



# Düzce Üniversitesi Bilim ve Teknoloji Dergisi

*Araştırma Makalesi*

## Renkli Görüntüler İçin Yusufçuk Algoritması Kullanılarak Benzerlik Görüntüsüne Dayalı Eşikleme

Ferzan KATIRCIOĞLU <sup>a</sup>

<sup>a</sup> *Elektronik ve Otomasyon Bölümü, Düzce Meslek Yüksekokulu, Düzce Üniversitesi, Düzce, TÜRKİYE*

\* *Sorumlu yazarın e-posta adresi: ferzankatircioglu@duzce.edu.tr*

### ÖZET

Bu çalışmada, Yusufçuk Algoritması (YA) kullanarak renkli görüntülerde eşik yöntemi ile kenar belirleme ve bölütleme gerçekleştirilmiştir. Kenar belirleme, görüntüdeki nesnelerin sınırlarını belirleme işlemi, bölütleme ise görüntü piksel yoğunluklarının önceden belirlenmiş sınıflara ayrılma işlemi yapmaktadır. Her iki işlem de görüntü üzerinden özellik çıkartma, görüntü tanıma ve nesne sınıflandırma gibi uygulamalarda ön işlem özelliklerini taşımaktadır. Kenar Belirleme ve bölütleme metotlarından en kolay uygulanabilir olanlarından biri olan eşikleme yönteminin uygulanabilmesi için ilk olarak, renkli görüntüden iki boyutlu bağıntı matrisine dayalı Benzerlik Görüntüsü (BG) elde edilmiştir. Elde edilen benzerlik görüntüsü üzerinde en uygun eşik değeri seçimi için amaç fonksiyonu olarak Kapur ve Otsu'nun metotları kullanılmıştır. Optimizasyon safhasından sonra, en iyi eşik değerler çoklu eşik bölütleme metoduna uygulanarak sınıflara ayrılmış görüntü ve kenarlar elde edilmiştir. Elde edilen sayısal ve görsel uygulama sonuçları, literatürde yer alan yerçekimi arama algoritması (YAA) ve Harmoni Arama Algoritması (HAA) ile karşılaştırılmıştır. Gerçekleştirilen çalışmanın sonuçları önerilen algoritmanın sayısal görüntü bölütleme ve kenar belirleme de güçlü olduğunu göstermiştir.

**Anahtar Kelimeler:** *Bölütleme ve Kenar Belirleme, Yusufçuk Algoritması, Benzerlik Görüntüsü.*

## Similar Image-Based Thresholding Using Dragonfly Algorithm For Color Images

### ABSTRACT

In this study, edge detection and segmentation are performed with threshold method using Dragonfly Algorithm (DA) in color images. While edge detection is the process of determining the boundaries of objects in an image, segmentation is the process of dividing the image pixel densities into predetermined classes. Both processes carry pre-processing features such as feature extraction, image recognition and object classification on the image. In order to apply the threshold method, which is one of the easiest to apply from the edge detection and segmentation methods, firstly Similarity Image (SI) based on the two-dimensional relation matrix from the color image is obtained. The methods of Kapur and Otsu have been used as the objective function for selecting the most appropriate threshold value on the obtained similarity image. After the optimization phase, the best threshold values are applied to the multiple threshold segmentation method to obtain class separated images and edges. The

obtained numerical and visual application results are compared with Gravitational Search Algorithm (GSA) and Harmony Search Algorithm (HSA) in the literature. The proposed algorithm shows that the digital image segmentation and edge detection are also robust.

*Keywords: Segmentation and Edge Detection, Dragonfly algorithm, Similarity Image.*

## I. GİRİŞ

Görüntüde parlaklık, yoğunluk ve renk bilgisi gibi belirlenen özellikler doğrultusunda birbirleriyle bağlı olan piksellerin türdeş gruplara ayırıştırma işlemine bölütleme denir [1]. Bölütleme yöntemlerinden biri olan eşikleme tekniğinde farklı gri seviye oranlarına sahip görüntünün histogramı hesaplandıktan sonra, eşik değerine tabii tutulmaktadır. Son yıllarda, uygun çok seviyeli eşik değer seçimini gerçekleştirmek için sezgisel optimizasyon teknikleri kullanılmıştır. İlk çalışmalardan biri olan minimum çapraz entropi yönteminde, uygun eşik değerini bulmak için parçacık sürü optimizasyonunu kullanılarak, gri görüntülere uygulanmıştır [2]. Cuevas vd. gri görüntünün histogramına yaklaşan her bir Gauss fonksiyonu bir piksel sınıfını ve dolayısıyla bir eşik noktasını temsil etmesini sağlamışlardır [3]. Birçok araştırmacı gri görüntülerde çeşitli sezgisel optimizasyon tekniklerini kullanarak bölütleme işlemini gerçekleştirmiştir [4-6].

Histogram üzerinde eşikleme tekniğinin renkli görüntülere uygulanması, gri görüntülere göre daha karmaşık ve farklılık göstermektedir. Üç boyutlu bir histogram içerisinde bütün boyutları içeren küresel bir tutma değeri seçmek, zor bir süreç olarak karşılaşılmaktadır. Renkli görüntülerde bölütleme işlemi ilk zamanlarda, üç boyutlu histogramlar iki boyuta dönüştürülerek veya her bir boyut yüzeyi ayrı değerlendirilerek gerçekleştirilmiştir [7]. Diğer bir yaklaşımda, üç boyutlu histogram, homojen bir bilgi alanına çevrilerek yeni farklı bir histogram elde edilmiştir [8]. Kurban ve arkadaşları renkli görüntülerde çoklu eşikleme yöntemi için sürü ve evrimsel tabanlı karşılaştırmalı bir çalışma sunmuşlardır [9]. Minimum çapraz entropi kavramı kullanılarak doğal renkli görüntü içerisinde bölütleme çalışması önerilmiştir [10]. Renkli görüntülerin mekan bağlamsal bilgisini içeren bir enerji fonksiyonu kullanılarak görüntünün enerji eğrisi oluşturulmuştur. Histogram eğrisine benzer özellikleri taşıyan enerji eğrisine çoklu eşik yöntemi uygulanmıştır [11].

Kenar, görüntü içerisinde yer alan nesnelerin sınırlarına, görüntüde yer alan gölge sınırlarına ve nesnelerin içindeki yansıma ve aydınlatma değişimlerine denir [12]. En yaygın kenar belirleme algoritmaları birinci dereceden eğimin maksimumun değerinin hesaplanması ve ikinci dereceden eğimin sıfır geçişinin belirlenmesi uygulamalarıdır. Robert, Prewitt ve Sobel operatörleri birincil operatörlerken Laplas ve LoG operatörleri ikincil operatörler sınıfına girmektedir [13,14]. Canny operatörü düşük yoğunluklarda kenar belirleme düzenleme ve filtreleme işlemlerini içeren çoklu fonksiyonel optimize edilmiş operatördür [15]. Renkli görüntülerde kenar belirleme için renkli görüntüler gri görüntülere dönüştürülmelidir. Ancak gri işlem süresi boyunca bilgi kaybı meydana gelmektedir. Bundan dolayı renkli görüntülerde kenar belirleme araştırmaları daha fazla dikkat çekici hale gelmiştir. İlk olarak Yu ve arkadaşları Kırmızı Yeşil Mavi (RGB) renk kanallarını üç bağımsız gri görüntüymüş gibi kullandı ve sonunda üç görüntünün sonuçlarını birleştirmişlerdir [16]. Vektör gradyan operatörü renk piksellerini 3 boyutlu bir vektör olarak alan klasik bir kenar belirleme algoritmasıdır [17]. RGB renk kanallarının korelasyonu gürültülü kenarlar ortadan kaldırılarak kullanılmıştır [18]. Kaybolmuş kenarları büyüten, güçlü belirgin kenarları da düzgülendiren ve her iki kenar tipini yerel süreklilik kullanımına sokan yeni bir Canny çalışması önerilmiştir [19]. Bir diğer çalışmada, bulanık hücresel otomat sistemine dayanan ve doku histogramı olarak adlandırılan kenar belirleme yöntemi sunulmuştur [20].

