

Araştırma Makalesi - Research Article

Riemann Anlamında Eğri Evrim Modeli İncelemesi: Görüntü Segmentasyonu Uygulaması

Riemannian Curve Model Analysis: Image Segmentation Application

Haydar Kılıç^{1*}, Salim Ceyhan²

Geliş / Received: 15/06/2021

Reviz / Revised: 16/09/2021

Kabul / Accepted: 22/09/2021

ÖZ

Görüntü bölütlemesinde görüntü üzerinde bir başlangıç eğrisi vererek, eğrinin hareketi ile görüntü üzerindeki objeleri sarması sağlanabilir. Burada eğri hareketine neden olan bir kısmi türevli yapı olduğu için, bu sınıfta bir bölütlemeye kısmi diferensiyel tabanlı bölütleme denilmektedir. Bu çalışmada, kısmi türevlerden oluşturulan bir matematik modelle görüntü segmentasyonu ile ilgili derin bir matematiksel analiz ve sayısal hesaplamalar bulunmaktadır. Sayısal hesaplamalarda, modele kullanıcı tarafından girilen parametrelerin incelemesi yapılmış, ayrıca bu parametrelerin yapay zeka algoritmaları ile optimizasyonu üzerinde durulmuştur. Ayrıca tüm nümerik hesapları yapan kullanıcı dostu bir arayüz uygulaması geliştirilmiştir. Uygulamadaki hesaplamalar yapay zeka algoritmaları ile yapılabilir, veya kullanıcı isterse arayüze gireceği değerlerle manuel bir hesaplamada yapabilir.

Anahtar Kelimeler- *Görüntü Bölütleme, Optimizasyon, Eğri Evrimi, Yapay Zekâ, Matematik Model*

ABSTRACT

By giving an initial curve on the image in image segmentation, it can be provided to wrap the objects on the image with the movement of the curve. A segmentation in this class is called partial differential-based segmentation, since there is a partial differential structure that causes the curve motion here. In this study, there is a deep mathematical analysis and numerical calculations related to image segmentation with a mathematical model created from partial derivatives. In numerical calculations, the parameters entered into the model by the user were examined, and the optimization of these parameters with artificial intelligence algorithms was emphasized. In addition, a user-friendly interface application that performs all numerical calculations has been developed. Calculations in the user interface can be made with artificial intelligence algorithms, or if the user wishes, user can make a manual calculation with the values entered into the interface.

Keywords- *Image Segmentation, Optimization, Curve Evolution, Artificial Intelligence, Mathematical Model*

^{1*}Sorumlu yazar iletişim: haydar.kilic@bilecik.edu.tr (<https://orcid.org/0000-0002-2551-3772>)

Bilgisayar Mühendisliği, Bilecik Şeyh Edebali Üniversitesi, Bilecik, Turkey

²İletişim: salim.ceyhan@bilecik.edu.tr (<https://orcid.org/0000-0003-0274-6175>)

Bilgisayar Mühendisliği, Bilecik Şeyh Edebali Üniversitesi, Bilecik, Turkey

I. GİRİŞ

Genel olarak herhangi bir görüntü iki boyutlu bir $I(x,y)$ fonksiyonu olarak açıklanabilir. Burada (x,y) uzamsal koordinatlarda $I(x,y)$ görüntü bilgisi temsil edilir. Görüntü tipine bağlı olarak bu bilgi ışık şiddeti, görüntü derinliği, dalga şiddeti veya sıcaklığı olabilir.

Dijital (Sayısal) bir görüntü iki boyutlu bir ayrık fonksiyondur ve uzamsal koordinatlarda sayısallaştırılabilir. Sayısal bir görüntüyü her bir elementinin belirttiği noktaya piksel denilen bir matris olarak görebiliriz. Görüntüde ise ilgili element oradaki görüntü özelliğinin yoğunluk seviyesini tanımlar. Örneğin,

$$\mathcal{F}_n = \{0,1,2, \dots, n - 1\} \quad (1)$$

Özellik seviyelerinin ayrık bir kümesi ve $I(x,y) \in \mathcal{F}_n$ olmak üzere, $[I(x,y)]_{p \times q}$ matrisi $p \times q$ boyutunda bir görüntüdür.

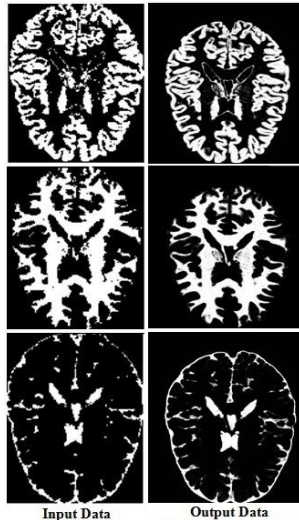
Segmentasyon (Bölütleme), görüntüyü birbiri ile kesişmeyen bölgelere ayırmak demektir. Her bir bölge homojen olmasına rağmen, herhangi iki bölgenin birleşimi homojen olamaz. Matematiksel olarak, diyelim ki, \mathcal{F} tüm piksellerin bir kümesi olsun. $H(\cdot)$ 'de homojenliği gösterebilir, yani $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_n$ homojen bölgeler olmak üzere,

$$\bigcup_{i=1}^n \mathcal{R}_i = \mathcal{F}, \mathcal{R}_i \cap \mathcal{R}_j = \emptyset, i \neq j \quad (1)$$

yazılabilir. Böylece her bir \mathcal{R}_i için $H(\mathcal{R}_i) = 1$ dir ve $\mathcal{R}_i, \mathcal{R}_j$ bölgesine komşu ise $H(\mathcal{R}_i \cup \mathcal{R}_j) = 0$ dır. Bir başka deyişle, herhangi bir bölgedeki pikseller görüntünün oradaki bir özelliğine (renk, gri seviye, kenar şiddeti, v.s.) göre benzerse o bölgeye homojen denilir [1].

Bu zamana kadar görüntüden kullanılan bilgilere göre bir çok segmentasyon teknikleri kullanılmıştır. Bunlar yaygın kullanımlarına göre bölge tabanlı, kenar tabanlı, eşikleme tabanlı, teori tabanlı, model tabanlı ve PDE (Kısmi diferensiyel denklem) tabanlı olarak altı sınıfa ayrılabilir [2].

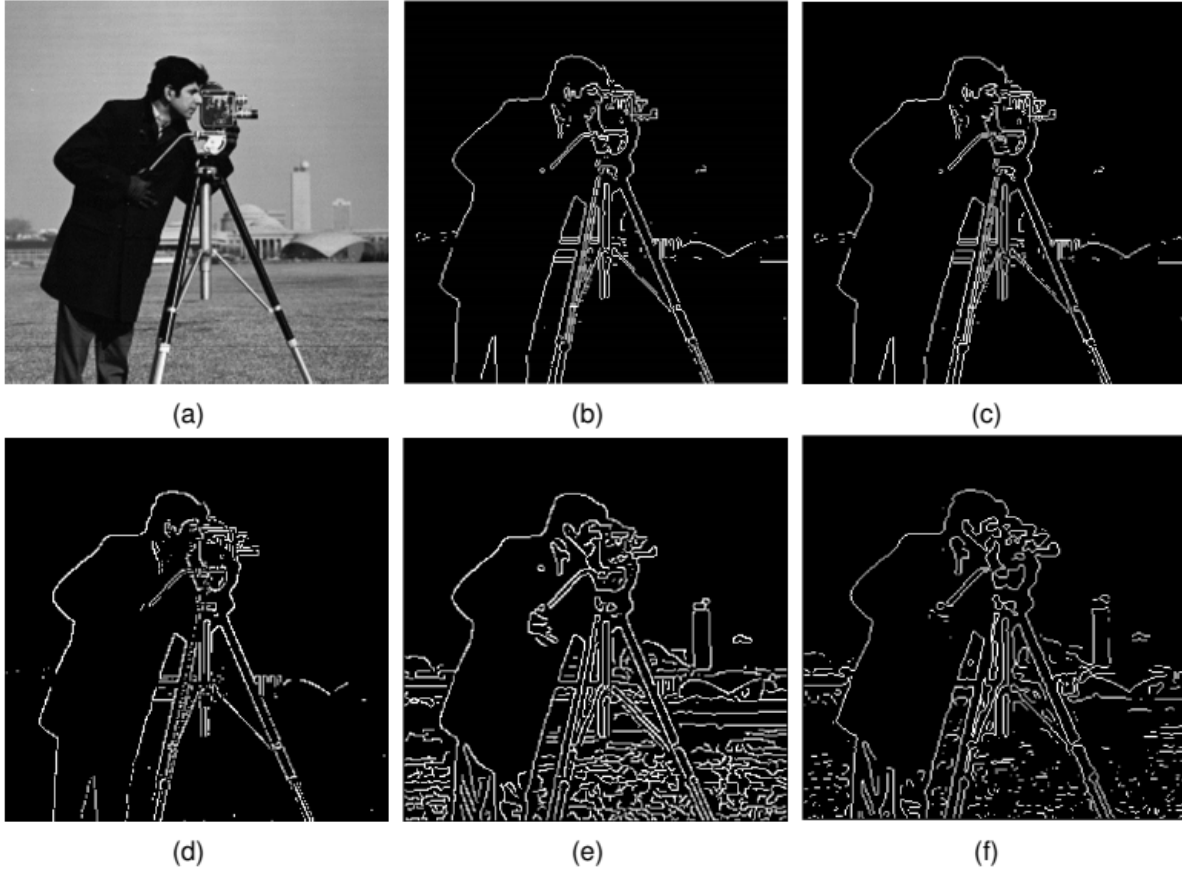
Bunlardan ilki olan "bölge tabanlı" teknikteki bölgeler aynı görüntü bileşenleri yoğunluğuna sahiptir. Yani bölgeler içerisinde komşu piksellerin oluşturduğu kümeler aynı yoğunluktaki görüntü bilgilerinden oluşur. Bölge büyütme, bölgeleri ayırma veya kaynaştırma yapılabilir. Komşu piksellerin homojenliğine veya benzerliğine bakılır. Örneğin bölge büyütme methodunda, görüntü homojenliklerine göre ayrı ayrı bölgelere bölütlenir. Her bir bölgedeki pikseller aynı kriterde benzerliğe sahiptir. Bu benzerlikler renk, yoğunluk gibi kriterler olabilir [3]. Aşağıda bu metolla bölütlenmiş bir figür görmekteyiz.



Şekil 1. Bölge tabanlı bölütlenmiş bir Beyin MR görüntüsü, sağda ilk girdiler, solda sonuçlar görülmektedir [3].

Bu figürde dikkat edilirse bölge büyütme metodu ile girdi görüntüsündeki gürültülü bölgelerin gürültüden arındırılarak çıktısı olarak daha detaylı ve alt bölgelerin daha belirgin olduğu görülür. Buda özellikle medical görüntüde hastalıkların daha iyi netice ile tanı konulup tedavisine neden olur. [3] makalesinin iddiasına göre bu yöntemin otomatik prosedürü nedeniyle zamandan da tasarruf sağladığı ve görüntü segmentasyonunu iyileştirmek işleme için sonraki adımları büyük ölçüde etkileyeceği için, büyük ölçekdeki görüntü segmentasyonlarında kullanılabilecektir.

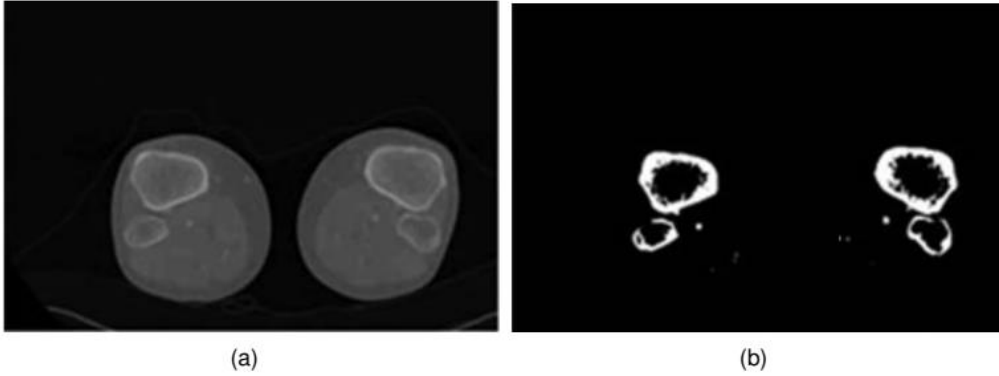
İkinci bir teknikte "kenar tabanlı" tekniktir. Kenar iki bölge arasındaki sınır olarak düşünüldüğünde, bir bölgeden diğer bir bölgeye geçişlerdeki ani yoğunluk değişimine sahip piksellerin tespiti yapıldığında kenarlar bulunmuş olur. Klasik kenar buluculardan bazıları Sobel, Prewitt, Roberts, Canny ve Laplacian'dir. Bu kenar bulucular görüntünün her bir noktasında gradyan büyüklüklerini yaklaşık olarak hesaplayarak ayırık bir gradyan operatörü oluştururlar. Böylece o noktadaki yoğunluk geçişinin büyüklüğü hesaplanabilmektedir [4-6]. Aşağıdaki figürde bu operatörlerle kenarlar MATLAB'in klasik 'cameraman' görüntüsü üzerinde bulunmuştur.



Şekil 2. Klasik kenar bulucular. (a) Orijinal görüntü, (b) Sobel, (c) Prewitt, (d) Roberts, (e) Canny, (f) Laplacian

Esas olarak figürde gösterilen klasik kenar bulucuların uygulama alanları bakımından birisinin diğerine üstünlüğünü karşılaştırmak haksızlık olur, bu nedenle bu kenar bulucular en iyiler arasında yer almaktadır. Kullanım amaçlarına göre birisi veya birileri seçilebilir.

Üçüncü teknik "eşikleme" tekniğidir. Bir eşik değeri ile binari görüntü denilen sadece iki yoğunluktan oluşan görüntü oluşturulur. Böylece bütün bir görüntü bölütlenmiş olur. Eşikleme tekniğinde farklı gri seviyelerdeki bölgeler bölütlenir. Eşik değeri görüntünün histogramındaki bir değerdir ve görüntüyü iki yoğunluklu bölgeye ayırır [7]. Aşağıda bir kemik görüntüsünün bu yöntemle bölütlendiğini görüyoruz.

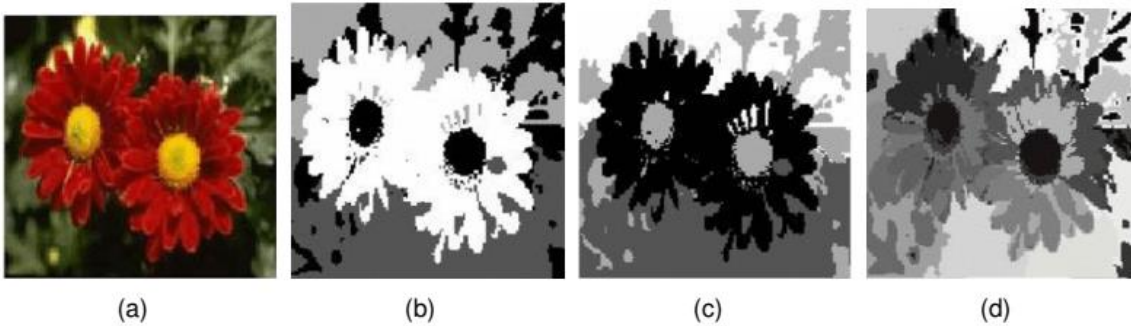


Şekil 3. Kemik resminde eşikleme uygulaması (a) Orijinal görüntü, (b) Bölütlenmiş görüntü [7]

Figürde, yerel eşikleme yöntemi ile görüntünün farklı bölümleri için farklı eşik değerleri bulmak amacıyla alt görüntülere bölünmüş ve ardından her bir bölüm için farklı eşik değerleri hesaplanmıştır. Yöntemde, görüntü yatay dikey çizgilere bölünürken, her parça hem arka planın hemde nesnenin bir parçasını içerir. Sonuçta enterpolasyon ile uygun sonuçlar elde edilir.

