



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MODİFİYE EDİLMİŞ
KARINCA ASLANI OPTİMİZASYON ALGORİTMASI KULLANILARAK
BÖLGE BÜYÜTME YÖNTEMİ İLE
GRI SEVİYE GÖRÜNTÜ SEGMENTASYONU

Bashir Sheikh Abdullahi JAMA

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz – 2021
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Bashir Sheikh Abdullahi JAMA tarafından hazırlanan “Modifiye Edilmiş Karınca Aslanı Optimizasyon Algoritması Kullanılarak Bölge Büyütme Yöntemi ile Gri Seviye Görüntü Segmentasyonu” adlı tez çalışması .12./07./2021. tarihinde aşağıdaki jüri tarafından OY BİRLİĞİ ile Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Dr. Öğr. Üyesi Ahmet ÖZKIŞ

.....

Danışman

Dr. Öğr. Üyesi Nurdan BAYKAN

.....

Üye

Dr. Öğr. Üyesi Hazim İŞCAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki 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.

Bashir Sheikh Abdullahi Jama

Tarih: Temmuz – 2021

ÖZET

YÜKSEK LİSANS TEZİ

MODİFİYE EDİLMİŞ KARINCA ASLANI OPTİMİZASYON ALGORİTMASI KULLANILARAK BÖLGE BÜYÜTME YÖNTEMİ İLE GRİ SEVİYE GÖRÜNTÜ SEGMENTASYONU

Bashir Sheikh Abdullahi JAMA

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Dr. Öğr. Üyesi Nurdan BAYKAN

2021, 68 Sayfa

Jüri

**Dr. Öğr. Üyesi Ahmet ÖZKIŞ
Dr. Öğr. Üyesi Hazim İŞCAN
Dr. Öğr. Üyesi Nurdan BAYKAN**

Görüntü segmentasyonu, çeşitli alanlar için geçerli olan görüntü işlemenin önemli bir adımıdır. Bu alanlar arasında makine görmesi, nesne algılama, astronomi, biyometrik tanıma sistemleri (yüz, parmak izi, plaka ve göz), tıbbi görüntüleme, video izleme ve diğer birçok görüntü tabanlı teknoloji bulunmaktadır. Etkili görüntü segmentleme, otomatik görüntü işlemede en önemli işlemlerden ve kritik rollerden biridir. Özellikle mühendislik çalışmalarında, problemlerde en uygun çözümleri bulmak önemli araştırma konularından biridir. Arama alanlarında en uygun çözümleri bulmak için Parçacık Sürü Optimizasyonu (PSO), Karınca Algoritması (KA), Yapay Arı Kolonisi (YAK) ve Yarasa Algoritması (YA) gibi biyo-esinlenmiş algoritmalar kullanılmaktadır ve Karınca Aslanı Optimizasyonu (KAO) bu algoritmalarından biridir. Son yıllarda, görüntülerin segmentleme parametrelerini optimize etmek için biyo-esinlenmiş algoritmalar kullanılmaktadır. Bu tez çalışması, bölge büyütme (BB) segmentasyon problemini çözmek için, biyo-esinlenmiş Karınca Aslan Optimizasyon algoritmasının (mKAO) modifiye edilmiş bir versiyonunu sunmaktadır. Algoritmanın modifikasyonu, yeni bir dengeli konum güncellemesi ve esnek rastgele yürüyüş sınırı yöntemi kullanılarak yapılmıştır. Uygulama sırasında, görüntülerin kalitesini artırmak için, giriş görüntülerine ortanca filtresi uygulanmıştır. Daha sonra mKAO yardımıyla optimum tohum noktaları bulunarak, bölge büyütme segmentasyon işlemi gerçekleştirilmiştir. Önerilen yaklaşımın başarısı, BSDS300 (Berkeley-300) veri setinden alınan görüntüler kullanılarak test edilmiştir. Ayrıca literatürdeki farklı algoritma sonuçları ile de önerilen algoritma karşılaştırılmıştır. Sonuçlar $J_e, d_{max}, d_{min}, DBI, XBI$ ve Yapısal Benzerlik İndeksi (YBİ), Tepe Sinyal-Gürültü Oranı (TSGO), Özellik Benzerlik İndeksi (ÖBİ), Sınır Yer Değiştirme Hatası (SYDH), Global Tutarlılık Hatası (GTH), Korelasyon Katsayısı (KK) olarak farklı karşılaştırma metrikleri ile sunulmuştur. Deneysel sonuçlar, önerilen yöntemin literatürdeki yöntemlerle rekabetçi sonuçlar ortaya koyduğunu göstermiştir.

Anahtar Kelimeler: Bölge büyütme, tohum seçimi, görüntü segmentasyonu, önışleme, Karınca Aslanı Optimizasyonu.

ABSTRACT

MS THESIS

GRAY LEVEL IMAGE SEGMENTATION WITH REGION GROWING METHOD USING MODIFIED ANT LION OPTIMIZATION ALGORITHM

Bashir Sheikh Abdullahi JAMA

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering**

Advisor: Asst. Prof. Dr. Nurdan BAYKAN

2021, 68 Pages

Jury

**Asst. Prof. Dr. Ahmet ÖZKIŞ
Asst. Prof. Dr. Hazim İŞCAN
Asst. Prof. Dr. Nurdan BAYKAN**

Image segmentation is a significant step in image processing that applies to various fields. These fields include machine vision, object detection, astronomy, biometric recognition systems (face, fingerprint, plate, and eye), medical imaging, video surveillance, and many other image-based technologies. Efficient image segmentation is one of the most important tasks and critical roles in automatic image processing. Especially in engineering studies, finding the most suitable solutions for problems is one of the important research topics. Bio-inspired algorithms such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), and Bat Algorithm (BAT), etc. are used to find the optimal solutions in search spaces and Ant Lion Optimization (ALO) is one of these algorithms. In recent years, bio-inspired algorithms are used to optimize the segmentation parameters of the images. In this thesis, a modified version of the bio-inspired ant-lion optimization algorithm (mALO) is introduced to solve the region growing (RG) segmentation problem. The modification of the algorithm is done using a new balanced position update and flexible random walk boundary method. During the implementation, the median filter was applied to the input images to improve the quality of the images. Then, by finding the optimum seed points with the help of mALO, region growing segmentation was performed. The success of the proposed approach has been tested using images from the BSDS300 (Berkeley-300) dataset. In addition, the proposed algorithm was compared with the results of different algorithms in the literature. The results are presented with different comparison metrics such as J_e , d_{max} , d_{min} , DBI , XBI and Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), Feature Similarity Index (FSIM), Boundary Displacement Error (BDE), Global Consistency Error (GCE), Correlation Coefficient (CC). Experimental results showed that the proposed method provided competitive results with those in the literature.

Key words:- Region growing, Seed point selection, Image Segmentation, pre-processing, Ant Lion Optimization.

ÖNSÖZ

Bu yüksek lisans çalışmasını birçok kurum ve bireyin desteği olmadan yapamazdım. Bu tez çalışmasında bana tavsiyeler verip, rehber olan ve büyük destek veren sayın danışman hocam Dr. Öğr. Üyesi Nurdan BAYKAN'a teşekkür ediyorum. Ayrıca, Türkiye bursları, aileme, arkadaşlara ve bu çalışmanın gerçekleştirilmesinde bana her türlü konuda yardımcı olan herkese teşekkür ediyorum.

Bashir Sheikh Abdullahi JAMA
KONYA-2021



İÇİNDEKİLER

| | |
|--|-----------|
| ÖZET | iv |
| ABSTRACT..... | v |
| ÖNSÖZ | vi |
| İÇİNDEKİLER | vii |
| SİMGELER VE KISALTMALAR | viii |
| 1. GİRİŞ..... | 1 |
| 1.1. Tezin Amacı:..... | 3 |
| 1.2. Tezin Önemi | 4 |
| 1.3. Tezin Organizasyonu | 4 |
| 2. KAYNAK ARAŞTIRMASI | 5 |
| 3. MATERYAL VE YÖNTEM..... | 13 |
| 3.1. Dijital Görüntü | 13 |
| 3.2. Temel Görüntü İşleme Adımları | 15 |
| 3.3. Görüntü segmentasyonu | 16 |
| 3.4. Bölge tabanlı görüntü segmentasyon yöntemleri (BTSY)..... | 20 |
| 3.5. Karınca Aslanı Algoritması | 23 |
| 3.5.1. Popülasyon oluşturma | 27 |
| 3.5.2. Karıncaların rastgele yürüyüşleri..... | 29 |
| 3.5.3. Tuzağın Kurulması | 30 |
| 3.5.4. Tuzaklardan etkilenme ve kapana kısılma..... | 31 |
| 3.5.5. Karınca aslanlarına doğru kayan karıncalar..... | 31 |
| 3.5.6. Avı yakalama ve çukuru yeniden inşa etme | 32 |
| 3.5.7. Elitizm..... | 33 |
| 3.6. Karşılaştırma Metrikleri..... | 34 |
| 3.7. Veri Kümesi | 39 |
| 4. ÖNERİLEN YÖNTEM | 42 |
| 5. DENEYSEL SONUÇLAR | 51 |
| 6. SONUÇLAR VE ÖNERİLER..... | 60 |
| 6.1 Sonuçlar | 60 |
| 6.2 Öneriler | 61 |
| KAYNAKLAR | 63 |
| ÖZGEÇMİŞ | 68 |

SİMGELER VE KISALTMALAR

Simgeler

| | |
|----------------------|--|
| n | : maksimum yineleme sayısı. |
| T | : rastgele yürüyüşün adımı. |
| a_i | : i 'inci değişkenin rastgele yürüyüşünün minimumu. |
| b_i | : i 'inci değişkenin rastgele yürüyüşünün maksimumu. |
| c_i^t | : i 'inci değişkenin t 'inci tekrarlamaadaki minimumu. |
| d_i^t | : i 'inci değişkenin t 'inci tekrarlamaadaki maksimumu. |
| Ant_j^t | : t 'inci tekrarlamaadaki seçilmiş j 'inci karınca aslanının pozisyonu. |
| Ant_t^i | : t 'inci tekrarlamaadaki i 'inci karıncanın pozisyonu. |
| P ve q | : iki nokta arasındaki Öklid mesafesi. |
| $Cumsum$ | : Kümülatif toplam. |
| X | : Kümelenecek görüntünün veri kümesi. |
| Q_{AI} | : Görüntü kümeleme sorununa aday bir çözüm. |
| $f(Q_{AI}, X)$ | : Q_{AI} için hesaplanan uygunluk değeri. |
| $d_{max}(X, Q_{AI})$ | : Tüm kümeler için merkezlerine göre hesaplanan ortalama uzaklıkların maksimum değeri. |
| $d_{min}(X, Q_{AI})$ | : Herhangi bir küme çifti arasındaki minimum ortalama mesafe. |
| Je | : Kümeleme işleminin kalitesini hesaplamak için tanımlanan niceleme hatası. |
| U_i^t | : Üst sınır. |
| l_i^t | : Alt sınır. |
| r | : Korelasyon katsayısı. |
| \bar{x} | : Bir örnekteki x değişkeninin değerleri. |
| y_i | : Bir örnekteki y değişkeninin değerleri. |
| μ_x | : x 'in ortalaması. |
| μ_{xy} | : y 'nin ortalaması. |
| σ_x^2 | : x 'in varyansı. |
| σ_y^2 | : y 'in varyansı. |
| $2\sigma_{xy}$ | : x ve y 'nin varyansı. |

Kısaltmalar

| | |
|------|---|
| NYA | : Nesne yönelimli analiz |
| KAO | : Karınca Aslanı Optimizasyon algoritması |
| BG | : Bölge Büyütme |
| BTSY | : Bölge tabanlı görüntü segmentasyon yöntemleri |
| RGB | : Kırmızı, yeşil, mavi renk uzayı |
| PSO | : Parçacık Sürüsü Optimizasyonu |
| OKH | : Ortalama Kare Hatası |
| KK | : Korelasyon Katsayısı |
| SYDH | : Sınır Yer Değiştirme Hatası |
| GB | : Gradyan büyüklüğü |
| FU | : Faz Uyumluluğu |
| ÖBİ | : Özellik Benzerlik İndeksi |
| TSGO | : Tepe Sinyal-Gürültü Oranı |
| YBİ | : Yapısal Benzerlik İndeksi |

1. GİRİŞ

Görüntü segmentasyonu (bölütme), görüntü analizinde kritik bir adımdır ve genellikle büyük miktarlarda veri üzerinde çalışırken yüksek bir hesaplama maliyeti içerir. Verimli görüntü bölümlendirme, nesne yönelimli analizde (NYA) kritik bir rol oynar ve otomatik görüntü işlemede en önemli görevlerden biridir (Zaitoun ve Aqel, 2015).

Görüntüleme cihazları ve bu cihazların teknolojilerindeki önemli ilerlemeler nedeniyle, dijital görüntüler hayatımızda giderek daha önemli bir rol oynamaktadır. Bir görüntü, gerçek dünyanın bir sahnesini, daha sonra saklanabilecek, iletilebilecek ve üzerinde çalışılabilecek sayısal bir gösterimde kaydeder (Haralick ve Shapiro, 1985). Türkçe’de “Bir resim bin kelimeye bedeldir” ifadesi, bir görüntünün taşıdığı zengin bilgiyi açıklamak için güçlü bir yeteneği olduğunu belirtmektedir.

Dijital görüntüleme birçok alanda kullanılmaktadır. Tıp alanında, Manyetik Rezonans (MR) Görüntüleme (Balafar ve ark., 2010), Bilgisayarlı Tomografi (BT) (Gao ve ark., 2011), ultrason gibi çeşitli cihazlar aracılığıyla görüntüler elde edilmektedir. Bu görüntüler, insan vücudunun içyapısını görmek için paha biçilmez bir erişim ve hekimlere daha doğru tanımlarda bulunabilme imkânı sağlamaktadır. Faydalı bilgilerin resimlerden çıkarılması bugün önemli bir görev haline gelmiştir. Ancak insanlar bu görevi doğal olarak çözebilse de bir bilgisayarın görüntüyü otomatik olarak yorumlaması hâlâ zordur. Bir görüntüyü yorumlamak veya anlamak için, görüntünün anlamlı bölümlere ayrılmasını içeren ve görüntüde neyin nerede olduğuna ait ipuçlarını sağlayan düşük düzeyden orta düzeye bir görme görevi, görüntü segmentasyonunun uygulama sistemlerinde önemli bir rol oynadığını göstermektedir.

Uydu görüntüleri alanında görüntü segmentasyonu, uydular aracılığıyla toplanan dünya veya diğer gezegenlere ait görüntüleri nesnelere bölmek için kullanılmaktadır. Bu nedenle, görüntü segmentasyonu, yeryüzüne ilişkin faydalı bilgiler sağlamak için uydu sistemlerini destekleyen önemli bir adımdır (Pare ve ark., 2015).

Video gözetimi, bir video kamera kullanarak hareketli cisimleri bir süre boyunca bulmak için kullanılan bir yöntemdir. Nesne segmentasyonundan oluşan, görüntüdeki kümeleri tespit eden ve bu görüntü kümelerini izlemek için kullanılan nesne izleme algoritmasını kullanırlar (Cheng ve Butler, 2005). Görüntü segmentasyonu gibi görüntü işleme teknikleri, gelişmiş trafik kontrol sistemlerini tasarlamak için kullanılabilir (Khan ve ark., 2010). Bu sistemlerde, trafik yoğunluğu hakkında

bilgi almak için kullanılan yolun bir görüntüsü alınır. Bu yeni tip trafik kontrol sistemleri, büyük şehirlerde trafik sıkışıklığı sorununu azaltmak için kullanılmaktadır.

Görüntü segmentasyonu; özerk sürüş, görüntü düzenleme, uzaktan algılama, artırılmış gerçeklik ve tıbbi görüntü analizi gibi birçok uygulamada kullanılmaktadır. Segmentasyonun doğruluğu bu uygulamalarda mutlak bir gerekliliktir. Bölümlendirme ne kadar doğru olursa, vizyon veya görüntünün analitik görevi başarabildiği performans o kadar güvenilir ve iyi olur. Çoğu durumda, görüntü uzmanların el ile segmentlendirmesi en iyi ve en güvenilir sonucu sağlayabilir, ancak bu zaman alıcı ve sıkıcıdır. Ayrıca, manuel segmentasyon operatör değişkenliğine tabidir. Her tür görüntünün segmentasyonu için kapsayıcı bir teknik yoktur. Her görüntüleme çerçevesinin kendine özgü kısıtlamaları vardır. Bu nedenle, araştırmacılar daha doğru segmentasyon sağlayabilecek yeni yöntemler geliştirmeye çalışmaktadırlar.

Bir resimdeki ayrıştırılan alt homojen bölgeler, karşılaştırılabilir özelliklere sahip benzer piksellerden oluşan bir gruptur. Bölge yaklaşımında, her piksel belirli bir nesneye veya bölgeye aittir. Kenar algılama tekniği ile karşılaştırıldığında, bölgeye bağlı segmentasyon algoritmaları genellikle basit ve gürültüye karşı daha güvenlidir (Zaitoun ve Aqel, 2015). Bölge tabanlı segmentasyonda, genellikle, nesneye uygun bölge tabanlı pikseller birleştirilir ve birlikte işaretlenir. Bölge tabanlı olarak aynı şekilde uygun eşikleme yöntemlerinin kullanılması gerekmektedir. Temel standartlar, yakınlık (gri değer kontrastlarını ve gri değer varyans değişimi içeren) ve bir bölgenin kompaktlarının yanı sıra Öklid aralığı da içeren uzamsal yakınlık ölçütüdür (Zaitoun ve Aqel, 2015).

Bölge tabanlı segmentasyon algoritmaları Bölge Büyütme (BB) (Region Growing-RG), Bölge Ayırma/Birleşme (Region Splitting/Merging) olmak üzere iki temel sınıfa ayrılır (Kang ve ark., 2009; Zaitoun ve Aqel, 2015). Bölge büyütme, önceden tanımlanmış bazı kriterlere bağlı olarak görüntünün bir alanını bulmak ve genişletmek için bir yöntemdir (Cui ve ark., 2008). Bu model daha çok yoğunluk bilgisine bağlıdır. Bölge büyütme, komşu piksellerin analiz edildiği ve kenarları algılanmayan bir alan sınıfına eklendiği resim segmentasyonu yapmanın bir yoludur (Zaitoun ve Aqel, 2015). Bu eylem dizisi, bölgenin her piksel sınırı için tekrarlanacaktır. Zayıf veya bitişik bölgelerin keşfedilmesi durumunda, zayıf kenarların çözüldüğü ve güçlü kenarların sağlam kaldığı bölge birleştirme algoritmaları kullanılır. Bölge ayırma/birleştirme, genişlemenin tersidir. Bu strateji, tüm görüntü üzerinde çalışır. Bölge bölme yukarıdan aşağıya çalışan bir metodolojidir. Süreç bütün bir

görüntü ile başlar, daha sonra izole bölge bütün resimden daha homojen olacak şekilde ayrılır. Bu nedenle, bölme kısmından sonra bir birleştirme aşaması sürekli gereklidir. Buna “bölme/birleştirme algoritması” denir (Cui ve ark., 2008). Herhangi bir alan alt bölgelere ayrılabilir ve uygun bölgeler daha sonra birleştirilebilir. Belli noktaları seçmek yerine, kullanıcı bir resmi birçok öznel ayrılmış bölgeye parçalayabilir ve daha sonra aynı işlevleri yerine getirmeye çalışan bölgeleri birleştirebilir (Qian Zhao ve ark., 2014; Malarvel ve ark., 2016; Merzougui ve El Allaoui, 2019).

Karınca Aslanı Optimizasyonu (KAO) (Ant Lion Optimization-ALO) algoritması, 2015 yılında Mirjalili tarafından önerilen ve karınca aslanlarının avlanma davranışına dayanan yeni bir popülasyon tabanlı optimizasyon algoritmasıdır. KAO’da, karınca aslanlarının davranışları ve doğadaki avları matematiksel olarak modellenmiştir (Mirjalili, 2015). Karınca aslanı algoritması; karıncaların rastgele yürüyüşü, tuzakların inşa edilmesi, karıncaların tuzaklara kaydırılması, avların yakalanması ve tuzakların yeniden inşa edilmesi gibi beş ana adımdan oluşmaktadır. KAO, özellik seçimi (Zawbaa ve ark., 2016), MR karaciğer görüntü segmentasyonu (Mostafa ve ark., 2016), yenilenebilir dağıtılmış güç (Ali ve ark., 2016), kablosuz sensör ağları (Yogarajan ve Revathi, 2018) gibi farklı alanlarda kullanılmıştır. Bu tez çalışmasının temel amacı, bölge büyütme (BB) ve KAO kullanarak yeni bir bölge tabanlı görüntü segmentasyon yöntemi ortaya koymaktır.

1.1. Tezin Amacı

Görüntü segmentasyonu (bölütleme), dijital görüntüyü birden çok bölüme ayıran tekniktir ve tipik olarak ilgilenilen bölgeleri (ROI) veya dijital görüntüdeki diğer ilgili bilgileri tanımlamak için kullanılır. Segmentasyonun temel amacı, görüntüyü alt bölgelere bölmek, böylece bu alt bölgelerin homojen olmasını sağlamaktır. En yaygın kullanılan homojenlik kriterleri yoğunluk, doku, renk, aralık, normal yüzey ve yüzey eğriliğidir. Görüntü segmentleme, görüntü analizinde kritik bir adımdır ve genellikle büyük miktarlarda veri üzerinde çalışırken yüksek bir hesaplama maliyeti içerir. Görüntü segmentasyonunun önemi, görüntüyü, tamamen anlaşılabilir ve analiz edilmesi daha kolay olan alt parçalara ayırmaktır (Saini ve Arora, 2014). Ayrıca, verimli görüntü segmentasyonu, nesne yönelimli analizde (NYA) kritik bir rol oynar ve son yıllarda ayrı bir araştırma alanı haline gelmeye başlamıştır.

Yapılacak tez çalışmasının temel amacı, bölge büyütme ve Karınca Aslanı Optimizasyon algoritması (KAO) kullanarak yeni bir bölge tabanlı görüntü segmentleme yöntemi ortaya koymaktır. Geliştirilen algoritmanın, literatürdeki mevcut algoritmalarla karşılaştırılması yapılarak, segmentasyon alanındaki verimliliği ortaya konmuştur.

1.2. Tezin Önemi

Bu çalışmada, verimli ve güvenilir bir görüntü segmentasyon işlemi gerçekleştiren bir yöntem ortaya konulması hedeflenmektedir. Bölge büyütme algoritması ilk olarak görüntüleri bölgelere ayırmak için kullanılacaktır. Bölge genişlemesinin ana zorluklarından biri doğru başlangıç noktasını seçmektir. Farklı başlangıç noktası seçenekleri farklı segmentasyon sonuçları verebilir. Bu nedenle, problem çözümünde en uygun başlangıç noktasını seçmek için Karınca Aslanı Optimizasyon (KAO) algoritması uygulanmıştır. Çalışmanın sonuçları, bölge büyütme segmentasyonu yöntemlerinin sonuçları ile karşılaştırılmıştır.

1.3. Tezin Organizasyonu

Tez, beş bölümden oluşmaktadır. İlk bölümde görüntü segmentasyonu ve segmentasyon türleri ile bu anlamda tez kapsamında önerilen modelin içeriği hakkında bilgi verilmiştir. Tezin amacı ve önemine ilişkin yer alan bilgiler de bu bölüm içerisinde verilmiştir.

İkinci bölümde literatür incelenmiştir. Görüntü segmentasyonu için geliştirilen literatürdeki yöntemler, özellikle metasezgisel algoritmaların uygulanması ve uygulama sonuçları bu bölümde incelenmiştir.

Üçüncü bölüm, tez çalışmasında kullanılan yöntemler ve veri setleri hakkında bilgi içermektedir. Kullanılan veri kümeleri ve Karınca Aslanı Optimizasyon (KAO) algoritması hakkındaki bilgilerin tümü bu bölümde sunulmuştur.

Dördüncü bölümde, geliştirilen yönteme ait uygulama aşamaları verilerek, deneysel sonuç analizleri sunulmuştur.

Son olarak beşinci bölümde ise tez kapsamında yapılan çalışmalar ve elde edilen sonuçlar genel olarak değerlendirilip, literatür ile karşılaştırmalar verilmiş ve sonraki çalışmalar için öneriler sunulmuştur.

2. KAYNAK ARAŞTIRMASI

Görüntü segmentasyonunda tek görüntü üzerine ilk çalışmalar 70'lerin başında yapılmıştır (Starr, 1976). Brice ve Fennema blok dünya görüntüleri için bir segmentasyon algoritması önermiştir. O zamandan beri birçok görüntü segmentasyon algoritması ortaya konulmuştur. Teknoloji ve yeniliklerin ortaya çıkması nedeniyle, çok çeşitli uygulamalarda farklı görüntü işleme yöntemlerinin gerekli olduğu anlaşılmıştır (Baxes, 1994). Görüntü segmentasyonu da, görüntü işleme alanında önemli bir konu haline gelmiştir ve görüntü işleme sistemlerinin odak noktasıdır. En sık kullanılan segmentasyon yöntemleri; bölge tabanlı, kenar tabanlı, bulanık tabanlı teori, yapay sinir sistemi ve eşik tabanlı segmentasyonu içermektedir (Zaitoun ve Aqel, 2015).

Son yıllarda, görüntü segmentasyonu süreci alanında sayısız araştırma yapılmıştır. Günümüzde, birbirinden oldukça farklı olan birçok segmentasyon algoritması vardır, ancak yine de her türlü dijital görüntüye uygulanabilecek ve her amaç için kullanılacak belirli bir algoritma yoktur (Li ve ark., 2011; Zaitoun ve Aqel, 2015).

Zhou ve ark. (2008), ortalama kayma kümeleme algoritmasıyla bölge tabanlı bir görüntü segmentasyon yöntemi önermişlerdir. Bu yöntem ilk olarak, uygun renk uzayını seçerek özellik vektörünü oluşturmak için her pikselden renk, doku ve konum özelliklerini çıkarmıştır. Çıkarılan özellik vektörleri, ortalama kayma kümeleme algoritması kullanılarak kümelendirilmiştir. Daha sonra aynı etikete sahip olan bölgeler pikseller için komşu bağlantı teorisine göre yeniden segmentlenmiş ve bölgeleri tanımlayan birçok özellik sağlanmıştır.

Viji ve JayaKumari (2013), biri yoğunluk kısıtlı ve diğeri doku kısıtlı olmak üzere iki yeni yöntemin, tümör bölgelerini tümör dışı bölgelerden ayırmak için etkinliklerini araştırmışlardır. MR görüntüleri için simülasyon sonuçları; doku ve yoğunluk bazlı bölge büyütme değerlerinin, MR görüntüsünde yoğunluk bazlı bölge büyütme yöntemine göre daha başarılı tümör segmentasyon sonuçları sunduğunu göstermiştir. Manuel segmentasyon ve derecelendirme ile karşılaştırıldığında, lezyon segmentasyonu için otomatik bir yaklaşımın avantajları arasında artan güvenilirlik, tutarlılık ve tekrarlanabilirlik sayılmıştır.