Bu çalışmada ikinci bölümde, renkli görüntüler için bağıntı matrisine dayalı BG, sınıfsal eşikleme metotları ve YA optimizasyon tekniğinden kısaca bahsedilmiştir. YA ve BG kullanılarak renkli görüntülerde yeni bir çok seviyeli eşikleme metodunun anlatımı bu bölümün en son alt konusunu oluşturmaktadır. Bağıntı matrisine dayalı BG görüntüsü üzerinden kenar belirleme ve bölütleme işlemini gerçekleştirmek, literatürde yer alan renkli görüntülerde çok eşikli kenar belirleme ve bölütleme yöntemlerinden farklılıklar içermektedir. Renkli görüntüden BG özellik görüntüsünün elde edilmesi ve bu özellik görüntüsü üzerinde çok eşikli kenar belirleme ve bölütleme gerçekleştirilmesi, farklılık özelliklerini oluşturmaktadır. Üçüncü bölümde, önerilen yöntem RGB renkli görüntülerine uygulanarak sonuçlar, karşılaştırmalı olarak verilmiştir. Elde edilen görsel ve sayısal sonuçlar Kenar Belirleme ve Bölütleme alt başlıklarında ayrı olarak incelenmiştir. Son bölümde ise yöntem ile ilgili değerlendirmeler yapılmış ve ileride yapılabilecek çalışmalar için öneriler sunulmuştur.

## II. YUSUFÇUK ALGORİTMASI KULLANILARAK BENZERLİK GÖRÜNTÜSÜNE DAYALI EŞİKLEME

Son yıllarda geliştirilen ve arama kabiliyeti yüksek olan YA sezgisel optimizasyon tekniği renkli görüntülerden elde edilen BG adı verilen özellik değerleri üzerine uygulanmıştır. Eşik değer seçimi için uygulamada Kapur ve Otsu sınıfsal eşikleme metotları kullanılmıştır.

### *A. BAĞINTI MATRİSİNE DAYALI BENZERLİK GÖRÜNTÜSÜ*

BG, piksellerin renk özellikleri esas alarak, renk fark değerlerinin analizi sonucu çıkan gri görüntü şeklinde tanımlanmaktadır. Üç boyutlu renk uzayındaki görüntü bilgilerini iki boyutlu özelliğe dönüştürme işlemini yapmaktadır [21]. BG elde etmek için renkli görüntü üzerinde 3x3'lük bir maske geçirilerek başlanmaktadır. Daha sonra, maske içerisindeki merkez piksel için, bağıntı matrisi ve ortalaması elde edilir, ortalama değer BG içerisinde merkez pikselin bulunduğu adrese yazılmaktadır [22]. Pikseller arasındaki mesafe, iki pikselin renkleri arasındaki geometrik fark şeklinde tanımlanır.  $P_i(R_i, G_i, B_i)$  ve  $P_j(R_j, G_j, B_j)$  şeklinde iki piksel arasındaki mesafe Öklid mesafesi kullanılarak bulunmaktadır.

$$d_{ij} = \sqrt{(R_i - R_j)^2 + (G_i - G_j)^2 + (B_i - B_j)^2} \quad (1)$$

Pikselin benzerlikleri için Denklem (2), (3) ve (4)'deki fonksiyonlar kullanılmıştır [23].

$$S(i, j) = 1 - \frac{d_{ij}}{D} \quad (2)$$

$$S(i, j) = \exp\left(\frac{-d_{ij}}{D}\right) \quad (3)$$

$$S(i, j) = \exp\left(\frac{-d_{ij}^2}{D}\right) \quad (4)$$

Denklem (2),(3) ve (4)' de verilen D değeri normalleştirilme katsayısıdır, 32, 64, 128 ve 255 değerleri verilerek elde edilen sonuçlar derecelendirilmektedir. Maske içerisindeki 9 piksel için kendileriyle ve komşularıyla beraber yapıldığında 9x9'luk bir benzeşim bağıntı matrisini oluşturmaktadır.

$$\begin{bmatrix} S(1,1) & S(1,2) \dots & S(1,9) \\ S(2,1) & S(2,2) \dots & S(2,9) \\ \vdots & \vdots & \vdots \\ S(9,1) & S(9,2) \dots & S(9,9) \end{bmatrix} \quad (5)$$

Denklem (5)'deki bağıntı matrisinin aritmetik ortalaması alınarak, merkez piksel için benzeşim görüntüsünde yer alacak olan benzeşim değeri Denklem (6) ya göre hesaplanmaktadır.

$$S_{merkez} = \frac{1}{81} \sum_{i=1}^9 \sum_{j=1}^9 S_{ij} \quad (6)$$

Denklem (6) da elde edilen değer, merkez pikselin adresi dikkate alınarak, yeni oluşturulan benzeşim görüntüsüne atanır ve maske hareket eder. İşlem gerçek görüntüdeki son pikselin benzeşim değeri bulununcaya kadar devam etmektedir [21].

## B. SINIFSAK EŞİKLEME METOTLARI

Eşikleme, gri tonlu görüntü piksellerinin L yoğunluk seviyelerine bağılı olarak kümelere veya sınıflara bölünme sürecidir, Denklem (7)'deki basit bir kural ile gerçekleştirilmektedir.

$$\begin{aligned} c_1 &\leftarrow p \quad \text{if } 0 \leq p < th \\ c_2 &\leftarrow p \quad \text{if } th \leq p < L - 1 \end{aligned} \quad (7)$$

Gri tonlu görüntü L gri ton seviyeleri  $L=\{0,1,2,\dots,L-1\}$  şeklinde temsil edilir ve eşitlikteki p bu görüntünün  $m \times n$  piksellerinden biridir. İlgili p pikseli  $c_1$  ve  $c_2$  sınıflarından birine yerleştirilir. L yoğunluk seviyelerini aldıktan sonra, yoğunluk değerlerinin olasılık dağılımları aşağıdaki gibi hesaplanmaktadır [24].

$$\begin{aligned} Ph_i^c &= \frac{h_i^c}{NP}, \quad \sum_{i=1}^{NP} Ph_i^c = 1, \\ c &= \begin{cases} 1,2,3, & \text{if } RGB \text{ görüntü} \\ 1, & \text{if } Gri \text{ görüntü} \end{cases} \end{aligned} \quad (8)$$

Denklem (8)'deki i yoğunluk seviyesini göstermektedir ve  $0 \leq i \leq L - 1$  arasındadır, c üzerinde çalışılan görüntü RGB ise görüntünün bileşen değerini, NP görüntünün toplam piksel sayısını temsil etmektedir.  $h_i^c$  (histogram) c bileşeni içerisindeki i yoğunluğuna karşılık gelen piksel sayısıdır. Bu durumda  $Ph_i^c$  olasılığı içerisinde histogram normalleştirilmiştir. İki sınıflı basit bir bölütleme aşağıda verilmektedir.