"Teori tabanlı" teknikler sınıfında "kümeleme tabanlı" ve "sinir ağları tabanlı" olmak üzere iki tip teknikten bahsedilebilir. Kümeleme tabanlı tekniklerde görüntünün renk, boyut, yoğunluk, doku gibi özelliklerine dayanarak sinir ağlarından farklı olarak denetimsiz bir öğrenme mekanizması ile gruplandırma vardır. Rasgele seçilen kümelerden yapılan hesaplar sonucunda birbirine yakın özellikteki kümeler grup yapılır.

Kümeleme tekniğini bir denetimsiz öğrenme tekniği olarak düşünebiliriz. Pikseller arasında bir benzerlik kriteri tanımlanır ve daha sonra benzer pikseller kümeler oluşturmak için birlikte gruplandırılır. Piksellerin kümeler halinde gruplandırılması, sınıf içi benzerliği en üst düzeye çıkarma ve sınıflar arası benzerliği en aza indirme ilkesine dayanır. Rasgele seçilen kümeler arasındaki uzaklıklar Öklid uzaklığı, Minkowski uzaklığı gibi uzaklık ölçü metodları ile ölçülür. Temel olarak K-means, Partitioning Around Medoids (PAM) ve Self Organizing Maps (SOM) metodları ile üç sınıfta incelenebilir. K-means metodunda bir veri kümesi, giriş olarak K sayıda kümeye bölünerek, kümeler arasında benzerliklerin en az, içinde ise en çok olması amaçlanmaktadır. PAM metodunda 'medoid' kavramı veri kümesinin temsilci nesnelere dir. Bunların veri kümesi içerisindeki ortalama benzersizlikleri en küçüktür. Ayrıca bu metodda K sayıda temsilci nesne kendisine en yakın nesne ile birlikte K sayıda küme oluşturur. Dolayısıyla bu metoda ayrıca K-medoids metoduda denir. SOM metodu bir çeşit denetimsiz olarak sinir ağı yapısından oluşmaktadır. Genellikle veri içerisindeki kümeleme ve sınıflandırma işlerindeki gizli biçimleri bulmak için kullanılmaktadır. Bu ağların temel özelliklerinden biri, oluşturulan kümeleri topolojik olarak sıralamasıdır. Yani veri kümelerini topolojik olarak sıralayarak bir grid oluşturur ve küme nesnelere grid bölgelerinin komşuluğu ile eşlenir [8].



Şekil 4. Kümeleme tabanlı segmentasyon örnekleri. (a) Orijinal Görüntü, (b) K-means, (c) PAM, (d) SOM [8]

Sinir ağları tabanlı teknikleri ise denetimli(supervised) ve denetimsiz(unsupervised) metodlar olarak ikiye ayırabiliriz. Denetimli metodlar uzman insan müdahalesi gerektirirken, denetimsiz metodlar gerektirmez. Bu iki kategorideki yapay sinir ağları metodlarının sınıflandırılması aşağıdaki gibi verilmiştir [9].

❖ Denetimli Teknikler

- İleri Beslemeli Sinir Ağları (Feed-Forward Neural Networks) [10,11]
- Geriye Yayılımlı Sinir Ağları (Back Propagation Neural Networks) [12]
- Kademeli Korelasyon Sinir Ağları (Cascade Correllation Neural Networks) [13]

❖ Denetimsiz Teknikler

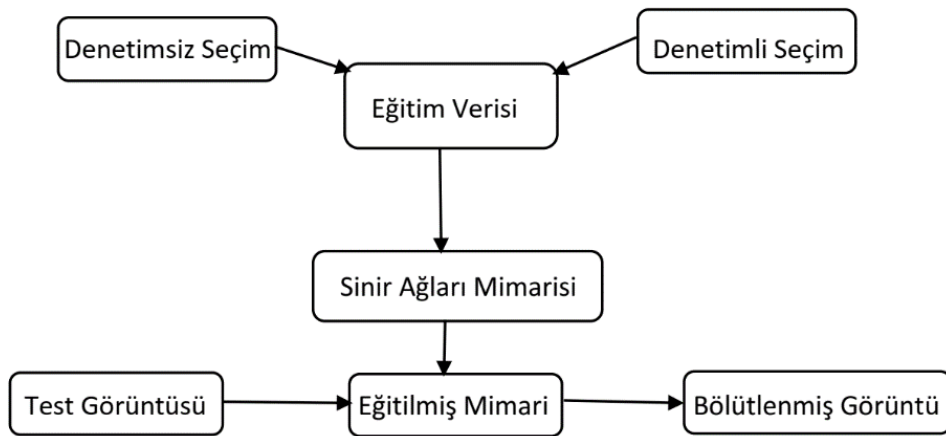
- Kısıtlı Yeterlikte Sinir Ağları (Constraint Satisfaction Neural Networks) [14]
- Darbe Bağlı Sinir Ağları (Pulse Coupled Neural Networks) [15]
- Salınımlı Sinir Ağları (Oscillatory Neural Networks) [16]

❖ Hibrit Teknikler

- Hopfield Sinir Ağları (Hopfield Neural Networks) [17]
- Kohonen Sinir Ağları (Kohonen Neural Networks) [18]

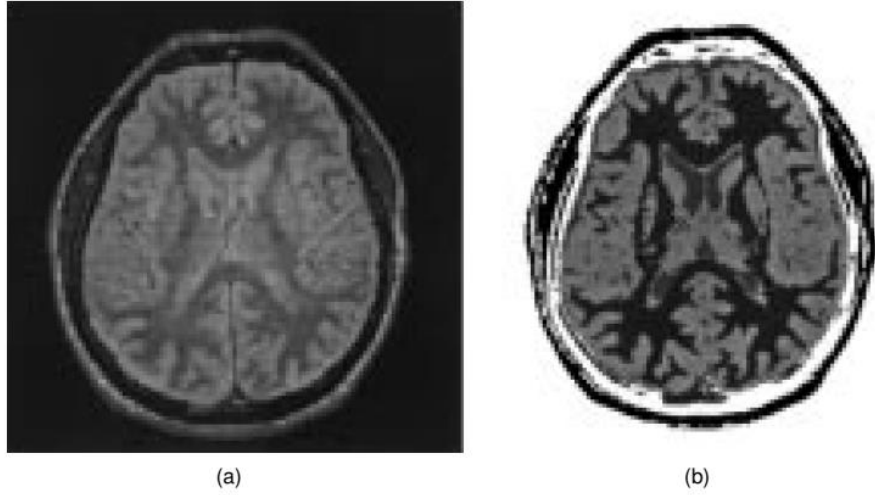
Denetimli(supervised) metodlar bir uzman yardımına ihtiyaç duyar, bunun anlamı görüntü bölütleme için seçilecek eğitim veri setini uzman kişi seçer. Denetimsiz(unsupervised) metodlarda ise bu işlem yarı veya tam otomatiktir. Eğer yarı otomatikse, yani zaman zaman uzman kişi katkısına gerek duyuluyorsa bu metodlar hibrittir diyebiliriz.

Denetimli metodlarda eğitim setini oluşturmak için uzman kişi eğitim görüntülerini seçerek onları manuel olarak belirli sayıda alt bölgelere ayırarak segmentasyonu sağlar. Bu eğitim görüntüleri sinir ağları mimarisine gönderilerek eğitimi yapılır. Daha sonra herhangi bir test görüntüsü girdisi alınarak görüntü bölütlenmiş olur. Denetimsiz metodların bundan farkı, eğitim seti uzman kişi tarafından değil otomatik oluşturulur. Aşağıda bunu anlatan bir diagram görülür:



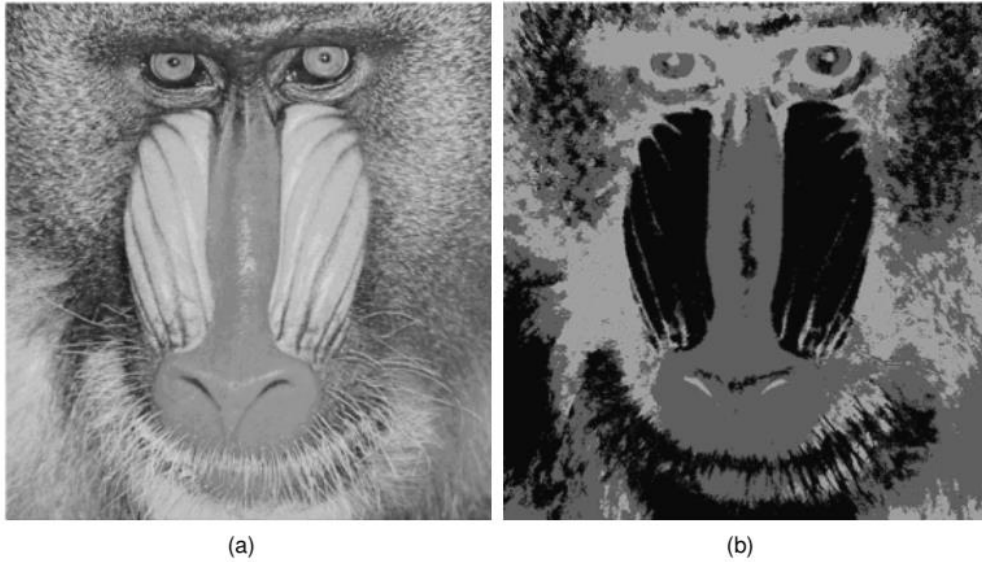
Şekil 5. Sinir ağları ile bölütleme

Yapay sinir ağları ile yapılan bölütlemeye bir örnekte aşağıdaki figürde görülmektedir. Burada back propagation tekniği ile bölütleme yapılmıştır. Beynin orijinal görüntüsü ile bölütlenmiş şekli görülmektedir.



Şekil 6. Beyin MR görüntüleri, tümör bölütleme (a) Orijinal Görüntü, (b) Back Propagation [19]

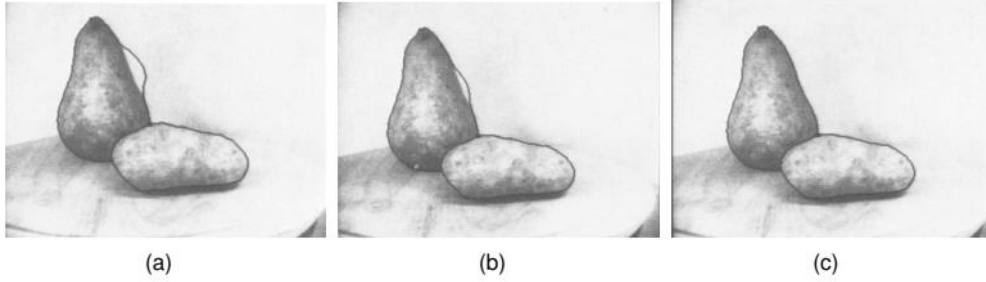
Model tabanlı bölütleme teknikleri aslında matematik istatistik modeller kullanılarak görüntü bölütlemidir. Burada en bilinen metodlar Markov Random Field Model (MRF) ve Gaussian Markov Random Field Model (GMRF) dir. Bu modellerde görüntünün bilinmeyen sayıda bölgelerden oluştuğu farzedilir ve her biri bağımsız Markov Random bölgeleri olarak modellenir. İlk olarak bölgelerin en mümkün olanları kabaca bölütlenir. Bu görüntüyü pencerelere bölerek yapılabilir. Özelliklerin veya model parametrelerin hesaplanması ile birbirine çok yakın sonuçlar veren pencereler birleştirilir. Bu modellerin genellikle pürüzlü veya gürültülü görüntülerde iyi sonuçlar verdiği görülmüştür [20,21]. Aşağıdaki figürde model tabanlı bölütlemeye bir örnek görülmektedir:



Şekil 7. Model tabanlı görüntü bölütleme (a) Orijinal görüntü (b) MRF [22]

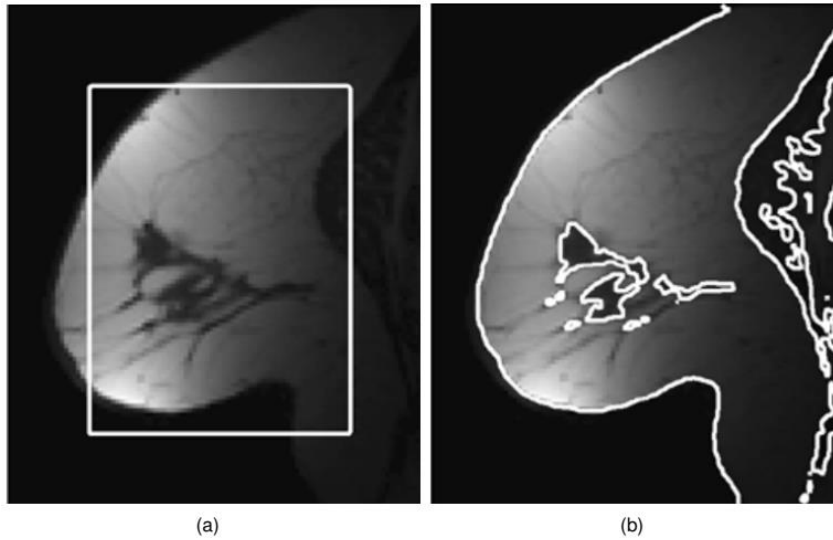
Son olarak inceleyeceğimiz segmentasyon metodları kısmi diferensiyel denklem (PDE) tabanlı segmentasyon metodlarıdır. PDE tabanlı segmentasyon metodlarının başında Aktif Kontur Model (Snakes) gelmektedir. Bu model ilk olarak 1987 yılında ortaya çıkmıştır. Bu modelin temel amacı görüntü bölütleme

problemini bir kısmi diferensiyel denklem çerçevesinde ele almaktır. Verilen bir eğrinin hareketi, kısmi diferensiyel denklemlerin bir yüzey veya görüntü üzerindeki çözümü olmaktadır. Dolayısıyla aktif konturlar görüntü içerisinde tanımlanan eğrilerdir ve kendi iç güçleri ile birlikte görüntü verisinden çıkarılan dış güçlerin etkisiyle hareket ederler [23]. Aşağıdaki figürde bu modele örnek verilmiştir.



Şekil 8. Aktif kontur model (a) Dışa doğru çekilmiş eğri (b) Nesneyi sarmaya başlayan eğri (c) Nesneyi sarmış eğri [23]

Başlangıçta bir aktif kontur enerji minimizasyon kriterine göre istenen konumlara doğru evrilen parametrize edilmiş bir eğri veya yüzeydir. Bu kontur modeli başlangıç koşullarına ve nesne sınırlarına uzaklıklarına duyarlı olması modelin dezavantajlarından biridir. Bu nedenle daha iyi bir aktif kontur modeli Level Set Metodu olarak bilinen modeldir. Level Set Metodu'nun temel fikri eğrileri veya yüzeyleri daha yüksek boyutlu bir hiperyüzeyin sıfır seviye kümesi olarak temsil edilmesidir. Dolayısıyla iki boyutlu bir yüzeydeki kapalı bir eğri üç boyutlu sürekli bir uzaydaki yüzeyin sıfır seviyesindeki eğrisi olabilir. Bu metod genellikle diğer modele göre daha doğru sayısal uygulamalar sunmasının yanında topolojik değişimlere kolayca adaptasyon sağlamaktadır [24,25].