Lu ve ark. (2014)'de, geleneksel bölge büyütme yöntemini geliştirmek için Quasi-Monte Carlo yöntemi tanıtılmıştır. Geliştirilmiş olan algoritma, uygun

tohumlama noktalarının seçilme verimliliğini ve iyileştirilmiş bölge büyütme kriterlerinin karaciğer segmentasyonuna daha iyi uymasını sağlamaktadır.

Happ ve ark (2015), MapReduce modeline (bulutta büyük miktarda veri işlemek için en çok kullanılan programlama modellerinden birisi) dayalı, bölgeye göre büyüyen görüntü segmentasyonu için yeni bir yöntem sunmuştur. Yöntem, doğrulama için Hadoop platformunda uygulanmıştır. Deney için Brezilya'nın São Paulo şehrinde bir alanı kapsayan 4000×4000 piksel boyutundaki bir uydu görüntüsü (QuickBird Satellite Image – DigitalGlobe'un sahibi olduğu ve işlettiği yüksek çözünürlüklü bir uyduya ait görüntü) kullanılmıştır. Ön deneyler, önerilen yöntemin çok büyük uzaktan algılama görüntüleri için ölçeklenebilir ve verimli olduğunu göstermiştir.

Zhongming ve Jun (2015)'de, Dalgacık Dönüşümü ve bölge büyütme dayanan kızılötesi görüntü segmentasyon yöntemi önerilmiştir. Geleneksel tohum genişletme yöntemi, görüntü piksellerini büyüyecek tohumlar olarak görür. Yapılan çalışmada, önerilen bir hesaplama yapılarak büyüme noktası ve büyüme eşiği elde edilmektedir. İlk olarak yöntem, görüntü segmentasyonu başlatma işlemini sürdürmek için dalgacık dönüşümü maksimum algoritmasını uygular ve segmentasyonun sonucunu oluşturur. Daha sonra yöntem; görüntü piksellerini, bölümlendirme sonrası gri değerlerin görüntü kenar noktasının ortalama gri değerinden daha büyük olduğu büyüme noktaları ve ortalama kare hatasını, büyümekte olan koşullar eşiği olarak kullanır. Bu, ortalama kare hatasının, orijinal görüntü ile bölgeyi genişletmek için her kenar noktası arasındaki ortalama gri değer olduğu anlamına gelir. Deney için kızılötesi insan vücudu görüntüsü kullanılmıştır.

Elmorsy ve ark (2015), BT görüntülerinden oldukça doğru karaciğer çıkarımı için yeni bir büyüklük seçimi bölgesi segmentasyon algoritması sunmaktadır. Bu çalışma iki ana adımda gerçekleştirildi. İlk aşamada veri setinin ön işleme tabi tutulmasıdır. Veri seti farklı hastalardan 69 vaka çalışmasından oluşmakta, bu çalışmalar arasında 10 farklı hasta vakasına, karaciğer pozisyonuna, normal ve anormal karaciğer dokularına odaklanmaktadır. İkinci aşama; üç eşikleme, bölge büyütme ve boyut seçimi morfolojik işlemi olmak üzere ardışık üç aşamada gerçekleştirilen BT görüntülerinden karaciğeri çıkarma aşamasıdır.

Singh ve Gupta (2015), mamogramdaki kanserli dokuların tespiti için basit ve kolay bir yaklaşım önermiştir. Saptama aşamasını, bir mamogram görüntüsünde tümör bölgesinin segmentasyonu izlemiştir. Bu yaklaşım, ortalama alma ve eşikleme gibi basit

görüntü işleme tekniklerini kullanmış, tümör tespiti için bir Max-Ortalama ve En Küçük Varyans tekniği tanıtılmıştır. Deneysel sonuçlar, yaklaşımın etkinliğini göstermiştir.

Li ve ark. (2015)'de, yüksek çözünürlüklü uzaktan algılama görüntüleri için geleneksel bölge büyütmeyle dayalı yeni segmentasyon yöntemi önerilmiştir. Bu yöntem, bölgesel minimumları görüntünün yatay ve dikey gradyan haritalarından almaktadır. WorldView-2 alt küme bölümlendirme haritaları test için kullanılmıştır.

Wu ve Guo (2015)'de, Markov Rastgele Alan (MRA) (Markov Random Field-MRF) modeli ve bölgesel genişletme yöntemi ile birleştirilen görüntü segmentasyon yöntemini, sonar görüntü segmentasyonunun doğruluğunu tespit etmek için önerilmiştir. Sonar görüntüsü (deniz dibi veya su altı nesnelere bir görüntüsünü verimli bir şekilde oluşturmak için kullanılan bir sonar sistemi kategorisi) deney için kullanılmıştır. İlk olarak, alanın özellikleri ve sonar görüntüsünün etiket alanı MRF modeli ile belirlenmiş; ikinci olarak, Tekrarlanmış Koşullu Modlar (ICM-Iterated Conditional Modes) algoritması ile ilk görüntü segmentasyonuna dayanan bölge büyümesiyle işlenmiştir. Bu, görüntü segmentasyonundaki MRF yönteminin neden olduğu segmentasyon plaklarını azaltmıştır.

Zhang ve ark. (2015)'de, iki yönlü büyüyen segmentasyon algoritması olarak gösterilen, tıbbi görüntü segmentasyonu için bölge büyütme segmentasyonunun basit bir çevirisi önerilmiştir. Algoritma, ilk olarak büyüyen bölge için ilk tohum olarak arka plan bölgesinde bir piksel seçmiştir. Daha sonra, ikincil bölge büyümesi, doku bölümünün merkezinde seçilen ilk tohumla ilk segmentasyon sonuçlarına dayanılarak gerçekleştirilmiştir. Önerilen algoritmanın çekici özellikleri, segmentasyon sonuçlarının seslere ve işlem piksellerinin sırasına duyarlı olmasıdır. Ayrıca hem arka plan hem de nesne bölgesi, deliksiz bağlı bölgelerdir ve tohumun seçilmesi kolaydır. Segmentasyon eşliğinin otomatik olarak seçilmesi için Komşu Fark Dönüşümü kavramı önerilmiştir.

Kotteswari ve Sathya (2016)'da, MR görüntülerinde ön plan algılama analizi hakkında bir çalışma önermişlerdir. Bu çalışmada iki ana tespit aşaması bulunmaktadır. İlk adım, giriş görüntüsünün, anizotropik difüzyon filtrelemesi ve bölge segmentasyonu ile gerçekleştirilen ön işleme kullanarak segmentlenmesini içermektedir. İkinci adım ise, Sobel kenar algılaması ve özellik çıkarımı katkısı ile DVM (Destek Vektör Makinesi) (SVM-Support Vector Machine) sınıflandırıcısını kullanarak sınıflandırmayı içermektedir. Deneysel analiz, önerilen yöntemin öngörülemez veya çok sayıda ilişkili ön plana sahip test sahnelerinde özel başarı ile farklı veri kümelerinde çalıştığını göstermiştir.

Malarvel ve ark. (2016)'da, giriş görüntüsünde ön işleme tekniklerinin kullanılması gerekmeden bölge büyütme yöntemine dayalı yeni bir segmentasyon yöntemi önerilmiştir. Bu yöntem, sektördeki X-Radyografi (X-Ray) görüntülerinde otomatik tohumlu bölge büyütme tekniği kullanılarak ön planı düzensiz arka plandan ayırarak segmentleyebilir. Otsu ve Kittler'in eşiklerinin kombinasyonu, tohumları otomatik olarak seçmek için kullanılır. Önerilen algoritma, Otsu ve Kittler'in eşik yöntemleriyle karşılaştırılmış ve Göreceli Mutlak Hata (GMH) (RAE-Relative Absolute Error) performansı değerlendirilmiştir. Önerilen yöntemle elde edilen sonuçlar diğer yöntemlerden daha iyi sonuçlar vermiştir.

Baghi ve Karami (2017)'de, Sentetik Açıklıklı Radar (SAR) görüntülerinin segmentasyonu için Bölge Büyütme (BB) ve Spektral Küme'ye (SK) (SC-Spectral Clustering) dayalı yeni bir yöntem tanıtılmıştır. Önerilen yöntemde, kenarı bulmak için önce SAR görüntülerine BG uygulanmış ve daha sonra SK yöntemi kullanılarak segmentleme yapılmıştır. Önerilen yöntem, gerçek SAR görüntüsü üzerindeki bazı son teknoloji segmentasyon algoritmaları ile karşılaştırılmıştır.

Duman ve Erdem (2017)'de, piksellerin kenarların mı yoksa dokuların mı olduğunu tespit etmek için, bölge büyütme yöntemi kullanmıştır. Bu şekilde hangi komşu piksellerin her bir piksel için hesaplamaya dâhil edilmesi gerektiği belirlenmiştir. Önerilen yöntem kenarları korurken, dokuların düzelmesini sağlar. Deneysel sonuçlar, önerilen yöntemin, Tepe Sinyal-Gürültü Oranı (PSNR-Peak Signal to Noise Ratio) değerleri açısından iyi bilinen ve yaygın olarak bulunabilen görüntü dengeleme algoritmalarından daha iyi performans gösterdiğini göstermiştir.

Jeevakala ve Rangasami (2018)'de, koklear sinir bölgesini doğru şekilde segmentlere ayıran modifiye edilmiş bir bölge büyütme algoritması önermişlerdir. Sri Ramachandra Medical Üniversitesi'nden 1.5 Tesla MR tarayıcısı (Siemens Avanto, Erlangen) kullanılarak elde edilen 20 hastanın görüntüleri test için kullanılmıştır. Bu çalışmanın segmentleme doğruluğu; Jaccard, Yanlış Pozitif ve Yanlış Negatif gibi parametreler kullanılarak değerlendirilmiştir. Segmentlenmiş bölge; uzun çap, kısa çap ve enine kesit alanı kullanılarak ölçülmüş ve değerlendirilmiştir. Otomatik ölçümün tekrarlanabilirliğini araştırmak için gözlemciler segment içi ve segmentler arası korelasyonun istatistiksel analizlerini ve kabul limitlerini, koklear sinirin kesit alanı üzerinde gerçekleştirmişlerdir.

Reddy ve Reddy (2018)'de, bölge büyütme yöntemine dayalı yeni bir algoritmanın performansını değerlendirmişlerdir. Bu yöntemin performansını test etmek

için beyin tümörü MR görüntüleri kullanılmış ve sonuç analizi yapılmıştır. Yeni algoritma; PSNR, Ortalama Kare Hata (OKH) (MSE-Mean Squared Error) ve Yapısal Benzerlik İndeksi (YBI) (SSIM-Structural Similarity Index Measure) gibi farklı parametreler kullanan mevcut K-Means ve FCM yöntemleriyle karşılaştırılmıştır.

Chondro ve ark. (2018), düz akciğer grafileri için bilgisayar destekli bölge segmentasyonu önermiştir. Yöntem, akciğer bölgelerinin opaklığını arttıran bir avangard kontrast geliştirmeyi içermektedir. İlgi Alanı (ROI-Region of Interest) hızlı bir şekilde blok temelli bir ikili hale getirme ve morfolojik işlemler uygulanarak yerleştirilir. Bölge sınırları için daha fazla gelişme, herhangi bir şüpheli gradyanda doğruluğunu artıran uyarlamalı bir grafik kesme tekniği ile büyüyen istatistiksel bir bölge kullanılarak gerçekleştirilir.

Charifi ve ark. (2018), RGB ve HSV renk uzaylarının Tohumlu Bölge Büyütme (TBB) (SRG-Seeded Region Growing) algoritmasının segmentasyon sonucu üzerindeki etkisi incelenmiştir. Uygulanan yöntem renk ve mekân özelliklerine dayalı otomatik tohum seçimi, Öklid mesafesine dayanan benzerliğine bağlı olarak bölge büyüme ve bölge birleştirme aşaması, aşırı bölümlenme sorununun üstesinden gelme ve sonuçların doğruluğunu artırma olarak üç adımdan oluşmaktadır.

Bruntha ve Kanimozhi (2018)'de BT kullanılarak akciğer kanserinin erken teşhisi için etkili bir metodoloji önerilmiştir. Başlangıçta, parazitleri gidermek için ön işlemlerde Gauss filtresi kullanılmıştır. Akciğer parankimini segmentlere ayırmak için uyumlu bir yoğunluk eşikleme metodu kullanılmıştır. Nodülleri tespit etmek için, Seçici Bölge Büyütme (SBB) (Selective Region Growing) algoritmasının yanı sıra morfolojik işlem kullanılmıştır. Anormal nodüller, bölgeye göre segmentlere ayrılan görüntü filtre edilerek tespit edilmiştir. Bu araştırmada, bir nodülün normal mi yoksa anormal mi olduğunu tespit etmek için sadece alan dikkate alınmıştır ve algılamada %95 doğruluk elde edilmiştir.

Merzougui ve El Allaoui (2019)'da yazarlar, bölge büyüme ve evrimsel algoritmalara dayanan bir segmentasyon yöntemi önermiştir. Segmentasyondan önce, sınıfların sayısı maksimum entropi ilkesine göre belirlenir. Birleştirilmiş Bölge Genişleme (CRG-Combined Region Growing) ve evrimsel algoritma, olası bölümlerin arasından, segmentlemeyi onaylamak için bir kriteri maksimize ederek en uygun bölümün seçilmesinden oluşur. Önerilen yöntem sentetik görüntü ve gerçek metal parçalar üzerinde test edilmiştir. Deneysel sonuçlar yakınsama hızının ve sunulan

segmentasyon yönteminin iyi performanslarını ve bunun kalite kontrol için verimli kullanıldığını göstermektedir.

Optimizasyon, maliyet ve enerji tüketimini en aza indirmek veya kârı, verimi, performansı ve verimliliği en üst seviyeye çıkarmak için en iyi girdileri bulma süreci olarak tanımlanabilir (Neri ve Cotta, 2012). Metaheuristik algoritmalar genellikle doğadan esinlenmiştir ve bunlar günümüzde optimizasyon için en yaygın olarak kullanılan algoritmalar arasındadır (Talbi, 2009). Ayrıca, metaheuristik algoritmalar çok çeşitlidir ve genetik algoritmalar (GA) (GA-Genetic Algorithm) (Mitchell, 1995), Benzetilmiş Tavlama (BT) (SA-Simulated Annealing) (Aarts ve ark., 2005), Diferansiyel Gelişim (DG) (DEA-Differential Evolution Algorithm) (Qin ve ark., 2008), Karınca Kolonisi Optimizasyonu (KKO) (ACO-Ant Colony Optimization) (Dorigo ve Stützle, 2003, Dorigo ve Blum, 2005, Dorigo ve Birattari, 2010), Parçacık Sürüsü Optimizasyonu (PSO) (PSO-Particle Swarm Optimization) (Eberhart ve Kennedy, 1995), Armoni Araştırması (AA) (HS-Harmony Search) (Lee ve Geem, 2004, Mahdavi ve ark., 2007), Yarasa Algoritması (YA) (BAT-Bat Algorithm) (Yang, 2010), Evrimsel Programlama (EP) (EC-Evolutionary Computations) (Back, 1996) ve diğerlerini içerirler. Son algoritmalarından bazıları ise Gri Kurt Optimizasyonu (GKO) (GWO-Gray Wolf Optimization) (Mirjalili ve ark., 2014), Yapay Arı Kolonisi (YAK) (ABC-Artificial Bee Colony) (Karaboga ve Basturk, 2007), Ateşböceği Algoritması (AA) (FA-Firefly Algorithm) (Rajinikanth ve Couceiro, 2015), Guguk Kuşu Arama (GKA) (CSA-Cuckoo Search Algorithm) (Gandomi ve ark., 2013), Yerçekimi Arama Algoritması (YAA) (GSA-Gravitational Search Algorithm) (Rashedi ve ark., 2009), Ücretli Sistem Arama (ÜSA) (CSS-Charged System Search) (Kaveh ve Talatahari, 2010), Ray Optimizasyonu (RO) (Kaveh ve Khayatazad, 2012), Çarpışan Cisimler Optimizasyonu (ÇCO) (CBO-Colliding Bodies Optimization) (Kaveh ve Mahdavi, 2014), Hibrit Parçacık Sürü Optimizasyonu (HPSO) (Neshat ve ark., 2013), Demokratik Parçacık Sürü Optimizasyonu (DPSO) (Kaveh ve Zolghadr, 2014), Yunus Ekolokasyonu (YE) (DE-Dolphin Echolocation) (Kaveh ve Farhoudi, 2013), Parçacıkların Kaotik Sürüsü (PKS) (Liu ve ark., 2005), Balina Optimizasyonu (BO) (Whale Optimization) (Mirjalili ve Lewis, 2016) algoritması ve Karınca Aslanı Optimizasyonu'dur (KAO) (ALO-AntLion Optimization) (Mirjalili, 2015).

Optimizasyon algoritmaları, belirli bir problem için bir veya daha fazla rastgele çözüm oluşturarak, bir arama alanında global optimum (en iyi) aramaktadır (Mirjalili, 2015). Bu sete “aday çözüm seti” adı verilmektedir. Aday set daha sonra, sona eren bir

koşul yerine getirilinceye kadar tekrarlı olarak geliştirilir. İyileşme, global optimumun ilk rastgele tahminlerden daha doğru bir yaklaşımını bulmak olarak düşünülebilir. Bu mekanizma, evrimsel algoritmalara problem bağımsızlığı, türev bağımsızlığı, yerel optimadan kaçınma ve basitlik gibi birçok avantaj getirmektedir (Mirjalili, 2015). Görüntü segmentasyonunda Ateşböceği Algoritması (Sharma ve Sehgal, 2016), Genetik Algoritma (Zanaty ve Ghiduk, 2013), Parçacık Sürüsü Optimizasyonu (Mirghasemi ve ark., 2013), Yapay Arı Koloni Optimizasyonu (Yigitbasi ve Baykan, 2013; Sağ ve Çunkaş, 2015), Karınca Aslanı Optimizasyonu (Mostafa ve ark., 2016), Gri Kurt Optimizasyonu (Li ve ark 2017; Koc ve ark., 2018), Karınca Kolonisi (Mao ve ark., 2009) ve diğerleri gibi birçok optimizasyon algoritması uygulanmıştır.

Mirghasemi ve ark. (2013)'de PSO'ya dayalı büyüyen bir tohum bölge tanımlanmıştır. Yöntem, PSO'nun görüntü segmentasyonuna yeni bir katkı sağlayan yaklaşımlarından biri olarak düşünülebilir. Berkeley Segmentasyon Veri Seti, deney için kullanılmıştır. Önerilen yöntem, literatürdeki veri setleri üzerinde başarılı bir şekilde test edilmiş ve sonuçlar mevcut popüler yöntemlerden çok daha iyi sonuçlar vermiştir.

Punitha ve ark (2018)'deki çalışmada, Yusufçuk optimizasyon algoritması (DA- Dragonfly Algorithm) adı verilen bir sürü optimizasyon tekniğini kullanmışlardır. Yapılan çalışmada ilk tohum noktalarının ve eşiklerin optimal olarak oluşturulduğu optimize edilmiş bir bölge büyütme tekniği kullanılmış ve tohumların otomatik tespit yöntemi önerilmiştir. Doku özellikleri, segmentlenmiş görüntülerden Gri Seviye eş Oluşum Matrisi (GLCM-Gray Level Co-occurrence Matrices) ve Gri Seviye Dizi Uzunluğu Matrisi (GLRLM-Gray Level Run Length Matrix) teknikleri kullanılarak çıkarılmış; görüntüler, iyi ve kötü olarak sınıflandıran geri yayılma algoritması kullanılarak eğitilmiş bir İleri Beslemeli Sinir Ağı (FFNN-Feed Forward Neural Network) sınıflandırıcısına aktarılmıştır. Önerilen tekniğin performansı, DDSM veri tabanından elde edilen görüntüler kullanılarak değerlendirilmiştir. Önerilen piksel tabanlı tekniğin elde ettiği sonuçlar, ROC analizi kullanılarak diğer bölge büyütme yöntemleriyle karşılaştırılmıştır. Önerilen sistemin duyarlılığı %98.1'e, özgüllüğü ise %97.8'e ulaşmıştır.

Mostafa ve ark. (2016) çalışmasında, Karınca Aslanı Optimizasyon algoritmasına bağlı olarak karaciğer segmentasyonu için bir yaklaşım önerilmiştir. MR görüntülerinde segmentasyon işlemini gerçekleştirmek için kümeleme tekniği olarak KAO kullanmıştır. KAO, tüm karaciğeri segmentlemek için karaciğerin istatistiksel bir

görüntüsü ile birleştirilmiş ve karaciğerin parçalanmış bölgesi bazı morfolojik işlemlerle iyileştirilmiştir. Daha sonra, ortalama kaydırma kümeleme tekniği bölümlenmiş karaciğeri bir dizi ROI ayırmıştır. Karaciğeri parçalamak ve önerilen yaklaşımı test etmek için bir dizi MR görüntüsü kullanılmıştır. Yapısal Benzerlik İndeksi yaklaşımın başarısını onaylamaktadır. Deneysel sonuçlar, önerilen yaklaşımın genel doğruluğunun %94.49 ile sonuçlandığını göstermiştir.



3. MATERYAL VE YÖNTEM

3.1. Dijital Görüntü

Görüntü $f(x,y)$ matematiksel fonksiyonu ile gösterilen iki boyutlu nesnelendir (Şekil 3.1). Burada x,y , yatay ve dikey olarak iki koordinattır ve “piksel” olarak adlandırılır. Her $f(x,y)$ noktasına resmin o noktasındaki piksel değeri verilir.



Şekil 3.1. Dijital renkli görüntü örneği

Şekil 3.1'deki resim, bilgisayardaki dijital görüntünün basit bir örneğidir. Bu resim, renkli bir görüntüdür. Dijital görüntüler Siyah-Beyaz (Black-White-BW), Gri Seviye (Grayscale) ve Renkli (Colored) olmak üzere 3 ayrı tipte olabilir. Siyah-Beyaz görüntüler için piksel değeri sadece 0 ya da 1 olurken; Gri Seviye görüntülerde piksel değeri $[0, 255]$ arasındadır ve 256 farklı durum için bilgi 8-bit ile gösterilir. Renkli görüntü için ise Şekil 3.1'de görüldüğü gibi R-G-B (Red-Green-Blue) (Kırmızı-Yeşil-Mavi) olmak üzere 3 ayrı renk kanalında aynı büyüklükte üç resim matrisi vardır ve renkli görüntü bu üç matrisin birleşiminden oluşur. Renkli görüntüde, her bir renk değeri bu 3 temel rengin belirli oranda karışmasıyla elde edilir. Örneğin RGB olarak her renk kanalında $[R=0, G=0, B=0]$ değeri alınırsa siyah renk elde edilirken; $[R=255, G=255, B=255]$ değerleri beyaz rengi ve $[R=255, G=255, B=0]$ değerleri ise sarı rengi vermektedir.

Bir kameradan resim yakalamanın işlenmesi fiziksel bir prosedürdür. Bir ışık kaynağı aydınlatma için kullanılır. Görüntüyü algılayan sensör kümesi, görüntüyü elde

etmek için kullanılır. Böylece ışık kaynağı ögenin üzerine geldiğinde, o noktada nesnenin yansıttığı ışık ölçüsü sensörler tarafından algılanır ve tespit edilen bilginin ölçülmesiyle sabit bir voltaj sinyali üretilir. Bilgisayarda dijital resim yapmak için bu bilgiyi gelişmiş bir yapıya dönüştürmek gerekir. Bu süreç örnekleme ve nicelemeyi içerir. Örneklemenin ve nicelemenin etkisi, sadece dijital bir resim olan iki boyutlu bir dizi veya sayıların kafeslenmesiyle sonuçlanır (Marques, 2011; Solomon ve Breckon, 2011).

Matematiksel bir bakış açısına göre, anlamlı 2 boyutlu sayılar dizisi görüntü olarak kabul edilebilir. Gerçek dünyada görüntüleri etkili bir şekilde göstermemiz, saklamamız, ağlar üzerinden iletmemiz ve sayılara karşılık gelen sayısal veri gövdelerini tanımlamamız gerekir. Bu, standart dijital görüntü formatlarının geliştirilmesine yol açmıştır. Basit bir ifadeyle, görüntü formatları bir dosya başlığını (görüntü verilerinin tam olarak nasıl depolandığına dair bilgi içeren) ve gerçek sayısal piksel değerlerinin kendilerini içerir. En yaygın kullanılan 2 boyutlu görüntü formatlarından bazıları Çizelge 3.1'de listelenmiştir (Solomon ve Breckon, 2011). Çizelgede, özellikler sütununda kısaca her formatın sıkıştırma türü kayıplı ve kayıpsız olarak açıklanmaktadır.

Çizelge 3.1. En yaygın görüntü formatları

| Kısaltma | İsim | Özellikleri |
|-----------------|------------------------------------|--|
| GIF | Grafik değişim formatı | Sadece 256 renkle sınırlıdır (8 bit); kayıpsız sıkıştırma |
| JPEG | Birleşmiş Fotoğraf Uzmanları Grubu | Günümüzde en sık kullanılan; kayıplı sıkıştırma; ayrıca kayıpsız değişkenler de mevcut |
| BMP | Bit harita resmi | Temel görüntü formatı; sınırlı (genellikle) kayıpsız sıkıştırma; ayrıca bazı kayıp varyantlar mevcut |
| PNG | Taşınabilir Ağ Grafikleri | Yeni kayıpsız sıkıştırma formatı; GIF'in yerini almak üzere tasarlandı. |
| TIF/TIFF | Etiketli resim (dosya) formatı | Son derece esnek, ayrıntılı ve uyarlanabilir format; sıkıştırılmış / sıkıştırılmamış varyantlar var |

Görüntü bilgi aktarmada en önemli yöntemlerden biridir ve görüntüde değerli bilgi yığınları vardır. Görüntüyü anlamak ve bazı hedeflere ulaşmak için görüntülerden veri çıkarmak, gelişmiş görüntü işleme ve bilgisayarla görmede kritik öneme sahiptir (Solomon ve Breckon, 2011). Görüntü işleme, gelişmiş bir görüntü elde etmek veya görüntüden bazı yararlı veriler elde etmek için bir resim üzerinde gerekli işlemleri gerçekleştirme stratejisidir. Giriş verisinin bir resim olduğu ve çıktı verisinin de resim veya resim ile ilgili nitelikleri/özellikleri olabileceği bir tür sinyal işlemedir. Günümüzde, görüntü işleme hızla gelişen yenilikler yer almaktadır (Marques, 2011)

ve bilgisayar ve mühendislik disiplinlerinde de merkezi bir araştırma alanı oluşturmaktadır. Görüntü işleme temel olarak üç aşamayı birleştirmektedir:

- Görüntü işleme araçlarıyla görüntünün yüklenmesi
- Görüntüyü analiz etmek ve gerekli işlemleri yapmak
- İşlenen görüntünün çıktısını sağlamak

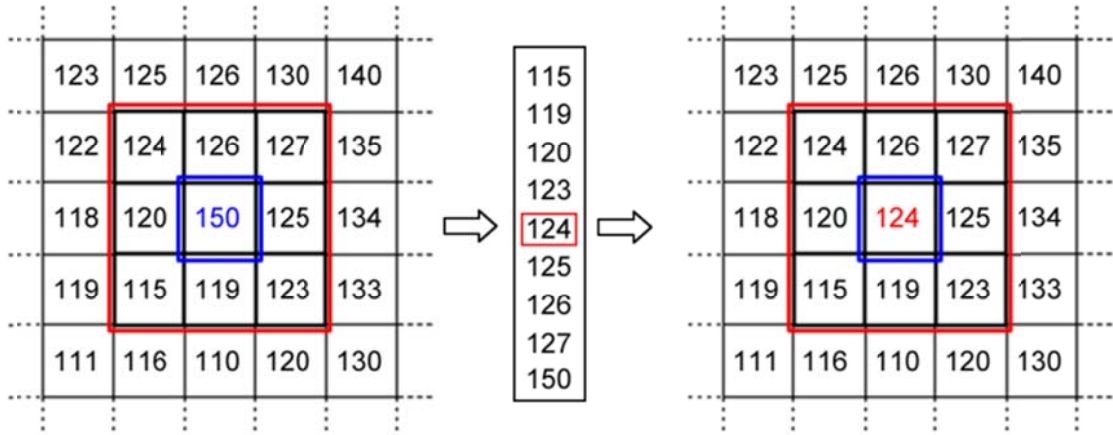
3.2. Temel Görüntü İşleme Adımları

Görüntü işlemede temel ilk adımlardan biri filtrelemedir. Filtreleme; bir görüntüde bazı özellikleri vurgulamak veya görüntüden kaldırmak, görüntünün kalitesini artırmak için kullanılan temel bir önışlem yöntemidir. Görüntü işlemede kenar bulma, keskinleştirme, düzleştirme, gürültü giderme gibi temel bazı işlemler bazı filtreleme işlemlerini kullanarak gerçekleştirilir. Her durumda, filtreler resimlerin görsel yorumlanmasına yardımcı olmakta ve benzer şekilde sonraki dijital işlem adımlarına kolaylık sağlamak için bir kılavuz olarak kullanılabilir (Bhattacharyya, 2011).