$$\begin{aligned}
c_1 &= \frac{Ph_1^c}{w_0^c(th)}, \dots, \frac{Ph_{th}^c}{w_0^c(th)}, \\
c_2 &= \frac{Ph_{th+1}^c}{w_1^c(th)}, \dots, \frac{Ph_L^c}{w_1^c(th)},
\end{aligned} \tag{9}$$

Denklem (9)'daki  $w_0(th)$  ve  $w_1(th)$   $c_1$  ve  $c_2$  için olasılık dağılımlarıdır ve aşağıdaki Denklem (10) da gösterilmiştir.

$$\begin{aligned}
w_0^c(th) &= \sum_{i=1}^{th} Ph_i^c, \\
w_1^c(th) &= \sum_{i=th+1}^L Ph_i^c,
\end{aligned} \tag{10}$$

Yaygın kullanım açısından Kapur ve Otsu tarafından önerilen iki sınıfsal eşikleme metodu mevcuttur [25,26].

### B.1. SINIF ARASI VARYANS (OTSU YÖNTEMİ)

Eşikleme için parametrik olmayan bir teknik olan bu metot Otsu tarafından önerilmiş olup, görüntüyü bölgelere ayırma kriteri olarak farklı sınıfların maksimum varyans değerlerini kullanmaktadır. Sınıfların belirlenmesinde  $\mu_0^c$  ve  $\mu_1^c$  ortalama seviyelerinin hesaplanması zorunluluktur.

$$\mu_0^c = \sum_{i=1}^{th} \frac{iPh_i^c}{w_0^c(th)}, \tag{11}$$

$$\mu_1^c = \sum_{i=th+1}^L \frac{iPh_i^c}{w_1^c(th)}, \tag{12}$$

$$\sigma^{2^c} = \sigma_1^c + \sigma_2^c \tag{12}$$

Denklem(11) ve (12)'nin ikisi de görüntünün tipi olan c'ye bağlıdır. Denklem (12)'deki iki sayısı Otsu'nun varyans operatörünün parçasıdır ve matematiksel anlamda üssel bir ifadeyi temsil etmez. Ayrıca  $c_1$  ve  $c_2$  nin varyansları Denklem (13) de tanımlanmıştır.

$$\begin{aligned}
\sigma_1^c &= w_0^c(\mu_0^c + \mu_T^c)^2, \\
\sigma_2^c &= w_1^c(\mu_1^c + \mu_T^c)^2,
\end{aligned} \tag{13}$$

Eşitlikteki  $\mu_T^c$  ifadesi  $\mu_T^c = w_0^c \cdot \mu_0^c + w_1^c \cdot \mu_1^c$  dir ve  $w_0^c + w_1^c = 1$  dir.  $\sigma_1^c$  ve  $\sigma_2^c$  değerleri esas alınarak amaç fonksiyonu Denklem (14) de sunulmaktadır.

$$J(th) = \max(\sigma^{2^c}(th)), \quad 0 \leq th \leq L - 1 \tag{14}$$

Denklem (14)'de ki  $\sigma^{2^c}$  verilen th değeri için Otsu varyansını temsil etmektedir. Bu yüzden optimizasyon problemi th yoğunluk değerini bulmak için maksimumdan azaltılmaktadır [25]. Çoklu

eşik yöntemi de yukarıdaki işlemlere benzer şekilde gerçekleştirilerek amaç fonksiyonu elde edilmektedir.

## B.2. ENTROPİ KRİTER YÖNTEMİ (KAPUR YÖNTEMİ)

Diğer bir parametrik olmayan optimal eşik değer belirleme metot da Kapur tarafından önerilmiştir. Bu metot görüntünün histogramını ve entropi ilkesini esas alır. Kapur probleminin amaç fonksiyonu aşağıdaki Denklem (15) de tanımlanmıştır.

$$J(th) = H_1^c + H_2^c \quad (15)$$

$$c = \begin{cases} 1,2,3, & \text{if } RGB \text{ görüntü} \\ 1, & \text{if } Gri \text{ görüntü} \end{cases}$$

Denklem (15)'deki H1 ve H2 entropileri aşağıdaki model kullanılarak hesaplanır.

$$H_1^c = \sum_{i=1}^{th} \frac{Ph_i^c}{w_0^c} \ln \left( \frac{Ph_i^c}{w_0^c} \right), \quad (16)$$

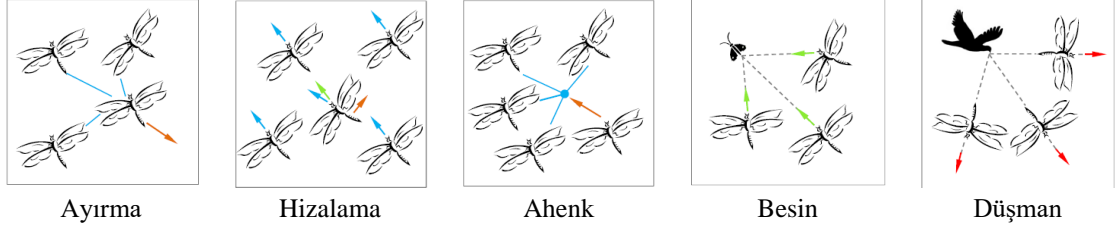
$$H_2^c = \sum_{i=th+1}^L \frac{Ph_i^c}{w_1^c} \ln \left( \frac{Ph_i^c}{w_1^c} \right),$$

Denklem (16)'deki  $Ph_i^c$  Denklem (8) kullanılarak elde edilmiş yoğunluk seviyelerinin olasılıkları,  $w_0(th)$  ve  $w_1(th)$  ise Denklem(10) kullanılarak elde edilmiş  $c_1$  ve  $c_2$  için olasılık dağılımlarını temsil etmektedir (26). Çoklu eşik yöntemi de yukarıdaki işlemlere benzer şekilde gerçekleştirilerek amaç fonksiyonu elde edilmektedir.

## C. YUSUFÇUK ALGORİTMASI

Yusufçuk sineği büyük birleşik gözleri, güçlü saydam kanatları, göz alıcı renkleri ve uzunca vücutlarıyla tanınır ve dünya da çok farklı türlere sahiptir. Yaşam döngüsü larva ve yetişkin olmak üzere iki önemli evreden oluşmaktadır. Avlanma özelliklerinde eşsiz ve nadir görülen zeki davranışlar görülmektedir. Statik sürüde yusufçuk sinekleri küçük gruplar halinde uçar ve avlanmak için ileri geri hareket ederler. Uçma yolu güzergahında yerel hareketler ve ani değişiklikler statik sürünün ana karakteristik özellikleridir. Bununla birlikte dinamik sürülerde çok sayıdaki yusufçuk büyük grup halinde göç için uzun yolculuk yaparlar.

Reynold'a göre sürü davranışları üç temel prensip içerisindedir [27]. Ayırma; Komşuluk içerisindeki diğer kişiler ile çarpışmayı önlemek anlamına gelmektedir. Hizalama; Komşuluk içerisindeki diğer bireyler ile hız uyumunu göstermektedir. Ahenk; Komşuluk kütesinin merkezine doğru bireylerin eğilimine atıf yapmaktadır. Herhangi bir sürünün asıl amacı hayatta kalabilmektir ve bundan dolayı bireyler gıda kaynaklarına doğru yönelmelidirler. Ayrıca bu ana eylemin yanında sürü dışarıdan düşmanları tarafından rahatsız edilebilirler. Bu iki davranış da eklenirse, konum güncelleme işleminde beş ana faktör kullanılmaktadır ve Şekil 1.'de gösterilmiştir [28].