Şekil 9. PDE tabanlı görüntü bölütleme. (a) Başlangıç eğrisi ile verilen original görüntü, (b) Level set metodu ile bölütlenmiş görüntü [24]

Aktif kontur metodları genellikle görüntü üzerinde bir başlangıç eğrisi alınarak görüntü üzerindeki nesne sınırlarına doğru bu eğrinin hareketini ve sonunda nesnelere bölütlenmesini ele alır. Bu method kısmi diferensiyel denklem ile formüle edilir. Önceki çalışmalar gösteriyor ki aktif konturlar parametrik ve geometrik olarak formüle edilmiştir.

Parametrik aktif konturlar, lagrange formülleri içerisinde parametrize edilmiş kontur eğrilerinin deformasyonu ile nesne sınırlarını belirlemesi modelidir [26]. Matematik olarak, bir parametrik aktif kontur, zamanla değişen

$$\mathcal{C}(s, t) = [x(s, t), y(s, t)], s \in [0, 1], t \in \mathbb{R}^+ \quad (2)$$

şeklinde ifade edilen bir fonksiyondur. Burada s bir yay uzunluğu, t ise zamandır. Bu fonksiyonaşağıdaki modelle yönetilir:

$$\gamma \mathcal{C}_t = \mathcal{F}_{int} + \mathcal{F}_{ext} \quad (4)$$

Bu denklemde \mathcal{C}_t , \mathcal{C} 'in zamana göre kısmi türevi, $\gamma \mathcal{C}_t$ sönmüleme kuvveti, $\gamma \geq 0$ rasgele bir sabit, \mathcal{F}_{int} iç kuvvet, \mathcal{F}_{ext} dış kuvvettir. Burada sönmüleme kuvveti ile iç ve dış kuvvetler sifıra ulaştığında kontur eğrisi hedefine ulaşır duracaktır.

Geometrik aktif konturlara gelince, bu model temelde eğri evrimi ve seviye seti metoduna dayanır. Eğriler eğri parametrizasyonundan bağımsız olarak geometrik ölçümlerle hareket ederler. Matematik olarak, $\mathcal{C}(x, t)$ iki boyutlu skalar bir seviye seti fonksiyonu olsun. Burada "sıfır" seviyesi bir aktif kontur tanımlar. Buna geometrik aktif kontur denir. Bu geometrik aktif kontur eğrisi aşağıdaki formül ile hareket eder.

$$\mathcal{C}_t = \varepsilon(\kappa + c)|\nabla \mathcal{C}| \quad (5)$$

Burada κ eğrilik, c bir sabittir. ε ise:

$$\varepsilon = \frac{1}{1 + |\nabla(\mathcal{G}_\sigma * I)|} \quad (6)$$

I görüntüsüne ait bir kenar potansiyel fonksiyonudur. $\nabla(\mathcal{G}_\sigma * I)$, \mathcal{G}_σ gauss çekirdeği ile I görüntüsünün konvolüsyonudur [27]. Seviye eğrisi modelindeki $\varepsilon(\kappa + c)$ ise \mathcal{C} seviye setinin normalleri boyunca hareket hızını tanımlar, κ eğrinin düzlüğünü, c ise büzülme ve açılmasını ifade eder.

Parametrik ve geometrik aktif kontur kavramlarına değindikten sonra, aktif kontur kavramı aslında bir enerji modelini çözen evrilen eğrilerin kavramıdır. Dolayısıyla bu konseptde bir eğri Riemann kurallarına göre veya Finsler kurallarına göre evrilir. Bu demektir ki, bir eğri konum bilgilerine göre evrilebilir veya hem konum hem yön bilgilerine göre evrilebilir. Özetle bir I görüntüsü içerisindeki enerjiyi minimize \mathcal{C} eğrileri, f yerel maliyet fonksiyonu olmak üzere, Riemann anlamında bir eğri evrimini:

$$E_{Riemann}(\mathcal{C}) = \int_{\mathcal{C}} f(I(\mathcal{C}(s))) ds \quad (7)$$

integralini minimize eden \mathcal{C} eğrilerinin hareketi ile incelerken, eğer f yöne bağımlı bir yerel maliyet fonksiyonu ise Finsler anlamında bir eğri evrimini:

$$E_{Finsler}(\mathcal{C}) = \int_{\mathcal{C}} f(I(\mathcal{C}(s)), I(\dot{\mathcal{C}}(s))) ds \quad (8)$$

integralini minimize eden \mathcal{C} eğrilerinin hareketi ile inceler.

II. RIEMANN ANLAMINDA EĞRİLERİN EVRİLMESİ

Bu bölümde ilgilendiğimiz problemlerden birisi objelerin etrafını sınırlayan en kısa kapalı eğrilerdir, bu eğrilere teminolojide geodezikler denilir. Öncelikler analitik olarak bu eğri sınıflarının modeli oluşturulup bu model nümerik olarak hesaplanacaktır. Nümerik hesaplardan elde edilen deney sonuçları detaylı olarak incelenecek ve sonra doğal parametreler optimize edilip en iyi neticeye ulaştırılacaktır.

Objeye çevresini sınırlayan en kısa eğrilere geodezikler dendiğini belirtmiştik. Geodezik aktif kontur kavramı ise en kısa kontur eğrileri olarak açıklanabilir. Matematikte x, y ve z koordinatlarında bir \mathcal{C} fonksiyonu yüzey belirterek, yüzeyin seviyesindeki veya yükseltisindeki yatay bir düzlemlerle kesişimi o düzlem üzerinde bir eğri koleksiyonu oluşturur, bu koleksiyona kontur eğrileri denilir ve aşağıdaki şekilde özetlenebilir:

$$\mathcal{C}(x, y) = z \quad (9)$$

Buradan görüldüğü gibi matematiksel olarak objenin veya üç boyutlu bir yüzeyin belli bir seviyede iz düşümlerini çizen eğrilerdir. Bilgisayarlı görme açısından baktığımızda objeleri sınırlayan kontur eğrileri görüntüdeki nesne kenarları olarak düşünülüp görüntüdeki yüksek ve alçak enerji geçişlerini ifade edebilir. Bu nedenle aktif kontur kavramı bir sebeple enerji ile hareket eden kontur eğrileri olarak düşünülüp obje sınırlarında ise optimum enerji ile objeyi çepeçevre saran enerji eğrileri olarak modellenmiştir. Bu modeli $\mathcal{C}(s) = (x(s), y(s))$ şeklinde bir parametrik eğri olmak üzere, bu eğrinin enerjisi aşağıdaki şekilde modellenmiştir [23]:

$$E = \int_0^1 E_{int}(\mathcal{C}(s)) + E_{image}(\mathcal{C}(s)) + E_{ext} ds \quad (10)$$

Burada E_{int} eğriyi büken iç enerjisi, E_{image} görüntüye neden olan enerji, E_{ext} eğri bükülmesini sınırlayan güçlerin enerjisini ifade eder.

Bu modelden sonra geodezik kavramı ile Geodezik Aktif Konturlar denen model ilk defa [28] tarafından sunulmuştur. Temel olarak zamanla evrilen ve görüntü içerisindeki objeleri saran en kısa kapalı eğriler ya da konturlar, hareket sırasında doğal olarak parçalanabilir ve daha sonra birleşerek nesne sınırlarını sarabilirler. Minimal seviye eğrileri veya geodezik aktif konturlar Riemann uzayındaki bir metriği görüntünün seviye eğrileri ile tanımlarlar.

Bu modelin $\mathcal{C}(s): [0,1] \rightarrow R^2$ eğrisi ile ilgili enerji tanımı, $I: [0, a] \times [0, b] \rightarrow R^+$ nesnelere sınırlarını belirlemek istediğimiz görüntü ve $\lambda_1, \lambda_2, \lambda_3$ pozitif sabitler olmak üzere aşağıdaki gibidir:

$$E(\mathcal{C}) = \lambda_1 \int_0^1 |\mathcal{C}'(s)|^2 ds + \lambda_2 \int_0^1 |\mathcal{C}''(s)|^2 ds - \lambda_3 \int_0^1 |\nabla I(\mathcal{C}(s))| ds \quad (11)$$

Bu integraldeki ilk iki terim iç enerji olarak bilinir ve konturların düzgünlüğünü kontrol eder. Üçüncü terim ise dış enerji olarak bilinir ve nesneye doğru yönelen konturları kontrol eder. Burada \mathcal{C} eğrisi E enerjisini minimize etmesi gerekir. Burada açık olarak görünen problemlerden birisi parametrelerin kontrolü olarak görülmektedir, öyle parametreler seçilmeli ki hem enerjisi minimize etmeli hem de düzgünlük ve konturların nesneye yaklaşımını temin etmelidir.

Bu denklemde $\lambda_2 = 0$ seçilerek ikinci dereceden süreksizlikler göz ardı edilerek kenarlar oluşturulmuş olur. Böylece eğri içerisinde belirli bir düzgünlük korunurken eğri maksimum kenar noktalarına yerleştirilmiş olur. Genelleştirilmiş bir kenar bulucu $\varepsilon: [0, +\infty) \rightarrow R^+$ azalan bir fonksiyon olarak tanımlansın. $-\lambda_3 |\nabla I| = \varepsilon(|\nabla I|)^2$ yazılarak genelleştirilmiş bir enerji ifadesi aşağıdaki şekilde yazılabilir.

$$\begin{aligned} E(\mathcal{C}) &= \lambda_1 \int_0^1 |\mathcal{C}'(s)|^2 ds + \lambda_3 \int_0^1 \varepsilon(|\nabla I(\mathcal{C}(s))|)^2 ds \\ &= \int_0^1 E_{int}(\mathcal{C}(s)) + E_{ext}(\mathcal{C}(s)) ds \end{aligned} \quad (12)$$

Bu eşitliği minimize eden \mathcal{C} eğrisini bulmak için Maupertuis Prensibi [29]'nden yararlanacağız. Bunun için

$$\mathcal{U}(\mathcal{C}) = -\lambda_3 \varepsilon(|\nabla I(\mathcal{C})|)^2 \text{ ve } \lambda_1 = \frac{m}{2}, m = sbt \text{ olsun, Maupertuis Prensibi'ne göre } \mathcal{C}(s) = \mathcal{C} \text{ ve}$$

$p(s) = m\mathcal{C}'(s)$ eğrileri $\int_0^1 \mathcal{L}(\mathcal{C}(s)) ds = \int_0^1 (p\mathcal{C}' - \mathcal{H}) ds$ enerji integralini \mathcal{H} enerji seviyesinde minimize eder. Buradaki $\mathcal{H} = \frac{p^2}{2m} + \mathcal{U}(\mathcal{C})$ Hamiltonian ve $\mathcal{L} = \frac{m}{2} |\mathcal{C}'|^2 - \mathcal{U}(\mathcal{C}) = p\mathcal{C}' - \mathcal{H}$ Langrangian'dır [28].

Burada $\mathcal{H} = \frac{p^2}{2m} + \mathcal{U}(\mathcal{C})$ eşitliğinden $|p| = \sqrt{2m(\mathcal{H} - \mathcal{U}(\mathcal{C}))}$ elde edilerek, bir sabit $E_0 = E_{int} - E_{ext}$ enerji seviyesinde;

$$\int_0^1 \mathcal{L}(\mathcal{C}(s)) ds = \int_0^1 (p\mathcal{C}' - E_0) ds = \int_0^1 p\mathcal{C}' ds - \int_0^1 E_0 ds \quad (13)$$

yazılabileceğinden $\int_0^1 E_0 ds$ sabit olduğundan, $\int_0^1 p\mathcal{C}' ds$ integralde minimize edilmiş olur. Böylece,

$$\begin{aligned} \int p C' ds &= \int \langle p, C' \rangle ds = \int |p| |C'| ds \\ &= \int \sqrt{2m(E_0 - U(C))} \sqrt{\delta_{ij} C'_i C'_j} ds \end{aligned} \quad (14)$$

elde edilir. Ayrıca

$$\int \langle p, C' \rangle ds = \int \sqrt{g_{ij} C'_i C'_j} ds \quad (15)$$

olur, (14) ve (15) denklemlerinin eşitliğinden,

$$g_{ij} = 2m(E_0 - U(C)) \delta_{ij} \quad (16)$$

bulunur. Bunun manası şudur, enerji minimizasyon problem, Riemann uzayında uygun bir geodezik eğri bulma ile eşdeğerdir. Yani enerji problemi verilen noktalar arasında minimum mesafeyi g_{ij} metriği ile ölçülerek bulunan bir eğrinin bulunması ile eşdeğerdir.

Enerjiyi minimuma indiren bir eğri bulma problemi, görüntü üzerindeki enerjilerle birlikte düşünüldüğünde, görüntüdeki enerji geçişleri yüksek hızlı ya da alçak hızlı olabilir. Yüksek hızlı bir enerji değişimi var ise oralarda keskinlikler veya kenarlar vardır diyebiliriz. Dolayısıyla bu kenarların her bir noktası en zamanda ilerlerken, en kısa yolu yaparlarsa bir çözüme ulaşılabilir. Bu tamda Fermat İlkesi'nin tanımıdır. Dolayısıyla Fermat'ın metrik tanımına göre $g_{ij} = \frac{1}{c^2(I)} \delta_{ij}$ olarak seçilebilir. Burada $c(I)$, I görüntüsü üzerindeki ışık geçişlerinin hızı olarak düşünüldüğünde, yukarıda aldığımız ε fonksiyonu ile $c(I) = \frac{1}{\varepsilon(\nabla I)}$ şeklinde tanımlanabilir [30] [31] [32]. Böylece (15) eşitliği;

$$\int \sqrt{g_{ij} C'_i C'_j} ds = \int \sqrt{\varepsilon^2(\nabla I) \delta_{ij} C'_i C'_j} ds \quad (17)$$

olur. Buradan enerji minimizasyon problemi;

$$\int_0^1 \varepsilon(\nabla I(C(s))) |C'(s)| ds \quad (18)$$

Integralinin minimizasyonu problemine dönüşmüş olur. C eğrisinin öklidyen olarak uzunluğu $L = \int |C'(x)| dx = \int ds$ şeklinde ifade edilir ve buradaki ds öklidyen yay uzunluğu veya öklidyen metrik olarak bilinir. κ , eğrinin öklidyen eğriliğini ve \mathcal{N} eğrinin herhangi bir noktadaki normalini temsil etsin. Böylece eğri akışı $C_t = \kappa \mathcal{N}$ şeklinde ifade edilebilir. Bunun için öncelikle q herhangi bir parametre olmak üzere $C(q) = C(s, t)$ olmak üzere, $\varepsilon(C) = \varepsilon(|\nabla I(C)|)$ ile gösterilsin, (18) eşitliği, L_R Riemannian eğri uzunluğu olmak üzere,

$$\begin{aligned} L_R(C) &= \int_0^1 \varepsilon(\nabla I(C(q))) |C'(q)| dq = \int_0^L \varepsilon(\nabla I(C(q))) ds \\ &= \int_0^1 \varepsilon(C(s, t)) C_q(s, t) dq \end{aligned} \quad (19)$$