Dijital görüntüde “gürültü”; görüntüde istenmeyen, görüntüyü bozan, ayrıntıları gizleyen ve görüntünün kötü görünmesine yol açan bilgiler olarak tanımlanabilir. Tuz-Biber gürültüsü, Gauss gürültüsü, Rayleigh gürültüsü, Üstel gürültü gibi çok farklı gürültü tipi bulunmaktadır. Tuz-Biber (salt and pepper) ve Gauss gürültülerinin yok edilmesi daha çok doğrusal filtreler ile yapılmaktadır. Gauss ve Ortalama (Mean), doğrusal filtrelerin örnekleridir. Doğrusal filtrelerin uygulanması kolaydır ama doğrusal olmayan filtreler kenarları korur ve çeşitli gereksiz gürültüleri çıkarmada daha güçlüdürler. Doğrusal olmayan filtreler, doğrusal filtrelerden daha güçlüdürler çünkü kenarları bulanıklaştırmadan gürültü seviyelerini azaltabilirler. Bu tür filtrelerin tasarımı zordur. Bir ortanca (medyan) (median) filtre doğrusal olmayan filtre için iyi bir örnektir (Bhattacharyya, 2011).

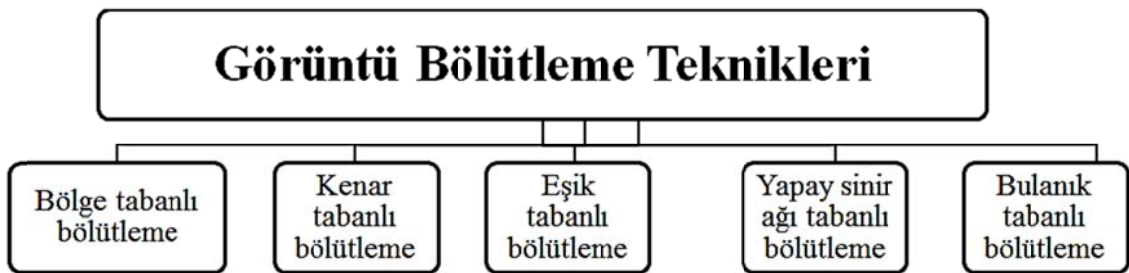
Ortanca (medyan) filtre, “tuz ve karabiber” (salt and pepper) gürültüsünü azaltmak için görüntü işlemede sıklıkla kullanılan temel bir doğrusal olmayan filtredir. Bir piksel ve diğer piksel arasındaki yoğunluk çeşitliliği ölçüsünü azaltmak için kullanılır. Bu tür bir filtre, belirlenen bir komşuluktaki piksellerin merkezinde bulunan piksel değerini elde etmeye çalışır (Teoh ve Ibrahim, 2012). Ortanca filtrede, hesaplama yapılacak pikselin değeri, belirlenen komşuluktaki piksellerin sıralanıp, sıranın ortasındaki değer alınarak hesaplanır (Şekil 3.2). Ortanca filtre, kenarları korurken gürültüyü gidermede çok etkilidir. Aynı zamanda, ortanca filtre ile filtrelemede amaç,

işlem zamanını azaltma ve kenarları korumada konvolüsyondan daha başarılı olmasıdır (Teoh ve Ibrahim, 2012).



Şekil 3.2. Ortanca filtre uygulama örneği

Filtrelemenin ardından yapılan işlem adımlarından biri segmentasyondur. Görüntü segmentasyonu (bölütleme), önışlemenin temel aşamalarından biridir ve görüntüyü birden fazla alt parçaya bölme işlemidir. Segmentasyonun amacı, bir görüntüyü daha anlamlı ve analiz edilmesi kolay olan alt parçalara dönüştürmektir. Görüntü segmentasyonu, görüntü işleme uygulamalarında kritik bir çalışma ve en sıkıntılı adımlardan biri olarak gösterilmektedir. Birçok görüntü segmentasyonu algoritması bulunmaktadır. Ancak genel olarak, görüntü segmentasyonu 5 temel başlıkta toplanmıştır (Zaitoun ve Aqel, 2015) (Şekil 3.3).



Şekil 3.3. Yaygın görüntü segmentasyon teknikleri (Zaitoun ve Aqel 2015)

3.3. Görüntü segmentasyonu

Görüntü segmentasyonu çalışma farklılıklarına göre, 5 temel başlıkta toplanmıştır.

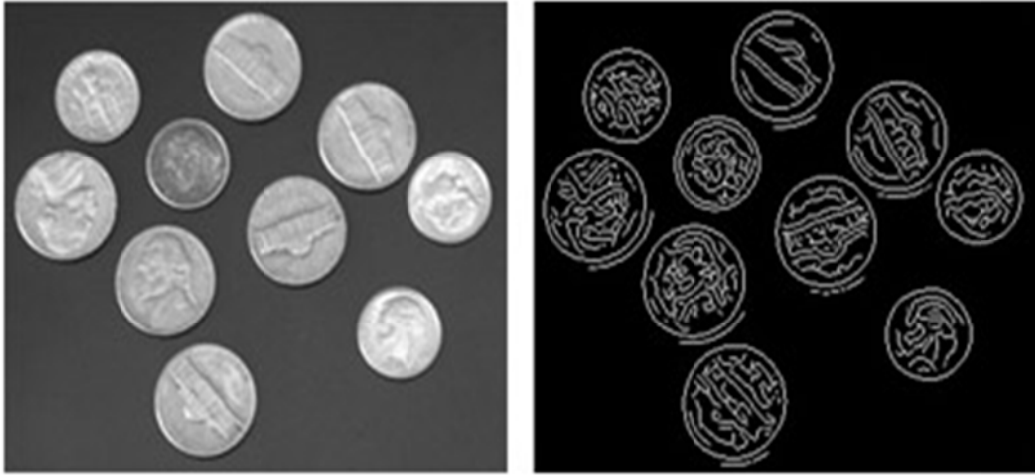
1) Bölge tabanlı segmentasyon: Bir resimde R ile gösterilen bölge, görüntünün ilişkili bir homojen alt kümesi olarak tanımlanmaktadır. Bu alt küme, gri boyut veya doku gibi bir özellikle tanımlanabilir. Bir resimdeki ayrıştırılan alt homojen bölgeler, karşılaştırılabilir özelliklere sahip benzer piksellerden oluşan bir gruptur. Bölge yaklaşımında, her piksel belirli bir nesneye veya bölgeye aittir. Kenar algılama tekniği ile karşılaştırıldığında, bölgeye bağlı segmentasyon algoritmaları genellikle basit ve gürültüye karşı daha güvenlidir (Zaitoun ve Aqel, 2015). Genellikle, nesneye uygun bölge tabanlı pikseller birleştirilir ve birlikte işaretlenir. Bölge tabanlı olarak aynı şekilde uygun eşikleme yöntemlerinin kullanılması gerekmektedir. Temel standartlar, yakınlık (gri değer kontrastlarını ve gri değer varyans değişimi içeren) ve bir bölgenin kompaktlarının yanı sıra Öklid aralığı da içeren uzamsal yakınlık ölçütüdür (Zaitoun ve Aqel, 2015).

Bölge tabanlı segmentasyon algoritmaları Bölge Büyütme (BB) (Region Growing-RG), Bölge Ayırma/Birleştirme (Region Splitting/Merging) olmak üzere iki temel sınıfa ayrılır (Kang ve ark., 2009; Zaitoun ve Aqel, 2015). Bölge büyütme, önceden tanımlanmış bazı kriterlere bağlı olarak görüntünün bir alanını bulmak ve genişletmek için bir yöntemdir (Cui ve ark., 2008). Bu model yoğunluk bilgisine bağlıdır. Bölge büyütme, komşu piksellerin analiz edildiği ve kenarları algılanmayan bir alan sınıfına eklendiği resim segmentasyonu yapmanın bir yoludur (Zaitoun ve Aqel, 2015). Bu eylem dizisi, bölgenin her piksel sınırı için tekrarlanacaktır. Zayıf veya bitişik bölgelerin keşfedilmesi durumunda, zayıf kenarların çözüldüğü ve güçlü kenarların sağlam kaldığı bir bölge birleştirme algoritması kullanılır. Bölge Ayırma/Birleştirme, genişlemenin tersidir. Bu strateji, tüm görüntü üzerinde çalışır. Bölge bölme yukarıdan aşağıya çalışan bir metodolojidir. Süreç bütün bir görüntü ile başlar, daha sonra izole bölge bütün resimden daha homojen olacak şekilde ayrılır. Bu nedenle, bölme kısmından sonra bir birleştirme aşaması sürekli gereklidir. Buna bölme ve birleştirme algoritması denir (Cui ve ark., 2008). Herhangi bir alan alt bölgelere ayrılabilir ve uygun bölgeler daha sonra birleştirilebilir. Belli noktaları seçmek yerine, kullanıcı bir resmi birçok öznel ayrılmış bölgeye parçalayabilir ve daha sonra aynı işlevleri yerine getirmeye çalışan bölgeleri birleştirebilir (Qian Zhao ve ark., 2014; Malarvel ve ark., 2016; Merzougui ve El Allaoui, 2019).

2) Kenar tabanlı segmentasyon: Kenar tabanlı segmentasyon, resimlerdeki nesnelerin sınırlarını kullanarak işlem yapan bir görüntü işleme prosedürüdür (Senthilkumaran ve Rajesh, 2009). Bu yöntemin avantajı, hesaplamaların basit ve aktivite

hızının daha hızlı olmasıdır. Parlaklıktaki süreksizlikleri kullanarak çalışır. Görüntü tanıma ve birçok alanda bilgi çıkarma için kenar tanıma özelliği kullanılır. Görüntü segmentasyonu için kenar algılama tekniğinde çok sayıda çalışma yapılmıştır. Ancak en genel olarak kullanılan kenar belirleme yöntemleri arasında Sobel, Laplacian, Prewitt, Robert Kirsh, Robinson, Marr-Hildreth, LoG ve Canny bulunmaktadır (Kaganami ve Beiji, 2009; Senthilkumaran ve Rajesh, 2009).

Görüntüdeki farklı gri değerlere sahip iki komşu bölge arasında bir kenar vardır (Senthilkumaran ve Rajesh, 2009). Bu kenarda, gri değer süreklilik göstermez. Ayrıklık veya kesilme, türev işlemlerin kullanılmasıyla kenar tespit edilen noktaları belirtir (Senthilkumaran ve Rajesh, 2009; Zaitoun ve Aqel, 2015). Kenar tespiti genellikle, görüntü segmentasyonunu gerçekleştirmek için uzamsal alan diferansiyel operatörler vasıtasıyla yapılır. Kenar algılama operatörleri, genel olarak birinci ve ikinci dereceden diferansiyel operatörler olarak iki gruba ayrılır (Senthilkumaran ve Rajesh, 2009). Birinci derece operatörler Prewitt, Roberts ve Sobel operatörü; ikincisi ise Laplacian, Kirsch ve Wallis operatörüdür. Şekil 3.4’de Canny kenar bulma işlemi görülmektedir.



Şekil 3.4 Canny ile yapılan örnek bir kenar tespiti (Marques 2011).

3) Eşik tabanlı segmentasyon: Eşik tabanlı segmentasyon, görüntü segmentasyonunda kullanılan en az karmaşık prosedürdür ve ayrıca en yaygın olarak bilinen paralel segmentasyon stratejilerinden biridir (Bhargavi ve Jyothi, 2014). Farklı nesnelerin gri seviye değerlerini kullanarak çalışan bir algoritmalardır. Eşik segmentasyonu, lokal ve global olmak üzere 2 farklı kenar tekniğine ayrılabilir. Global kenar tekniği, resmi, bir objenin ön ve arka planı olmak üzere iki bölgesine ayırır (Zhang ve ark., 2008; Bhargavi ve Jyothi, 2014). Lokal kenar stratejisinde ise; çeşitli

segmentasyon kenarları seçilir ve görüntü çok sayıda eşik ile farklı hedef alanlara ayrılır (Bhargavi ve Jyothi, 2014).

Genellikle kullanılan eşik segmentasyon algoritması çoğu zaman sınıflar arası en büyük varyans stratejisini kullanan Otsu metodudur (Kang ve ark., 2009). Bu yöntem, gruplar arasında varyansı artırarak en uygun eşığı seçer. Bunun haricinde, entropi temelli kenar belirleme, en az hata ve eşzamanlılık matrisi, moment koruması, temel istatistik, olasılık, bulanık küme ve farklı tekniklerle birleştirilmiş eşik stratejileri gibi birçok eşikleme segmentasyon stratejisi de bulunmaktadır. Eşiklemeye özellikle ön ile arka plan arasında yüksek fark olduğu zaman, bölme etkisi kolayca elde edilebilir. Buradaki dezavantaj, resimdeki çok büyük gri tonlama ayrımının olmadığı veya resimdeki gri tonlama değerinin büyük bir maskesinin olmadığı durumlarda resim bölümü sorunları için kesin sonuçlar elde etmenin zor olmasıdır (Petitjean ve Dacher, 2011). Bu segmentasyon algoritmaları, resmin uzaysal verilerini düşünmeden sadece resmin gri seviye verilerini göz önünde bulundurduğundan, gri tondaki değişim düzensizliğine karşı hassastırlar (Petitjean ve Dacher, 2011).

4) Yapay Sinir Ağı tabanlı segmentasyon: Yapay Sinir Ağları (YSA), insan beyninin çalışma şeklini taklit eden, belirli süreç içinde bir dizi veriden çıkarım yaparak çalışan algoritmalarıdır (Kang ve ark., 2009). YSA genellikle bir sinir sistemi veya sadece bir sinir ağı olarak adlandırılır. Son yıllarda, yapay sinir sistemleri, birçok farklı görüntü işleme sorununu çözmek için kullanılmıştır. İnsan sinir sisteminin taklit edilmesine bağlı YSA, benzer şekilde çok sayıda paralel merkezden oluşur. YSA, özellikle insan öğrenme prosedürünü temel alarak, kendisine gelen bilgilerden önce öğrenme ve sonra da tanımadığı verileri edindiği tecrübelerden analiz etme işlemlerini yapmaktadır. YSA'da "nöron" adı verilen her birim bazı temel işlemleri yapabilir. Öğrenme prosedürü, birimler arası bağlantıların ağırlıkları değiştirilerek gerçekleştirilebilir (Ghosh ve ark., 1991). YSA'nın görüntü işlemede kullanılma amaçları; görüntü yeniden yapılandırma (reconstruction), restorasyon (restoration), görüntü içeriğinin tanınması, segmentasyon, görüntü zenginleştirme başta olmak üzere tüm farklı işlem adımlarıdır. YSA, görüntü işlemede farklı sinir ağları katmanları eğitilerek uygulanır (Ghosh ve ark., 1991).

5) Bulanık tabanlı segmentasyon: Bulanık kümelemede verinin öğeleri aynı anda birden fazla kümeye dâhil olabilir ve hangi kümeye ne kadar dâhil olduğu "üyelik seviyesi" ile belirlenir (Kang ve ark., 2009; Yang ve ark., 2009). Üyelik seviyesi, belirli bir küme ile o veri bileşeni arasındaki ilişkinin gücünü ve kalitesini gösterir. Bulanık

kümeleme, bu üyelik seviyelerini tahsis etme prosedürüdür ve bunları kullanarak en az bir veya daha fazla kümeye verileri atamak için kullanılmaktadır (Yang ve ark., 2009). Çoğunlukla resim segmentasyonunda kullanılan en çok bilinen bulanık küme algoritması Bulanık C-Ortalamalar (BCO) (Fuzzy C-Means – FCM) algoritmasıdır. BCO algoritması, belirlenen bir kritere göre C tane bulanık kümeye N adet bileşeni bölmek için kullanılır. Verilerin düzenlenmesi göz önüne alındığında, BCO algoritması C küme merkezlerinin ve W segment matrisinin (N adet verinin C kümeye bölünmesi matrisi) sonucunu döndürür (Yang ve ark., 2009; Naz ve ark., 2010). Bulanık kümelemede her bir noktanın, tamamen bir grupta bir yere sahip olmak yerine, bulanık mantıkta olduğu gibi kümelerde yer alma düzeyi vardır. Bulanık mantık prensibi gereği her verinin, kümelerin her birine aitliği $[0,1]$ arasında değişen birer üyelik değeri ile belirlenir. Bir verinin tüm sınıflara olan üyelik değerleri toplamı “1” olmalıdır. Nesne hangi küme merkezine yakın ise o kümeye ait olma üyeliği diğer kümelere ait olma üyeliğinden daha büyük olacaktır. BCO, özellikle bir resimdeki nesnelerin kümelenmesinde yaygın olarak kullanılmıştır (Yang ve ark., 2009).

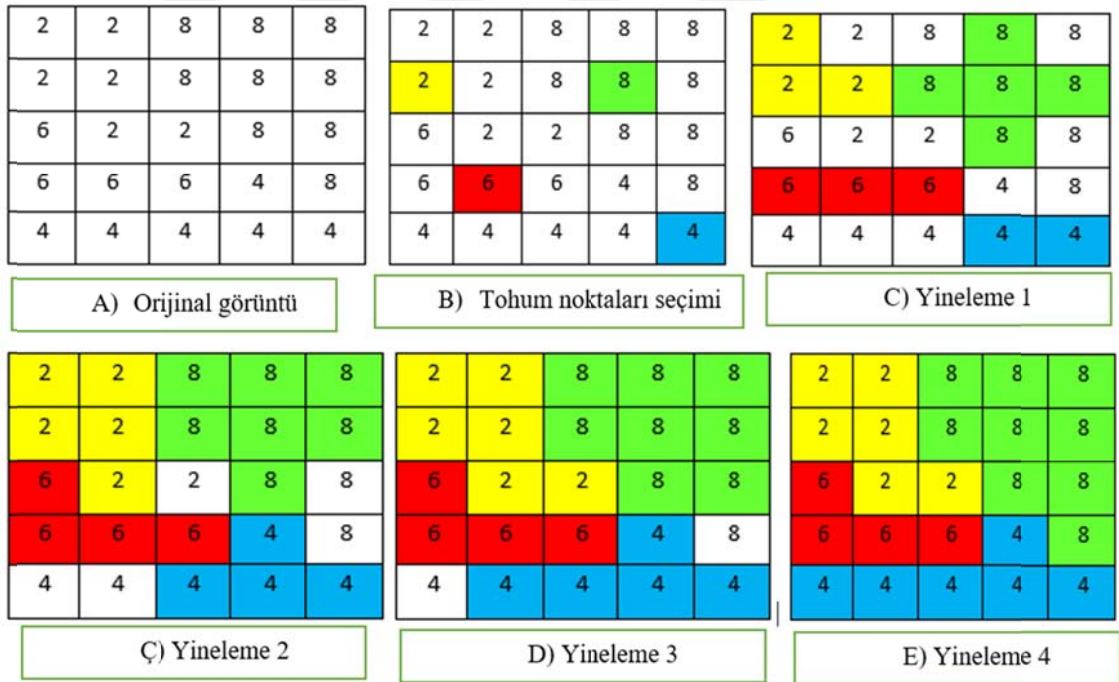
3.4. Bölge tabanlı görüntü segmentasyon yöntemleri (BTSY)

Bölge tabanlı segmentasyonda, bir bölge, tamamen aynı özelliklere sahip birbirine bağlı bir piksel grubu olarak tanımlanır. Genel olarak, segmentlenmiş bölgeler görüntünün içeriğini parçalı olarak analiz etmek için kullanılırlar. Bir bölge çoğunlukla görüntünün belirli bir kısmı veya bölümü ile ilgilidir (Zaitoun ve Aqel, 2015). Genelde bölgelerin birbirleri ile paylaştığı bağlantı, kompaktlık, sınırların tutarlılığı, renk veya dokudaki homojenlik ve komşu bölgelerden ayrılma gibi özellikler vardır. BTSY'in temel amacı resimleri alanlara bölmektir ve kenar tabanlı segmentasyon teknikleri gibi, gri tonlama veya ton özelliklerinde düzensizliğe bağlı bölgeler arasındaki sınırlara göz atmak suretiyle bu amacı gerçekleştirir (Cui ve ark., 2008). BTSY, bölgeye özel olarak karar verilmesine ilişkin bir prosedürdür (Zaitoun ve Aqel, 2015).

Bölge tabanlı segmentasyon algoritmaları Bölge Büyümesi, Bölge Ayırma/Birleştirme olmak üzere iki temel sınıfa ayrılır (Cui ve ark., 2008).

1) Bölge Büyütme (BB): Bölge temelli stratejilerdeki en zor tekniktir (Kelkar ve Gupta, 2008; Kang ve ark., 2009). Bu algoritmadaki temel fikir, görüntüdeki aynı özelliğe sahip bitişik pikselleri kontrol etmektir. Bu noktada algoritma, komşu piksellerin aynı bölge içine eklenip eklenmeyeceğine karar verir (Zaitoun ve Aqel,

2015). Dolayısıyla, benzer piksellere ulaşılan ve bir araya getirilinceye kadar tekrarlı bir prosedür uygulanır. Bu algoritma ilk önce başlangıç noktasını rasgele olarak seçer (Cui ve ark., 2008). Daha sonra, eğer başlangıç noktasının komşu pikselleri; yoğunluk, gri ya da renkli tonlama gibi bir benzerlik kriterini karşılırsa, komşu pikseller başlangıç noktası ile birleştirilerek, başlangıç noktasının alanı büyütülür (Cui ve ark., 2008). Bölge büyütmenin çalışma örneği Şekil 3.5'te gösterilmektedir. Şekilde ilk önce 4 farklı başlangıç piksel noktası seçilmiştir (Şekil 3.5-A). Ardından, her bir başlangıç pikseli, komşu pikseline göre kontrol edilmektedir. Şekil 3.5'teki örnekte, ana başlangıç noktalarının komşu piksellerini genişletmek için 4 bağlantılı komşuluk kullanılmıştır. Komşu piksel değeri belirlenen kritere göre başlangıç noktalarından biriyle aynı olduğu zaman, eşleştiği başlangıç noktasına eklenmektedir (Şekil 3.5-C, Ç, D) (Kang ve ark., 2009). Tüm benzer pikseller kontrol edilip, uygun noktalara eklenene kadar işlem devam etmektedir (Şekil 3.5-E).



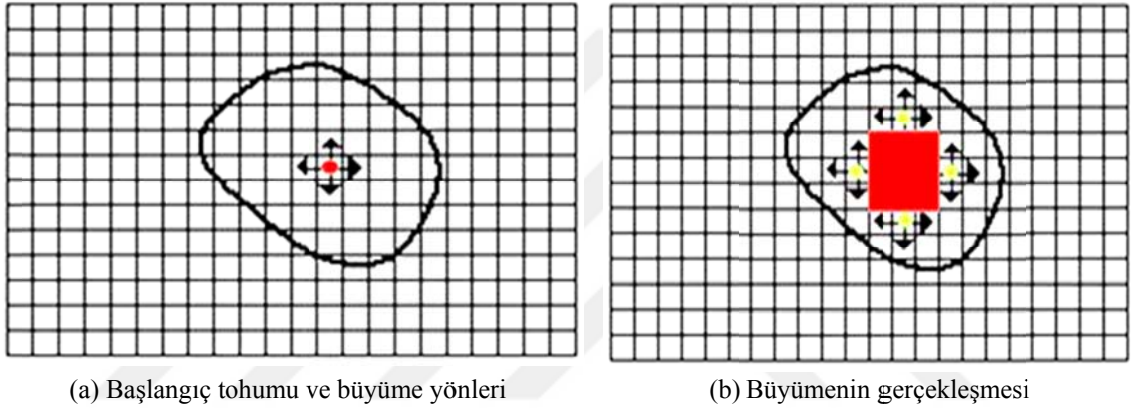
Şekil 3.5 Bölge büyütme örneği (Kelkar ve Gupta 2008)

Tohum noktası seçimi, bölge büyütme (BB) sürecinin önemli bir parçasıdır, çünkü bölge büyütme algoritmasına yaklaşım, başlangıç tohumunun seçilmesiyle başlar. Doğru tohum noktasını seçmek daha başarılı bir görüntü segmentasyon sonucu sağlar. BB, bölümlere ayrılacak olan görüntüye her bir tohumun tek bir piksel veya bir dizi bağlı piksel olabileceği, bir dizi tohum yerleştirilmesi ile başlar. Daha sonra bu

tohumlara art arda komşu pikseller eklenerek bölgelere dönüştürülür. Rastgele tohumlar aşağıdaki üç kritere göre seçilir:

- Tohum pikselinin komşularıyla yüksek benzerliği olmalıdır.
- Hedeflenen bir bölgenin üretimi için en az bir tohum üretilmelidir.
- Farklı bölgelerdeki tohumların bağlantısı kesilmelidir.

İlk rastgele tohum seçilmesinden sonra, tohumlar ve komşu pikseller arasındaki mesafe ölçülerek segmentler oluşturulmaktadır. BB sürecinde ilk rastgele tohumların seçimi ve ilk tohum noktasından itibaren bölge büyütmenin örnek gösterimi Şekil 3.6'da verilmiştir.