Şekil 1. Sürü içerisinde bireyler arasındaki beş ana faktör [28].

Bu davranışların her birini matematiksel olarak aşağıdaki gibi modellenmektedir.

$$S_i = - \sum_{j=1}^N X - X_j \quad (17)$$

Denklem (17) deki  $X$  mevcut bireyin pozisyonu,  $X_j$  ise  $j$ . komşu bireyin pozisyonunu göstermektedir.  $N$  komşu bireylerin sayısını vermektedir.

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (18)$$

Denklem (18) de yer alan  $V_j$  değeri  $j$ . komşu bireyin hızını göstermektedir.

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (19)$$

Besin kaynağına doğru çekim aşağıdaki gibi hesaplanmaktadır.

$$F_i = X^+ - X \quad (20)$$

Denklem (20) deki  $X^+$  ifade besin kaynağının pozisyonunu göstermektedir. Bir düşman tehdidi karşısında dışa doğru davranış aşağıdaki gibi hesaplanmaktadır. Denklem (21) deki  $X^-$  ifadesi tehdit eden düşmanın pozisyonunu göstermektedir.

$$E_i = X^- + X \quad (21)$$

Yapay yusufluk sineklerinin pozisyonlarını güncellemek ve onların hareketlerine benzetmek için  $\Delta X$  adım ve  $X$  pozisyon olmak üzere iki vektör düşünülmüştür.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (22)$$

Denklem (22) de ki  $s$  ayırma,  $a$  hizalama,  $c$  uyum,  $f$  besin,  $e$  düşman faktörlerinin ağırlık değerlerini temsil etmektedir. Ayrıca  $w$  değeri  $t$ . döngüdeki atalet ağırlığını göstermektedir. Adım vektörünü hesapladıktan sonra aşağıda Denklem (23) de yer verilen pozisyon vektörü hesaplanmaktadır.

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (23)$$

Rassallığı, stokastik davranışı ve yapay yusufçukların keşif özelliklerini geliştirmek için rastgele yürüyüş (Le'vy uçuşu) kullanılmıştır. Bu durumda yusufçukların pozisyonları aşağıdaki eşitlik kullanılarak güncellenmiştir [28].

$$X_{t+1} = X_t + Le'vy(d) * X_t \quad (24)$$

Denklem (24) deki t mevcut döngüyü, d ise pozisyon vektörünün boyutunu temsil eder.

#### D. YA VE BG KULLANILARAK RENKLİ GÖRÜNTÜLERDE KENAR BELİRLEME VE BÖLÜTLEME

Renkli görüntü okunduktan sonra, gri tonlu benzeşim görüntüsü Denklem (2), (3) ve (4) de ki fonksiyonlar kullanılarak üç farklı şekilde elde edilmektedir. BG, D=16, D=32 ve D=64 olarak hassasiyetleri derecelendirilmektedir. BG histogramı elde edildikten sonra, Denklem (14) Otsu veya Denklem(15)'deki Kapur amaç fonksiyonu kullanılarak YA optimizasyon tekniğine uygulanmaktadır.

```

Adım 1. I görüntüsünü oku.
        for piksel=1 den görüntünün uzunluğuna kadar
        for piksel=1 den görüntünün genişliğine kadar
            3x3 maske üzerinden Benzerlik Görüntüsünü elde et;
        son
    son
Adım 2. Benzerlik Görüntüsünün Histogramını elde et
Adım 3. Olasılık  $Ph_f$  ve  $w_0(th)$   $w_1(th)$  olasılık dağılımları hesapla
Adım 4. Yusufçuk Algoritmasının ajan sayısı N, döngü sayısı ve amaç fonksiyonu belirle
Adım 5. Rastgele başlatma
Adım 6. Ajanların uygunluk değerlerinin tespiti
        for döngü=1 den maksimum döngüye
             $J(th) = (Kapur)$  veya  $J(th) = (Otsu)$ ; Eşikleri Bul;
        son
Adım 7. Optimizasyon sürecinde çıkan en uygun değerleri th eşik olarak kaydet.
        for piksel=1 den görüntünün uzunluğuna kadar
        for piksel=1 den görüntünün genişliğine kadar
            Kenar Belirleme yap; veya Bölütleme yap;
        son
    son
Adım 8. out im cıkıs görüntüsünü ekranda göster

```

Şekil 2. YA ve BG kullanılarak renkli görüntülerde kenar belirleme ve bölütleme pseudo- kodu

Şekil 2'de pseudo kodunda verildiği gibi, kenar belirleme çalışması için sadece bir eşik değerine ihtiyaç duyarken, bölütleme için ikiden fazla girilmesi gerekmektedir.

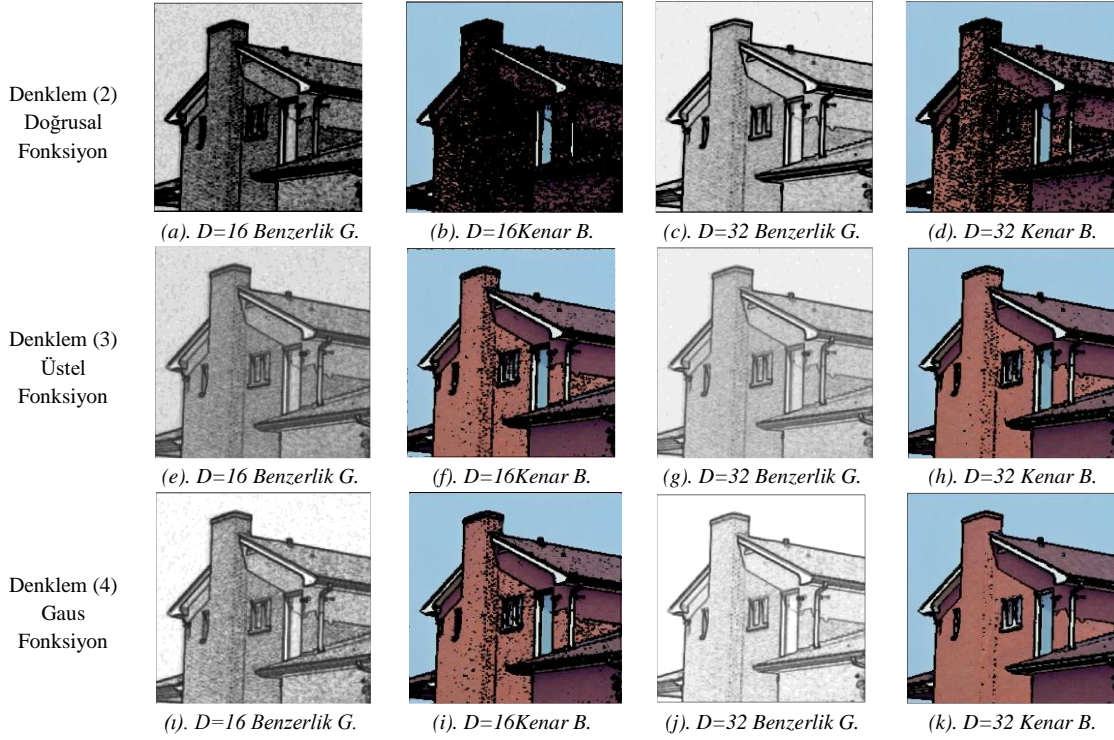
### III. BULGULAR ve TARTIŞMA

Kenar belirleme ve bölütleme işlemi için Şekil 3'de yer alan ev, Lena ve su RGB renkli görüntülerinde test edilmiştir. Bu görüntüler 256x256 piksel büyüklüğünde ve jpeg formatındadır.