Burada şunu söyleyebiliriz, gradyan minimizasyonunu düşünerek, yani eğri akışının minimizasyonu hesaba katılarak, en dik azalış (steepest descent) metoduna göre bir diferensiyel hesap yapılacağından, (19) denkleminde her iki tarafın türevi alınarak;

$$\frac{d(L_R(C(q)))}{dt} = \int_0^1 \frac{d(\varepsilon(C(s, t)))}{dt} |C_q(s, t)| dq + \int_0^1 \varepsilon(C(s, t)) \frac{d(|C_q(s, t)|)}{dt} dq \quad (20)$$

elde edilir. \mathcal{T} , C eğrisinin birim teğeti olmak üzere, $\frac{d(|C_q(s, t)|)}{dt} = \mathcal{T}(s, t) C_{qt}(s, t)$ ve $\frac{d(\varepsilon(C(s, t)))}{dt} = \nabla \varepsilon(C) C_t(s, t)$ olduğundan,

$$\begin{aligned} \frac{d(L_R(\mathcal{C}(q)))}{dt} &= \int_0^1 (\nabla \varepsilon(\mathcal{C}(s, t)) \cdot \mathcal{C}_t(s, t)) |\mathcal{C}_q(s, t)| dq \\ &+ \int_0^1 \varepsilon(\mathcal{C}(s, t)) (\mathcal{J}(s, t) \mathcal{C}_{qt}(s, t)) dq \end{aligned} \quad (21)$$

elde edilir. Burada sağ taraftaki ikinci integralde kısmi integrasyon uygulanırsa;

$$\begin{aligned} \frac{d(L_R(\mathcal{C}(q)))}{dt} &= \int_0^1 (\nabla \varepsilon(\mathcal{C}(s, t)) \mathcal{C}_t(s, t)) |\mathcal{C}_q(s, t)| dq \\ &- \int_0^1 (\varepsilon(\mathcal{C}(s, t)) \mathcal{J}(s, t))_q \mathcal{C}_t(s, t) dq \\ &= \int_0^1 [(\nabla \varepsilon(\mathcal{C}) \mathcal{C}_t) |\mathcal{C}_q| - (\nabla \varepsilon(\mathcal{C}) \mathcal{C}_q) (\mathcal{J} \mathcal{C}_t) - \varepsilon(\mathcal{C}) \mathcal{J}_q \mathcal{C}_t] dq \end{aligned} \quad (22)$$

Bulunur ki, burada s , $\mathcal{C}(t)$ eğrisinin yay uzunluğu, $\mathcal{J}_q = \mathcal{J}_s |\mathcal{C}_q|$ ve $\mathcal{J}_s = \kappa \mathcal{N}$ olduğundan,

$$\frac{d(L_R(\mathcal{C}(q)))}{dt} = \int_0^{L(\mathcal{C}(q))} [\nabla \varepsilon(\mathcal{C}) - (\nabla \varepsilon(\mathcal{C}) \mathcal{J}) \mathcal{J} - \varepsilon(\mathcal{C}) \kappa \mathcal{N}] \mathcal{C}_t ds \quad (23)$$

elde edilir. $\nabla \varepsilon = a\mathcal{J} + b\mathcal{N}$ olduğundan $a = \nabla \varepsilon \mathcal{J}$, $b = \nabla \varepsilon \mathcal{N}$ dir ve $\nabla \varepsilon - (\nabla \varepsilon \mathcal{J}) \mathcal{J} = \nabla \varepsilon - a\mathcal{J} = b\mathcal{N} = (\nabla \varepsilon \mathcal{N}) \mathcal{N}$ olur. Böylece;

$$\frac{d(L_R(\mathcal{C}(q)))}{dt} \Big|_{t=0} = \int_0^{L(\mathcal{C}_0)} [(\nabla \varepsilon(\mathcal{C}_0) \mathcal{N}) \mathcal{N} - \varepsilon(\mathcal{C}_0) \kappa \mathcal{N}] \mathcal{C}_t(0) ds \quad (24)$$

bulunur. Burada $L_R(\mathcal{C})$, \mathcal{C} 'nin bir fonksiyoneldir, dolayısıyla (24) integrali L_R 'nin $\mathcal{C} = \mathcal{C}_0$ 'daki Gateaux türevidir [33]. Yani, en dik iniş (steepest-descent) metoduna göre, $L_R(\mathcal{C})$ 'nin lokal minimumu olan başlangıç \mathcal{C}_0 eğrisini bulmak için aşağıdaki diferensiyel eşitlikteki eğri ailesini incelemek gerekecektir. Steepest-descent metodunun detaylarına [34] makalesinden ulaşılabilir.

$$\mathcal{C}_t = \varepsilon(\mathcal{C}) \kappa \mathcal{N} - (\nabla \varepsilon(\mathcal{C}) \mathcal{N}) \mathcal{N} \quad (25)$$

III. SAYISAL HESAPLAR VE DENEYSEL SONUÇLAR

\mathcal{C} eğrisi, $u: [0, a] \times [0, b] \rightarrow R$ fonksiyonunun bir seviye eğrisi olduğu düşünülürse, \mathcal{C} bir \mathcal{N} normal boyunca evrilir. Yani, bir Σ fonksiyonu için, $\mathcal{C}_t = \Sigma \mathcal{N}$ yazılabilir. Bu nedenle u fonksiyonu $u_t = \Sigma |\nabla u|$ ile yeniden şekillenir. u 'nun evrimi aşağıdaki denklemle yazılabilir [35]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= |\nabla u| \div \left(\varepsilon(I) \frac{\nabla u}{|\nabla u|} \right) = \varepsilon(I) |\nabla u| \div \left(\varepsilon(I) \frac{\nabla u}{|\nabla u|} \right) + \nabla \varepsilon(I) \nabla u \\ &= \varepsilon(I) |\nabla u| \kappa + \nabla \varepsilon(I) \nabla u \end{aligned} \quad (26)$$

denklemi $\varepsilon = \frac{1}{1+|\nabla u|^p}$ ve c hız sabiti, \hat{I} düzgünleştirilmiş bir I görüntüsüdür, c negative seçildiğinde eğri içeri doğru evrilir, c pozitif seçildiğinde eğri dışarı doğru evrilir ve sıfırdan büyük olmalıdır [27,36,37]. Buna göre (26) denklemi;

$$\frac{\partial u}{\partial t} = \varepsilon(I) (c + \kappa) |\nabla u| + \nabla u \nabla \varepsilon \quad (27)$$

olduğundan, \mathcal{C} eğrileri,

$$\mathcal{C}_t = \varepsilon(I) (c + \kappa) \mathcal{N} - (\nabla \varepsilon(I) \mathcal{N}) \mathcal{N} \quad (28)$$

denklemi ile zamanla evrilirler.

Bir $I: [0, a] \times [0, b] \rightarrow R^+$ görüntüsü üzerinde C eğrilerinin evrimini gözlemlemek için, $C: [0,1] \rightarrow R^2$ eğrileri görüntü üzerinde bir $u: [0, a] \times [0, b] \rightarrow R$ fonksiyonunun seviye eğrileri olduğundan, u 'nun I içindeki hareketini gözlemlemek yeter. Dolayısıyla $u = (x(s, t), y(s, t))$ 'nin x ve y yönündeki sayısal türevleri;

$$u_x = \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2\Delta t} \quad (29)$$

$$u_y = \frac{u(x_i, y_{j+1}) - u(x_i, y_{j-1})}{2\Delta t} \quad (30)$$

$$u_{xx} = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{\Delta t^2} \quad (31)$$

$$u_{xy} = \frac{u(x_{i+1}, y_{j+1}) + u(x_{i-1}, y_{j-1}) - u(x_{i+1}, y_{j-1}) - u(x_{i-1}, y_{j+1})}{4\Delta t^2} \quad (32)$$

şeklinde. Bu sayısal türevleri kullanarak (28) denkleminin sayısal çözümü için aşağıdaki algoritma izlenir.

Algoritma : Görüntü Üzerinde Eğri Evrimi Algoritması

- 1 Üzerine çalışacağımız bir örnek $I(x, y)$ görüntüsü yüklenir;
 - 2 Görüntü $\mathcal{G}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$ gauss filtresi ile düzgünleştirilir. Düzgünleştirilmiş görüntü $\hat{I} = I \otimes \mathcal{G}$;
 - 3 $\varepsilon = \frac{1}{1 + |\nabla \hat{I}|^\rho}$ kenar fonksiyonunu hesaplanır;
 - 4 Görüntü içinde her bir iterasyonda değişecek u maskı başlangıç eğrisi ile oluşturulur;
 - 5 **for her bir iterasyon do**
 - 6 | Eğrilik $\kappa = \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}$ ifadesinden κ hesaplanır[38];
 - 7 | $u = u + \Delta t u_t$ şeklinde güncellenir;
 - 8 **end**
-

Algoritma 1. Görüntü üzerinde bir Riemann modeli ile eğri evrimi

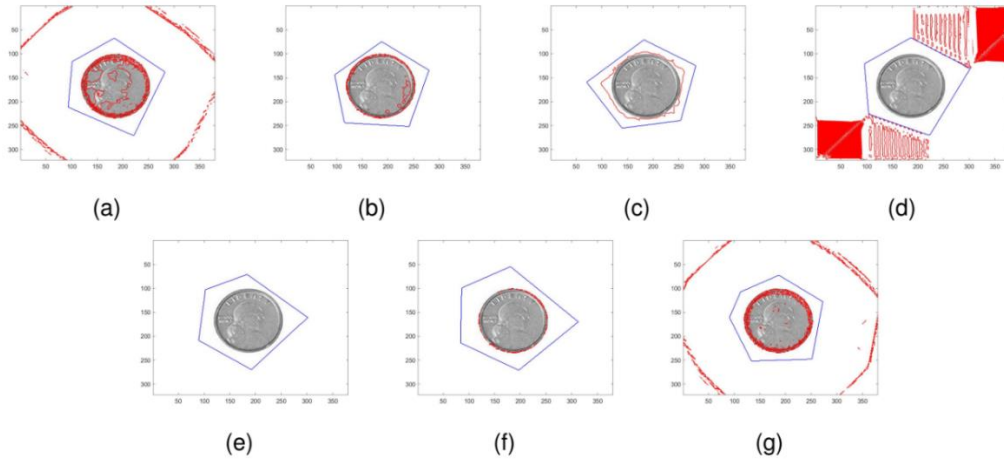
Bu algoritma ile Tablo 1' de verilen deney setleri eğri evrimini nasıl etkileyecek sonuçlarını inceleyeceğiz. Tablo 1'de görüldüğü gibi kullanıcı tarafından girilen 13 adet deney bulunmaktadır. Bu deneyler, eğri evrimi modelinde kullanıcı tanımlı girilen parametrelere göre eğri değişimini inceleyecektir. Bu deneylerin işletilmesinde MATLAB R2016b kullanılmış, adım uzunluğu 0.1 olmakla beraber 200 tekrar işletilmiştir.

Tablo 1. Deney verileri tablosu

Deney No.	ρ	c	σ	Ort. Karesele Hata
1	2	-10	1	0.019
2	2	-5	1	0.018
3	2	-1	1	0.04
4	2	0	1	0.08
5	2	1	1	0.023
6	2	5	1	0.018
7	2	10	1	0.019
8	1	-5	1	0.019
9	5	-5	1	0.019
10	10	-5	1	0.019
11	2	-5	1	0.018
12	2	-5	5	0.024
13	2	-5	10	0.023

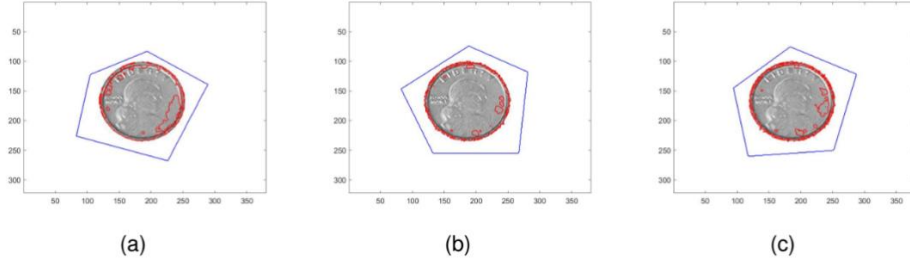
Yukarıdaki tablodan görüldüğü gibi, c evrilme hızı, σ gauss filtre varyansı, ρ kenar fonksiyonunun büyüklük derecesi parametrelerinin başlangıç eğrilerinden sonra değişimleri incelenmiştir. 1-7 deneylerinde $\sigma = 1$, $\rho = 2$ sabit tutulurken, c 'nin -10, -5, -1, 0, 1, 5, 10 değerlerine konturların cevabı incelenmiştir. 8-10 deneylerinde $c = -5$, $\sigma = 1$ sabit tutulurken, ρ 'nin 1, 5, 10 değerlerinde konturların cevabı incelenmiş ayrıca kenar görüntüleri alınmıştır. 11-13 deneylerinde $c = -5$, $\rho = 2$ sabit tutulurken σ 1, 5 ve 10 değerlerinde konturların davranışları incelenmiş olup, kenar görüntüleri ve düzgünleştirilmiş görüntüler alınmıştır.

Aşağıda Şekil 10. da c evrilme parametresinin sırasıyla -10, -5, -1, 0, 1, 5, 10 olduğu durumlarda eğrinin cevapları gözlemlenmektedir. Dikkat edilirse (a) ve (g) gibi çok küçük ve çok büyük değerlerde eğrinin parçalandığı ve resim üzerine dağıldığı gözlemlenmektedir. Bu durum eğrinin obje sarıldıktan sonra resimde ilgisiz yerlere dağılmasına neden olmuştur. Ayrıca (b) ve (c) görüntülerine bakılırsa, (b) de eğri (c) ye göre daha hızlı giderek objeyi sarmış ve parçalanmaya başlamıştır, buradan c 'nin eğrinin evrilme hızını etkilediği açıkça görülmektedir. Bu parametrenin etkisiz olduğu durum ise (d) seçeneğinde görülmektedir. Buradan eğri davranışının bozulduğu, eğrinin içe veya dışa doğru evrilme kararının olmadığı dolayısıyla rasgele hareket ettiği gözlemlenmektedir. (e), (f), (g) seçeneklerine bakılırsa eğri dışa doğru hareket ederek uzay içerisine parçalanmış halde dönmüştür, sırasıylaobjeyi parçalanmış eğriler hızına göre sarmaya çalışmıştır.



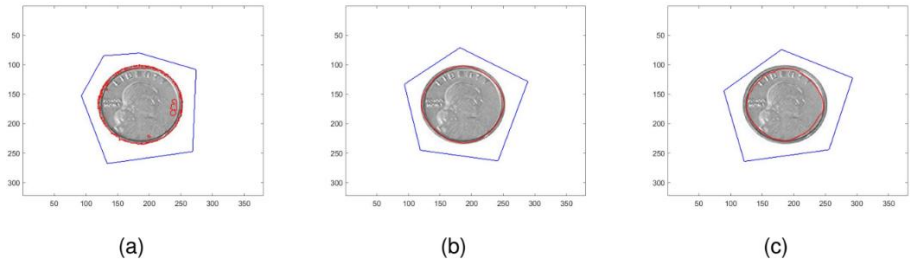
Şekil 10. c evrilme hızının değişimine göre eğri evrimi sonuçlarıdır. Sırasıyla deney 1-7 sonuçlarıdır. Mavi: Başlangıç eğrisi, Kırmızı: Sonuç

Şekil 11’de kenarları güçlendiren veya zayıflatan pozitif bir parametrenin değişimi gözlemlenmektedir. Esasen bir görüntüde görüntünün kullanım amacına göre çok fazla detay istenebilir veya daha az detayda istenebilir. Bu parametre o nedenle görüntü kullanım amacına göre kullanıcı tanımlı olarak görüntü yorumlamaya neden olur. Şekilde görüldüğü kadarıyla parametre büyüdükçe detayla ve eğri parçalanmaları artmakta, bunun tersi olarak küçüldükçe ise detayları bulma ihtimali azaldığından eğrinin hedefe yönelmesi azalacağından eğrilerin rasgele parçalanmalarına sebebiyet vermektedir. Bu nedenle makul bir ayar gerekmektedir.