Şekil 3.6 Bölge büyütmede ilk tohum seçimi ve büyütme

Şekil 3.6 (a)'da kırmızı daire ilk başlangıç tohum pikselidir ve siyah oklar o bölgenin büyüyeceği yönleri göstermektedir. Şekil 3.6 (b)'de ise bölge büyütmesi görülmektedir.

Çizelge 3.2'de bölge büyütme algoritmasının avantaj ve dezavantajları verilmiştir.

Çizelge 3.2 Bölge büyütme algoritmasının avantaj-dezavantajı

| BB'nin Avantajları | BB'nin Dezavantajları |
|---|--|
| Aynı özelliklere sahip olan pikselleri kolay bir şekilde ayırt eder. | Hesaplamanın maliyeti pahalıdır, çünkü güçlü bir bellek ve işlemci gerektiren özyinelemeli bir işlemdir. |
| Net kenarlara sahip bir çıktı sunar. | Görüntü işlemede ortak bir perspektif sorunu olan komşu tabanlı bir tekniktir. |
| Yöntem son derece basittir ve az sayıda başlangıç noktası seçmek gerekir. | Gürültüye karşı daha hassastır. |

2) Bölge Ayırma/Birleştirme: Bu algoritmada bölme ve birleştirme işlemleri beraber uygulanmaktadır. Bölge bölme ve birleştirme, görüntünün homojenliğini ayırt etmek için kullanılan bölge segmentasyon yöntemidir (Zaitoun ve Aqel, 2015). Bu algoritmada, görüntü önce homojen küçük bölgelere ayrılmakta, daha sonra benzer özelliklere sahip komşu alanlar belirlenen bir kritere göre birleştirilmektedir (Kelkar ve Gupta, 2008; Zaitoun ve Aqel, 2015).

Bölge bölme algoritması, görüntünün genelinden başlayarak, her biri kendi içinde homojen olan alt bölgelere görüntünün ayrılmasıdır. Bütün görüntü daha küçük bölgelere ayrılmakta, bir ağaç yapısı içinde bütünü oluşturan parçalar şeklinde değerlendirildikten sonra benzerlik ölçüsünde nesne ve bölgeler tespit edilmeye çalışılmaktadır. Yöntemin doğru ve kabul edilebilir sonuçlar verebilmesi için görüntünün mümkün olduğu kadar küçük parçalara ayrılabilir olması gerekmektedir. Yöntemin bu dezavantajı yüksek karmaşıklık ve işlem kalabalığına neden olmaktadır (Kelkar ve Gupta, 2008).

Bölge birleşmesi, benzer başlangıç noktalarından başlamalıdır. Uygun bir birleşme bölgesi bulmak için farklı çalışmalar yapılmıştır. Bir çalışmada, görüntü 2x2 veya 4x4 bloklara ayrılarak, bloklar kontrol edilmektedir. Bir başka strateji ise, görüntüyü şeritlere ayırdıktan sonra, görüntüyü şeritler halinde izole etmektedir (Kelkar ve Gupta, 2008; Zaitoun ve Aqel, 2015). Burada en kötü durumda, başlangıç noktası yalnız bir piksel olarak kalacaktır. Bir başlangıç noktası keşfedildiğinde, komşu pikselleri bir başka komşu bölge, tutarlılık standardına uymayana kadar bir araya getirilir. Bu durumda, bölge tüm resimden izole edilmiştir ve diğer başlangıç noktası başka bir bölgeyi birleştirmek için kullanılmaktadır. Birleştirme yönteminin birkaç dezavantajına dikkat edilmesi gerekmektedir. Birleştirme işlemi doğal olarak ardışıktır ve segmentasyonda iyi bir sonuç gerekirse, hesaplama süresi uzar. Ayrıca, bir kural olarak, iki alanın birleştirilmesi tahmin edilen özelliğin değerini değiştireceğinden, sonraki bölge komşular arasında kullanılan toplama tekniğine ve toplanan piksellere dayanacaktır (Zaitoun ve Aqel, 2015).

3.5. Karınca Aslanı Algoritması

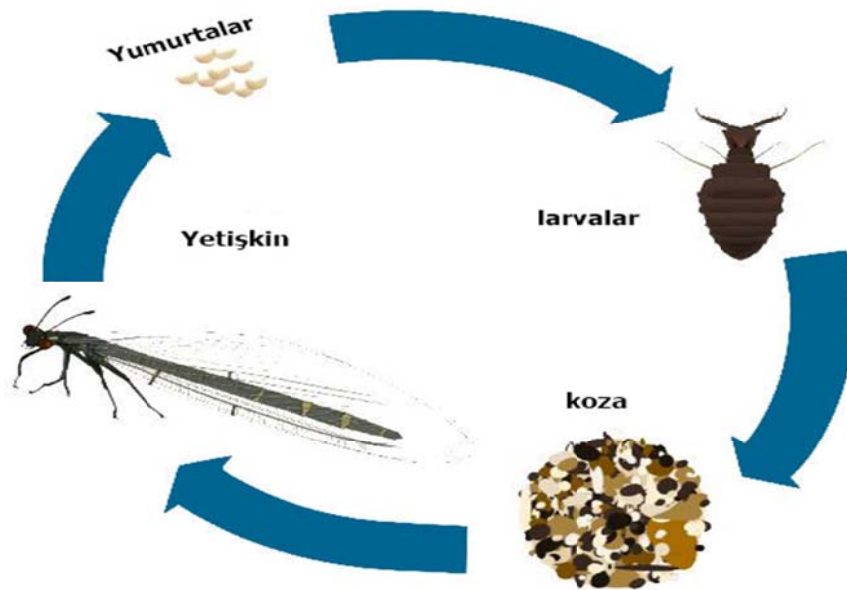
Metasezgisel algoritmalar, optimizasyon problemlerini, geniş bir olası çözümler kümesi içinde en iyi çözümü bularak çözmek için kullanılırlar (Kaveh ve Khayatazad, 2012). Optimizasyon problemleri üç özelliikle tanımlanabilir:

- Minimize veya maksimize edilecek başarı ölçüsü (“maliyet fonksiyonu” olarak isimlendirilir).
- Problemin örnekleri (tasarım değişkenleri cinsinden ifade edilen probleme girdiler). Bu örnekler bazı kısıtlamalara tabi olabilir, bu nedenle düzenleme keşfedilecek olası değerler üzerinde sınırlar koyar.
- En iyi çözümü bulmak için keşfedilecek alanı tanımlayan olası çözümler kümesi

Genel olarak, çözüm kümesini araştırmak hesaplama açısından ağırdır ve çok zaman alıcıdır. Bu nedenle bu sorunların üstesinden gelmek ve olası kısa sürede uygun ve kabul edilebilir bir çözüm bulmak için çeşitli algoritmalar bulunmaktadır (Kaveh ve Khayatazad, 2012). Bu algoritmalar tekrarlayıcıdır ve bazı kriterlere göre, bulunan çözümün optimizasyon algoritmasını durduracak kadar iyi olup olmadığına karar verilebilir. Bu mümkün olduğunda, algoritma yakınsamasından söz edebiliriz. En yaygın metasezgisel algoritmaları; Genetik Algoritmalar (GA) (Genetic Algorithm – GA) (Mitchell, 1995), Benzetilmiş Tavlama (BT) (Simulated Annealing – SA) (Aarts ve ark., 2005), Evrimsel Programlama (EP) (Back, 1996), Diferansiyel Gelişim (DG) (Differential Evolution – DE) (Qin ve ark., 2008), Karınca Kolonisi Optimizasyonu (KKO) (Ant Colony Optimization – ACO) (Dorigo ve Stützle, 2003; Dorigo ve Blum, 2005; Dorigo ve Birattari, 2010), Parçacık Sürü Optimizasyonu (PSO) (Particle Swarm Optimization – PSO) (Eberhart ve Kennedy, 1995), Armoni Araması (AA) (Harmony Search – HS) (Lee ve Geem, 2004; Mahdavi ve ark., 2007), Yarasa Algoritması (YA) (Bat Algorithm – BA) (Yang, 2010) ve diğerleridir. Son yıllarda geliştirilen diğer algoritmalarından bazıları ise Gri Kurt Optimizasyon algoritması (GKO) (Gray Wolf Optimizer – GWO) (Mirjalili ve ark., 2014), Yapay Arı Kolonisi (YAK) (Artificial Bee Colony – ABC) (Karaboga ve Basturk, 2007), Ateş Böceği Algoritması (ABA) (Firefly Algorithm – FA) (Rajinikanth ve Couceiro, 2015), Guguk Kuşu Arama (GKA) (Cuckoo Search – CS) (Gandomi ve ark., 2013), Gravitasyonel Arama Algoritması (GAA) (Gravity Search Algorithm – GSA) (Rashedi ve ark., 2009), Ücretli Sistem Arama (ÜSA) (Kaveh ve Talatahari, 2010), Çarpışan Cisimler Optimizasyonu (ÇCO) (Kaveh ve Mahdavi, 2014), Yunus Ekolokasyonu (YE) (Dolphin Echolocation – DE) (Kaveh ve Farhoudi, 2013), Parçacıkların Kaotik Sürüsü (PKS) (Liu ve ark., 2005), Balina optimizasyonu (BO) (Whale Optimization Algorithm – WOA) (Mirjalili ve Lewis, 2016) algoritması ve Karınca Aslanı Optimizasyonu’dur (KAO) (Ant Lion

Optimization – ALO) (Mirjalili, 2015). Bu tez kapsamında Karınca Aslanı Optimizasyon algoritması kullanılmıştır.

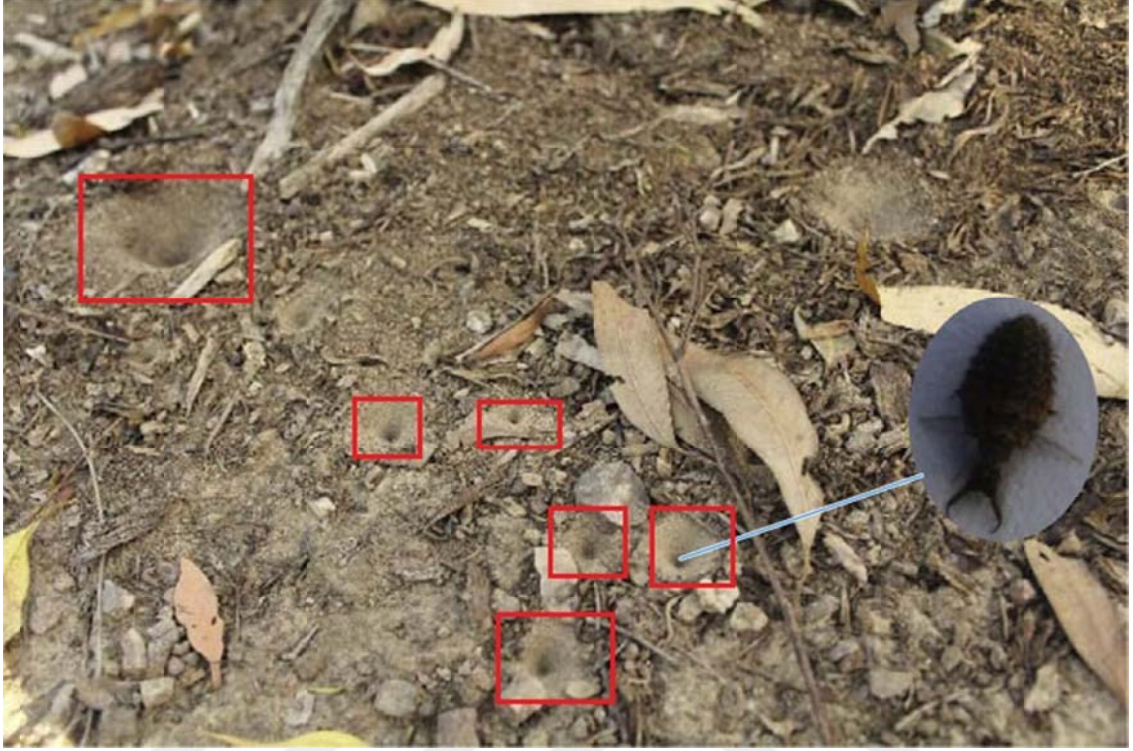
Karınca aslanları taksonomik sınıflandırmada Myrmeleontidae ailesi altındaki bir böcek grubudur. Myrmeleontidae ailesi, Dobsonflies ve Lacewings’i içeren Neuroptera sınıfının bir parçasıdır (Imms ve ark., 2012). “Karınca Aslanı” adı, özellikle karıncaları ve diğer böcekleri yiyen açgözlü avcı olan larvalara denilmektedir. Bu avcılar stratejik olarak enkaz parçaları altına ya da kendi kendine oluşan kum çukurlarının altına saklanırlar, böceklerin tuzağa düşmesini beklerler ve daha sonra hemen avlarını tüketirler. Karınca Aslanlarının yaşam döngüsü iki temel aşamadan oluşur: larva ve yetişkinlik. Karakteristik bir mutlak yaşam beklentisi, genellikle larva döneminde sürdürülen 3 yıl kadardır (yetişkinlik sadece 3-5 haftadır). Karınca Aslanları bir yetişkine dönüşmek için bir döngüde metamorfoza maruz kalmaktadır. Çoğunlukla larvadayken avlanırlar ve yetişkinlik döneminde üremeye odaklanırlar (Mirjalili, 2015) (Şekil 3.7).



Şekil 3.7 Karınca aslanının yaşam döngüsü (Kelkar ve Gupta, 2008)

“Karınca Aslanı” ismi, benzersiz avlanma davranışlarından ve en sevdikleri avdan gelmektedir. Bir Karınca Aslanı larvası, dairesel bir yol boyunca ilerleyerek ve çenesiyle kumları fırlatarak koni şeklindeki bir çukur kazar. Şekil 3.8, farklı ebatlardaki koni biçimli çukurları göstermektedir. Tuzağı kazdıktan sonra, larva koninin tabanının

altına gizlenir (bir otur-ve-bekle yırtıcı hayvanı olarak) ve böceklerin (tercihen karınca) çukurun içinde sıkışmasını bekler (Mirjalili, 2015).



Şekil 3.8 Karınca aslanları tuzak çukurları (Mirjalili, 2015).

Koninin kenarı, böceklerin tuzağın dibine kolayca düşmesi için yeterince keskindir. Karınca aslanı bir avın tuzağa düştüğünü fark ettiğinde, onu yakalamaya çalışır. Ancak, böcekler genellikle hemen yakalanmaz ve tuzaktan kaçmaya çalışırlar. Bu durumda, karınca aslanları avı çukurun dibine kaydırmak için çukurun kenarına doğru kum atarlar. Av çukurdaki çenenin içine girdiğinde ise toprağın altına çekilir ve tüketilir (Mirjalili, 2015).

Avları tükettikten sonra, karınca aslanları, kalan artıkları çukurun dışına atar ve çukuru bir sonraki av için değiştirir. Karınca aslanlarının yaşam tarzında gözlenen bir başka ilginç davranış ise, kazdıkları tuzağın boyutunun, karınca aslanlarının açlık seviyesine ve/veya ayın şekline bağlıdır. Karınca aslanları, daha aç olduklarında ve/veya dolunayda daha büyük tuzaklar kazma eğilimindedirler (Mirjalili, 2015). Hayatta kalma şanslarını arttırmak için tuzak kazmayı bu şekilde geliştirmişlerdir ve adapte olmuşlardır. Ayrıca, bir karınca aslanının, tuzağın boyutuna karar vermek için doğrudan ay şeklini gözlemediği, ancak bu kararları almak için bir iç ay saatine sahip olduğu keşfedilmiştir (Mirjalili, 2015).

3.5.1. Popülasyon oluşturma

Karınca Aslanı Optimizasyon (KAO) algoritması, karınca aslanı larvalarının yiyecek arama davranışından esinlenerek önerilmiş bir algoritmadır (Mirjalili, 2015). KAO algoritması, tuzaktaki karıncalar ve karınca aslanları arasındaki etkileşimi taklit etmektedir. Doğadaki bu etkileşimleri modellemek için, karıncaların arama alanı üzerinde hareket etmeleri ve karınca aslanlarının tuzakları kullanarak onları avlamaları örnek alınmıştır (Mirjalili, 2015). Karıncalar yiyecek ararken, rastgele olarak hareket ettiklerinden dolayı, karıncaların hareketi Denklem 3.1'deki gibi rastgele belirlenir.

$$x(t) = \pi_r \quad (3.1)$$

$$x^t = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \dots, \text{cumsum}(2r(t_n) - 1)]$$

Burada, *cumsum* kümülatif toplamı hesaplamaktadır. *n* maksimum yineleme sayısı, *t* iterasyon sayısı ve *r(t)* Denklem 3.2'deki gibi tanımlanmış bir rastgele fonksiyondur.

$$r(t) = \begin{cases} 1; & \text{eğer } \text{rand} > 0.5 \\ 0; & \text{eğer } \text{rand} \leq 0.5 \end{cases} \quad (3.2)$$

Burada, *t* iterasyon sayısı ve *rand* ise [0,1] aralığında tekdüze dağılımla oluşturulan rastgele bir sayıyı göstermektedir. Karıncaların yerleri, Denklem 3.3'teki matristeki optimizasyon sırasında kaydedilir.

$$M_{\text{karınca}} = \begin{bmatrix} k_{1,1} & k_{1,2} & \dots & \dots & \dots & k_{1,d} \\ k_{2,1} & k_{2,2} & \dots & \dots & \dots & k_{2,d} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot \\ k_{n,1} & k_{n,2} & \dots & \dots & \dots & k_{n,d} \end{bmatrix} \quad (3.3)$$

Burada $M_{\text{karınca}}$ her bir karıncanın konumunu kaydetmek için kullanılan matristir. $k_{i,j}$; *i* karıncanın *j*. değişkeninin (boyut) değerini gösterir. *n* karınca sayısı ve *d* ise değişken sayısıdır.

Karıncaların PSO'daki partiküllere veya GA'daki bireylere benzer olduğu not edilmelidir. Bir karıncanın konumu, belirli bir çözüm için parametreleri belirtir. Denklem 3.3'te $M_{\text{karınca}}$, optimizasyon sırasında tüm karıncaların (tüm çözümlerin değişkenleri) konumunu saklamaktadır. Her karıncayı değerlendirmek için, optimizasyon sırasında uygunluk (objektif) işlevi kullanılır ve Denklem 3.4'teki matris karıncaların amaç fonksiyon değerlerini saklar.

$$M_{\text{karıncaZindelik}} = \begin{bmatrix} f([k_{1,1} & k_{1,2} & \dots & \dots & \dots & k_{1,d}]) \\ f([k_{2,1} & k_{2,2} & \dots & \dots & \dots & k_{2,d}]) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot \\ f([k_{n,1} & k_{n,2} & \dots & \dots & \dots & k_{n,d}]) \end{bmatrix} \quad (3.4)$$

Burada, $k_{i,j}$; i . karıncanın j değişkeninin (boyut) değerini gösterir. n karınca sayısı ve f amaç fonksiyonudur.

Karıncalara ek olarak karınca aslanlarının da arama alanında saklandıkları konumlarını ve uygunluk değerlerini kaydetmek için Denklem 3.5 ve Denklem 3.6'daki matrisler kullanılır (KA : karınca aslanı).

$$M_{\text{karıncaAslanı}} = \begin{bmatrix} KA_{1,1} & KA_{1,2} & \dots & \dots & \dots & KA_{1,d} \\ KA_{2,1} & KA_{2,2} & \dots & \dots & \dots & KA_{2,d} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot \\ KA_{n,1} & KA_{n,2} & \dots & \dots & \dots & KA_{n,d} \end{bmatrix} \quad (3.5)$$

$$M_{\text{karıncaAslanıZindelik}} = \begin{bmatrix} f([KA_{1,1} & KA_{1,2} & \dots & \dots & \dots & KA_{1,d}]) \\ f([KA_{2,1} & KA_{2,2} & \dots & \dots & \dots & KA_{2,d}]) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot \\ f([KA_{n,1} & KA_{n,2} & \dots & \dots & \dots & KA_{n,d}]) \end{bmatrix} \quad (3.6)$$

Optimizasyon sırasında aşağıdaki koşullar uygulanır:

1) Karıncalar, farklı rastgele yürüyüşleri kullanarak arama alanı içinde hareket eder.

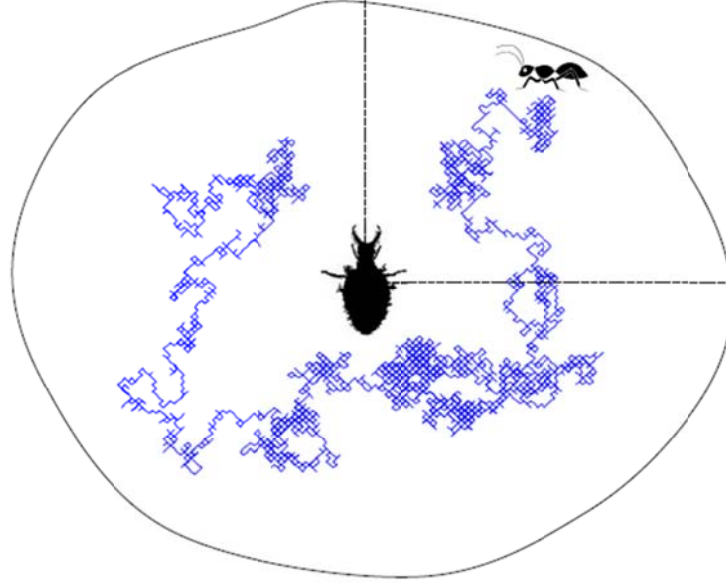
- 2) Karıncaların tüm boyutlarına rastgele yürüyüşler uygulanır.
- 3) Rastgele yürüyüşler, karınca aslanlarının tuzaklarından etkilenir.
- 4) Karınca aslanları, zindelikleri ile orantılı çukurlar oluşturabilirler (daha yüksek zindelik, daha büyük çukur demektir).
- 5) Daha büyük çukurlu karınca aslanlarının karıncaları yakalama olasılıkları daha yüksektir.
- 6) Her bir karınca, her bir yinelemede ve seçimde bir karınca aslanı (en zinde karınca aslanı) tarafından yakalanabilir.
- 7) Kayan karıncaları karınca aslanlarına doğru simüle etmek için rastgele yürüyüş mesafesi uyarlanır şekilde azaltılır.
- 8) Eğer bir karınca, bir karınca aslanından daha zinde (kurnaz) hale gelirse; bu, karıncanın, karınca aslanı tarafından kumun altına çekildiği ve yakalandığı anlamına gelir.
- 9) Bir karınca aslanı kendisini en son yakalanan avına yeniden konumlandırır ve her avdan sonra başka bir av yakalama değişikliğini geliştirmek için bir çukur oluşturur.

3.5.2. Karıncaların rastgele yürüyüşleri

Rastgele yürüyüşlerin tamamı Denklem 3.1'e dayanmaktadır. Karıncalar pozisyonlarını, optimizasyonun her aşamasında rastgele yürüyüşle güncellerler. Ancak, her arama alanının bir sınırı olduğu için (değişken aralığı), Denklem 3.1 karıncaların pozisyonlarını güncellemek için doğrudan kullanılamaz. Rastgele yürüyüşleri arama alanı içinde tutmak için (Şekil 3.9), Denklem 3.7 kullanılarak normalleştirilirler (minimum – maksimum normalizasyon):

$$X_i^t = \frac{(X_i^t - a_i) \times (U_i^t - L_i^t)}{(b_i - a_i)} + L_i^t \quad (3.7)$$

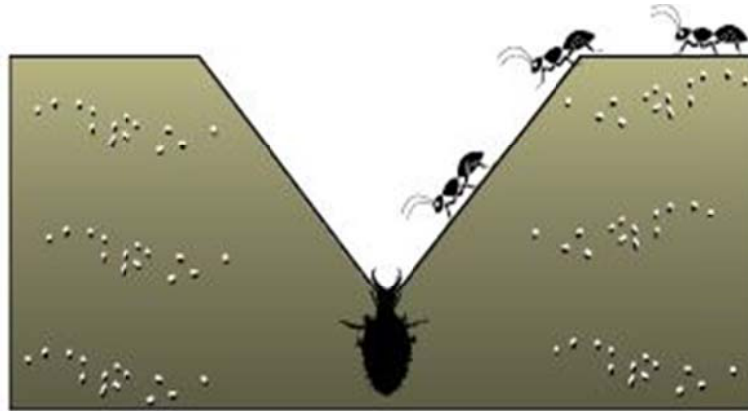
b_i ve a_i , X_i^t dizisindeki maksimum ve minimum değerlerdir. U_i^t ve L_i^t , t . iterasyonda i . boyutun üst ve alt sınırındır.



Şekil 3.9 Bir karıncanın, karınca aslanının tuzağının içinde rastgele yürüyüşü (Mirjalili 2015)

3.5.3. Tuzağın Kurulması

Karıncı aslanının avlanma kabiliyetini modellemek için bir rulet çarkı kullanılmaktadır. Şekil 3.10'da gösterildiği gibi, karıncaların seçilen tek bir karıncı aslanına yakalandıkları varsayılmaktadır (Mirjalili, 2015). KAO algoritmasında, optimizasyon sırasındaki zindeliğe bağlı bölgeleri seçmek için bir rulet tekerleği operatörü kullanımı gereklidir. Bu mekanizma, en zinde karıncı aslanlarına karıncaların yakalanması için yüksek şans verir.



Şekil 3.10. Karıncı aslanlarının avlanma davranışı(Mirjalili, 2015)

3.5.4. Tuzaklardan etkilenme ve kapana kısılma

Rastgele karınca yürüyüşleri, karınca aslanlarının tuzaklarından etkilenir. Bu durumu matematiksel olarak modellemek için Denklem 3.8 ve 3.9 önerilmiştir.

$$L_i^t = KA_j^t + L^t \quad (3.8)$$

$$U_i^t = KA_j^t + U^t \quad (3.9)$$

Burada, L^t , t . iterasyonda bütün değişkenlerin minimumunu, U^t ise maksimumunu içeren vektördür. L_i^t , i . karınca için t . iterasyondaki minimumunu, U_i^t ise maksimumunu göstermektedir. KA_j^t ise t . iterasyonda seçilen j . karınca aslanının pozisyonudur. Denklem 3.8 ve 3.9 karıncaların rastgele seçilmiş bir karınca aslanı etrafında ve vektörleri tarafından tanımlanan bir hiper kürede yürüdüğünü göstermektedir (Şekil 3.10).