Şekil 3. Uygulamada kullanılan test görüntüleri (a). Ev (b). Lena (c). Su



Şekil 4. YA optimizasyon algoritmasının benzerlik fonksiyonlarına uygulanması sonucu Benzerlik ve Kenar Belirleme sonuçları

Öncelikle Denklem (2), (3) ve (4)'deki, en iyi sonuçların hangi fonksiyonda elde edildiği tespit edilmiştir. Şekil 4'de yer alan görsel sonuçlara göre Denklem (2) sonuçlarında kirlilik çok fazladır, bu durumda kenarların kalın çıkması ile sonuçlanmıştır. Denklem (4) sonuçları, BG verilerinin kaybolması ile sonuçlanmıştır, Şekil 4. (j)'de görsel olarak verilmiştir. Dolayısı ile optimizasyon algoritması eşik bulmakta zorlanmıştır. Bu nedenle, bundan sonraki bütün test uygulamaları için Denklem (3) üstel fonksiyon kullanılmıştır.

Önerilen çalışmanın istikrar ve tutarlılığını değerlendirmek için elde edilen sonuçların standart sapma değerleri ve orijinal görüntü ile bölütlenmiş görüntü arasındaki benzerliği belirleyen Tepe Sinyal Oranı (TSO) kullanılmıştır.

$$TSO = 20 \log_{10} \left( \frac{255}{RMSE} \right) \quad RMSE = \sqrt{\frac{\sum_{i=1}^{ro} \sum_{j=1}^{co} (I_0^c(i,j) - I_{th}^c(i,j))^2}{ro \cdot co}} \quad (25)$$

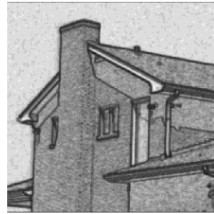
Denklem (25) deki  $I_0^c$  orijinal görüntüyü,  $I_{th}^c$  ise bölütlenmiş görüntüyü, ro ve co görüntünün sırasıyla sütun ve satır sayısını vermektedir.

## A. KENAR BELİRLEME SONUÇLARI

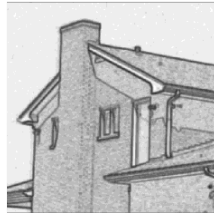
D=16 ve D=32 normalleştirme katsayıları ile elde edilen BG, YA optimizasyon algoritmasına Kapur ve Otsu metotları için ayrı sokulmuştur. Her bir test görüntüsü için optimizasyon sonucunda elde edilen eşik değeri, maksimum değeri, standart sapma, TSO ve işlem süresine yer verilmiştir.

**Tablo 1.** YA Kapur kenar belirleme sonuçları

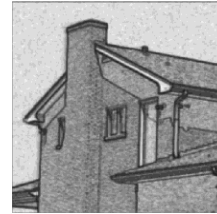
Görüntü	D	Eşik	TSO			SP	Zaman	En iyi değer
			Kırmızı	Yeşil	Mavi			
Ev	16	119	15,77	16,86	16,70	0.0040	12.90	12.64
	32	161	15,99	16,65	16,37	0.0147	12.86	12.53
Lena	16	132	12,35	16,91	16,33	0.0020	12.54	12.83
	32	151	13,09	17,63	16,96	0.0069	13.21	12.72
Su	16	138	13,84	15,55	15,44	0.0128	13.29	13.25
	32	158	14,87	16,87	16,50	0.0042	13.58	13.02



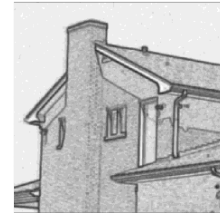
(a). D=16 Benzerlik G.



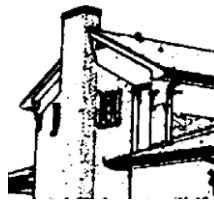
(b). D=32 Benzerlik G.



(c). D=16 Kapur Kenar B.



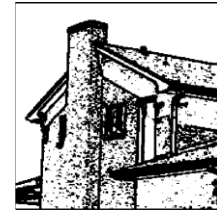
(d). D=32 Otsu Kenar B.



(e). D=16 Kapur Kenar B.



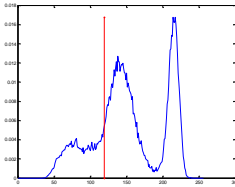
(f). D=32 Otsu Kenar B.



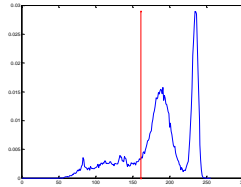
(g). D=16 Kapur Kenar B.



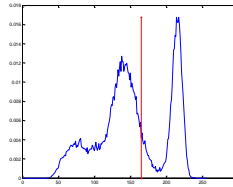
(h). D=32 Otsu Kenar B.



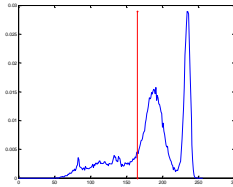
(i). D=16 Kapur Kenar B.



(j). D=32 Otsu Kenar B.



(k). D=16 Kapur Kenar B.



(l). D=32 Otsu Kenar B.

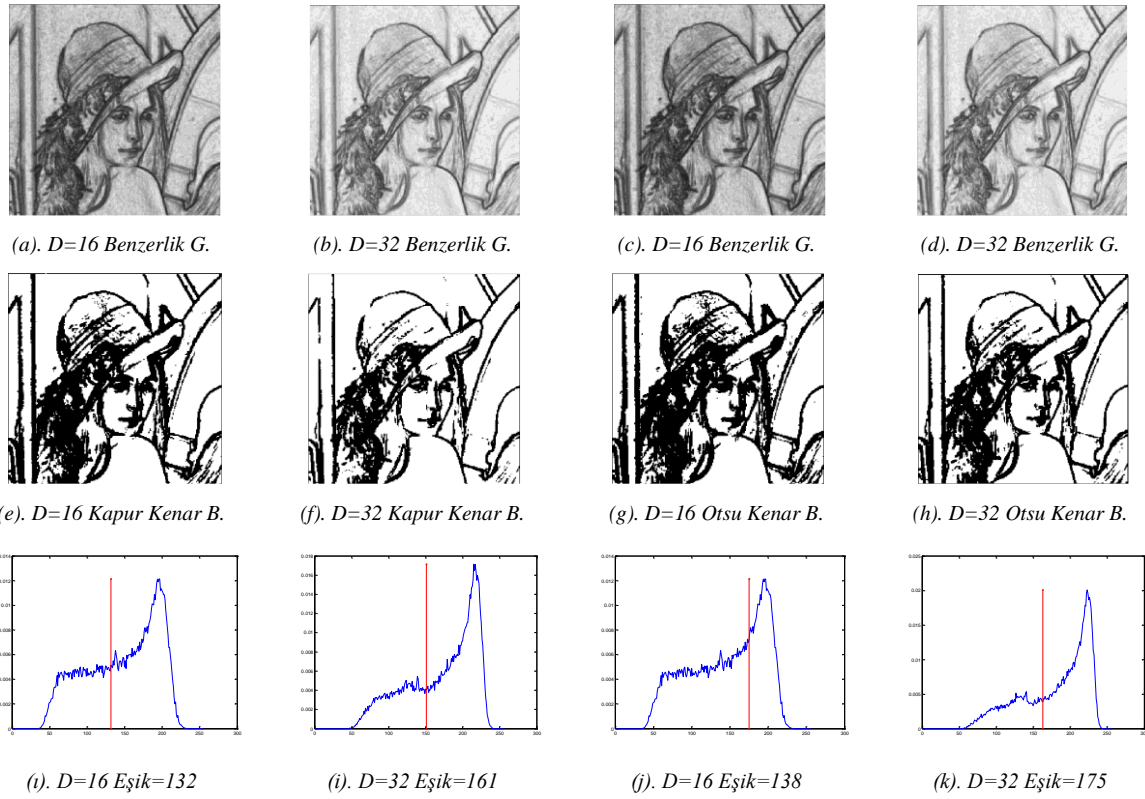
**Şekil 5.** YA Ev görüntüsü için Kapur ve Otsu kenar belirleme görüntüleri