Şekil 11. ρ kenar bulma parametresinin değişimine göre eğri evrimi sonuçlarıdır. Sırasıyla deney 8-10 sonuçlarıdır. Mavi: Başlangıç eğrisi, Kırmızı: Sonuç

Şekil 12’de resmi düzgünleştiren bir parametre incelenmiştir. Bu filtre varyansı arttıkça resim gürültüden kurtularak daha düzgün bir görüntü olur. Parametre azaldıkça resimdeki gürültü çoğalacağından resimde eğrilerin hareketi rasgeleşmeye başlar, dolayısıyla eğri hedefi olan kenarlar yerine resimde parçalanarak rasgele bölgelere yerleşir. Burada (a)’dan (c)’ye tamamen bu gözlemlenmektedir. (b) ile (c) ye dikkat edilirse parametrenin çok yüksek değeri görüntüde obje kenarlarını yok ederek eğriyi kenarı bulamadan evrilmesine neden olur. Dolayısıyla görüntüyü düzgünleştirmek, bir başka deyişle gürültüden arındırmak önemlidir, fakat filtrelemede aşırıya kaçılmamalıdır. O nedenle bu parametrenin resme göre optimum değerini kullanmak önemlidir.

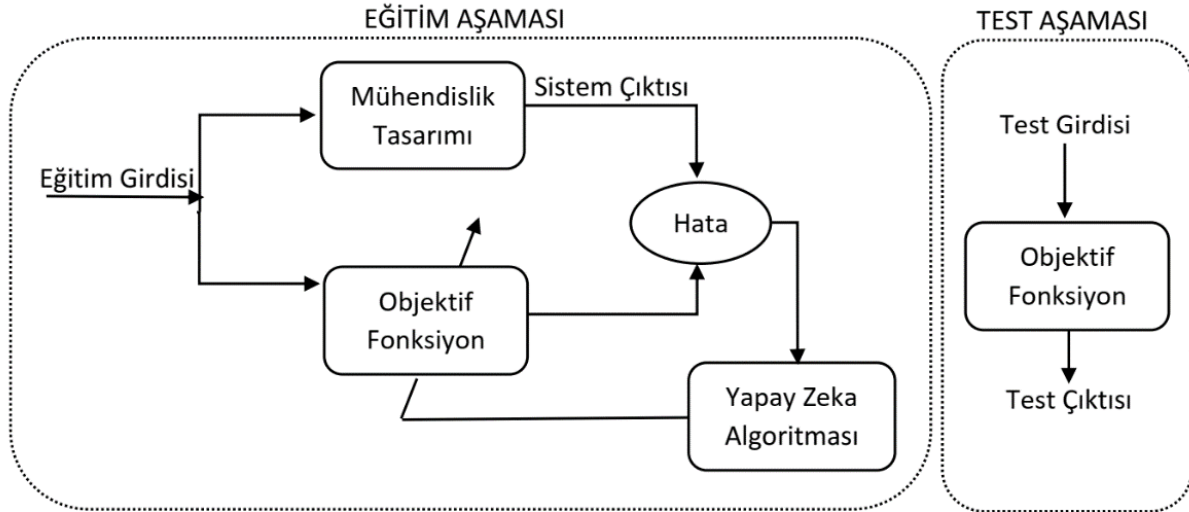


Şekil 12. σ düzgünleştirme parametresinin değişimine göre eğri evrimi sonuçlarıdır. Sırasıyla deney 11-13 sonuçlarıdır. Mavi: Başlangıç eğrisi, Kırmızı: Sonuç

Tablo 1’de kullanılan veriler kullanıcı tanımlı olarak üretilen verilerdir. Buradaki esas amaç modelde kullanıcı tarafından girilen bilgilerin en uygununu bulmaktır. Yapılan deneylerde gözlemlenmiştir ki, bu veriler resimlere göre değişkenlik gösterebilir. Dolayısı ile bu verilerin optimumunu araştırmak doğal bir sonuç olacaktır. Yani bu parametreler optimizasyon algoritmaları ile optimize edilebilir. Buradan hareketle, gelecek bölümde bu parametrelerin yapay zeka optimizasyon algoritmaları ile nasıl optimize edildiğini gözlemleyeceğiz.

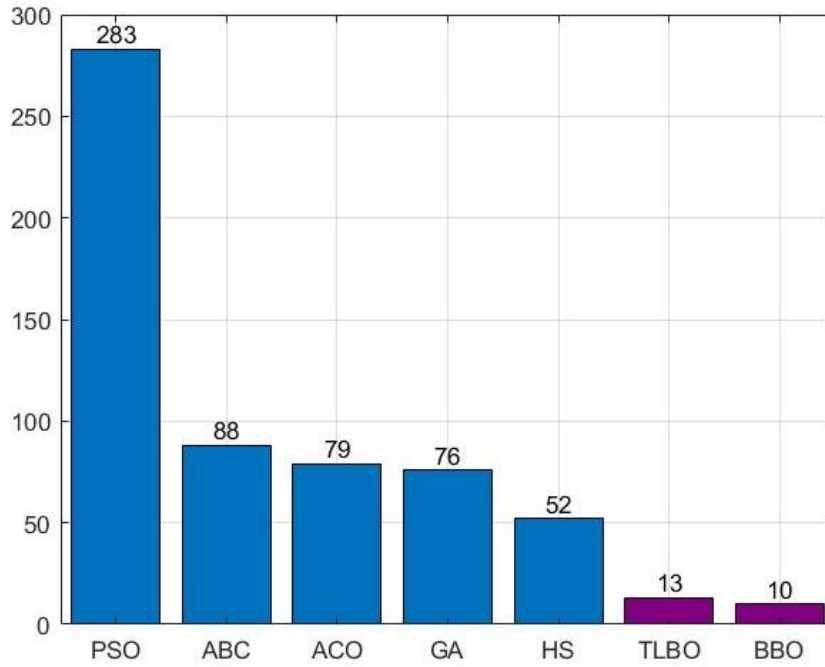
IV. OPTİMİZASYON

Önceki bölümde c, ρ, σ değerlerinin değişiminde görüntüdeki eğri hareketinin ve bölütlemenin nasıl olduğu tartışılmıştı. Bu bölümde bu değişimlerin optimum olduğu değerler neler olabilir, yapay zeka optimizasyon algoritmalarından meta-sezgisel yaklaşımlarla bu parametrelerin değişimlerinde neler olabilir, bunlar tartışılacaktır. Yapay zeka optimizasyon algoritmaları son zamanlarda kompleks optimizasyon problemlerinin çözümünde büyük rol almaktadır. Yapay zeka tabanlı algoritmalar genellikle doğal fenomenlerin, doğadaki canlıların sosyal davranışlarının benzetimi ile ortaya çıkmıştır. Dolayısıyla daha sezgisel veya meta-sezgisel adını alırlar ve sezgisel algoritmalar olarak bilinirler. Bir yapay zeka optimizasyonunda temelde iki aşama vardır, bunlar eğitim(training) ve test aşamalarıdır. Eğitim aşamasında optimizasyonunu sağlayacağımız veya optimum parametrelerini bulacağımız mühendislik probleminden veri setleri oluşturulur, bunlara eğitim seti denilir. Eğitim seti, yani birçok veri ile mühendislik problemi yapay zeka algoritması ile çözülür. Her bir veri için bir sonuç bulunur. Bulunan sonuçlardan en iyisi test aşaması için saklanır. Bu sonuç yapay zeka algoritmasının bulduğu optimum sonuçtur. Test aşamasında bulunan optimum parametreler yine mühendislik problemi için oluşturulan eğitim setinden farklı bir test seti ile test edilir. Tüm bu aşamaları anlatan diagram aşağıda görülmektedir:



Şekil 13. Yapay zeka optimizasyonu ilüstrasyonu

Optimizasyon problemlerinde optimal çözümü bulmak için bu zamana kadar bir çok algoritma geliştirilmiş, yapıları ve çözüm metodları incelenmiştir. Fakat bunlardan en ilginç olanları sezgisel ve meta-sezgisel yada sezgiüstü algoritmalar olmuştur. Algoritmaların bilim adamlarına ilham veren tarafı ise tabiatın işleyisi içerisinde kendiliğinden gelişen optimallik arayışı olmuştur. Bu algoritmaların optimal çözümü bulmak için arama prosesleri özel bir ilgi alanına yönelik yapısal işleyişleri gerçekten inanılmazdır. Bu nedenle bazı modellenen sezgisel algoritmalar en optimal çözüme veya gerçek çözüme çok kısa zamanlarda ulaşabilir. Aşağıdaki grafikte son yıllardaki sezgisel algoritmalar ile ilgili yayınların sayısı görülmektedir [39]. Bu çalışmada kullanılacak yapay zeka algoritmalarından son zamanlarda en çok yayın çıkaran ilk beş algoritma sırasıyla Particle Swarm Optimization(PSO) [40],Artificial Bee Colony(ABC) [41], Ant ColonyOptimization(ACO) [42], Genetic Algorithm(GA) [43] ve Harmony Search Algorithm(HS) [44] dir. Bu çalışmada diğer algoritmalara ek olarak daha az yayın çıkaranlar arasında yer alan Teaching Learning Based Optimization(TLBO) [45], Biogeography Based Optimization(BBO) [46] algoritmaları da kullanılacaktır.



Şekil 14. Bu çalışmada kullanılan algoritmaların yayın sayıları [39]

Parçacık sürü optimizasyonu gıda kaynakları için yiyecek arayan sürü kuşların davranışlarından esinlenmiştir, PSO'da arama ajanları yani parçacıkların her biri üç boyutlu vektör kümeleridir ve parçacığın anlık pozisyonu, önceki en iyi pozisyonu ve hızından oluşur. Ayrıca parçacık sürüsü içindeki her üye arama işlemi sırasında ulaşılabildiği komşusu hakkında bilgiye sahiptir.

Algoritma : Parçacık Sürü Optimizasyonu(Particle Swarm Optimization)

```
1 Başlangıç hız limitlerini hesapla;  
2 Başlangıç pozisyonunu belirle;  
3 Başlangıç hızını belirle;  
4 while Durdurma kriteri tamamlanmadığında do  
5   foreach parçacık do  
6     hızı hesapla;  
7     hızı sınırla;  
8     pozisyonu hesapla;  
9     if pozisyon uygun bölgede then  
10      pozisyonu değerlendir;  
11   end  
12 end  
13 Yeni hız sınırlarını hesapla;  
14 end
```

Algoritma 2. Parçacık Sürü Optimizasyonu

Arılar, yiyecek aramak veya çiftleşmek için toplu davranış sergileyen böceklerin en bilinen örnekleri arasındadır. Yapay arı kolonisi algoritmasında, arama ajanları bir arama uzayında optimal yiyecek arayan yapay bal arılarını temsil eder. Bu yiyecek kaynaklarından her biri, belirli bir optimizasyon problemi için olası çözümleri temsil eder, yiyeceğin miktarı ise kaliteyi temsil eder. Arı kolonisinin üyeleri işçi arılar, gözcü arılar ve kaşif arılar

olmak üzere üç gruba ayrılır. Bu arı grupları, arıların yiyecek ararken kullandıkları mekanizmalara göre farklı görevlere sahiptir. Örneğin işçi arılar bilinen gıda kaynaklarının çevresini keşfetme ve kovana getirme görevini üstlenirler. Kaşif arılar bilinmeyen yeni besin kaynakları için tüm araziye keşfetme görevini üstlenirler. Gözcü arılar kovana gözetirler, işçi arıların kovanda ilettikleri bilgiye göre yiyecek arayışına çıkarlar.

Algoritma : Yapay Arı Kolonisi(Artificial Bee Colony)

```
1 Başlangıç yiyecek kaynaklarını rasgele belirle;  
2 İşçi arıları bu kaynaklara rasgele gönder;  
3 while Sonlandırma kriteri oluşmadıkça do  
4   /*İşçi arıların fazı*/;  
5   for her bir işçi arı do  
6     Komşu bölgelerde yiyecek kaynağı bul;  
7     Yeni yiyecek kaynağının fitnessını hesapla;  
8     En iyi seçimi uygula;  
9   end  
10  Her bir yiyecek kaynağı için olasılıkları hesapla;  
11  /*Gözcü arıların fazı*/;  
12  for her bir gözcü arı do  
13    Yukarıda hesaplanan yiyecek kaynağı olasılığına göre, gözcü arıları kaynaklara  
14    gönder;  
15    Komşu bölgelerde yiyecek kaynağı bul; Yeni yiyecek kaynağının fitnessını hesapla;  
16    En iyi seçimi uygula;  
17  end  
18  /*Kaşif arıların fazı*/;  
19  if herhangi bir işçi arı kaşif arı olursa then  
20    Kaşif arıyı rasgele üretilen bir yiyecek kaynağına gönder;  
21  end  
22  Buraya elde edilen en iyi çözümü sakla;  
23 end  
23 En iyi çözümü çıkar;
```

Algoritma 3. Yapay Arı Kolonisi Algoritması

Doğada karıncalar rasgele hareket ederler ve uygun bir kaynak bulunduğunda arkalarında bir feromon izi bırakarak kolonilerine dönerler. Böylece karıncalar bıraktıkları feromonları veya diğer karıncaların izlediği yolu takip ederek daha önce bulunan besin kaynaklarına yönelebilirler. Doğal olarak bir karınca bir yoldan ne kadar çok gidip gelirse, feromonun dağılması içinde o kadar çok zaman gerekir. Bu nedenle daha kısa yollarda daha fazla feromon yoğunluğu vardır. Bunun anlamı şudur, eğer bir karınca koloniden besin kaynağına kısa bir yol bulursa, diğer karıncalarında bu yolu takip etme olasılığı yüksektir. Böylece kolonideki karıncalardan çoğunluğunun takip ettiği bir yol, feromon yoğunluğunu artırıp tümünün takip etmesine yol açacağından optimal yol bulunmuş olacaktır.

Algoritma : Karınca Kolonisi Optimizasyonu(Ant Colony Optimization)

- 1 Başlangıç karınca kolonisini rasgele oluştur;
- 2 Başlangıç feromon izlerini ve parametrelerini belirle;
- 3 Başlangıç kolonisinin fitness değerlerini hesapla;
- 4 Koloninin en iyi çözümünü bul;
- 5 **while** *Sonlandırma kriteri oluşmadıkça do*
- 6 | **for** *her bir karınca do*
- 7 | | Çözümü bul;
- 8 | | Bölgesel feromon izlerini güncelle;
- 9 | **end**
- 10 | Feromonu güncelle;
- 11 | En iyi karıncayı belirle;
- 12 **end**
- 13 En iyi çözümü çıkar;

Algoritma 4. Karınca Kolonisi Optimizasyonu Algoritması

Genetik Algoritma, adından da anlaşılacağı üzere doğadaki birey oluşumunda genetik yapının işlevinden esinlenerek modellenmiştir. En iyi sonucu üreten kromozomun yaşaması prensibinden yola çıkarak en iyi sonuca ulaşma metodudur. Genetik algoritmada bireyler sayı dizilerinden oluşan vektörler, bireyler topluluğunda popülasyon adı verilen matrislerdir. Bireylerdeki yani vektörlerdeki her bir bileşene gen denilir. Bireylerin problem için çözüm olup olmayacağına fitness fonksiyonu karar verir, en iyi fitness değerlerine sahip bireyler çoğalmaları için seçilir. Böylece bu bireyler çaprazlama neticesinde çocuk bireyler oluşur, bunlar yeni bireylerdir. Yeni gelişip büyümeleri yani mutasyona uğramaları neticesinde popülasyona dahil edilirler. Yeni çocukların oluşması için bu işlemler yeniden tekrarlanır.