3.5.5. Karınca aslanlarına doğru kayan karıncalar

KAO algoritmasında, karınca aslanları zindelikleriyle orantılı olarak tuzaklar kurabilmektedir ve karıncaların rastgele hareket etmeleri gerekmektedir. Ancak, karınca aslanları bir karıncanın tuzağa düştüğünü anladıklarında çukurun merkezinden dışarıya doğru kum atarlar. Bu davranış, kapana kısılmış ve kaçmaya çalışan karıncayı aşağıya doğru kaydırır. Bu davranışın matematiksel olarak modellenmesi için, karıncaların rastgele yürüme hiperküresinin yarıçapı uyarlanabilir şekilde azaltılır ve karıncanın çukurların içindeki kayma süreci simüle edilir (Mirjalili, 2015). Bu işlem Denklem 3.10 ile modellenmiştir.

$$L^t = \frac{L^t}{p} \quad (3.10)$$

$$U^t = \frac{U^t}{p}$$

Burada p bir orandır ve Denklem 3.12'de verilmiştir. L^t , t . iterasyonda bütün değişkenlerin minimumunu ve U^t , t . iterasyonda bütün değişkenlerin maksimumunu

içeren vektörü belirtir. L^t ve U^t adaptif olarak azaltılınca, Denklem 3.8 ve 3.9; Denklem 3.11'e göre yeniden hesaplanır.

$$\begin{cases} U_i^t = KA_j^t + \frac{U^t}{p} \\ L_i^t = KA_j^t + \frac{L^t}{p} \end{cases} \quad (3.11)$$

$$p = 10^w * \frac{t}{T} \quad (3.12)$$

Burada KA_j^t , karınca aslanının konumudur. U^t ve L^t , t . iterasyondaki üst ve alt sınırlarıdır. p ise sınırın oranıdır. Burada t mevcut iterasyon, T ise maksimum iterasyon sayısıdır. w , sömürü seviyesini ayarlamak için kullanılan ağırlık parametresidir (ağırlık) ve Denklem 3.13'e göre hesaplanmaktadır. w 'nin değeri [2-6] arasındadır (Toz, 2019) ve tez kapsamında bu değerler kullanılmıştır.

$$\begin{aligned} \text{Eğer } t > 0.01T; \quad w &= 2 \\ \text{Eğer } t > 0.5T; \quad w &= 3 \\ \text{Eğer } t > 0.75T; \quad w &= 4 \\ \text{Eğer } t > 0.9T; \quad w &= 5 \\ \text{Eğer } t > 0.95T; \quad w &= 6 \end{aligned} \quad (3.13)$$

3.5.6. Avı yakalama ve çukuru yeniden inşa etme

KAO'da av yakalama, karıncalar tuzağa düştüğü zaman ortaya çıkar. Sonra karınca aslanı, karıncayı kumun içine çeker ve kısa sürede tüketir. Bundan sonra, karınca aslanı yeni av yakalama şansını arttırmak için konumunu avlanan karıncanın en son konumuna getirmeye başlar (Mirjalili, 2015). Denklem 3.14 bu davranışı simüle etmektedir.

$$KA_j^t = k_i^t; \quad \text{eğer } f(k_i^t) > f(KA_j^t) \quad (3.14)$$

Burada t mevcut tekrarlamayı (iterasyonu) gösterir. KA_j^t , t iterasyondaki seçilmiş j 'inci karınca aslanını ve k_i^t ise t . iterasyondaki i . karıncayı belirtmektedir.

3.5.7. Elitizm

Elitizm, optimizasyon sürecinin herhangi bir aşamasında elde edilen en iyi çözümlü/çözümleri sürdürmelerini sağlayan evrimsel algoritmaların önemli bir özelliğidir. Bu çalışmada, her bir tekrarlama şimdiki kadar elde edilen en iyi karınca aslanı kaydedilmekte ve elit (en iyi birey) olarak kabul edilmektedir (Mirjalili, 2015). Elite, en zinde karınca aslanı olduğu için, yinelemeler sırasında tüm karıncaların hareketlerini etkilemektedir. Bu nedenle, her karıncanın rulet tekerleği ve elit tarafından eş zamanlı olarak seçilen bir karınca aslanı etrafında Denklem 3.15'e göre rastgele yürüdüğü varsayılmaktadır.

$$k_i^t = \frac{R_{KA}^t + R_E^t}{2} \quad (3.15)$$

Burada R_{KA}^t , t . iterasyonda rulet tekerleği ile seçilmiş karınca aslanının etrafındaki rastgele yürüyüşü; R_E^t , t . iterasyonda elit tarafındaki rastgele yürüyüşü ve k_i^t ise t . iterasyondaki i . karıncanın konumunu temsil etmektedir.

KAO algoritmasının işlem adımları Şekil 3.11'de verilmektedir.

| Algoritma – 1: KAO algoritması | |
|---------------------------------------|--|
| 1) | İlk karınca ve karınca aslanı popülasyonunu rasgele başlat |
| 2) | Karıncaların ve karınca aslanlarının uygunluğunu hesapla |
| 3) | En iyi karınca aslanlarını bul ve “elit” olarak kabul et (optimum olarak belirlenmiş) |
| 4) | Son kriter yerine getirilemediği zaman başla (while the end criterion is not achieved) |
| 4.1) | Her karınca için başla (for each ant) |
| 4.1.1) | Rulet çarkı kullanarak bir karınca aslanı seç |
| 4.1.2) | U_i^t ve L_i^t 'yi Denklem 3.8 ve Denklem 3.9'a göre güncelle |
| 4.1.3) | U ve L 'yi Denklem 3.10'a göre güncelle |
| 4.1.4) | Rasgele bir yürüyüş oluşturun. Denklem 3.1 ve 3.7'ye göre normalleştir |
| 4.1.5) | Karınca konumunu Denklem 3.15'e göre güncelle |
| | Her karınca için bitir |
| 4.2) | Tüm karıncaların zindeliğini hesapla |
| 4.3) | Bir karınca aslanı soluk hale gelirse, uygun bir karınca ile değiştir (Denklem 3.14) |
| 4.4) | Eğer bir karınca aslanı “elitten” daha zinde olursa, “eliti” güncelle |
| | Son kriter yerine getirildiği zaman bitir |
| 5) | “Elit” döndür |

Şekil 3.11 KAO algoritması işlem adımları

3.6. Karşılaştırma Metrikleri

Tez kapsamında önerilen algoritmanın başarısını ölçmek için farklı karşılaştırma metrikleri kullanılmıştır. Kullanılan metrikler görüntü segmentasyonu için literatürde en yaygın kullanılan metriklerdir.

Yapısal Benzerlik İndeksi (YBİ): YBİ, (SSIM-Structural Similarity Index), verilen iki görüntü arasındaki benzerliği ölçmek için kullanılan bir görüntü kalitesi değerlendirme yöntemidir. Verilen 2 görüntü arasındaki yapısal benzerliği bulmak için YBİ'nin sistematik hesaplaması Denklem 3.16'da verilmiştir. YBİ değeri [-1, 1] arasındadır. "1" değeri verilen 2 görüntünün çok benzer veya aynı olduğunu, "-1" değeri ise verilen 2 görüntünün tamamen farklı olduğunu gösterir (Sara ve ark., 2019).

$$YBİ(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3.16)$$

Burada μ_x , x 'in ortalaması; μ_y , y 'nin ortalaması; σ_x^2 , x 'in varyansı; σ_y^2 , y 'nin varyansı; $2\sigma_{xy}$ ise x ve y 'nin varyansıdır. c_1 ve c_2 ise 2 değişkendir ve denklemde sıfıra bölünmekten kaçınma adına kullanılan küçük değerlerdir (Denklem 3.17).

$$\begin{aligned} c_1 &= (k_1 L)^2 \\ c_2 &= (k_2 L)^2 \end{aligned} \quad (3.17)$$

Burada L , piksellerin alabileceği maksimum değer; $k_1 = 0.01$ ve $k_2 = 0.03$ ise sabit katsayılarıdır.

Tepe sinyal-gürültü oranı (TSGO): TSGO (PSNR-Peak Signal-Noise Ratio), bir görüntünün mümkün olan maksimum gücü ile temsilinin kalitesini etkileyen bozucu gürültünün gücü arasındaki orandır (Denklem 3.18). Yani; sinyal gücünün gürültü gücüne oranıdır. Görüntü sıkıştırma kalitesini karşılaştırmak için kullanılan metriklerden biridir. TSGO hesaplamasında Ortalama Kare Hata (OKH) (MSE-Mean Squared Error) kullanılmaktadır. OKH, makine öğrenmesinde sistem çıktısının gerçek değere ne kadar yakın olduğunu hesaplar (Denklem 3.19).

$$TSGO = 10 \log_{10} \left(\frac{255^2}{\sqrt{OKH}} \right) \quad (3.18)$$

$$OKH = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N} \quad (3.19)$$

Burada M, N görüntü boyutları ve (m, n) ise belirlenen koordinattaki piksel değeridir.

Özellik Benzerlik İndeksi (ÖBİ): ÖBİ (FSIM-Feature Similarity Index), orijinal görüntü ile segmentlenen resim arasındaki benzerliği değerlendiren metriktir. ÖBİ değeri $[-1, 1]$ aralığında değer almaktadır. Yüksek ÖBİ değeri, daha iyi segmentasyon yapıldığını göstermektedir. Segmentasyon için belirlenen eşik değeri sayısı arttıkça, ÖBİ değeri de artmaktadır. ÖBİ hesaplaması Denklem 3.20'ye göre yapılmaktadır (Zhang ve ark., 2011).

$$\text{ÖBİ} = \frac{\sum_{X \in \Omega} S_L(X) PC_m(X)}{\sum_{X \in \Omega} PC_m(X)} \quad (3.20)$$

Burada Ω , görüntü uzaysal alanını göstermektedir. PC , Faz uyumluluğu (Phase Compatibility), S_L (Similarity) ise benzerliktir ve Denklem 3.21, 3.22, 3.23'e göre hesaplanır.

$$S_L(x) = [S_{PC}(X)]^a \cdot [S_G(X)]^\beta \quad (3.21)$$

$$S_{pc} = \frac{2PC_1PC_2 + T_1}{PC_1^2 + PC_2^2 + T_1} \quad (3.22)$$

$$S_G(x) = \frac{2G_1(x) \cdot G_2(x) + T_2}{G_1^2(x) + G_2^2(x) + T_2} \quad (3.23)$$

Burada T_1 , S_{PC} 'nin stabilitesini artırmak için pozitif bir sabittir. T_1 'in belirlenmesi, PC değerlerinin dinamik aralığına bağlıdır ve iki pozitif gerçekte sayının benzerliğini ve $(0, 1]$ içindeki sonuç aralıklarını tanımlamak için yaygın olarak kullanılan bir ölçüdür. T_2 , gradyan büyüklük değerlerinin dinamik aralığına bağlı olan

pozitif bir sabittir. G_1 ve G_2 , karşılaştırılan 2 görüntüdür. α ve β , PC ve GM (Gradient Magnitude), gradyan büyüklüğü arasındaki özelliklerin göreceli önemini ayarlamak için kullanılan parametrelerdir ve genellikle $\alpha = \beta = 1$ olarak alınmaktadır.

Olasılık Rand İndeksi (ORİ): ORİ (PRI-Probabilistic Rand Index), farklı sayıda sınıfa sahip iki bölümü/segmenti karşılaştırma problemini ikili etiket ilişkilerini hesaplama problemine dönüştüren bir benzerlik fonksiyonudur. Gerçek görüntü ile segmentlenmiş görüntü arasındaki piksel çiftlerinin ağırlıklandırılması ile karşılaştırma yapmak için kullanılır (Denklem 3.24) (Unnikrishnan ve ark., 2007).

$$ORİ(G_{seg}, \{G_K\}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i < j}} [c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij})] \quad (3.24)$$

N pikselden oluşan $X = \{x_1, x_2, \dots, x_N\}$ görüntüsü için bir dizi gerçek görüntü $\{G_1, G_2, \dots, G_K\}$ alt alanlarına sahip olduğu düşünülürse, G_{seg} ise segmentlenmiş görüntüdür. p_{ij} , X görüntüsünün piksel çiftlerini ve c_{ij} ise segmentlenmiş görüntüdeki piksel çiftlerini tanımlamaktadır. Burada $\{G_K\}$ ile G_{seg} arasında benzersiz etiket sayısının aynı olması konusunda herhangi bir kısıtlama bulunmamaktadır.

Sınır Yer Değiştirme Hatası (SYDH): SYDH, (BDE-Boundary Displacement Error), bir sınır pikselin hatasını diğer sınır görüntüdeki piksel ile en yakın piksel arasındaki mesafe olarak tanımlayarak bölütlendirilmiş iki görüntü arasındaki sınır piksellerinin ortalama yer değiştirme hatasını ölçmektedir (Denklem 3.25).

$$\mu LA^{(u,v)} = \begin{cases} \frac{u-v}{L-1} & 0 < u-v \end{cases} \quad (3.25)$$

Burada L belirli bir bölgenin piksel sınırıdır. Sınır u , mevcut piksel sınır mesafesidir ve v yer değiştirme değeridir.

Global Tutarlılık Hatası (GTH): GTH (GCE-Global Consistency Error), bir segmentasyonun diğerinin ne kadar inceltmesi olarak görülebileceğini ölçer. Bu şekilde ilişkili olan segmentasyonlar, farklı ölçeklerde segmentlere ayrılmış aynı doğal görüntüyü temsil edebildikleri için tutarlı olarak kabul edilir. Değer ne kadar küçükse, segmentasyon o kadar iyidir (Denklem 3.26).

$$GTH = \frac{1}{n} \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\} \quad (3.26)$$

Burada, segmentasyon hatası ölçümü girdi olarak iki segmentasyon S_1 ve S_2 olur ve sıfırın hata olmadığını gösterdiği $[0, 1]$ aralığında gerçek değerli bir çıktı üretir. Belirli bir p_i pikseli için, o pikseli içeren S_1 ve S_2 'deki segmentleri göz önünde bulundurur.

Korelasyon Katsayısı (KK): KK (CC-Correlation Coefficient), iki değişken arasındaki ilişkinin ne kadar güçlü olduğunu ölçmek için kullanılır. KK, $[-1, 1]$ arasında bir değer döndürür (Denklem 3.27). Burada:

- Arti (+1) güçlü bir pozitif ilişkiyi gösterir.
- Eksi (-1) güçlü bir negatif ilişkiyi gösterir.
- Sıfır (0) sonucu hiçbir ilişki olmadığını gösterir.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (3.27)$$

Burada r , korelasyon katsayısı; x_i , bir örnekteki x değişkeninin değerlerini;, \bar{x} , x değişkeninin değerlerinin ortalamasını; y_i , bir örnekteki y değişkeninin değerlerini ve \bar{y} ise y değişkeninin değerlerinin ortalamasını göstermektedir.

Bu metrikler dışında, segmentasyon işlemlerinin sonuçlarını analiz etmek için, literatürde kümeleme yöntemi için çalışan algoritmalarda kullanılan, farklı performans ölçüm metrikleri de bulunmaktadır (Ozturk ve ark., 2015). Bu metrikler aynı zamanda kümeleme problemlerinde uygunluk fonksiyonlarının hesaplanması için de kullanılmaktadır. Değerlendirme metrikleri için kullanılan temel parametreler Çizelge 3.3'te verilmiştir.

Küme içi mesafe değeri (d_{max}): Kümeleme sonunda elde edilen tüm kümeler için, küme merkezlerine göre hesaplanan uzaklıkların ortalamasının maksimum değeridir (Denklem 3.28). İyi ayrılmış kümeler elde etmek için bu mesafe değerinin minimum olması istenir.

$$d_{max}(Z, x_i) = \max \left(\sum_{\forall Z_p \in C_{i,k}} \frac{d(Z_p, m_{i,k})}{n_{i,k}} \right) \quad (3.28)$$

Çizelge 3.3. Kümeleme yöntemi ile çalışan algoritmaların performans ölçümlerinde kullanılan parametreler

| Parametre | Açıklaması |
|-----------|---|
| P_A | Kullanılan metasezgisel algoritma |
| N | Desen sayısı |
| K | Toplam elde edilen küme sayısı. $C = \{C_1, C_2, \dots, C_K\}$ |
| Z | Kümelenecek görüntünün veri kümesi (piksel kümesi) $Z = \{z_1, z_2, \dots, z_p, \dots, z_N\}$ |
| x_i | i . Yiyecek kaynağı. $x_i = (m_{i,1}, m_{i,2}, \dots, m_{i,k}, \dots, m_{i,K})$ |
| z_p | D boyutlu desen |
| $C_{i,k}$ | i . Yiyecek kaynağının k . Küme |
| $m_{i,k}$ | i . Yiyecek kaynağının k . Küme merkezi |
| $n_{i,k}$ | $C_{i,k}$ kümesindeki toplam piksel sayısı |
| S_i | i . Küme içi dağılım |
| R_{ij} | i . Ve j . Kümeler arası dağılım |
| R_k | Maksimum R_{ij} değeri |
| SSW | Küme için kompaktlık değeri. Küme için toplam kareler (Sum Squares Within (SSW) cluster). |
| SSB | Kümelere arası ayrışım değeri. Kümelere arasındaki toplam kareler (Sum Squares Between (SSB) clusters). |
| M | Kümelenecek görüntüdeki veri kümesinin ortalama değeri |

Kümelere arası ortalama Öklid mesafesinin minimum değeri (d_{min}): Kümeleme sonunda elde edilen herhangi iki küme arasındaki ortalama Öklid mesafesinin minimum değeridir ve iyi segmente edilmiş kümelere elde etmek için bu mesafenin maksimum olması istenir (Denklem 3.29).

$$d_{min}(Z, x_i) = \min\{d(m_{i,j}, m_{i,k})\} \quad j \neq k \quad (3.29)$$

Kümeleme kalitesinin nicelendirme hatası (J_e): Kümeleme sürecinin kalitesini hesaplamak için tanımlanan nicelendirme hatasıdır (Denklem 3.30). İyi bir kümeleme için hatanın minimum olması istenir.

$$J_e = \frac{\sum_{k=1}^K \sum_{z_p \in C_k} d(z_p, m_k) / n_k}{K} \quad (3.30)$$

Davies–Bouldin İndeksi (DBI): *DBI*, küme içi dağılım toplamının, kümeler arası ayrışma oranıdır (Denklem 3.31, 3.32, 3.33). İyi bir kümeleme için bu oranın minimum olması istenir.

$$DBI = \frac{1}{K} \sum_{k=1}^K R_k, \quad (R_k = \max(R_{ij})) \quad (3.31)$$

$$S_i = \frac{1}{n_i} \sum_{x_j \in C_i} d(x_j, m_i)^2 \quad (3.32)$$

$$R_{ij} = \frac{S_i + S_j}{d(m_i, m_j)^2}, \quad i \neq j \quad (3.33)$$

XBI indeksi: Küme içi kompaktlık değerinin, kümeler arası ayrışım değerinin oranıdır. Kümeleme sonucunun kalitesini vermektedir. İndeks değeri ne kadar küçükse, kümeleme sonucu o kadar iyi olmaktadır (Denklem 3.34, 3.35, 3.36).

$$XBI = K * \frac{SSW}{SSB} \quad (3.34)$$

$$SSW = \sum_{k=1}^K \sum_{z_p \in C_k} d(z_p, m_k)^2 \quad (3.35)$$

$$SSB = \sum_{k=1}^K n_k * d(m_k, M)^2 \quad (3.36)$$

3.7. Veri Kümesi

Tez kapsamında, literatürde kullanılan farklı segmentasyon görüntüleri ve Berkeley Gri Seviye görüntü veri seti BSDS300 (Martin ve ark., 2001) kullanılmıştır. Kullanılan görüntüler literatürde kullanılan ve karşılaştırmalar için kullanılacak görüntülerdir. Tez kapsamında BSDS300 veri setinde bulunan 300 görüntünün tamamında geliştirilen algoritma test edilip, sonuçlar verilmiştir. Görüntüler 481x321 ya da 321x481 piksel boyutlarında gri seviye görüntülerdir. Ancak Berkeley veri setinden seçilmiş Şekil 3.12’de verilen görüntüler de ayrıca literatürle karşılaştırma için kullanılmıştır (görüntü isimleri Berkeley veri setinde ve literatürde kullanılan orijinal isimlerdir).



(a) 42049



(b) 48025



(c) 35008



(d) 35010



(e) 43051



(f) 176039



(g) 253036



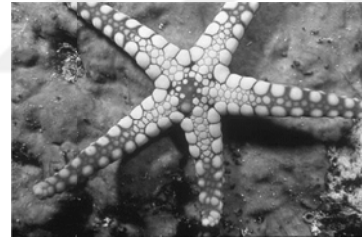
(h) 55067



(i) 3096



(j) 175043



(k) 12003



(l) 138078



(m) 253027



(n) 66075

Şekil 3.12 Berkeley veri setinden alınmış ve literatür karşılaştırması için kullanılan görüntüler

Tez kapsamında Berkeley haricinde kullanılan görüntüler Lena ve Uçak (jetplane) görüntüleridir. Görüntüler [0-255] arası piksel değerlerine sahip, 512x512 piksel boyutlarında gri seviye görüntülerdir (Şekil 3.13).



(a) Lena görüntüsü



(b) Uçak görüntüsü

Şekil 3.13 Gri seviye görüntüler



4. ÖNERİLEN YÖNTEM

Bölge büyütme, piksellerin bölgelere bölünmesi amacıyla yoğunluk, doku vb. görüntü özellikleri gibi homojenlik koşullarının dikkate alındığı piksel tabanlı bir görüntü segmentasyon yaklaşımıdır. Başlangıçta çok sayıda piksel, benzerlik kısıtlamalarına dayalı olarak tekrarlama prosedürü ile birleştirilir. Bir tohum pikseli seçilir ve daha sonra bölge boyutunun arttığı yerde benzer olan komşu pikseller toplanarak bölge büyütme işlemi gerçekleştirilir. Komşu piksellerin hiçbiri homojenlik kriterlerini karşılamadığında bölge büyütme durdurulur ve farklı bir yeni tohum pikseli seçilir. Bu işlem, görüntüdeki tüm pikseller farklı bölgelere ait oluncaya kadar devam eder. Bölge büyütmeyle dayalı segmentasyon tekniğinde, homojenlik kısıtlamalarına karar vermek için tohum noktalarının ve eşiklerin seçimi, segmentasyon doğruluğunu artırmada önemli bir rol oynar.

Bölge büyütme üç ana sorunla karşı karşıyadır:

- 1) Tohumların yeri (Farklı tohum noktaları farklı segmentasyon sonuçları verir).
- 2) Tohum sayısı.
- 3) Bölge büyütme stratejisi

Bölge büyütme ile ilgili bir diğer sorun da gürültü verilerinin bölgelere bölünmesi gereken görüntüler üzerindeki etkisidir. Bu sorun sadece bölge büyütme için değil, aynı zamanda sıklıkla başvurulan diğer segmentasyon teknikleri için de yaygın bir sorundur. Bu nedenle, tüm araştırmacılar bu sorunları çözmeye ve gürültülü görüntülerde iyi sonuçlar veren segmentleme algoritmalarını ortaya çıkarmaya çalışmaktadırlar. Bu nedenle, bu tez çalışmasında önerilen yaklaşım, belirtilen sorunları çözmeyi amaçlamaktadır. Çalışma, en iyi tohum noktaları belirlemek için bir karınca aslanı optimizasyon algoritmasına dayanan geliştirilmiş bir bölge büyütme yaklaşımı önermektedir. Bölge büyütme algoritması genel işlem adımları aşağıdaki gibidir:

Adım 1: R_1, R_2, \dots, R_n olarak adlandırılan alanlara dağılmış çeşitli tohum odakları (başlangıç noktası) ile başlanır. Her başlangıç noktasının yerleri p_1, p_2, \dots, p_n olarak belirlenir.

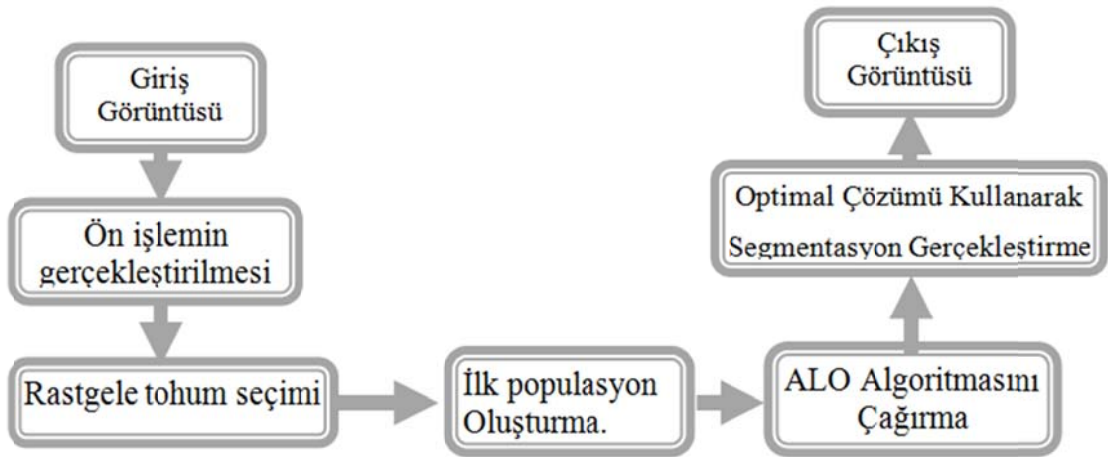
Adım 2: Temeldeki başlangıç noktası p_i ve komşu noktalarının piksel kestirimini hesaplamak için, eğer 2 nokta arasındaki fark, Bölge Büyütme (BB) segmentasyon işlemi sırasında belirlenebilen eşik değerden daha küçükse, komşu noktanın R_i olarak sınıflandırılabilceğini tanımlanır. Burada $I = 1, 2, \dots, n$ 'dir.

Adım 3: R_i alanının sınırları yeniden hesaplanır ve bu sınır noktalarının yeni tohum odakları p_i (ler) olarak ayarlanır. Ayrıca, R_i ortalama piksel değerlerinin sırasıyla yeniden hesaplanması gerekir.

Adım 4. Görüntüdeki tüm pikseller uygun bir bölgeye atanana kadar Adım 2 ve Adım 3 tekrarlanır.

Bölge Büyütme (BB) yönteminin avantaj ve dezavantajları da bulunmaktadır. Bölge büyütme yönteminin avantajları arasında, aynı özelliklere sahip bölgelerin ayrılması, iyi segmentasyon sonuçları ile net kenarlara sahip orijinal görüntülerin sağlanması, bölgeyi büyütme için sadece az sayıda tohum noktasına ihtiyaç duyulması, bölge büyütme için ölçütlerin belirlenmesi, aynı anda birden fazla ölçüt seçilebilmesi ve her pikseli sınırlı sayıda ziyaret ettiği için teorik olarak verimli olmaları yer alır. Öte yandan, bölge büyütme yönteminin dezavantajları arasında küresel bir sorun bakış açısına sahip olmayan yerel bir yöntem olması ve parazitlere duyarlı olması bulunmaktadır. Ayrıca görüntüye bir eşik fonksiyonu uygulanmadığı sürece, renkle ilgili görüntüdeki iki noktayı birleştiren sürekli bir nokta yolu vardır ve pratik rastgele bellek erişimi algoritmayı yavaşlattığından adaptasyon gereklidir. Bu amaçla tez kapsamında Karınca Aslanı Optimizasyon (KAO) algoritmasına dayalı bir BB yöntemi önerilmiştir.

Tez kapsamında önerilen yöntemle ait akış diyagramı Şekil 4.1’de verilmiştir.