Tablo 1 ve Tablo 2'nin sayısal sonuçlarını karşılaştırıldığında Kapur yönteminin TSO değerlerinin yüksek çıkması orijinal görüntüye yakınlık derecesinin daha fazla olduğunu göstermektedir. Standart sapma değerleri küçük olması ise optimizasyon sonucu üretilen uygunluk değerlerinin birbirine daha yakın sonuçlar ürettiğini belirtmektedir. Ayrıca, bütün test görüntülerinde Kapur yöntemi Otsu yöntemine göre işlem süresi küçük çıkmıştır.

**Tablo 2.** YA Otsu kenar belirleme sonuçları

Görüntü	D	Eşik	TSO			SP	Zaman	En iyi değer
			Kırmızı	Yeşil	Mavi			
Ev	16	165	11,37	13,60	13,50	0.0703	14.37	1662.7
	32	163	15,82	16,54	16,27	0.1993	16.08	1242.2
Lena	16	138	11,93	16,47	16,14	0.1290	15.07	1947.5
	32	175	12,39	16,95	16,37	0.2140	13.20	1573.7
Su	16	142	13,70	15,35	15,31	0.1037	14.16	2268.1
	32	166	14,53	16,45	16,13	0.0014	16.92	1897.9

Şekil 5'de ev görüntüsü için kenar belirleme sonuçlarına yer verilmiştir. Şekil 5 (f) Kapur ve (h) Otsu kenar belirleme sonuçlarında gürültü çok az olup, kenarlar yakalanmıştır. Şekil 5 (i), (j) ve (k) da benzerlik görüntülerinin histogramları üzerinde YA ile elde edilen eşik değerleri gösterilmiştir. Şekil 6 (f) Kapur kenar belirleme sonuçları temiz ve belirgin kenar olarak göze çarpmaktadır.



**Şekil 6.** YA Lena görüntüsü için Kapur ve Otsu kenar belirleme görüntüleri

YA'nın uygunluğunu ve performansını test etmek amacıyla yerçekimi arama algoritması (YAA) ve Harmoni Arama Algoritması (HAA) ile karşılaştırılmıştır [30,31]. Tablo 3'de YA optimizasyon algoritması literatürdeki diğer algoritmalara göre TSO ve SP değerleri için Kapur ve Otsu yönteminde iyi sonuçlar vermiştir. TSO ortalama değer bakımından HAA Ev Otsu 16, Lena Kapur 32, Lena Otsu 32 çalışmalarında YAA Su Kapur 32 çalışmasında diğerlerinden farklı sonuçlar vermiştir. SP değerinin küçük çıkması kullanılan algoritmanın performansının iyi olduğunu göstermektedir. SP küçük değer sıralaması, YA, HAA ve YAA olarak elde edilmiştir.

**Tablo 3.** YA Kenar belirleme sonuçları literatür karşılaştırılması(TSO, SP, En iyi)

Görüntü	Yöntem	D değeri	YAA			HAA			YA		
			TSO	SP	En iyi	TSO	SP	En iyi	TSO	SP	En iyi
Ev	Kapur	16	16,44	0,0365	12,64	16,44	0,0272	12,64	<b>16,44</b>	<b>0,0040</b>	<b>12,64</b>
		32	16,33	0,0269	12,53	16,33	0,0204	12,53	<b>16,33</b>	<b>0,0147</b>	<b>12,53</b>
	Otsu	16	12,82	0,1807	1662,7	11,94	0,0813	1662,7	<b>12,82</b>	<b>0,0703</b>	<b>1662,7</b>
		32	16,21	0,6587	1242,2	16,21	0,6029	1242,2	<b>16,21</b>	<b>0,1993</b>	<b>1242,2</b>
Lena	Kapur	16	15,19	0,0153	12,83	15,19	0,0316	12,83	<b>15,19</b>	<b>0,0020</b>	<b>12,83</b>
		32	15,21	0,0236	12,60	7,66	0,0290	12,77	<b>15,89</b>	<b>0,0069</b>	<b>12,72</b>
	Otsu	16	14,78	0,6604	1947,5	14,78	0,8440	1775,1	<b>14,78</b>	<b>0,0090</b>	<b>1947,5</b>
		32	15,18	0,0954	1573,7	15,08	0,0903	1573,6	<b>15,23</b>	<b>0,0140</b>	<b>1573,7</b>
Su	Kapur	16	14,94	0,0963	13,25	14,94	0,0976	13,25	<b>14,94</b>	<b>0,0128</b>	<b>13,25</b>
		32	15,37	0,0103	12,97	16,08	0,0542	13,03	<b>16,08</b>	<b>0,0042</b>	<b>13,02</b>
	Otsu	16	14,78	0,1693	2268,1	14,78	1,0586	2268,1	<b>14,78</b>	<b>0,1037</b>	<b>2268,1</b>
		32	15,70	0,0189	1897,9	15,70	1,0184	1897,9	<b>15,70</b>	<b>0,0014</b>	<b>1897,9</b>

**Tablo 4.** YA Kenar belirleme sonuçları literatür karşılaştırılması(Eşik, Zaman)

Görüntü	Yöntem	D değeri	Eşik değeri			Geçen Zaman		
			YAA	HAA	YA	YAA	HAA	YA
Ev	Kapur	16	119	119	<b>119</b>	31,12	14,32	<b>12,90</b>
		32	161	161	<b>161</b>	28,62	14,25	<b>12,86</b>
	Otsu	16	165	165	<b>165</b>	18,36	17,53	<b>14,37</b>
		32	163	163	<b>163</b>	16,27	17,60	<b>16,08</b>
Lena	Kapur	16	132	132	<b>132</b>	31,68	16,16	<b>12,54</b>
		32	159	51	<b>151</b>	15,71	15,18	<b>13,21</b>
	Otsu	16	138	138	<b>138</b>	15,15	14,59	<b>14,07</b>
		32	166	166	<b>175</b>	15,35	14,68	<b>13,20</b>
Su	Kapur	16	138	138	<b>138</b>	35,93	15,22	<b>13,29</b>
		32	173	158	<b>158</b>	16,28	17,04	<b>13,58</b>
	Otsu	16	142	142	<b>142</b>	15,23	14,98	<b>14,16</b>
		32	166	166	<b>166</b>	15,75	14,14	<b>13,92</b>

Tablo 4'de görüldüğü gibi HAA Lena Kapur 32 çalışmasında eşik 51 değeri gibi çok küçük bir değer elde edilmesi görüntünü kirli ve kalın kenarlar elde edilmesi ile sonuçlanmıştır. YAA Su Kapur 32 çalışmasında 173 gibi büyük bir eşik değeri bulunması ise kenar kaybolması ve çok ince kenar elde edilmesi şeklinde kötü sonuçlanmasını sağlamıştır. İşlem sürelerinde en kısa zamanda YA'nın sonuç bulduğu görülmektedir.