Algoritma : Genetik Algoritma(Genetic Algorithm)

- 1 Başlangıç popülasyonunu rasgele belirle;
- 2 Her bir kişinin fitnessını hesapla;
- 3 En iyi bireyi belirle;
- 4 **while** *Sonlandırma kriteri oluşmadıkça do*
- 5 | Eşleşecek bireylerin seçimini yap;
- 6 | Yeni bireylerin oluşması için çaprazlama işlemini uygula;
- 7 | Yeni bireyleri mutasyona uğrat;
- 8 | Yeni bireylerin fitnessını hesapla;
- 9 | Yeni bireyleri eski popülasyona gönder;
- 10 | Düşük fitnesslı bireyleri ele;
- 11 **end**

Algoritma 5. Genetik Algoritma

Armoni Arama Algoritması, müzisyenlerin mükemmel armoniye bulmak amacıyla hafızalarında depolanan farklı müzik perdelerini birleştirerek bir armoni oluşturduğu söylenen, armoni doğaçlama sürecinin arkasındaki ilkelerden esinlenen metasezgisel bir yaklaşımdır. Algoritmada mükemmel uyumu bulma süreci, bir optimizasyon probleminde optimal çözümü bulmaya benzer.

Algoritma : Armoni Arama Algoritması(Harmony Search Algorithm)

```
1 Armoni hafızasını dikkate alma oranını tanımla(Harmony Memory Considering
  Rate(HMCR));
2 Ton ayarlama oranını tanımla(Pitch Adjusting Rate(PAR));
3 Rasgele armonilerle armoni belleği(Harmony Memory(HM)) üret;
4 while Maksimum iterasyona ulaşılmadıysa do
5   while Maksimum değışkene ulaşılmadıysa do
6     if (0, 1) aralığında seçilen rasgele bir sayı<HMCR then
7       Bu değışken için HM'den bir değer seç;
8       if (0, 1) aralığında seçilen rasgele bir sayı<PAR then
9         HM'den seçtiğın değere bir miktar ekleyerek ayarla;
10      end
11     else
12       Rasgele bir değer üret;
13     end
14   end
15   Yeni armoni çözümlü daha iyiyse al;
16 end
17 Güncel en iyi çözümlü bul;
```

Algoritma 6. Armoni Arama Algoritması

Öğretme Öğrenme Tabanlı Optimizasyonda, bir öğrenci grubundan en iyisi öğretmen olarak seçilir ve öğretmen diğer öğrencilerin oluşturduğu grubun ortalamasını yükseltmek için, bilgisini onlarla paylaşır. Bu aşamaya öğretmen aşaması veya fazı denir. Öğrenci fazında ise, rasgele bir öğrenci diğer öğrencilerle iletişime geçerek onların performansını artırmaya çalışır.

Algoritma : Öğrenme Öğretme Tabanlı Optimizasyon(Teaching Learning Based Optimization)

```
1 Başlangıç öğrencilerini belirle;
2 Öğrencilerin fitnesslarını hesapla;
3 while durdurma kriteri oluşmadıkça do
4   En iyi öğrenciyi öğretmen olarak seç;
5   Tüm öğrencilerin ortalamasını hesapla;
6   for her bir öğrenci do
7     /*Öğretmen Fazı*/;
8     Öğretmen çarpanını hesapla(1 yada 2 olmalı);
9     Öğrenciyi bu çarpana göre güncelle;
10    Yeni öğrencinin fitness değerini bul;
11    Eğer eski öğrenciden daha iyiyse yenisini al;
12    /*Öğrenci Fazı*/;
13    Rasgele diğerinden farklı bir öğrenci seç;
14    Yeni öğrenciyi eskileri ile kıyasla;
15    Eğer eski öğrenciden daha iyi ise eskisi ile değıştir;
16  end
17 end
```

Algoritma 7. Öğrenme Öğretme Tabanlı Optimizasyon Algoritması

Biocoğrafya canlıların coğrafi dağılımının incelenmesidir. Biocoğrafik modeller türlerin habitatlar arasında nasıl göç ettiklerini, görünümünü, adaptasyonlarını, evrimlerini ve yok oluşlarını açıklamaya çalışır. Habitatteki tür sayısı arttıkça daha kalabalık hale gelir, daha az tür habitata göç ederek başarılı bir şekilde hayatta kalabilir ve göç oranı azalır. Habitatta tür yoksa göç oranı sıfır olmalıdır. Tür sayısı arttıkça habitat kalabalık hale gelir, daha fazla tür diğer olası habitatları keşfetmek için habitatı terk edebilir ve göç hızı artar.

Algoritma : Biocoğrafya Tabanlı Optimizasyon(Biogeography Based Optimization)

- 1 Başlangıç popülasyonunu oluştur;
- 2 Bireylerin fitnesslarını bul;
- 3 **while** Durdurma kriteri oluşmadıkça **do**
- 4 | Çözümleri en iyi fitness değerlerine göre sırala;
- 5 | Dış ve iç göç oranlarının objektif fonksiyon değerlerine göre belirle;
- 6 | Çözümleri bu orana göre güncelle;
- 7 | Bireyleri mutasyona uğrat;
- 8 | Mutasyona uğrayan bireylerin fitnesslarını bul;
- 9 | Fitness değerlerine göre bireyleri sırala;
- 10 **end**
- 11 En iyi sonucu çıkar;

Algoritma 8. Biocoğrafya Tabanlı Optimizasyon Algoritması

ABC, ACO, BBO, GA, HS, PSO, TLBO algoritmaları ile parametreler eğitilerek en optimal maliyet değerleri aşağıdaki tabloda verilmiştir.

Tablo 3. Eğitimler sonucunda ortaya çıkan maliyetler

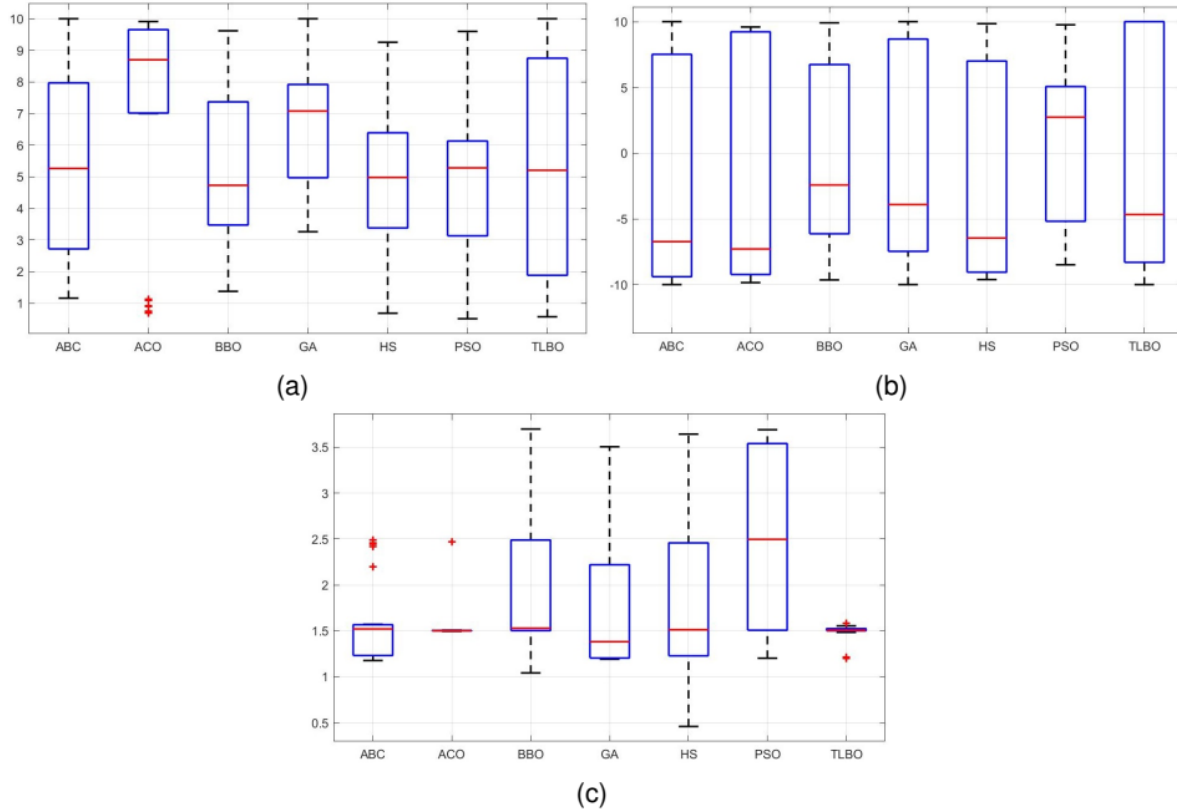
Algoritma	Ortalama maliyet	En iyi maliyet	En kötü maliyet	Standart Sapma
ABC	0.038413041	0.038384332	0.038452148	1.38E-05
ACO	0.038388627	0.038370768	0.038411458	7.45E-06
BBO	0.038416432	0.038391113	0.038452148	2.01E-05
GA	0.038413945	0.038391113	0.038452148	1.85E-05
HS	0.038416884	0.038370768	0.038445367	2.09E-05
PSO	0.038422083	0.038384332	0.038465712	2.31E-05
TLBO	0.038398573	0.03837755	0.038404677	6.51E-06

Tablodaki değerler, algoritmalar 30 eğitimden geçirildikten sonra elde edilmiştir. Aşağıdaki figürde her bir eğitimde algoritmalarda çıkan en iyi maliyet değerleri çizdirilmiştir. Tablo 3'ün ilk sütununda eğitimlerde kullanılan yedi algoritma ve iki, üç, dört ve beş sütunlarda onların eğitimleri sonucunda bu algoritmalara karşılık sırasıyla ortalama maliyet, en iyi maliyet, en kötü maliyet ve popülasyonun standart sapması verilmiştir. Tablodan ortalama maliyetin en iyi değeri ACO algoritmasında olduğu görülmektedir. Yine tablonun üçüncü sütunundan en kötü maliyetleri sıraladığımızda PSO, ABC, BBO, GA, HS, ACO, TLBO şeklinde bir sıralama görmekteyiz ki, en kötünün en iyisi olarak TLBO, ve en kötünün ikinci en iyisi olarak ACO çıkmaktadır. Özetle en iyi maliyet sıralaması ACO, HS, TLBO, ABC, PSO, GA, BBO olarak çıkarken, ortalama en iyi maliyet sıralaması ACO, TLBO, ABC, GA, BBO, HS, PSO çıkmaktadır. Dolayısıyla bu çalışmada ACO ve TLBO sonuçlarını ön planda değerlendirmek isabet olacaktır. Tüm bu eğitimlerden sonra algoritmaların bulunduğu parametre değerleri aşağıdaki tabloda verilmiştir.

Tablo 4. Eğitimler sonucunda elde edilen en iyi parametre değerleri

Algoritma	ρ	c	σ
ABC	1.15977333	-9.84715543	1.4948825
ACO	0.6852938	-5.20059159	1.49711835
BBO	3.27284251	-4.45830569	1.20512633
GA	8.5542024	9.77491109	1.5023567
HS	0.78781235	-5.94257386	1.49646862
PSO	5.1788286	-7.16793581	2.49949652
TLBO	0.87238302	-6.60224216	1.48399116

Tüm algoritmaların eğitimlerdeki performansını istatistiki olarak ilk eğitimden sonuncuya kadar geçen skalada veren kutu grafikleri aşağıda sunulmuştur. Kutu grafikleri hakkında detaylı bilgi için [47] makalesi okunabilir.

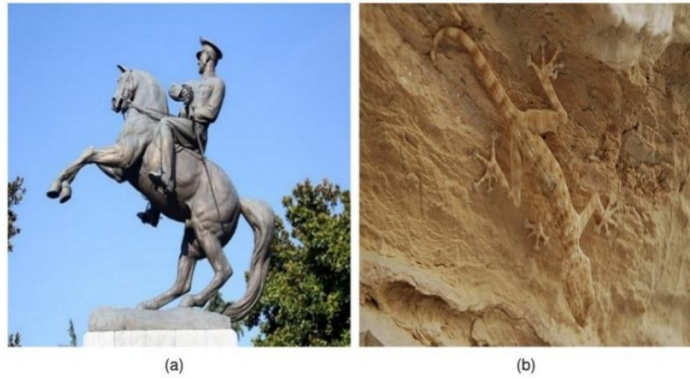


Şekil 15. Sırasıyla ρ , c , ve σ parametrelerinin eğitimlerdeki değişiminin kutu grafiği

Yukarıda Şekil 15'de kutu grafiklerinden ilki olan kenar parametresine bakıldığında en geniş veri dağılımının TLBO ve ABC algoritmalarında, en dar veri dağılımlarından birisinin ACO algoritmasında olduğu söylenebilir. Ayrıca BBO, HS, PSO algoritmalarının medyanlarının birbirine yakın olması, veri dağılım genişliğinin yakın aralıklarda seyretmesi, daha öncede bahsettiğimiz gibi, ABC, ACO ve TLBO algoritmalarının bu algoritmalarından daha iyi neticelere ulaşmasına neden olmuş olabilir. Anlaşılan o ki bu problem için BBO, HS ve PSO parametre bulma performansında aynı seyirde gitmiştir. İkinci seçenekte eğri evrimi parametresinin

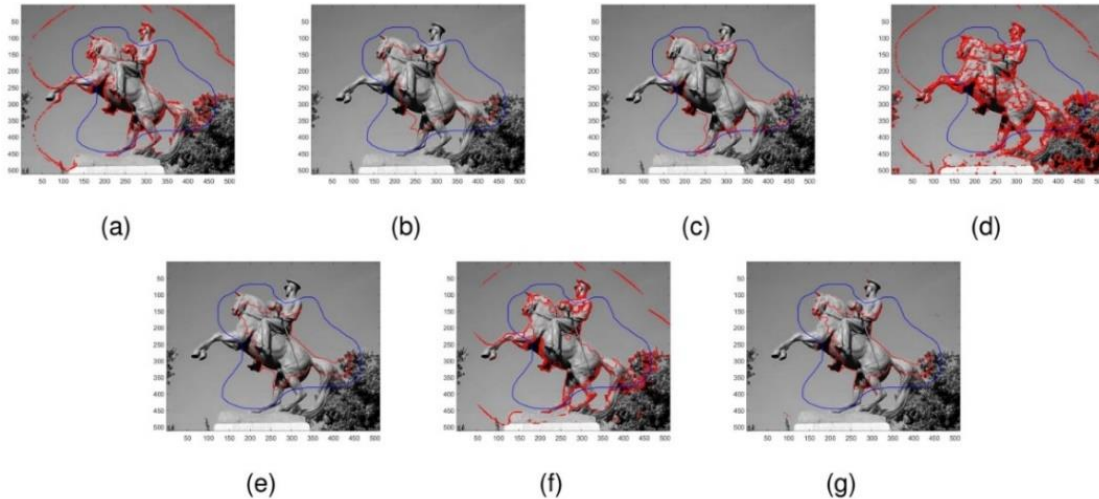
diagramına dikkat edilirse PSO hariç tüm algoritmaların medyan değerleri negatife doğru yoğun dağılım göstermektedir, buda daha önceki bölümde incelenen deneylerde, evrim hızının bu aralıkta olmasının daha iyi neticeler vermesi tezini kanıtlamaktadır. Ayrıca dağılımlar hemen hemen bütün bir aralığa, yani incelenmesi istenilen aralığın tamamına kadar veri taraması yapılmış olması çoğunlukla ABC, ACO, HS ve TLBO arasında çok bir fark olmadığını gösterir. Üçüncü diyagram olan düzleştirme parametresinin dağılımına bakıldığında, algoritmalar arasında ACO ve TLBO'nun birbirine çok yakın dağılım gösterdiği, bunun haricinde BBO, GA ve HS'nin de yine birbirine yakın dağılım gösterdikleri görülmüş, fakat ABC ve PSO'nun tamamen diğerlerinden farklı bir dağılıma yöneldiği görülmektedir. Bu ise varyansın kutu grafiğinde, algoritmaların bu parametredeki performansları açısından üç gruba ayrıldığını göstermektedir.