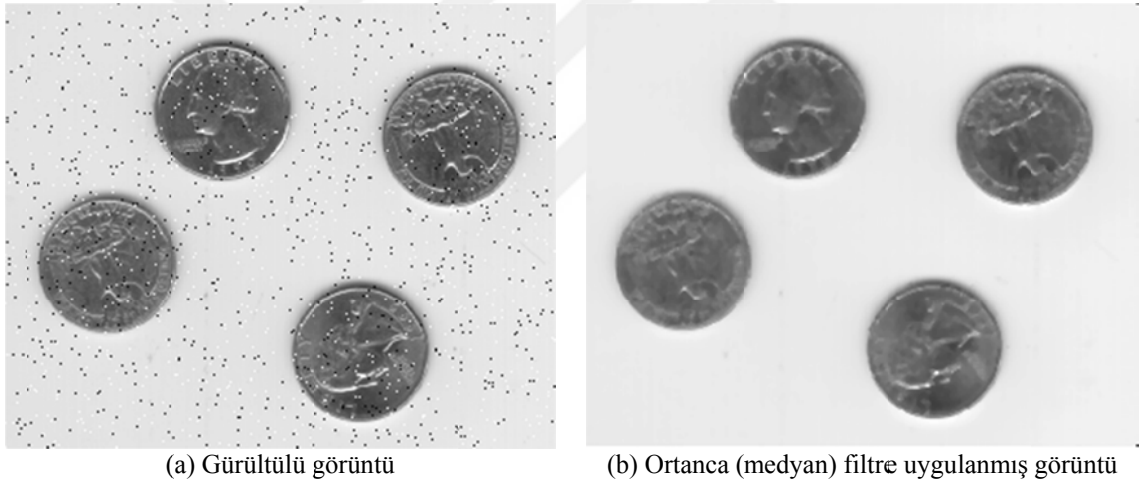


Şekil 4.1. Önerilen yöntemin akış şeması

Bölge büyütmeyle dayalı segmentasyon stratejisi, aynı bölgedeki piksellerin aynı ortak özellikleri paylaşmasını beklemektedir. Bölge büyütme, tüm görüntü piksellerini kendi içinde başka bir bölümün elde edilemediği ve görüntüde farklı alanları belirten ayrı ayrı bölümlere ayırdığımız birleştirmeli sıralama algoritmasına benzemektedir. Tez kapsamında geliştirilen uygulamalar Matlab ortamında kodlanmıştır. Şekil 4.1’de gösterildiği gibi, önerilen yöntemin işlem adımları sırasıyla aşağıda verilmiştir.

1) Giriş görüntülerini alma: Uygulamanın gerçekleştirilen ilk adımudur. Bu adımda, işlenecek görüntüler sisteme yüklenmektedir.

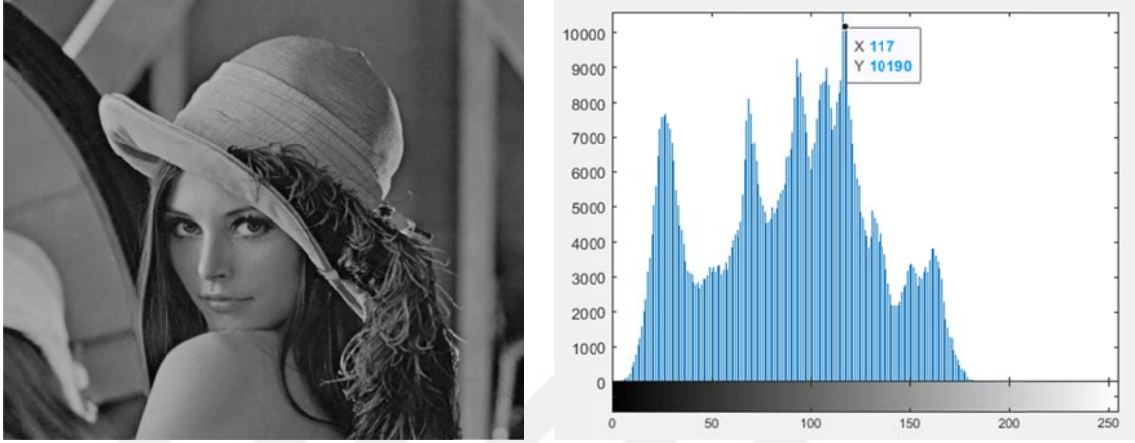
2) Önişlem: Segmentasyondan önce giriş görüntülerine bazı ön işlem adımları uygulanmıştır. Önişleme amacı; istenmeyen gürültüleri önlemek, görüntü verilerinin kalitesini artırmak veya sonraki işlemler için önemli olan bazı görüntü özelliklerini iyileştirmektir. Bu çalışmada ön işleme filtresi olarak ortanca (medyan) filtresi kullanılmıştır (Şekil 4.2).



Şekil 4.2. Ortanca filtre uygulanma örneği

3) Rastgele başlangıç tohum noktası seçimi: Başlangıç noktalarının seçiminde kullanılan küresel bir strateji yoktur ve probleme bağlıdır. Bununla birlikte, başlangıç tohumları seçilirken dikkate alınan iki önemli strateji vardır. Birincisi, kıvılcık görüntüler kullanılarak hedeflerin tanınması gerekiyorsa, görüntünün/görüntülerin en parlak piksel(ler)i seçilir. İkinci olarak, önceden herhangi bir bilginin olmadığı durumlarda, histogram hesaplanır ve en güçlü tepe noktalarına ilişkin gri seviye değerleri, başlangıç tohum noktaları olarak seçilir. Bu çalışmada, başlangıç tohum noktasını seçmek için histogram temelli yaklaşım kullanılmıştır. Bu amaçla öncelikle, görüntülerin histogramı belirlenmiş ve en yüksek tepe değerlerine karşılık gelen gri

seviye deęerleri ilk tohum noktaları olarak kabul edilmiřtir (řekil 4.3). Ka adet eřik deęeri kullanılacaksa, en yksek deęere sahip piksel deęerinden itibaren sıralanarak, istenilen sayıda deęer, ilk tohum noktası olarak alınmıřtır. řekil 4.3'te X deęeri gri seviye deęeri olarak 117 olarak alınmıř bir rnek verilmektedir.



řekil 4.3. Histogramdan bařlangı pikseli (seed pixel) seme rneęi

Tez alıřmasında rastgele ilk tohum noktalarının seiminden sonra, tohumlar ve komřu pikseller arasındaki mesafe llerek segmentler oluřturulmuřtur. Grntdeki piksel deęerleri min-max fonksiyonuna gre $[0,1]$ arasına normalize edilerek kullanılmıřtır (Denklem 4.1).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

Burada, x_{norm} grntdeki pikselin normalize edilmiř deęeri; x pikselin orijinal gri seviye deęeri; x_{min} grntdeki en kk gri seviye deęeri ve x_{max} ise grntdeki en byk gri seviye deęerini gstermektedir.

Tez alıřmasında, bařlangı tohum noktası ile komřu pikseller arasındaki mesafeyi lmek iin klid uzaklık forml kullanılmıřtır (Denklem 4.2).

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4.2)$$

Burada x ve y olarak iki nokta arasındaki Öklid uzaklığı, noktaların karşılıklı gelen koordinatları arasındaki değerlerin farkların kareleri toplamının karekökü olarak tanımlanır.

4) Başlangıç popülasyonu oluşturma: Popülasyon oluşturma, Karınca Aslanı Optimizasyon (KAO) algoritmasının ilk adımınıdır. İlk karınca ve karınca aslanı popülasyonunun rastgele başlatılması ile gerçekleştirilir. Tez kapsamında önerilen modifiye KAO'nun (mKAO) ilk adımı olarak (n) karıncaların ve karınca aslanının (m) ilk popülasyonu, boyutsal arama uzayında (S) rastgele oluşturulur (Mirjalili, 2015) (Denklem 4.3). Karınca ve karınca aslanlarının pozisyonları Denklem 4.4 ve 4.5'te verilen matrislere kaydedilir.

$$X_{i,s} = L_s + rand(0,1)(U_s - L_s) \quad (4.3)$$

$$M_{\text{karınca}} = \begin{bmatrix} k_{1,1} & k_{1,2} & \dots & \dots & \dots & k_{1,s} \\ k_{2,1} & k_{2,2} & \dots & \dots & \dots & k_{2,s} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot \\ k_{n,1} & k_{n,2} & \dots & \dots & \dots & k_{n,s} \end{bmatrix} \quad (4.4)$$

$$M_{\text{karıncaAslanı}} = \begin{bmatrix} KA_{1,1} & KA_{1,2} & \dots & \dots & \dots & KA_{1,s} \\ KA_{2,1} & KA_{2,2} & \dots & \dots & \dots & KA_{2,s} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot \\ KA_{m,1} & KA_{m,2} & \dots & \dots & \dots & KA_{m,s} \end{bmatrix} \quad (4.5)$$

Burada L_s ve U_s , s boyutlu arama uzayının alt ve üst sınırlarıdır. Tez kapsamında görüntülerdeki piksel değerleri normalize edildiği için, alt sınır 0 ve üst sınır değeri olarak 1 alınmıştır.

5) Uygunluk değerlerinin hesaplanması: Popülasyon oluşturulduktan sonra uygunluk değerleri Denklem 4.6-4.7'ye göre hesaplanarak, Denklem 4.8 ve Denklem 4.9'da verilen matrislere kaydedilir.

$$F(PA_i, S) = J_e \frac{d_{\max}(S, PA_i)}{d_{\min}(S, PA_i)} (d_{\max}(S, PA_i) + z_{\max} - d_{\min}(S, PA_i) + OKH_{\text{küme}}) \quad (4.6)$$

$$OKH_{k\ddot{u}me} = \frac{1}{N} \sum_{k=1}^K \sum_{\forall z_p \in C_k} d(z_p, m_k)^2 \quad (4.7)$$

$$M_{karıncaZindelik} = \begin{bmatrix} f([k_{1,1} & k_{1,2} & \dots & \dots & \dots & k_{1,d}]) \\ f([k_{2,1} & k_{2,2} & \dots & \dots & \dots & k_{2,d}]) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & & & \vdots & & \vdots \\ f([k_{n,1} & k_{n,2} & \dots & \dots & \dots & k_{n,d}]) \end{bmatrix} \quad (4.8)$$

$$M_{karıncaAslanıZindelik} = \begin{bmatrix} f([KA_{1,1} & KA_{1,2} & \dots & \dots & \dots & KA_{1,d}]) \\ f([KA_{2,1} & KA_{2,2} & \dots & \dots & \dots & KA_{2,d}]) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & & & \vdots & & \vdots \\ f([KA_{n,1} & KA_{n,2} & \dots & \dots & \dots & KA_{n,d}]) \end{bmatrix} \quad (4.9)$$

Tez kapsamında kullanılan uygunluk fonksiyonu kümeleme problemlerinde kullanılan ve literatürdeki en başarılı fonksiyondur (Ozturk ve ark., 2015). Uygunluk fonksiyonu, ortalama karesel hatayı ($OKH_{k\ddot{u}me}$) ve niceleme (quantization) hatasını verimli bir şekilde ele almaktadır. Ayrıca, d_{max} maksimize ederken ve d_{min} kriterlerini minimize edip, hata oranını minimum oranlarda tutmayı amaçlamaktadır.

6) Karınca arama uzayının içinde tutulması: Her arama alanının belirli bir sınırı vardır ve bu nedenle karınca arama alanının rastgele yürüyüşlerini arama alanı içinde tutmak gerekmektedir. Karınca arama alanını belirli bir sınırdan tutmak için Denklem 4.10'daki min-maks normalleştirme fonksiyonu kullanılmıştır (Mirjalili, 2015).

$$X_i^t = \frac{(X_i^t - a_i) \times (u_i^t - L_i^t)}{(b_i - a_i)} + L_i \quad (4.10)$$

Burada, b_i ve a_i , X_i^t dizisindeki maksimum ve minimum değerlerdir. u_i^t ve L_i^t , t . iterasyonda i . boyutun üst ve alt sınırıdır. Alt ve üst sınırları hesaplaması, karınca aslanının konumuna bağlı olarak Denklem 4.11'e göre yapılır.

7) Arama uzayında karınca aslan konumunu bulma: Karınca aslanlarının mevcut konumu orijinal KAO algoritmasında Denklem 3.11 ve 3.12 kullanılarak belirlenmektedir (Mirjalili, 2015). Orijinal KAO'nun sınırlaması, algoritmanın kullandığı kademeli olarak azalan strateji, iterasyon sayısı arttıkça karınca arama alanının rastgele yürüyüşünün sınırının azalmasına neden olur. Denklem 3.11'de gösterildiği gibi, bazı

noktalarda iterasyonun değeri her arttığında, algoritma aday çözümler etrafında rastgele dolaşmanın sınır daralmasıyla karşı karşıya kalır. Çözümlerin etrafındaki sınırı azaltma stratejisi, aslında rastgele yürüme yeteneğinin ve performansının düşmesine neden olur, bu da karınca aslanının avlanma verimliliğini ve genel algoritmayı etkiler.

Bu problemi çözmek için, yapılan tez çalışmasında esnek bir yürüme limiti (sınır) yöntemi önerilmektedir. Önerilen esnek strateji, karınca yürüyüşü ve sınır değişikliği arasındaki etkileşimi geliştirir. Önerilen objektif işlevi, karıncaların rastgele yürüyüş marjı azaldığında sınırı arttırır. Bu nedenle, kötü karınca aslanı, yerel arama kapasitesini ve hızlı yerel optimumları iyileştirmek için tuzakları azaltacaktır. Diğer tarafta iyi karınca aslanı keşif kapasitesini geliştirmek için tuzakları artıracaktır. Önerilen yöntemin hesaplanması Denklem 4.11, 4.12 ve 4.13'deki gibidir.

$$\begin{cases} U_i^t = KA_i^t + \frac{U_i}{p} \\ L_i^t = KA_i^t + \frac{L_i}{p} \end{cases} \quad (4.11)$$

$$p = \partial(v) * 10^w * \frac{t}{T} \quad (4.12)$$

$$\partial(v) = x + \frac{1}{1 + e^{\left(\frac{2KA - KA_{pop}}{KA}\right)}} \quad (4.13)$$

Burada KA_i^t , karınca aslanının konumudur. U_i ve L_i , i . boyutlu çözüm uzayının üst ve alt sınırlarıdır. p ise sınırın oranıdır. Burada t mevcut iterasyon, T ise maksimum iterasyon sayısıdır. Burada KA , uygunluğu kullanılarak elde edilen mevcut karınca aslanı, KA_{pop} toplam karınca aslan popülasyonudur. x sabit değerdir ve tezde x değeri 0.5'tir. w , sömürü seviyesini ayarlamak için kullanılan ağırlık parametresidir (ağırlık) ve Denklem 4.14'e göre hesaplanmaktadır. w 'nin değeri [2-6] arasındadır (Toz, 2019) ve tez kapsamında bu değerler kullanılmıştır.

Eğer $t > 0.01T$; $w = 2$

Eğer $t > 0.5T$; $w = 3$

Eğer $t > 0.75T$; $w = 4$

Eğer $t > 0.9T$; $w = 5$

Eğer $t > 0.95T$; $w = 6$

(4.14)

Denklem 4.13’de $KA = 1$ ve $\partial(v) \approx 1.5$ değeri olduğunda p sınır oranı artacak, aynı zamanda karınca aslanının yürümesi azalacaktır. Bu noktada, kötü karınca aslanı, yerel aramayı iyileştirmek için tuzaklarını azaltacaktır. Ayrıca $KA = KA_{Pop}$ ve $\partial(v) \approx 0.5$ değeri olduğunda; p sınır oranı azalacak, aynı zamanda karınca aslanının yürüyüşü artacaktır. Bu nedenle iyi karınca aslanı, avlanma fırsatlarını iyileştirmek ve yerel optimuma düşmekten kaçınmak için tuzakları artıracaktır.

8) Karınca aslanı konum güncellemesi: Konum güncellemesi etkinliğini dengelemek için orijinal KAO (Mirjalili, 2015), karınca aslanını elit karınca aslanına yaklaştırarak, karınca aslanının konumunu değiştirmek için Denklem 3.15’i kullanır. Burada, algoritmanın ilerleme kapasitesini iyileştirmiş olsa da, KAO’nun vurgunun altında yatan aşamada daha fazla temelli keşif kapasitesine ihtiyaç duyduğunu ve döngünün sonraki aşamasında daha temelli iyileştirme yeteneği gerektirdiğini tamamen dikkate almaz. Aynı şekilde, sabit noktalar açısından algoritmanın düzensiz avlanma kapasitelerini sınırlar. Bunun için Denklem 4.15 önerilmiştir.

9) Elitizm: Seçkin (Elite) prosedürü kullanarak, optimal karınca aslanı, arama alanındaki tüm karıncaların rastgele yürüyüşünü etkileyecektir. Konum güncelleme etkinliğini ve karıncanın rastgele yürüyüşünü dengelemek için Denklem 3.15 yerine Denklem 4.15 önerilmiştir. Denklem 4.15, karınca aslanının Denklem 3.15 denklemi ile karşılaştırıldığında daha verimli çalışmaktadır.

$$k_j^t = R_K^t \left(1 - \text{Rand}^{(1-\frac{t}{T})}\right) + R_E^t \left(1 - \text{Rand}^{(1-\frac{t}{T})^2}\right) \quad (4.15)$$

Burada R_K^t , fitness rulet tekerleği tarafından t . iterasyonunda seçilen karınca aslanının etrafındaki rastgele yürüyüştür. R_E^t elit karınca aslanının t . iterasyonda rastgele yürüyüşüdür. Güncellenen karınca, seçilen karınca aslanından daha fit hale gelirse, o zaman, karınca aslanı konumunu Denklem 4.16’ya göre en son karıncaya göre günceller (Mirjalili, 2015).

$$KA_j^t = k_i^t; \quad \text{eğer } f(k_i^t) > f(KA_j^t) \quad (4.16)$$

Tez kapsamında önerilen yöntemle ait işlem adımları Şekil 4.3’te verilmiştir.

Algoritma – 2: mKAO algoritması

- 1) $[r, c] \leftarrow$ Giriş resmini yükle
- 2) $X \leftarrow$ boyutlandırılmış gri tonlamalı test görüntüsü
- 3) $[u, l] \leftarrow X$ verisinin üst ve alt sınırı
- 4) $Q_{AL} \leftarrow X$ 'in u ve p arasında rastgele oluşturulmuş karınca popülasyonu. (4,2)
- 5) $Q_A \leftarrow X$ 'in u ve p arasında rastgele oluşturulmuş karınca aslanı popülasyonu(4,3)
- 6) Karınca aslanının uygunluk değerini hesaplayın
 - 6.1) Q_{AL} 'in her karınca aslanı için başla
 - 6.1.1) En yakın karınca aslanı kümesine X_p ekleyin
 - 6.1.2) Denklemi (3,4) kullanarak karınca aslanının uygunluğunu hesaplayın
 Her karınca aslanı için bitir
 - 6.2) Son kriter karşılanmamışken Q_A 'nın her karıncası için başla
 - 6.2.1) Denklem'de önerilen u ve l 'yi kullanarak karıncaların rastgele yürüyüş sınırı oluşturun. Denklemi (4.13) ve (4.14)
 - 6.2.2) Rastgele yürüyüşe göre karıncaların pozisyonunu güncelle
 Her karınca için bitir
 - 6.3) Karıncaların uygunluğunu hesaplayın
 - 6.4) Q_A 'nın her karıncası için başla
 - 6.4.1) En yakın karınca aslanı kümesine X_p ekleyin
 - 6.4.2) Denklem (3.4) kullanarak karınca aslanının uygunluğunu hesaplayın
 Her karınca için bitir
 - 6.5) Karınca aslanlarının pozisyonlarını uygunluklarına göre güncelleyin.
 - 6.6) popülasyondaki Elitist karınca aslanını güncelle
 Son kriter yerine getirilemediği zaman bitir
 “Elit” döndür

Şekil 4.3. mKAO algoritması işlem adımları

5. DENEYSSEL SONUÇLAR

Tez kapsamında önerilen yöntemin performansı, hem BSDS300 hem de literatürde kullanılan gri seviye görüntüler üzerinde test edilmiştir. Tez kapsamındaki uygulamalar MATLAB kullanılarak geliştirilmiştir. MATLAB, görüntü analistlerinin analiz, gösteri ve geliştirme gibi görüntü ile ilgili bazı aktiviteleri gerçekleştirmesini ve görüntü işleme için standart algoritmalar ve görüntü işleme uygulamalarının kapsamlı bir düzenlemesini sağlayan bir yazılım ortamıdır. Yapılan tüm uygulamalar, Intel (R) Core™ i3-2370m CPU @ 2.40GHZ işlemci ve 4 GB RAM'e sahip bir bilgisayarda gerçekleştirilmiştir.

Tez kapsamında önerilen yöntemin başarısını değerlendirmek için BSDS300 veri setindeki görüntüler kullanılarak, literatürdeki temel segmentasyon yöntemlerinden K-Ortalamlar ve temel KAO ile karşılaştırılmıştır. Karşılaştırma işlemlerinde, algoritmalarda kullanılan parametre değerleri (Ozturk ve ark., 2015) çalışması referans alınarak belirlenmiş ve Çizelge 5.1'de verilmiştir.

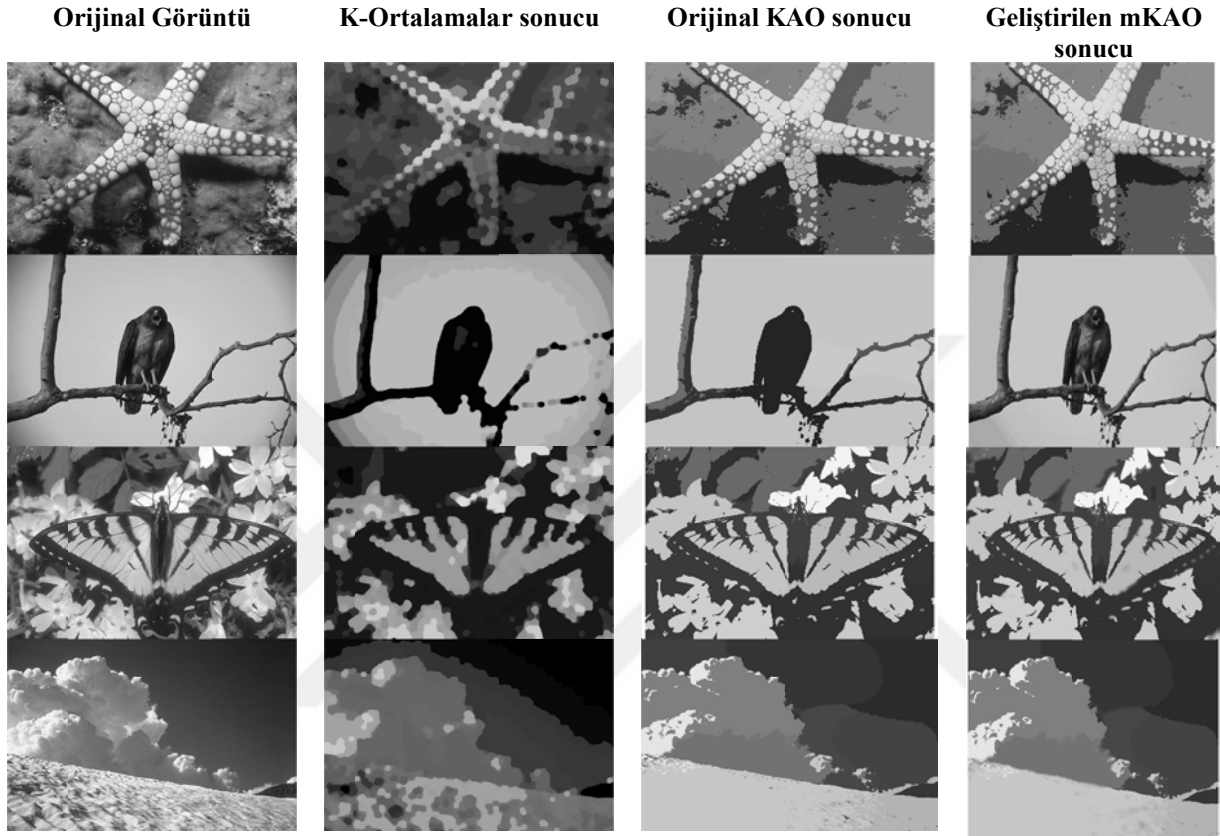
Çizelge 5.1. BSDS300 ile yapılan işlemlerde algoritmalara ait parametre değerleri

| | Orijinal KAO | Geliştirilen Yöntem (mKAO) | K-Ortalamlar |
|---------------------------|--------------|----------------------------|--------------|
| Karınca sayısı | 30 | 30 | X |
| Karınca aslanı sayısı | 30 | 30 | X |
| Eşik değeri sayısı | 2-3-4-5 | 2-3-4-5 | X |
| Max İterasyon sayısı | 100 | 100 | X |
| Çadır haritası (Tent Map) | 0.01 | 0.01 | X |
| Tekrarlama sayısı | 30 | 30 | X |
| K küme sayısı | X | X | 5 |

Çizelge 5.2. Geliştirilen yöntemin BSDS300 veri seti üzerindeki sonuçları

| Veri seti | Karşılaştırma ölçüsü | | Yöntem | | |
|--|----------------------|----------------|--------------------|--------------|----------------------------|
| | | | K-Ortalamlar (K=5) | Orijinal KAO | Geliştirilen yöntem (mKAO) |
| Berkeley BSDS300 veri kümesindeki 300 görüntü için | J_e | Ortalama | 9.4597 | 9.1696 | 8.9961 |
| | | Standart Sapma | 1.2079 | 0.3025 | 0.18744 |
| | d_{max} | Ortalama | 12.927 | 10.9049 | 10.4667 |
| | | Standart Sapma | 0.5436 | 0.6494 | 0.1879 |
| | d_{min} | Ortalama | 29.758 | 41.2025 | 41.813 |
| | | Standart Sapma | 1.783 | 2.6938 | 0.3873 |
| | DBI | Ortalama | 0.2018 | 0.1585 | 0.1451 |
| | | Standart Sapma | 0.03581 | 0.0233 | 0.0055 |
| | XBI | Ortalama | 0.197467 | 3.46E-5 | 1.93E-5 |
| | | Standart Sapma | 1.96E-3 | 2.6E-6 | 3.61E-7 |

Geliştirilen yöntemin BSDS300 veri setinden alınan görüntüler üzerinde elde edilen sonuçlar Çizelge 5.2’de ve örnek sonuç görüntüleri de Şekil 5.1’de verilmiştir. Segmentasyon için BSDS300 veri setindeki 300 görüntü kullanılmış ve ortalama değerler alınmıştır.



Şekil 5.1. BSDS300 veri seti için örnek segmentasyon sonuçları

KAO, başlangıçta sürekli optimizasyon problemlerini çözmek için tasarlanmış, biyo-esinlenmiş bir algoritma olmasına rağmen, temel olarak ayrık optimizasyon problemlerinin çözümü için KAO’yu kullanan bazı çalışmalar bulunmaktadır (Yogarajan ve Revathi, 2016; Barma ve ark., 2019; Mukherjee ve ark., 2020). Görüntü segmentasyonunda ise Iyer ve ark. (2019), eşik değeri bulma problemini çözmek için KAO ve Balina Optimizasyon Algoritmalarını kullanmış ve eşik değerlerini ayrık olarak sunmuşlardır. Bu nedenle, literatürdeki algoritmalar temel alınarak, tez çalışmasında en yakın tamsayıya yuvarlama yaklaşımı (Rounding approach) kullanılarak eşik değeri bulma işlemi belirlenerek segmentasyon yapılmıştır.