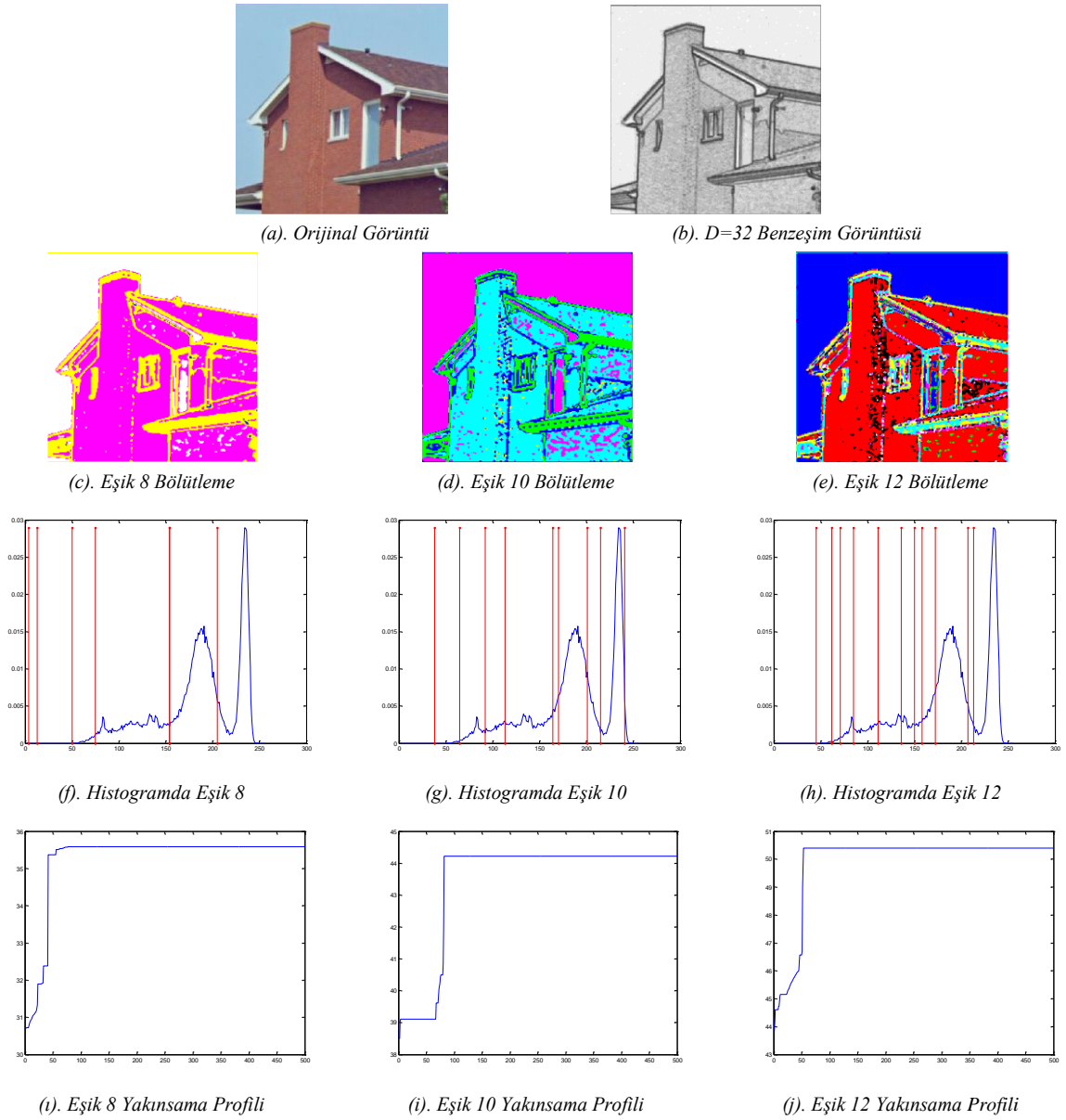
### B. BÖLÜTLEME SONUÇLARI

Bu bölümde, renkli görüntülerden elde edilen BG görüntüsü Denklem (3) üstel fonksiyonu ve  $D=32$  normalleştirme katsayısı sabit kabul edilerek bölütleme gerçekleştirilmiştir. Bu iki değişken çalışmalarında BG görüntülerinde gürültü oranı az ve daha belirgin sonuçlar vermiştir. BG görüntüsü üzerine Kapur ve Otsu metotları Eşik=8, Eşik=10 ve Eşik=12 için YA optimizasyon tekniği çalıştırılmıştır.

**Tablo 5. YA Kapur ve Otsu bölütleme sonuçları**

Görüntü	Yöntem	Çoklu Eşikler		TSO			SP	Zaman	En iyi değer
				Kırmızı	Yeşil	Mavi			
Ev	Kapur	Eşik 8	4 13 50 75 154 154 205 255	11,83	12,86	10,48	1.136	22,90	35,58
		Eşik 10	38 65 92 113 164 170 201 215 241 255	10,21	9,48	7,93	1,82	24,72	44,22
		Eşik 12	45 62 71 85 111 136 150 158 172 207 213 243	10,11	9,82	13,20	1,52	27,38	50,39
	Otsu	Eşik 8	47 60 134 154 160 228 244 255	10,32	8,49	10,47	3,06	24,80	3750,7
		Eşik 10	37 51 95 113 123 156 167 174 202 226	9,66	12,38	8,98	5,71	26,77	3668,8
		Eşik 12	54 59 62 70 71 94 117 139 144 146 160 176	11,22	10,54	9,04	2,5	29,18	4721,6
Lena	Kapur	Eşik 8	62 133 170 176 190 240 255 255	8,17	9,40	10,29	0,311	22,41	35,24
		Eşik 10	39 42 79 143 157 187 204 225 255 255	9,73	9,82	9,44	0,781	25,11	43,62
		Eşik 12	75 84 117 137 138 151 157 189 197 229 237 255	10,93	10,23	10,69	0,358	27,53	49,88
	Otsu	Eşik 8	46 57 72 154 159 171 237 247	11,38	11,46	9,27	1,70	25,08	4046,6
		Eşik 10	34 54 71 71 89 119 161 178 183 193	8,53	9,54	10,64	1,36	26,23	4670,3
		Eşik 12	41 51 73 77 78 81 116 150 151 173 185 255	8,38	10,59	9,64	1,58	30,12	5101,2
Su	Kapur	Eşik 8	4 13 50 75 154 154 205 255	11,21	9,22	6,75	0,61	22,41	37,55
		Eşik 10	30 66 107 120 173 197 255 255 255 255	8,39	9,90	6,76	0,81	24,38	44,24
		Eşik 12	43 52 60 71 77 110 119 164 199 219 238 255	9,85	9,76	11,01	1,064	27,72	53,08
	Otsu	Eşik 8	2 17 32 115 152 187 201 251	8,39	9,32	6,47	1,27	25,82	5208,3
		Eşik 10	51 56 57 65 112 136 156 157 170 198	11,18	9,30	12,42	4,6	25,11	5607,2
		Eşik 12	12 19 35 112 123 159 164 170 255 255 255 255	11,11	10,44	11,68	3,8	30,21	6329,7

Uygulamada çalışılan literatürdeki görüntülerin doğal ve çok renkli olmalarından dolayı bölgelerin daha iyi tespiti için eşik değerleri yüksek değerler olarak seçilmiştir.