Testlerde yapılan incelemelerde iki sınıf resimler üzerinde durulmuştur. Bunlar kamufle ve kamufle olmayan resimlerdir. Kamufle resimlerde, resim içerisindeki nesnelere pek belirgin değildir, ışık geçişleri daha azdır, kamufle olmayan resimlerde ise, resim içerisindeki nesnelere daha belirgin olup, ışık geçişleri daha fazladır. Aşağıdaki Şekil'de bunlara örnek iki resim görülmektedir.

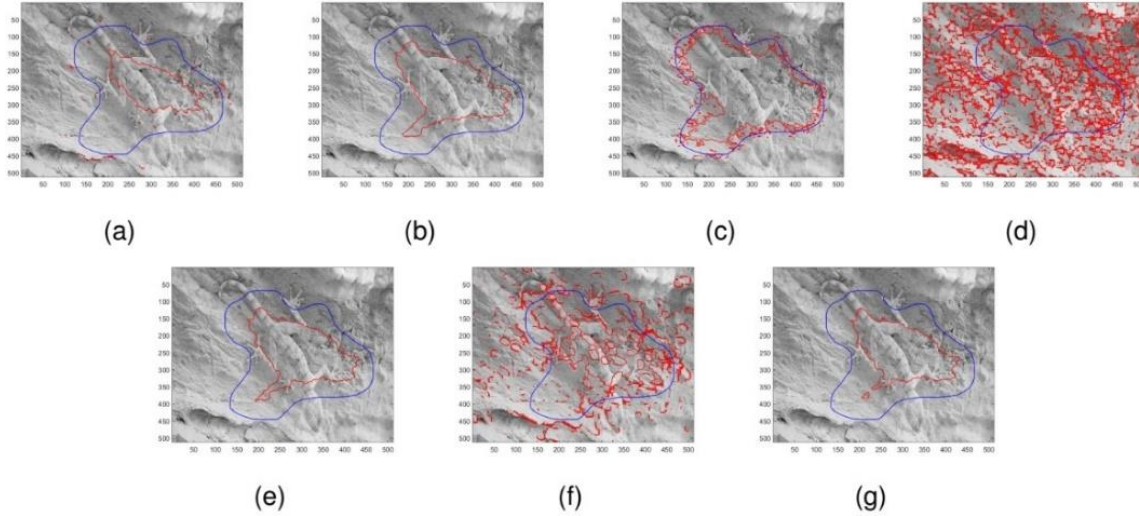


Şekil 16. (a) Kamufle olmayan görüntü (b) Kamufle görüntü

Aşağıdaki Şekil 17. ve Şekil 18'de algoritmaların görüntü içerisindeki nesnelere belirgin olan bir resimdeki eğri evrimi sonuçları ve görüntü içerisindeki nesnelere belirgin olmayan bir resimdeki eğri evrimi sonuçları yer almaktadır.



Şekil 17. Sırasıyla ABC, ACO, BBO, GA, HS, PSO, TLBO algoritmalarının nesnelere belirgin bir görüntüdeki performansları. Mavi: Başlangıç eğrisi, Kırmızı: Sonuç

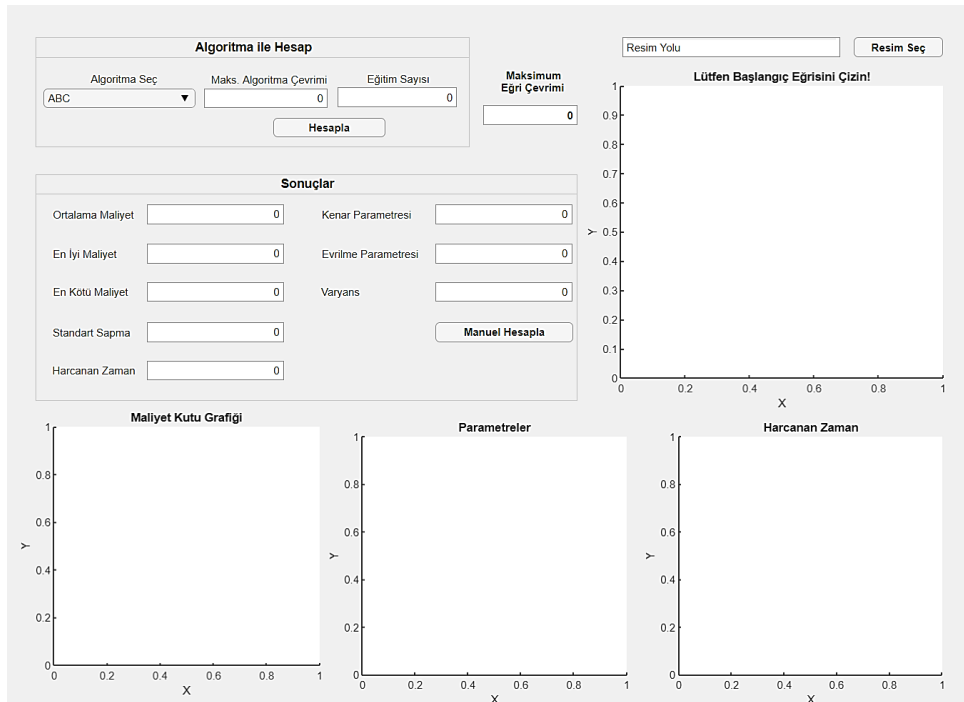


Şekil 18. Sırasıyla ABC, ACO, BBO, GA, HS, PSO, TLBO algoritmalarının nesnelere belirgin olmayan bir görüntüdeki performansları. Mavi: Başlangıç eğrisi, Kırmızı: Sonuç

V. BİR KULLANICI ARAYÜZÜ TASARIMI

Burada yapılan çalışmalar detaylı analizler ile oluşturulduğundan bu incelemeleri herhangi bir kullanıcıya yapabilmek için bir kullanıcı arayüzü uygulaması yapıldı. Bu uygulama kişilerin görebilmesi ve paylaşabilmesi için online platformlarda paylaşılacaktır.

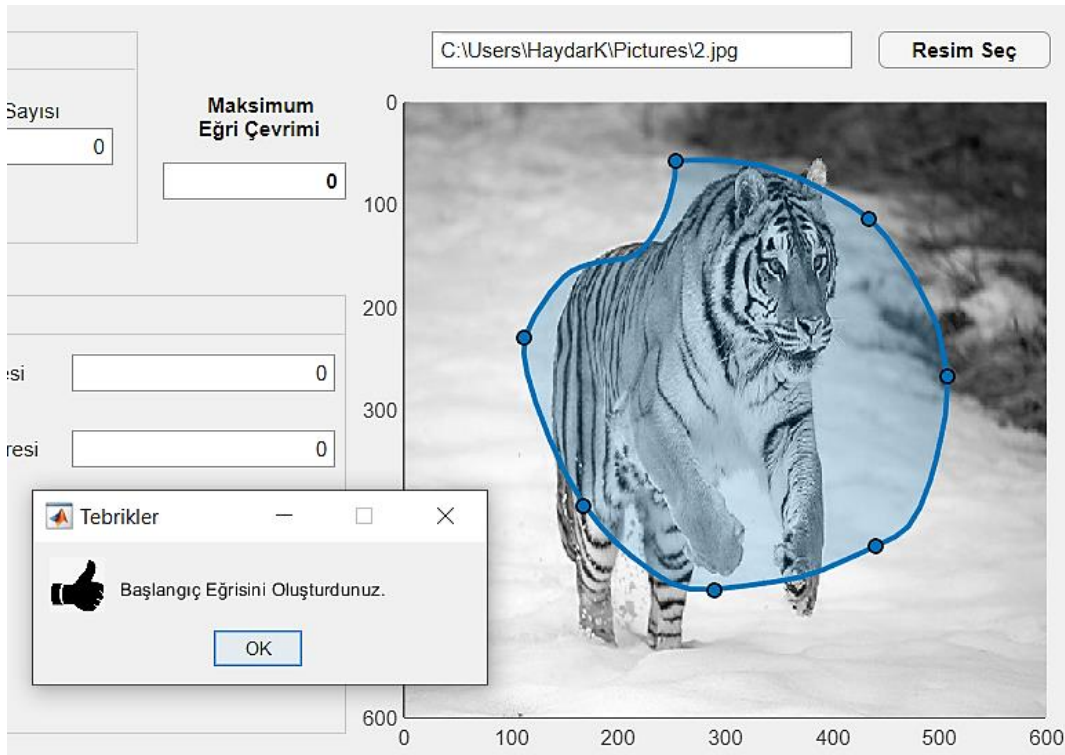
Uygulamanın aşağıdaki şekilde görüldüğü gibi bir ana pano içerisinde değişik kullanıcı girdileri ile hesaplamaların yapılacağı alt panolar yerleştirilmiştir.



Şekil 19. Uygulama Anapano

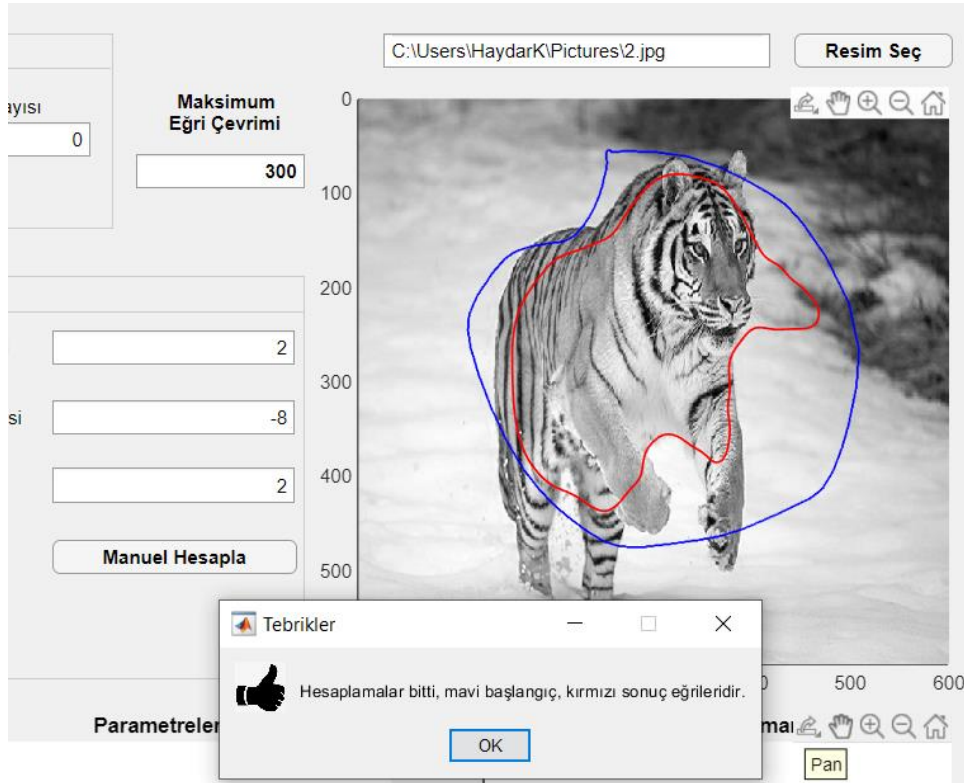
Kullanıcı uygulamayı çalıştırdığında ilk olarak gelen boş ana panoda, işleyeceği görüntüyü sağ üst köşedeki 'Resim Seç' butonu ile bilgisayarında bulunan herhangi bir yerdeki görüntüyü seçebilecektir. Resim Seç butonunun yanında ayrıca kullanıcı bilgisine sunulan 'Resim Yolu' alanında resmin bilgisayar içerisindeki alt klasörler bilgisi yer almaktadır.

Kullanıcı görüntüyü seçtikten sonra evrilmek istenen başlangıç eğrisini oluşturacaktır. Başlangıç eğrisi kullanıcının resmi yerleştiği yerde fare imleci ile serbest olarak çizeceği bir kapalı eğri olacaktır. Tüm bunların ilüstrasyonu aşağıdaki şekilde görülmektedir. Bu şekilde dikkat çeken bir diğer hususta 'Manuel Eğri Çevrimi' alanıdır, bu alana kullanıcı başlangıç eğrisinin diferensiyel evriminin kaç çevrimde oluşturulması gerektiği bilgisini girecektir. Bu alan hem manuel hesaplarda hem de yapay zeka algoritmaları ile yapılan hesaplarda doldurulacaktır.



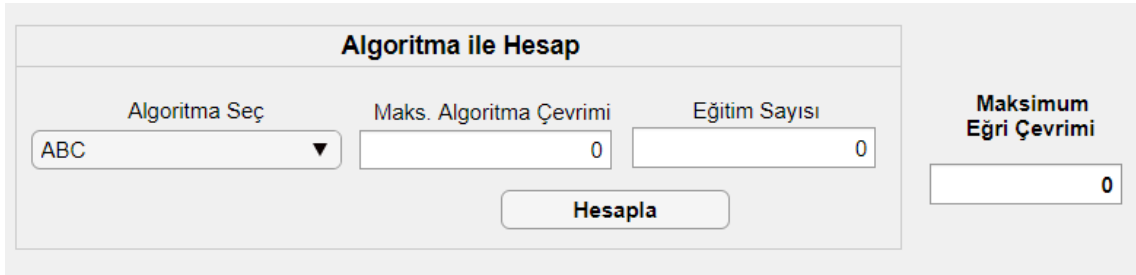
Şekil 20. Resim yükleme ve başlangıç eğrisi oluşturma

Başlangıç eğrisi oluşturduktan sonra, iki türlü hesaplama yapılabilecektir. Bunlardan ilki manuel olarak hesap yapmadır, burada parametreler kullanıcı tarafından girilip manuel hesapla butonuna tıklandığında aşağıdaki şekilde görüldüğü gibi resmin bulunduğu bölgede, hesaplamaların sonucu mavi başlangıç eğrisi, kırmızı sonuç eğrisi gösterilecektir.