Geliştirilen yöntem, BSDS300 veri setindeki 300 görüntüde, farklı eşik değer sayıları kullanılarak da test edilmiştir. Bu amaçla, görüntü segmentasyonunda literatürde kullanılan eşik değerleri sayıları olan 2-3-4-5 eşik değerleri kullanılarak

işlem yapılmıştır (Akay, 2013). Elde edilen sonuçlar farklı karşılaştırma değerleri ile test edilmiş ve sonuçlar Çizelge 5.3'te verilmiştir. Eşik değeri belirlenerek yapılan segmentasyon sonuçları, literatürde eşik değeri belirlemek amacıyla kullanılan Otsu ve Kapur entropi yöntemleri ile karşılaştırmalı olarak verilmiştir.

Çizelge 5.3 BSDS300 için farklı eşik değer sayılarında ortalama sonuçlar

| | Algoritma | Eşik değeri sayısı | | | |
|---|-----------|--------------------|----------------|----------------|----------------|
| | | 2 | 3 | 4 | 5 |
| Yapısal Benzerlik İndeksi (YBİ) (SSIM-Structural Similarity Index Measure) | Otsu | 0.81132 | 0.86087 | 0.85356 | 0.89258 |
| | Kapur | 0.91586 | 0.94791 | 0.91181 | 0.96446 |
| | KAO | 0.88025 | 0.93023 | 0.85453 | 0.90251 |
| | mKAO | 0.94537 | 0.92926 | 0.91389 | 0.97702 |
| Özellik Benzerlik İndeksi (ÖBİ) (FSIM-Feature Similarity Index) | Otsu | 0.74914 | 0.80399 | 0.82551 | 0.81953 |
| | Kapur | 0.75739 | 0.78944 | 0.76424 | 0.81004 |
| | KAO | 0.88631 | 0.85301 | 0.87801 | 0.85401 |
| | mKAO | 0.89161 | 0.85726 | 0.86831 | 0.90216 |
| Tepe Sinyal-Gürültü Oranı (TSGO) (PSNR-Peak signal-to-noise ratio) | Otsu | 21.5458 | 19.454 | 24.7125 | 24.7825 |
| | Kapur | 24.1762 | 23.9176 | 24.0421 | 24.6863 |
| | KAO | 26.0185 | 25.0545 | 24.05457 | 26.1177 |
| | mKAO | 25.3012 | 23.9641 | 26.7957 | 27.0281 |
| Korelasyon Katsayısı (KK) (CC-Correlation Coefficient) | Otsu | 0.8495 | 0.6882 | 0.7472 | 0.8312 |
| | Kapur | 0.8595 | 0.8780 | 0.85807 | 0.7182 |
| | KAO | 0.8561 | 0.8608 | 0.8457 | 0.8146 |
| | mKAO | 0.8916 | 0.8812 | 0.8087 | 0.8717 |
| Olasılık Rand İndeksi (ORİ) (PRI-Probabilistic Rand Index) | Otsu | 0.6304 | 0.6546 | 0.7065 | 0.7184 |
| | Kapur | 0.6937 | 0.6925 | 0.6710 | 0.6387 |
| | KAO | 0.7110 | 0.7188 | 0.6888 | 0.7488 |
| | mKAO | 0.7060 | 0.7238 | 0.7262 | 0.7311 |
| Sınır Yer Değiştirme Hatası (SYDH) (BDE-Boundary Displacement Error) | Otsu | 8.9724 | 9.0403 | 9.2137 | 9.112 |
| | Kapur | 9.1010 | 9.1142 | 9.2301 | 9.3076 |
| | KAO | 7.8631 | 8.0018 | 7.6040 | 7.5022 |
| | mKAO | 7.0434 | 7.7201 | 7.6012 | 7.5214 |
| Global Tutarlılık Hatası (GTH) (GCE-Global Consistency Error) | Otsu | 0.3539 | 0.3621 | 0.4003 | 0.3106 |
| | Kapur | 0.3031 | 0.3086 | 0.3152 | 0.3124 |
| | KAO | 0.2139 | 0.2029 | 0.2110 | 0.2012 |
| | mKAO | 0.1205 | 0.1059 | 0.1353 | 0.1074 |

Tez kapsamında sunulan mKAO algoritması sonuçları literatürle de karşılaştırılmıştır. Bu amaçla literatürdeki K-Ortalamlar (K-Means), Parçacık Sürü Optimizasyonu (PSO) (Particle Swarm Optimization-PSO), Genetik Algoritmalar (GA) (Genetic Algorithm-GA), Yapay Arı Kolonisi (YAK) (Artificial Bee Colony-ABC) ve Geliştirilmiş Karınca Aslanı Optimizasyon (GKAO) (Improved Ant Lion Optimization-IALO) algoritmaları kullanılarak yapılan literatürdeki çalışmalarla karşılaştırmalar

yapılmıştır. Bu algoritmalar için kullanılan parametre değerleri (Ozturk ve ark., 2015; Toz, 2019) çalışması referans alınarak belirlenmiş ve parametre değerleri Çizelge 5.4'te verilmiştir.

Çizelge 5.4. Segmentasyon sonuçlarının karşılaştırılmasında kullanılan literatürdeki farklı algoritmaların parametre değerleri

| Algoritma | Parametre | Değerler |
|---------------|--------------------------------------|--------------------|
| K-Ortalamalar | K küme sayısı | 5 |
| | Fonksiyon değerlendirme sayısı (FEs) | 7500 |
| PSO | İterasyon sayısı | 250 |
| | Parçacık sayısı | 30 |
| | Fonksiyon değerlendirme sayısı (FEs) | 7500 |
| | Tekrarlama sayısı | 30 |
| | c_1, c_2 | $c_1 = c_2 = 1.49$ |
| | Atalet ağırlığı (inertia weight w) | [0.2- 0.9] |
| | Hız (velocity V) | 255 |
| GA | İterasyon sayısı | 250 |
| | Kromozom sayısı | 30 |
| | Tekrarlama sayısı | 30 |
| | Fonksiyon değerlendirme sayısı (FEs) | 7500 |
| | Çaprazlama oranı | 0.8 |
| | Mutasyon oranı | 0.2 |
| YAK | İterasyon sayısı | 250 |
| | Koloni boyutu | 30 |
| | Tekrarlama sayısı | 30 |
| | Fonksiyon değerlendirme sayısı (FEs) | 7500 |
| GKAO | İterasyon sayısı | 250 |
| | Karınca / Karınca aslanı sayısı | 30 / 30 |
| | Tekrarlama sayısı | 30 |
| | Çadır haritası (Tent map) | 0.001 |
| | Fonksiyon değerlendirme sayısı (FEs) | 7500 |
| Önerilen mKAO | İterasyon sayısı | 250 |
| | Karınca / Karınca aslanı sayısı | 30 / 30 |
| | Tekrarlama sayısı | 30 |
| | Çadır haritası (Tent map) | 0.001 |
| | Fonksiyon değerlendirme sayısı (FEs) | 7500 |

Literatürdeki farklı algoritmalarla yapılan çalışmalara ait sonuçlar Çizelge 5.5'te verilmiştir.

Çizelge 5.5. Gri seviye görüntü segmentasyonu için önerilen yöntemin literatürdeki farklı algoritmalarla karşılaştırılması (Ozturk ve ark., 2015; Toz, 2019)

| Metrik | Yöntem | Görüntü ismi | | | | | | | |
|-----------|---------------|----------------|---------|----------------|---------|----------------|---------|----------------|---------|
| | | Lena | | Uçak | | 48025 | | 42049 | |
| | | Ortalama | SS | Ortalama | SS | Ortalama | SS | Ortalama | SS |
| J_e | K-Ortalamlar | 9.430 | 0.036 | 9.854 | 0.121 | 10.526 | 0.021 | 8.989 | 0.244 |
| | PSO | 9.793 | 0.198 | 9.723 | 0.169 | 10.872 | 0.098 | 8.744 | 0.139 |
| | GA | 10.211 | 0.489 | 10.083 | 0.502 | 11.277 | 0.701 | 9.158 | 0.718 |
| | YAK | 9.736 | 0.108 | 9.736 | 0.108 | 10.895 | 0.103 | 8.699 | 0.053 |
| | GKAO | 8.5974 | 0.0315 | 9.737 | 0.0321 | 10.7139 | 0.089 | 8.6817 | 0.022 |
| | Önerilen mKAO | 8.476 | 0.0445 | 7.8973 | 0.4604 | 8.9968 | 0.0466 | 8.0854 | 0.0401 |
| d_{max} | K-Ortalamlar | 11.712 | 0.261 | 15.434 | 0.577 | 12.512 | 0.232 | 12.759 | 1.470 |
| | PSO | 10.691 | 0.224 | 10.835 | 0.361 | 12.079 | 0.272 | 10.026 | 0.254 |
| | GA | 11.046 | 0.583 | 11.435 | 0.659 | 13.082 | 1.274 | 10.521 | 0.990 |
| | YAK | 10.676 | 0.133 | 10.734 | 0.322 | 12.276 | 0.302 | 9.900 | 0.158 |
| | GKAO | 8.8565 | 0.0476 | 10.7497 | 0.1618 | 11.6726 | 0.2061 | 9.938 | 0.098 |
| | Önerilen mKAO | 8.598 | 0.0565 | 8.6306 | 0.0573 | 8.3347 | 0.0577 | 8.0453 | 0.0595 |
| d_{min} | K-Ortalamlar | 34.833 | 0.461 | 21.167 | 3.573 | 31.350 | 0.249 | 19.838 | 7.271 |
| | PSO | 40.564 | 0.503 | 41.329 | 1.202 | 41.917 | 2.750 | 38.930 | 0.863 |
| | GA | 40.223 | 1.945 | 39.697 | 2.302 | 45.798 | 4.553 | 37.543 | 2.878 |
| | YAK | 39.955 | 0.967 | 40.531 | 1.234 | 44.716 | 1.914 | 38.024 | 0.766 |
| | GKAO | 34.8374 | 0.3347 | 40.4416 | 0.6103 | 39.6405 | 1.6537 | 38.5154 | 0.3026 |
| | Önerilen mKAO | 41.5662 | 0.0521 | 41.654 | 0.0551 | 41.9101 | 0.0522 | 41.0052 | 0.0529 |
| DBI | K-Ortalamlar | 0.154 | 0.001 | 0.233 | 0.001 | 0.1683 | 0.0012 | 0.2088 | 0.2088 |
| | PSO | 0.1554 | 0.0045 | 0.1571 | 0.0012 | 0.1552 | 0.0052 | 0.1558 | 0.0022 |
| | GA | 0.1629 | 0.0073 | 0.166 | 0.0126 | 0.1594 | 0.0089 | 0.1655 | 0.0173 |
| | YAK | 0.1548 | 0.0044 | 0.1569 | 0.0008 | 0.1504 | 0.0023 | 0.1558 | 0.0025 |
| | GKAO | 0.153 | 0.0011 | 0.158 | 0.002 | 0.1561 | 0.0039 | 0.1542 | 0.0018 |
| | Önerilen mKAO | 0.1475 | 0.0059 | 0.1429 | 0.0058 | 0.1406 | 0.0050 | 0.14733 | 0.0055 |
| XBI | K-Ortalamlar | 0.2266 | 5.9E-5 | 0.2048 | 0.0003 | 0.2440 | 0.0002 | 0.1298 | 0.0089 |
| | PSO | 0.2361 | 0.0083 | 0.2566 | 0.0153 | 0.2699 | 0.0141 | 0.1286 | 0.0036 |
| | GA | 0.2585 | 0.0287 | 0.2535 | 0.0236 | 0.2871 | 0.0294 | 0.1486 | 0.0314 |
| | YAK | 0.2297 | 0.0047 | 0.2533 | 0.0111 | 0.2781 | 0.0100 | 0.1281 | 0.0025 |
| | GKAO | 2.18E-5 | 1.33E-7 | 1.14E-5 | 2.91E-8 | 4.33E-5 | 6.40E-7 | 1.32E-5 | 5.04E-8 |
| | Önerilen mKAO | 3.24E-6 | 1.12E-7 | 2.35E-6 | 2.62E-8 | 3.21E-5 | 8.20E-8 | 1.21E-5 | 3.21E-8 |

Literatürden verilen yöntemler, yalnızca belirtilen görüntüler üzerinde sonuçlar vermiştir.

Çizelge 5.5'ten de görüleceği gibi, tez kapsamında önerilen yöntem, literatürdeki diğer yöntemlerden daha başarılı sonuçlar vermiştir.

Önerilen yöntem, literatürdeki farklı algoritmalarla karşılaştırıldıktan sonra farklı KAO algoritmaları ile de karşılaştırılmıştır. Bu algoritmalar için kullanılan parametre değerleri, referans çalışmalardaki değerler ile aynı alınmıştır ve parametre değerleri Çizelge 5.6'da verilmiştir (Zawbaa ve ark., 2016; Toz, 2019). Karşılaştırma sonuçları ise Çizelge 5.7'de verilmiştir.

Çizelge 5.6. Segmentasyon sonuçlarının karşılaştırılmasında kullanılan literatürdeki farklı KAO algoritmaların parametre değerleri

| Algoritma | Parametre | Değerler |
|---|--------------------------------------|----------|
| Orijinal KAO, GKAO, Kaotik KAO, Önerilen mKAO | İterasyon sayısı | 100 |
| | Karınca / Karınca aslanı sayısı | 30 / 30 |
| | Tekrarlama sayısı | 30 |
| | Çadır haritası (Tent map) | 0.001 |
| | Fonksiyon değerlendirme sayısı (FEs) | 3000 |

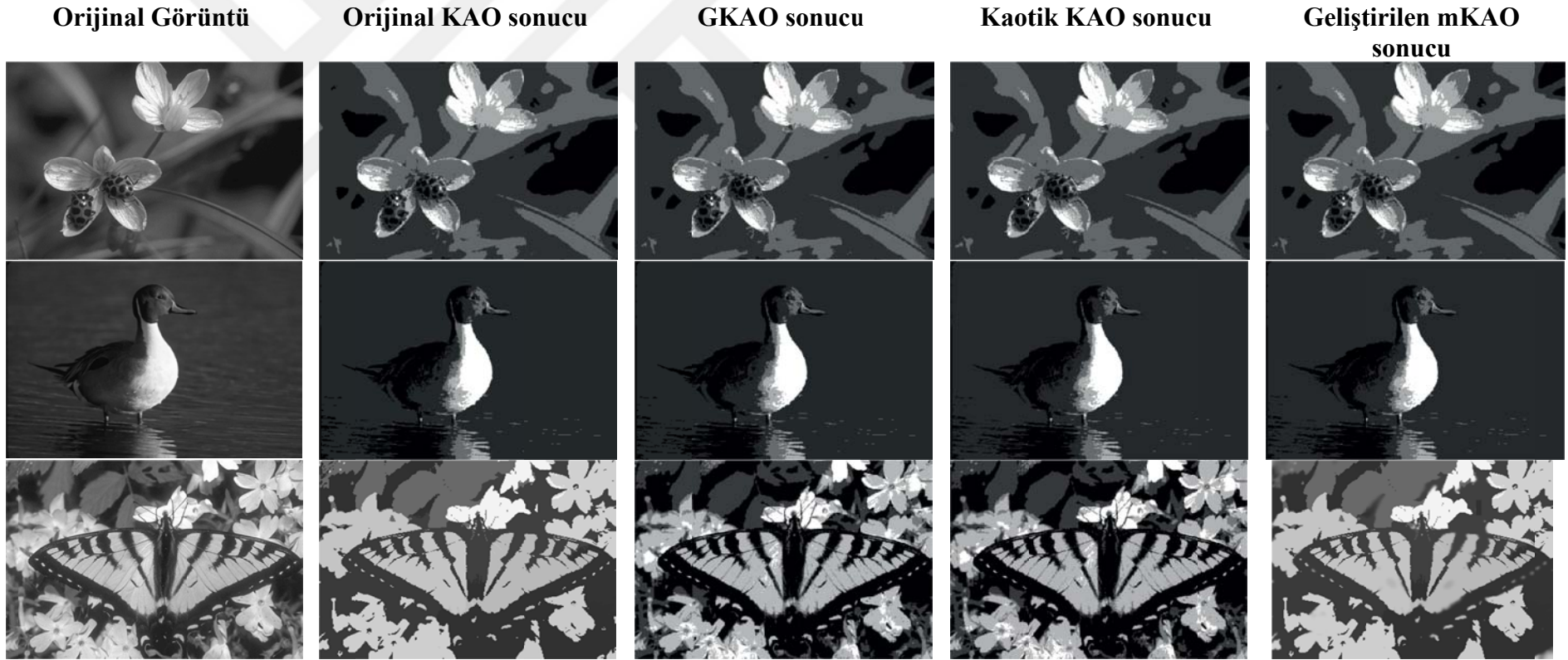
Önerilen yöntem, literatürdeki farklı KAO algoritmaları ile karşılaştırma sonuçları ise Çizelge 5.7’de verilmiştir (Zawbaa ve ark., 2016; Toz, 2019).

Çizelge 5.7. Gri seviye görüntü segmentasyonu için farklı KAO algoritmalarının karşılaştırılması (Zawbaa ve ark., 2016; Toz, 2019)

| Metrik | Yöntem | Görüntü ismi | | | | | |
|------------------------|---------------|-----------------|----------|-----------------|-----------|-----------------|-----------|
| | | 35008 | | 43051 | | 35010 | |
| | | Ortalama | SS | Ortalama | SS | Ortalama | SS |
| Amaç fonksiyonu değeri | Orijinal KAO | 736.2994 | 38.3905 | 400.6007 | 21.3415 | 559.1882 | 39.72 |
| | GKAO | 728.1738 | 23.0159 | 395.1102 | 11.357 | 561.3453 | 23.0585 |
| | Kaotik KAO | 753.3415 | 12.3611 | 410.1765 | 22.6691 | 581.8477 | 28.6938 |
| | Önerilen mKAO | 470.7116 | 38.781 | 425.0685 | 31.9173 | 699.0855 | 70.454 |
| J_e | Orijinal KAO | 11.4493 | 0.2374 | 10.7991 | 0.2792 | 10.7657 | 0.2249 |
| | GKAO | 11.3975 | 0.1639 | 10.8623 | 0.1642 | 10.7414 | 0.1476 |
| | Kaotik KAO | 11.5223 | 0.1347 | 10.91244 | 0.29674 | 10.8047 | 0.2555 |
| | Önerilen mKAO | 10.4115 | 0.27545 | 10.1096 | 0.25742 | 11.0727 | 0.3363 |
| d_{max} | Orijinal KAO | 14.7926 | 0.5961 | 14.6963 | 1.113 | 12.2865 | 0.1599 |
| | GKAO | 14.9358 | 0.3776 | 15.0607 | 0.6034 | 12.4293 | 0.2381 |
| | Kaotik KAO | 15.1945 | 0.2935 | 15.0289 | 1.12092 | 12.45159 | 0.2691 |
| | Önerilen mKAO | 14.6614 | 0.3489 | 14.5488 | 0.54781 | 12.3415 | 0.57614 |
| d_{min} | Orijinal KAO | 43.1965 | 0.9604 | 37.7084 | 3.2374 | 43.6025 | 4.7838 |
| | GKAO | 43.3313 | 0.9187 | 37.0075 | 2.0069 | 42.5042 | 3.3127 |
| | Kaotik KAO | 43.0898 | 1.586 | 38.0184 | 2.556 | 42.42049 | 4.4382 |
| | Önerilen mKAO | 43.5703 | 1.0639 | 41.1736 | 1.0476 | 43.484 | 1.0508 |
| DBI | Orijinal KAO | 0.1559 | 0.0125 | 0.14216 | 0.0082 | 0.1734 | 0.0483 |
| | GKAO | 0.1515 | 0.0082 | 0.1405 | 0.0061 | 0.1664 | 0.0216 |
| | Kaotik KAO | 0.1531 | 0.0071 | 0.1439 | 0.0061 | 0.1735 | 0.0416 |
| | Önerilen mKAO | 0.14406 | 0.005009 | 0.14869 | 0.0059293 | 0.1405 | 0.0058746 |
| XBI | Orijinal KAO | 4.359E-5 | 6.13E-7 | 3.613E-5 | 2.36E-6 | 2.116E-5 | 1.73E-6 |
| | GKAO | 4.357E-5 | 6.17E-7 | 3.677E-5 | 1.30E-6 | 2.079E-5 | 7.941E-7 |
| | Kaotik KAO | 4.393E-5 | 5.52E-7 | 3.654E-5 | 2.233E-6 | 2.114E-5 | 1.62E-6 |
| | Önerilen mKAO | 3.9E-5 | 5.13E-7 | 2.7E-5 | 1.4E-6 | 1.74E-5 | 6.82E-7 |

Literatürden verilen yöntemler, yalnızca belirtilen görüntüler üzerinde sonuçlar vermiştir.

Önerilen yöntemin, farklı KAO algoritmaları ile de karşılaştırılmasından elde edilen bazı örnek sonuç görüntüleri de Şekil 5.2’de verilmiştir.



Şekil 5.2. Farklı KAO algoritmaları için örnek segmentasyon sonuçları

Berkeley BSDS300 veri setinden alınan 10 farklı görüntü üzerinde literatürde sınıflar arası varyans ve farklı sayıda eşik değerlerine göre karşılaştırmalar da yapılmıştır. Karşılaştırmalar için kullanılan parametre değerleri (Akay, 2013) çalışmasından alınmıştır ve parametre değerleri Çizelge 5.8’te verilmiştir. Tez kapsamında önerilen yöntemin literatürde bulunan bu sonuçlarla karşılaştırma sonuçları Çizelge 5.9’da verilmiştir.

Çizelge 5.8. BSDS300 gri seviye resimler için segmentasyon sonuçlarının karşılaştırılmasında kullanılan literatürdeki farklı algoritmaların parametre değerleri

| Algoritma | Parametre | Değerler |
|---------------|--------------------------------------|--------------------|
| PSO | İterasyon sayısı | 100 |
| | Parçacık sayısı | 30 |
| | Fonksiyon değerlendirme sayısı (FEs) | 3000 |
| | Tekrarlama sayısı | 30 |
| | c_1, c_2 | $c_1 = c_2 = 1.49$ |
| | Atalet ağırlığı (inertia weight w) | [0.2- 0.9] |
| | Hız (velocity V) | 255 |
| YAK | İterasyon sayısı | 100 |
| | Koloni boyutu | 30 |
| | Tekrarlama sayısı | 30 |
| | Fonksiyon değerlendirme sayısı (FEs) | 3000 |
| Orijinal KAO | İterasyon sayısı | 100 |
| | Karınca / Karınca aslanı sayısı | 30 / 30 |
| | Tekrarlama sayısı | 30 |
| | Çadır haritası (Tent map) | 0.001 |
| | Fonksiyon değerlendirme sayısı (FEs) | 3000 |
| Önerilen mKAO | İterasyon sayısı | 100 |
| | Karınca / Karınca aslanı sayısı | 30 / 30 |
| | Tekrarlama sayısı | 30 |
| | Çadır haritası (Tent map) | 0.001 |
| | Fonksiyon değerlendirme sayısı (FEs) | 3000 |

