
- Parekh, H. S., Thakore, D. G., ve Jaliya, U. K. (2007). A Survey on Object Detection and Tracking Methods. *International Journal of Innovative Research in Computer and Communication Engineering (An ISO, 3297(2))*. [www.ijircce.com](http://www.ijircce.com)
- Portes de Albuquerque, M., Esquef, I. A., Gesualdi Mello, A. R., ve Portes de Albuquerque, M. (2004). Image thresholding using Tsallis entropy. *Pattern Recognition Letters*, 25(9), 1059-1065. <https://doi.org/10.1016/J.PATREC.2004.03.003>
- Raj, A., Gautam, G., Huda Sheikh Abdullah, S. N., Salimi Zaini, A., ve Mukhopadhyay, S. (2019). Multi-level thresholding based on differential evolution and Tsallis Fuzzy entropy. *Image and Vision Computing*, 91, 103792. <https://doi.org/10.1016/J.IMAVIS.2019.07.004>
- Rajinikanth, V., Raja, N. S. M., ve Kamalanand, K. (2017). Firefly algorithm assisted segmentation of tumor from brain MRI using Tsallis function and Markov Random Field. *Control Engineering and Applied Informatics*, 19(3), 97-106.  
<http://ceai.srait.ro/index.php?journal=ceai&page=article&op=view&path%5B%5D=4850>
- Sathya, P. D., ve Kayalvizhi, R. (2010). Optimum multilevel image thresholding based on tsallis entropy method with bacterial foraging algorithm. *International Journal of Computer Science Issues (IJCSI)*.  
<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=3a4a2cb2328e2f416be0be012e5d580975943554#page=359>
- Sezgin, M., ve Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 146-168.  
<https://doi.org/10.1117/1.1631316>

- Storn, R., ve Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341-359. <https://doi.org/10.1023/A:1008202821328/METRICS>
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*. <https://books.google.com/books?hl=tr&lr=&id=SIsa6zi5XV8C&oi=fnd&pg=PR7&dq=Metaheuristic:+from+Design+to+Implementation&ots=-cQMvQlyHt&sig=eU8wvnyKpc7NFYEie9nmYLKYIIE>
- Tsallis, C. (1988). Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52(1-2), 479-487. <https://doi.org/10.1007/BF01016429>
- Ünlü, A., ve İlhan, İ. (2023). A Novel Hybrid Gray Wolf Optimization Algorithm with Harmony Search to Solve Multi-Level Image Thresholding Problem. *Necmettin Erbakan Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 5(2), 230-245. <https://doi.org/10.47112/neufmbd.2023.21>
- Yang, X. S. (2009). Firefly Algorithms for Multimodal Optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5792 LNCS, 169-178. [https://doi.org/10.1007/978-3-642-04944-6\\_14](https://doi.org/10.1007/978-3-642-04944-6_14)
- Yuan, R., Luo, M., Sun, Z., Shi, S., Xiao, P., ve Xie, Q. (2017). RayPlus: a Web-Based Platform for Medical Image Processing. *Journal of Digital Imaging*, 30(2), 197-203. <https://doi.org/10.1007/S10278-016-9920-Y>