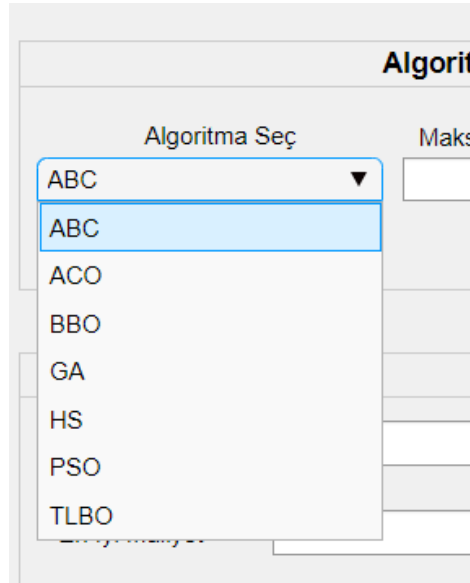
Şekil 21. Manuel olarak hesap yapma

Uygulama kullanıcısı manuel hesapla değilse, yapay zeka algoritmaları ile parametreleri bulmak isterse, aşağıdaki şekilde görülen 'Algoritma ile Hesap' panosunu ve 'Maksimum Eğri Çevrimi' alanını dolduracaktır.



Şekil 22. Algoritma ile hesaplama bölümü

Algoritma ile hesapla panosundan 'Algoritma Seç' menüsünden hesapları hangi algoritma ile yapmak istediğini belirterek 'Hesapla' butonuna basması yeterli olacaktır.



Algoritma Seç

ABC

ACO

BBO

GA

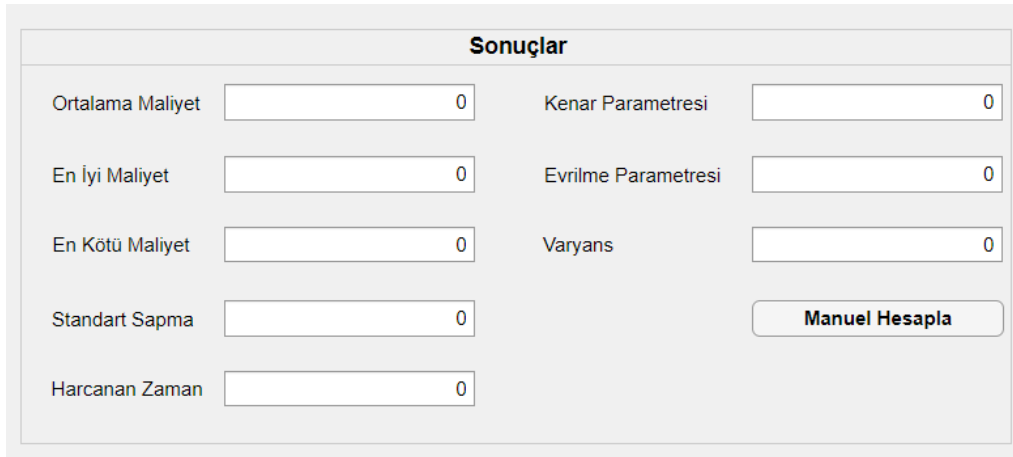
HS

PSO

TLBO

Şekil 23. Algoritma seçme bölümü

Hesaplamaya başladıktan sonra aşağıdaki 'Sonuçlar' panosu yapay zeka algoritmasının bulduğu sonuçlar ile otomatik doldurulacaktır.

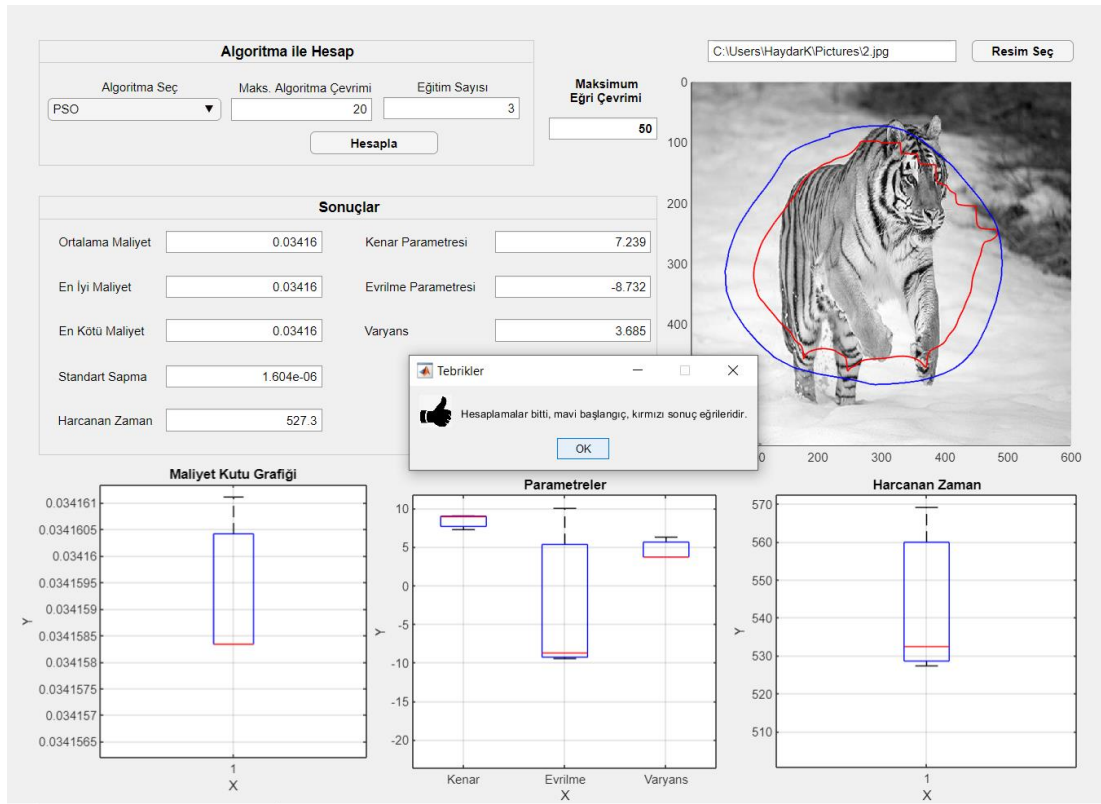


Sonuçlar

Ortalama Maliyet	<input type="text" value="0"/>	Kenar Parametresi	<input type="text" value="0"/>
En İyi Maliyet	<input type="text" value="0"/>	Evrilme Parametresi	<input type="text" value="0"/>
En Kötü Maliyet	<input type="text" value="0"/>	Varyans	<input type="text" value="0"/>
Standart Sapma	<input type="text" value="0"/>	<input type="button" value="Manuel Hesapla"/>	
Harcanan Zaman	<input type="text" value="0"/>		

Şekil 24. Sonuçlar panosu

Yapay zeka algoritması ile yapılan hesaplamaların neticesinde uygulama bir uyarı ile kullanıcıya hesapların bittiğini bildirecektir. Bu bildirim aynı zamanda manuel hesaplar bittiğinde de yapılır. Hesapların neticesinde Ana panonun en altında bulunan grafikler bölümünde Maliyet, Parametreler ve Harcanan zamanın tüm çevrim yada iterasyonlar bittiğinde istatistiksel sonuçları sunulacaktır. Ayrıca görüntünün son neticeleride aşağıdaki şekilde görüldüğü gibi resim bölümünde yer alacaktır.



Şekil 25. Uygulama yapay zeka algoritması sonuçları

VI. SONUÇ

Sayısal bir görüntü iki boyutlu bir ayrık fonksiyon olup uzamsal koordinatlarda sayısallaştırılabilir. Sayısallaştırılan görüntü üzerinde, görüntü işleme teknikleri kullanılarak görüntü üzerinde bölütleme yapılabilir. Bölütleme işleme görüntü üzerinde objeleri tanıma işlemi olarak da görülebilir. Bu çalışmanın giriş kısmında bu tekniklerin en yaygın kullanılanlarından bahsedilmiştir. Kısmi türevli denklemler yardımıyla yapılan bölütleme tekniklerinden biri olan aktif konturlar yardımıyla obje belirleme ise matematiksel detayları ile işlenmiştir.

Matematik modelin analitik yapısı incelendikten sonra, görüldü ki analitik olarak bir kapalı eğri evrimi gerçekleştirilerek objeleri tanıma yapılmaktadır. Bu model sayısallaştırılarak numerik analizi yapılmıştır. Numerik incelemeler sonunda, modele kullanıcı tarafından tanımlanan üç adet parametre bulunmaktadır. Her bir görüntüde gürültü yoğunluğu, objelerdeki ışık geçişleri gibi özelliklere göre bu parametreler kullanıcı tarafından tanımlanmak zorundadır. Bu nedenle bu parametrelerin yapay zeka algoritmaları ile optimizasyonu yapılarak, bu parametrelerin resme göre optimumlarının neler olabileceği değerlendirilmiştir. Ayrıca bu çalışmanın son bölümünde bununla ilgili bir kullanıcı arayüzü tasarlanarak sunulmuştur.

Sonuç olarak burada bir matematik modelle açıklanan segmentasyon modeli tartışılmış, detayları ile incelenmiştir. Bu model Riemann ölçümleri ile kurulduğu için, bu çalışmanın bir bölümünde bu modelden bahsedilirken 'Riemann anlamında eğri evrimi' olarak bahsedilmiştir. Bu çalışmanın ilerisinde bir çalışma olarak bu modelin Finsler ölçümlerine göre detaylı incelemesi yapılarak geliştirilecektir.

KAYNAKLAR

- [1] Pal, N. R. & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern recognition*, 26, 1277–1294.
- [2] Jaglan, P., Dass, R., & Duhan, M. (2019). A comparative analysis of various image segmentation techniques. *Proceedings of 2nd International Conference on Communication, Computing and Networking*, Springer, 359–374.
- [3] Javadpour, A., & Mohammadi, A. (2016). Improving brain magnetic resonance image (mri) segmentation via a novel algorithm based on genetic and regional growth. *Journal of biomedical physics & engineering*, 6, 95.
- [4] Ziou, D., & Tabbone, S. (1998). Edge detection techniques-an overview. *Pattern Recognition and Image Analysis C/C of Raspoznvaniye Obrazov I Analiz Izobrazhenii*, 8, 537–559.
- [5] Al-Amri, S. S., & Kalyankar, N. V. (2010). Image segmentation by using threshold techniques. *arXiv:1005.4020*.
- [6] Senthilkumaran, N., & Rajesh, R. (2008). Edge detection techniques for image segmentation-a survey. *Proceedings of the International Conference on Managing Next Generation Software Applications (MNGSA-08)*, 749–760.
- [7] Norouzi, A., Rahim, M. S. M., Altameem, A., Saba, T., Rad, A. E., Rehman, A., & Uddin, M. (2014). Medical image segmentation methods, algorithms, and applications. *IETE Technical Review*, 31, 199–213
- [8] Chebbout, S., & Merouani, H. F. (2012). Comparative study of clustering-based colour image segmentation techniques. *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*, IEEE, 839–844.
- [9] Amza, C. (2012). A review on neural network-based image segmentation techniques. *De Montfort University, Mechanical and Manufacturing Engg., The Gateway Leicester, LE1 9BH, United Kingdom*, 1–23.
- [10] Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, 13, 27–31.
- [11] Montana, D. J., & Davis, L. (1989). Training feedforward neural networks using genetic algorithms. *IJCAI*, 89, 762–767.
- [12] Erb, R. J. (1993). Introduction to backpropagation neural network computation. *Pharmaceutical research*, 10, 165–170.
- [13] Fahlman, C. L. (1990). The cascade-correlation learning architecture. *Advances in Neural Information Processing Systems*, 2.
- [14] Lin, W. C., Tsao, E. C. K., & Chen, C. T. (1992). Constraint satisfaction neural networks for image segmentation. *Pattern Recognition*, 25, 679–693.
- [15] Ranganath, H., Kuntimad, G., & Johnson, J. (1995). Pulse coupled neural networks for image processing. *Proceedings IEEE Southeastcon '95. Visualize the Future*, 37–43.
- [16] Selverston, A. I., & Moulins, M. (1985). Oscillatory neural networks. *Annual review of physiology*, 47, 29–48.
- [17] Sulehria, H. K., & Zhang, Y. (2007). Hopfield neural networks: A survey. *Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, Citeseer, 6, 125–130.
- [18] Dekker, A. H. (1994). Kohonen neural networks for optimal colour quantization. *Network: Computation in Neural Systems*, 5, 351–367.
- [19] Alirezaie, J., Jernigan, M., & Nahmias, C. (1997). Neural network-based segmentation of magnetic resonance images of the brain. *IEEE Transactions on Nuclear Science*, 44, 194–198.
- [20] Azmi, R., & Norozi, N. (2011). A new markov random field segmentation method for breast lesion segmentation in mr images. *Journal of medical signals and sensors*, 1, 156.
- [21] Barker, S. A., & Rayner, P. J. (2000). Unsupervised image segmentation using markov random field models. *Pattern Recognition*, 33, 587–602.
- [22] Yang, F., & Jiang, T. (2003). Pixon-based image segmentation with markov random fields. *IEEE Transactions on Image Processing*, 12, 1552–1559.
- [23] Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1, 321–331.

- [24] Li, C., Huang, R., Ding, Z., Gatenby, J. C., Metaxas, D. N., & J. C., Gore. (2011). A level set method for image segmentation in the presence of intensity inhomogeneities with application to mri. *IEEE transactions on image processing*, 20, 2007–2016.
- [25] Jiang, X., Zhang, R., & Nie, S. (2009). Image segmentation based on pdes model: A survey. *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, IEEE, 1–4.
- [26] Xu, C., Yezzi, A., & Prince, J. L. (2000). On the relationship between parametric and geometric active contours. *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No. 00CH37154)*, IEEE, 1, 483–489.
- [27] Caselles, V., Catté, F., Coll, T., & Dibos, F. (1993). A geometric model for active contours in image processing. *Numerische mathematik*, 66, 1–31.
- [28] Caselles, V., Kimmel, R., & Sapiro, G. (1997). Geodesic active contours. *International journal of computer vision*, 22, 61–79.
- [29] Bolsinov, A. V., Kozlov, V. V. E., & Fomenko, A. T. (1995). The Maupertuis principle and geodesic flows on the sphere arising from integrable cases in the dynamics of a rigid body. *Russian Mathematical Surveys*, 50, 473.
- [30] Javaloyes, M. A. (2012). Conformally standard stationary spacetimes and fermat metrics. *Recent Trends in Lorentzian Geometry*, Springer, 207–230.
- [31] Perlick, V. (1990). On fermat's principle in general relativity. i. the general case. *Classical and Quantum Gravity*, 7, 1319.
- [32] Torromé, R. G., Piccione P., & Vitória, H. (2012). On fermat's principle for causal curves in time oriented finler spacetimes. *Journal of mathematical physics*, 53, 123.
- [33] Zhao, D., & Luo, M. (2017). General conformable fractional derivative and its physical interpretation. *Calcolo*, 54, 903–917.
- [34] Petrova, S. S., & Solov'ev, A. D. (1997). The origin of the method of steepest descent. *Historia Mathematica*, 24, 361–375.
- [35] Chen, Y. G., Giga, Y., Goto, S. (1991). Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *Journal of differential geometry*, 33, 749–786.
- [36] Malladi, R., Sethian, J. A., & Vemuri, B. C. (1995). Shape modeling with front propagation: A level set approach. *IEEE transactions on pattern analysis and machine intelligence*, 17, 158–175.
- [37] Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12, 629–639.
- [38] Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79, 12–49.
- [39] Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). Metaheuristic research: A comprehensive survey. *Artificial Intelligence Review*, 52, 2191–2233.
- [40] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, 4, 1942–1948.
- [41] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 200, 1-10
- [42] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1, 28–39.
- [43] Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267, 66–73.
- [44] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, 76, 60–68.
- [45] Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43, 303–315.
- [46] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12, 702–713.
- [47] Potter, K., Hagen, H., Kerren, A., & Dannenmann, P. (2006). Methods for presenting statistical information: The box plot. *Visualization of large and unstructured data sets*, 4, 97–106.