Çizelge 5.9. Gri seviye görüntü segmentasyonu için önerilen yöntemin literatürdeki farklı algoritmalarla, sınıflar arası varyans kriterine dayalı olarak elde edilen en iyi eşik değerlerine göre karşılaştırılması (Akay, 2013)

| Görüntü ismi | Eşik sayısı | Sınıflar arası varyans | | | | | Eşik değerler | | | | |
|--------------|-------------|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------------|------------------------|------------------------|------------------------|------------------------|
| | | Otsu | PSO | YAK | Orijinal KAO | Önerilen mKAO | Otsu | PSO | YAK | Orijinal KAO | Önerilen mKAO |
| 35010 | 2 | 0.923426 | 0.923426 | 0.923426 | 0.923426 | 0.923426 | 83,155 | 83,155 | 83,155 | 83,155 | 83,155 |
| | 3 | 0.951457 | 0.950834 | 0.951471 | 0.945325 | 0.959303 | 72, 137, 196 | 72, 143, 198 | 71, 137, 195 | 88, 141, 197 | 75, 148, 199 |
| | 4 | 0.964544 | 0.963202 | 0.965896 | 0.970192 | 0.979008 | 59, 98, 150, 200 | 75, 117, 164, 197 | 66, 113, 161, 203 | 76, 122, 198, 200 | 63, 127, 196, 201 |
| | 5 | 0.975649 | 0.972272 | 0.975807 | 0.975961 | 0.98135 | 52, 81, 122, 168, 207 | 39, 74, 112, 154, 192 | 48, 79, 121, 166, 205 | 64, 111, 176, 190, 209 | 52, 109, 157, 208, 215 |
| 176039 | 2 | 0.910900 | 0.910900 | 0.910900 | 0.910900 | 0.910900 | 94, 167 | 94, 167 | 94, 167 | 94, 167 | 94, 167 |
| | 3 | 0.945091 | 0.944911 | 0.945211 | 0.948387 | 0.959894 | 73, 134, 185 | 77, 136, 190 | 74, 134, 187 | 75, 87, 141, 189 | 78, 107, 144, 187 |
| | 4 | 0.956891 | 0.960631 | 0.962228 | 0.968536 | 0.978053 | 58, 100, 142, 192 | 38, 71, 136, 184 | 40, 81, 135, 188 | 61, 90, 127, 193 | 57, 84, 143, 189 |
| | 5 | 0.974485 | 0.973205 | 0.974524 | 0.977139 | 0.981637 | 38, 78, 120, 159, 200 | 43, 74, 123, 155, 202 | 38, 77, 120, 158, 200 | 44, 77, 129, 163, 201 | 39, 84, 115, 185, 205 |
| 253036 | 2 | 0.914539 | 0.914539 | 0.914539 | 0.914539 | 0.914539 | 141, 204 | 141, 204 | 141, 204 | 141, 204 | 141, 204 |
| | 3 | 0.934884 | 0.944093 | 0.944904 | 0.933578 | 0.969033 | 68, 129, 194 | --- | 115, 177, 215 | 77, 137, 191 | 66, 1141, 197 |
| | 4 | 0.964154 | 0.960368 | 0.964222 | 0.965586 | 0.967028 | 70, 123, 177, 214 | 83, 117, 181, 220 | 71, 124, 178, 215 | 73, 115, 177, 212 | 78, 129, 187, 218 |
| | 5 | 0.969813 | 0.969674 | 0.974094 | 0.967307 | 0.985401 | 46, 80, 128, 179, 215 | 83, 117, 181, 220 | 70, 117, 163, 190, 220 | 54, 125, 171, 194, 219 | 73, 124, 169, 218 |
| 55067 | 2 | 0.913001 | 0.913001 | 0.913001 | 0.938937 | 0.915814 | 53, 143 | 53, 143 | 53, 143 | 53, 143 | 53, 143 |
| | 3 | 0.970362 | 0.913001 | 0.970678 | 0.974906 | 0.998657 | 60, 128, 207 | 48, 123, 192 | 54, 125, 199 | 62, 143, 197 | 68, 130, 196 |
| | 4 | 0.984290 | 0.983053 | 0.984402 | 0.987663 | 0.997391 | 51, 102, 153, 204 | 39, 111, 152, 201 | 50, 103, 150, 209 | 47, 113, 149, 200 | 48, 110, 153, 203 |
| | 5 | 0.992186 | 0.989866 | 0.992229 | 0.993467 | 0.997319 | 49, 92, 121, 156, 212 | 38, 95, 123, 144, 203 | 48, 92, 122, 158, 210 | 46, 93, 126, 160, 214 | 47, 95, 123, 157, 213 |
| 66075 | 2 | 0.968245 | 0.968245 | 0.968245 | 0.968245 | 0.968245 | 74, 135 | 74, 135 | 74, 135 | 74, 135 | 74, 135 |
| | 3 | 0.900417 | 0.892161 | 0.900453 | 0.903608 | 0.928443 | 68, 101, 150 | 59, 91, 136 | 68, 100, 149 | 62, 93, 139 | 63, 101, 137 |
| | 4 | 0.931263 | 0.928326 | 0.931586 | 0.937553 | 0.959253 | 66, 94, 126, 178 | 64, 91, 116, 177 | 64, 91, 124, 176 | 65, 96, 128, 179 | 64, 97, 127, 177 |
| | 5 | 0.949724 | 0.943309 | 0.951424 | 0.954842 | 0.970618 | 60, 83, 103, 136, 186 | 50, 75, 105, 124, 189 | 55, 77, 99, 131, 179 | 54, 88, 126, 154, 181 | 55, 78, 100, 138, 180 |
| 3096 | 2 | 0.856388 | 0.856388 | 0.856388 | 0.856388 | 0.856388 | 65, 118 | 65, 118 | 65, 118 | 65, 118 | 65, 118 |
| | 3 | 0.919361 | 0.918892 | 0.920436 | 0.928594 | 0.927345 | 76, 152, 180 | 92, 155, 180 | 85, 152, 180 | 84, 154, 181 | 85, 155, 179 |
| | 4 | 0.941907 | 0.942728 | 0.949283 | 0.949818 | 0.972883 | 51, 127, 161, 183 | 89, 127, 156, 185 | 75, 135, 163, 185 | 88, 129, 156, 187 | 78, 132, 164, 184 |
| | 5 | 0.957237 | 0.955297 | 0.963686 | 0.964139 | 0.98484 | 44, 96, 138, 165, 186 | 73, 120, 155, 172, 197 | 72, 128, 155, 173, 190 | 75, 123, 157, 173, 196 | 73, 122, 158, 176, 195 |
| 175043 | 2 | 0.803111 | 0.803111 | 0.803111 | 0.803111 | 0.803111 | 86, 134 | 86, 134 | 86, 134 | 86, 134 | 86, 134 |
| | 3 | 0.883920 | 0.880593 | 0.884028 | 0.893335 | 0.90093 | 69, 106, 148 | 60, 104, 148 | 68, 107, 149 | 73, 105, 147 | 70, 108, 148 |
| | 4 | 0.923659 | 0.911147 | 0.923684 | 0.926884 | 0.93749 | 60, 93, 122, 160 | 51, 75, 112, 157 | 60, 94, 123, 161 | 60, 92, 126, 157 | 61, 96, 124, 165 |
| | 5 | 0.945009 | 0.939712 | 0.945552 | 0.939185 | 0.979219 | 58, 86, 111, 137, 175 | 56, 85, 101, 123, 165 | 55, 84, 108, 134, 171 | 56, 88, 106, 129, 173 | 58, 89, 102, 134, 171 |
| 12003 | 2 | 0.843043 | 0.843043 | 0.843043 | 0.843043 | 0.843043 | 84, 157 | 84, 157 | 84, 157 | 84, 157 | 84, 157 |
| | 3 | 0.920170 | 0.918130 | 0.920181 | 0.91924 | 0.929631 | 66, 117, 176 | 61, 118, 170 | 66, 117, 175 | 66, 116, 169 | 63, 121, 175 |
| | 4 | 0.948285 | 0.947286 | 0.948576 | 0.94627 | 0.978166 | 60, 101, 139, 187 | 60, 100, 143, 190 | 57, 99, 136, 185 | 62, 105, 144, 184 | 59, 98, 124, 187 |
| | 5 | 0.963940 | 0.960563 | 0.964184 | 0.963841 | 0.967957 | 52, 87, 117, 151, 195 | 49, 91, 118, 153, 183 | 49, 84, 115, 148, 193 | 48, 89, 122, 169, 195 | 49, 88, 119, 155, 198 |
| 138078 | 2 | 0.890490 | 0.890490 | 0.890490 | 0.890490 | 0.890490 | 107, 189 | 107, 189 | 107, 189 | 107, 189 | 107, 189 |
| | 3 | 0.952274 | 0.951487 | 0.952338 | 0.958355 | 0.966236 | 98, 157, 217 | 100, 160, 211 | 97, 158, 216 | 99, 154, 213 | 100, 158, 215 |
| | 4 | 0.967525 | 0.964610 | 0.967565 | 0.970065 | 0.970932 | 78, 117, 160, 218 | 89, 124, 164, 208 | 78, 117, 160, 216 | 76, 119, 163, 217 | 78, 118, 162, 220 |
| | 5 | 0.972141 | 0.973751 | 0.976908 | 0.989593 | 0.998454 | 49, 88, 124, 161, 218 | 71, 114, 144, 181, 237 | 76, 114, 150, 182, 223 | 77, 117, 158, 195, 229 | 79, 123, 165, 199, 234 |
| 253027 | 2 | 0.827683 | 0.827683 | 0.827683 | 0.827683 | 0.827683 | 99, 173 | 99, 173 | 99, 173 | 99, 173 | 99, 173 |
| | 3 | 0.905167 | 0.903116 | 0.905790 | 0.913742 | 0.949146 | 84, 121, 183 | 88, 120, 186 | 86, 125, 187 | 87, 125, 185 | 86, 127, 188 |
| | 4 | 0.939160 | 0.934875 | 0.939189 | 0.939022 | 0.964324 | 77, 105, 136, 195 | 80, 104, 134, 207 | 78, 107, 137, 196 | 81, 122, 167, 198 | 83, 123, 164, 197 |
| | 5 | 0.956017 | 0.951261 | 0.956046 | 0.967121 | 0.998889 | 70, 95, 118, 147, 202 | 78, 109, 137, 181, 224 | 71, 95, 118, 146, 201 | 74, 99, 123, 156, 205 | 77, 103, 144, 187, 220 |

6. SONUÇLAR VE ÖNERİLER

6.1 Sonuçlar

Görüntü segmentasyonu; görüntü işleme, örüntü tanıma ve yapay zekâ gibi birçok alanda çok önemli ve zor bir sorundur. Bilgisayarlı görme teknolojisindeki temel ve önemli adım, görüntü anlam bilimsel anlayışın da önemli bir parçası olan görüntü segmentlemedir. Doğru segmentleme olmadan doğru görüntü işleme mümkün değildir. Bu nedenle görüntü segmentleme farklı alanlarda önemli bir görüntü analizi tekniğidir ve birçok farklı alanda yaygın olarak uygulanmaktadır. Tez çalışmasında, gri seviye görüntü segmentasyonu için yeni bir bölge büyütme tabanlı modifiye Karınca Aslanı Optimizasyon (mKAO) algoritması sunulmuştur. Tez kapsamında yapılan uygulamada öncelikle alınan gri seviye görüntülere görüntü kalitesini artırmak ve görüntülerdeki paraziti gidermek için ortanca (medyan) önışleme filtresi uygulanmıştır. Daha sonra bölge büyütme için doğru tohum piksellerini optimize etmek ve seçmek için modifiye KAO (mKAO) algoritması kullanılarak, görüntüler segmentlenmiştir. Yapılan işlemler literatürde kullanılan farklı karşılaştırma ölçütlerine göre değerlendirilmiş, elde edilen sonuçlar geliştirilen yöntemin gri seviye görüntü segmentasyonunda başarılı sonuçlar verdiğini göstermiştir.

Tez kapsamında önerilen mKAO ile Berkeley BSDS300 veri kümesindeki 300 görüntü kullanılarak yapılan segmentasyon sonuçları, hem temel segmentasyon algoritması olan K-Ortalamalar (K-Means) ile hem de orijinal Karınca Aslanı Optimizasyon (KAO) algoritması ile karşılaştırılmıştır. Elde edilen sayısal sonuçlar ve elde edilen sonuç görüntülerine göre, tez kapsamında önerilen modifiye KAO algoritmasının (mKAO) daha iyi segmentasyon sonuçları verdiği görülmektedir.

Eşik değeri kullanarak görüntü segmentasyonu için literatürde en çok kullanılan yöntemler olarak Otsu ve Kapur kullanılmaktadır. mKAO kullanılarak 2-3-4-5 eşik değeri kullanılarak da segmentasyon yapılmış ve sonuçlar hem Otsu ve Kapur ile hem de temel KAO ile karşılaştırılmıştır. Karşılaştırma sonuçlarına göre Özellik Benzerlik İndeksi (ÖBİ), Korelasyon Katsayısı (KK), Sınır Yer Değiştirme Hatası (SYDH) ve Global Tutarlılık Hatası (GTH) değerlerine göre mKAO, diğer algoritmalarından daha üstün performans göstermiştir.

Önerilen mKAO'nun PSO, GA, YAK ve GKAO ile yapılan karşılaştırmalarından da görüleceği üzere bir görüntü hariç diğer tüm görüntülerde

mKAO'nun tüm algoritmalarından daha başarılı olduğu görülmektedir. Ayrıca mKAO'nun literatürdeki diğer geliştirilen KAO algoritmaları ile karşılaştırılmasında da, mKAO'nun rakabetçi sonuçlar verdiği görülmüştür.

Gri seviye görüntü segmentasyonu için literatürdeki çalışmalarla farklı eşik değer sayıları ve bu eşik değerleri temel alınarak yapılan karşılaştırmada ise, mKAO'nun sınıflar arası varyans değerine göre en iyi sonuçlar verdiği görülmektedir.

6.2 Öneriler

Segmentasyon genellikle görüntü analizinin ilk aşamasıdır. Görüntü segmentasyonu, bir görüntüyü kendi içlerinde aynı ama diğer bölgelerle (segmentlerle) farklı özelliklere sahip, anlamlı bölgelere ayırmak işlemi olarak tanımlanabilir. Segmentasyon yöntemleri ve bu yöntemlerin başarıları; uygulamaya, probleme veya amaca göre değişebilmektedir. Bu nedenle tüm görüntülere uygulanabilecek genel ve ortak bir segmentasyon yöntemi bulunmamaktadır. Bu nedenle de bilgisayarlı otomatik görüntü segmentasyonu, görüntü işlemenin en zor işlem adımlarından biridir.

Gri seviye görüntü segmentasyonu, temel olarak gri seviye piksel değerlerinin süreksizlik (discontinuity) ve benzerlik (similarity) olarak iki temel özelliğine göre yapılır. Gri seviye değerlerindeki süreksizlikler dikkate alındığında, gri seviyelerdeki ani değişikliklere yani görüntüdeki kenar ve hatların belirlenmesine (edge/line detection) dayanılarak segmentasyon işlemi yapılır. Benzerliklere göre yapılan segmentasyon işlemi ise; bölge segmentleme (region segmentation) olarak isimlendirilir ve eşikleme (thresholding), bölge büyütme (region growing), bölge ayırma/birleştirme (region splitting/merging) işlemlerine dayalı olarak yapılır.

Bölge büyütme ile görüntü segmentasyonunda, bir veya belirli sayıda başlangıç (seed) pikseli rastgele ya da otomatik olarak seçilir. Bu başlangıç pikselleri ile komşu piksellerin benzerliği, belirlenecek bir benzerlik kriterine göre test edilerek, benzer özellikteki piksellerin birbirine bağlanması sağlanır ve böylelikle bölge büyütülmesi gerçekleştirilir.

Tez kapsamında önerilen modifiye bir KAO algoritması (mKAO), bölge büyütme ile segmentasyon işleminde, başlangıç tohumların seçilmesinde kullanılmıştır.

Tez kapsamında verilen bilgiler, başta KAO algoritması olmak üzere metasegisel algoritmalarla görüntü işleme ve özellikle segmentasyon alanında çalışmak isteyen araştırmacılara alt yapı niteliğinde bilgiler sunmaktadır. Gri seviye

görüntü segmentasyonu için ilerleyen zamanlarda hem farklı algoritmalar kullanılabilir hem de farklı alanlarda yapılacak çalışmalarda hibrit algoritmalar ortaya konabilir. Histogram temelli çalışan segmentasyon algoritmaları için eşik değerlerin bulunmasında, daha doğru sonuçlar ortaya konması amacıyla metasezgisel algoritmalar kullanılabilir. Ayrıca renkli görüntülerde segmentasyon işlemi için metasezgisel algoritmalar, renk kanallarında ayrı ayrı kullanılarak, elde edilen sonuçlar tekrar birleştirilerek, renkli segmentasyon yapılabilir. Buna benzer olarak, farklı dalga boylarında alınan termal ya da infrared görüntülerde de nesne tespiti için bölge büyütme algoritmaları kullanılarak farklı alanlarda, farklı türde görüntüler segmentlenebilir.



KAYNAKLAR

- Aarts E, Korst J, Michiels W, 2005. Simulated annealing. In: Search methodologies. Eds: Springer, 187-210.
- Ali E, Abd Elazim S, Abdelaziz A, 2016. Ant lion optimization algorithm for renewable distributed generations, *Energy* (116), syf. 445-58.
- Back T, 1996. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford university press, p.
- Baghi A, Karami A. SAR image segmentation using region growing and spectral cluster. 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), 229-32.
- Balafar MA, Ramli AR, Saripan MI, Mashohor S, 2010. Review of brain MRI image segmentation methods. *Artificial Intelligence Review*, 33, 3, 261-74.
- Barma PS, Dutta J, Mukherjee A, 2019. A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem. *Decision Making: Applications in Management and Engineering*, 2 (2), 112-25.
- Baxes GA, 1994. Digital image processing: principles and applications, Wiley New York, p.
- Bhargavi K, Jyothi S, 2014. A survey on threshold based segmentation technique in image processing. *International Journal of Innovative Research and Development*, 3 (12), 234-239.
- Bhattacharyya S, 2011. A brief survey of color image preprocessing and segmentation techniques. *Journal of Pattern Recognition Research*, 1 (1), 120-129
- Bruntha PM, Kanimozhi M. Application Of Selective Region Growing Algorithm In Lung Nodule Segmentation. 2018 4th International Conference on Devices, Circuits and Systems (ICDCS), 319-22.
- Charifi R, Essbai N, Mansouri A, Zennayi Y. Comparative Study of Color Image Segmentation by the Seeded Region Growing Algorithm. 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), 279-84.
- Cheng H, Butler D. Segmentation of aerial surveillance video using a mixture of experts. *Digital Image Computing: Techniques and Applications (DICTA'05)*, 66-.
- Chondro P, Yao C-Y, Ruan S-J, Chien L-C, 2018. Low order adaptive region growing for lung segmentation on plain chest radiographs. *Neurocomputing*, 275, 1002-11.
- Cui W, Guan Z, Zhang Z. An improved region growing algorithm for image segmentation. 2008 International Conference on Computer Science and Software Engineering, 93-6.
- Dorigo M, Birattari M, 2010. Ant colony optimization, Springer, p.
- Dorigo M, Blum C, 2005. Ant colony optimization theory: A survey. *Theoretical computer science*, 344 (2-3), 243-78.
- Dorigo M, Stützle T, 2003. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In: Handbook of metaheuristics. Eds: Springer, p. 250-85.
- Duman E, Erdem OA. A new image denoising method based on region growing segmentation. 2017 25th Signal Processing and Communications Applications Conference (SIU), 1-4.
- Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 39-43.

- Elmorsy SA, Abdou MA, Hassan YF, Elsayed A. K3. A region growing liver segmentation method with advanced morphological enhancement. 2015 32nd National Radio Science Conference (NRSC), 418-25.
- Gandomi AH, Yang X-S, Alavi AH, 2013. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29 (1), 17-35.
- Gao H, Dou L, Chen W, Xie G. The applications of image segmentation techniques in medical CT images. Proceedings of the 30th Chinese Control Conference, 3296-9.
- Ghosh A, Pal NR, Pal SK, 1991. Image segmentation using a neural network. *Biological Cybernetics*, 66 (2), 151-8.
- Happ PN, Ferreira RS, Costa GA, Feitosa RQ, Bentes C, Gamba P. Towards distributed region growing image segmentation based on MapReduce. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 4352-5.
- Haralick RM, Shapiro LG, 1985. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29 (1), 100-32.
- Imms AD, Richards OW, Davies RG, 2012. Imms' General Textbook of Entomology: Volume 2: Classification and Biology, Springer Science & Business Media, p.
- Iyer S, Nadkarni AP, Padmini T. Antlion optimization and Whale optimization Algorithm for multilevel thresholding segmentation. 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), 1-8.
- Jeevakala S, Rangasami R, 2018. A novel segmentation of cochlear nerve using region growing algorithm. *Biomedical Signal Processing and Control*, 39, 117-29.
- Kaganami HG, Beiji Z. Region-based segmentation versus edge detection. 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 1217-21.
- Kang W-X, Yang Q-Q, Liang R-P. The comparative research on image segmentation algorithms. 2009 First International Workshop on Education Technology and Computer Science, 703-7.
- Karaboga D, Basturk B, 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39 (3), 459-71.
- Kaveh A, Farhoudi N, 2013. A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, 59, 53-70.
- Kaveh A, Khayatazad M, 2012. A new meta-heuristic method: ray optimization. *Computers & structures*, 112, 283-94.
- Kaveh A, Mahdavi V, 2014. Colliding bodies optimization: a novel meta-heuristic method. *Computers & Structures*, 139, 18-27.
- Kaveh A, Talatahari S, 2010. A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213 (3-4), 267-89.
- Kaveh A, Zolghadr A, 2014. Democratic PSO for truss layout and size optimization with frequency constraints. *Computers & Structures*, 130, 10-21.
- Kelkar D, Gupta S. Improved quadtree method for split merge image segmentation. 2008 First International Conference on Emerging Trends in Engineering and Technology, 44-7.
- Khan JF, Bhuiyan SM, Adhami RR, 2010. Image segmentation and shape analysis for road-sign detection. *IEEE Transactions on Intelligent Transportation Systems*, 12 (1), 83-96.
- Koc I, Baykan OK, Babaoglu I, Gri kurt optimizasyon algoritmasına dayanan çok seviyeli imge eşik seçimi. *Politeknik Dergisi*, 21 (4), 841-7.

- Kotteswari R, Sathiya K. Analysis of foreground detection in MRI images using region based segmentation. 2016 International Conference on Communication and Signal Processing (ICCSP), 1062-5.
- Lee KS, Geem ZW, 2004. A new structural optimization method based on the harmony search algorithm. *Computers & structures*, 82, 9-10, 781-98.
- Li BN, Chui CK, Chang S, Ong SH, 2011. Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation. *Computers in biology and medicine*, 41 (1), 1-10.
- Li L, Sun L, Guo J, Qi J, Xu B, Li S, 2017. Modified discrete grey wolf optimizer algorithm for multilevel image thresholding. *Computational intelligence and neuroscience*, 2017.
- Li X, Jing L, Lin Q, Li H, Xu R, Tang Y, Ding H, Liu Q. A new region growing-based segmentation method for high resolution remote sensing imagery. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 4328-31.
- Liu B, Wang L, Jin Y-H, Tang F, Huang D-X, 2005. Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals*, 25 (5), 1261-71.
- Lu X, Wu J, Ren X, Zhang B, Li Y, 2014. The study and application of the improved region growing algorithm for liver segmentation. *Optik-International Journal for Light and Electron Optics*, 125 (9), 2142-7.
- Mahdavi M, Fesanghary M, Damangir E, 2007. An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, 188 (2), 1567-79.
- Malarvel M, Sethumadhavan G, Bhagi PCR, Thangavel S, Krishnan A. Region growing based segmentation with automatic seed selection using threshold techniques on X-radiography images. 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 1-4.
- Mao X, Zhang Y, Hu Y, Sun B. Color image segmentation method based on region growing and ant colony clustering. 2009 WRI Global Congress on Intelligent Systems, 173-7.
- Marques O, 2011. Practical image and video processing using MATLAB, John Wiley & Sons, p.
- Martin D, Fowlkes C, Tal D, Malik J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.
- Merzougui M, El Allaoui A, 2019. Region growing segmentation optimized by evolutionary approach and Maximum Entropy. *Procedia Computer Science*, 151, 1046-51.
- Mirghasemi S, Rayudu R, Zhang M. A new image segmentation algorithm based on modified seeded region growing and particle swarm optimization. 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), 382-7.
- Mirjalili S, 2015. The ant lion optimizer. *Advances in Engineering Software*, 83, 80-98.
- Mirjalili S, Lewis A, 2016. The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.
- Mirjalili S, Mirjalili SM, Lewis A, 2014. Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Mitchell M, 1995. Genetic algorithms: An overview. *Complexity*, 1 (1), 31-9.
- Mostafa A, Houseni M, Allam N, Hassanien AE, Hefny H, Tsai P-W. Antlion optimization based segmentation for MRI liver images. International Conference on Genetic and Evolutionary Computing, 265-72.

- Mukherjee A, Barma PS, Dutta J, Panigrahi G, Kar S, Maiti M, 2020. A modified discrete antlion optimizer for the ring star problem with secondary sub-depots. *Neural Computing and Applications*, 32 (12), 8143-56.
- Naz S, Majeed H, Irshad H. Image segmentation using fuzzy clustering: A survey. 2010 6th international conference on emerging technologies (ICET), 181-6.
- Neri F, Cotta C, 2012. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2, 1-14.
- Neshat M, Sepidnam G, Sargolzaei M, 2013. Swallow swarm optimization algorithm: a new method to optimization. *Neural Computing and Applications*, 23 (2), 429-54.
- Ozturk C, Hancer E, Karaboga D, 2015. Improved clustering criterion for image clustering with artificial bee colony algorithm. *Pattern Analysis and Applications*, 18 (3), 587-99.
- Pare S, Bhandari AK, Kumar A, Singh GK, Khare S. Satellite image segmentation based on different objective functions using genetic algorithm: a comparative study. 2015 IEEE international conference on digital signal processing (DSP), 730-4.
- Petitjean C, Dacher J-N, 2011. A review of segmentation methods in short axis cardiac MR images. *Medical image analysis*, 15 (2), 169-84.
- Punitha S, Amuthan A, Joseph KS, 2018. Benign and malignant breast cancer segmentation using optimized region growing technique. *Future Computing and Informatics Journal*, 3 (2), 348-58.
- Qian Zhao Y, Hong Wang X, Fang Wang X, Shih FY, 2014. Retinal vessels segmentation based on level set and region growing. 47, 7, 2437-46.
- Qin AK, Huang VL, Suganthan PN, 2008. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation*, 13 (2), 398-417.
- Rajinikanth V, Couceiro M, 2015. RGB histogram based color image segmentation using firefly algorithm. *Procedia Computer Science*, 46, 1449-57.
- Rashedi E, Nezamabadi-Pour H, Saryazdi S, 2009. GSA: a gravitational search algorithm. *Information sciences*, 179, 13, 2232-48.
- Reddy AS, Reddy PC. Novel Algorithm based on Region Growing Method for Better Image Segmentation. 2018 3rd International Conference on Communication and Electronics Systems (ICCES), 229-34.
- Sağ T, Çunkaş M, 2015. Color image segmentation based on multiobjective artificial bee colony optimization. *Applied soft computing*, 34, 389-401.
- Senthilkumaran N, Rajesh R, 2009. Edge detection techniques for image segmentation-a survey of soft computing approaches. *International journal of recent trends in engineering*, 1 (2), 250.
- Sharma A, Sehgal S. Image segmentation using firefly algorithm. 2016 International Conference on Information Technology (InCITe)-The Next Generation IT Summit on the Theme-Internet of Things: Connect your Worlds, 99-102.
- Singh AK, Gupta B, 2015. A novel approach for breast cancer detection and segmentation in a mammogram. *Procedia Computer Science*, 54, 676-82.
- Solomon C, Breckon T, 2011. Fundamentals of Digital Image Processing: A practical approach with examples in Matlab, John Wiley & Sons, p.
- Starr DW, 1976. Automatic interpretation of Landsat images using context sensitive region merging, University of British Columbia.
- Talbi E-G, 2009. Metaheuristics: from design to implementation, John Wiley & Sons, p.

- Teoh SH, Ibrahim H, 2012. Median filtering frameworks for reducing impulse noise from grayscale digital images: a literature survey. *International Journal of Future Computer and Communication*, 1 (4), 323.
- Toz M, 2019. An improved form of the ant lion optimization algorithm for image clustering problems. *Turkish Journal of Electrical Engineering & Computer Sciences*, 27 (2), 1445-60.
- Viji KA, JayaKumari J. Modified texture based region growing segmentation of MR brain images. 2013 IEEE Conference on Information & Communication Technologies, 691-5.
- Wu J, Guo H. A method for sonar image segmentation based on combination of MRF and region growing. 2015 Fifth International Conference on Communication Systems and Network Technologies, 457-60.
- Yang X-S, 2010. A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Eds: Springer, p. 65-74.
- Yang Z, Chung F-L, Shitong W, 2009. Robust fuzzy clustering-based image segmentation. *Applied soft computing*, 9 (1), 80-4.
- Yigitbasi ED, Baykan NA, 2013. Edge detection using artificial bee colony algorithm (ABC). *International Journal of Information and Electronics Engineering*, 3 (6), 634.
- Yogarajan G, Revathi T, 2016. A discrete ant lion optimization (DALO) algorithm for solving data gathering tour problem in wireless sensor networks. *Middle-East Journal of Scientific Research*, 24 (10), 3113-20.
- Yogarajan G, Revathi T, 2018. Improved cluster based data gathering using ant lion optimization in wireless sensor networks. *Wireless Personal Communications*, 98, 3, 2711-31.
- Zaitoun NM, Aqel MJ, 2015. Survey on image segmentation techniques. *Procedia Computer Science*, 65, 797-806.
- Zanaty EA, Ghiduk AS, 2013. A novel approach based on genetic algorithms and region growing for magnetic resonance image (MRI) segmentation. *Comput. Sci. Inf. Syst.*, 10 (3), 1319-42.
- Zawbaa HM, Emary E, Grosan C, 2016. Feature selection via chaotic antlion optimization. *PloS one*, 11 (3), e0150652.
- Zhang H, Fritts JE, Goldman SA, 2008. Image segmentation evaluation: A survey of unsupervised methods. *Computer vision and image understanding*, 110 (2), 260-80.
- Zhang X, Li X, Feng Y, 2015. A medical image segmentation algorithm based on bi-directional region growing. *Optik*, 126 (20), 2398-404.
- Zhongming L, Jun W. The image segmentation algorithm of region growing and wavelet transform modulus maximum. 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 1171-4.
- Zhou Y-m, Jiang S-y, Yin M-l. A region-based image segmentation method with mean-shift clustering algorithm. 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 366-70.