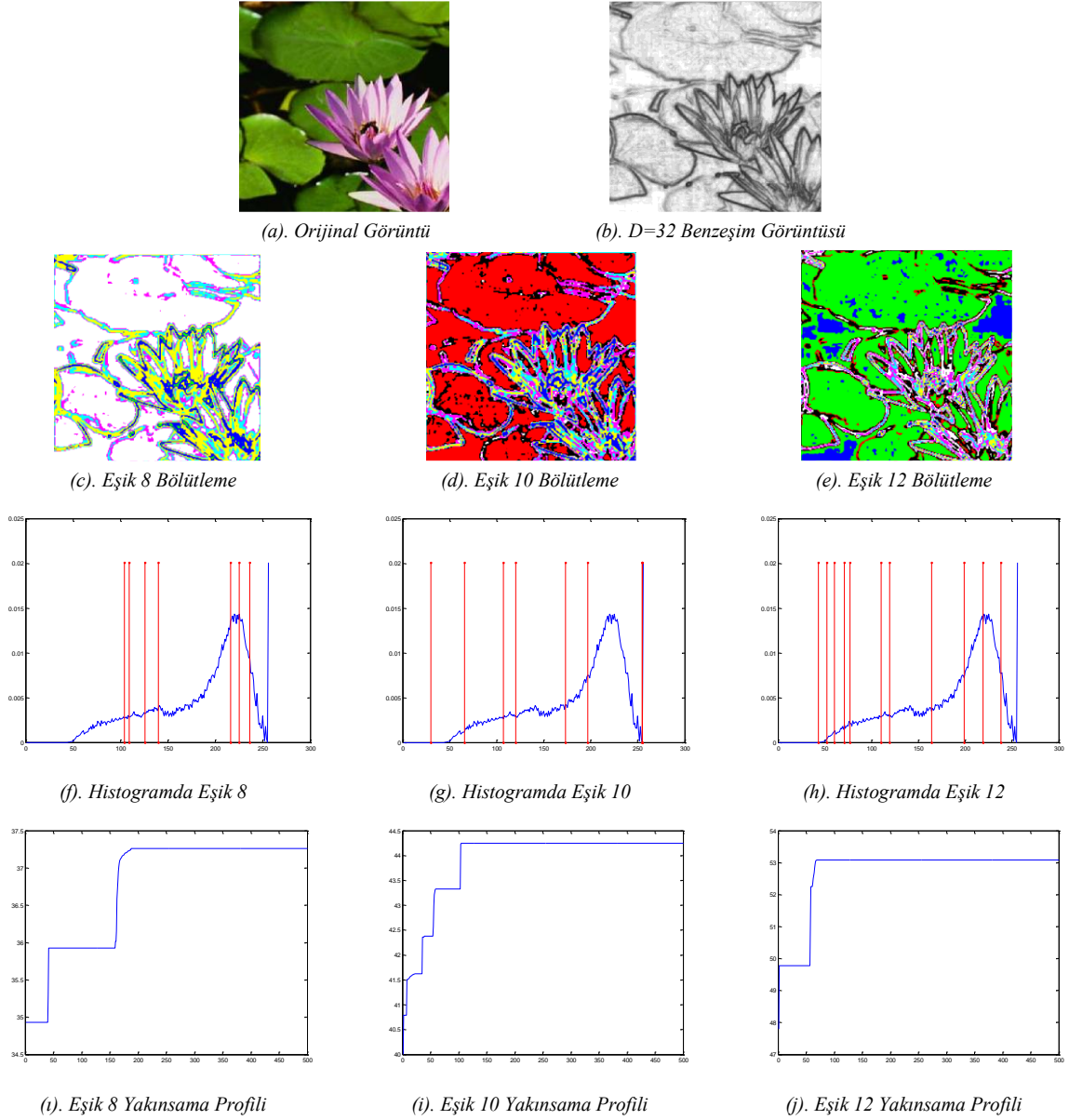


**Şekil 7.** YA Ev görüntüsü için Kapur bölütleme görsel sonuçları

Tablo 5'in sayısal sonuçlarını karşılaştırdığımızda, Kapur yönteminin de TSO değerleri yüksek çıkmıştır. Bu durum orijinal görüntüye yakınlığın daha fazla olduğunu, standart sapma değerleri küçük olması elde edilen uygunluk değerlerinin birbirine daha yakın olduğunu benzer sonuçlar ürettiğini göstermektedir. Kapur yönteminin sonuçlardaki üstünlüğü, çoklu eşik kullanımında daha belirgin olarak ortaya çıkmıştır. Eşik sayısı arttıkça YA optimizasyon tekniğindeki eşik arama işlem süresi artmıştır.

Şekil 7'de 8,10 ve 12 eşik değerlerinde ev görüntüsündeki bölütleme sonuçlarına yer verilmiştir. Eşik sayısı arttıkça bölgelerin belirginliği ve farklı bölgeleri yakalama özelliği artmıştır. Şekil 7. (a), (i) ve (j)'de YA optimizasyon tekniğinin yakınsama hızının yüksek olduğu genel olarak 500 döngülü çalıştırmada 100 ve yakın değerlerinde sonucu elde ettiği görülmektedir. Ayrıca Şekil 7. (g) ve (h)'da eşik değerlerin BG histogramı üzerinde uygun yayıldığı tespit edilmiştir.





**Şekil 8.** YA Su görüntüsü için Kapur bölütleme görsel sonuçları

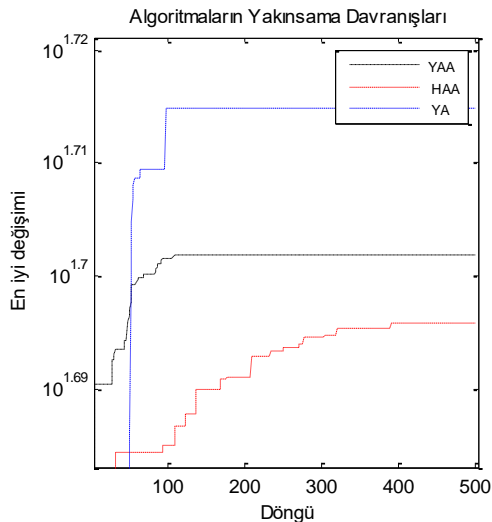
Eşik sayısı arttıkça bölgelerin belirginliği ve farklı bölgeleri yakalama özelliğinin arttığını Şekil 8. (e)'de ki nilüfer yaprak damarlarının dahi yakalanması ile gözlemlenmektedir. Şekil 8. (f), (g) ve (h)'de YA optimizasyon tekniğinin yakınsama hızı yüksek olduğu tespit edilmiştir.

YA'nın uygunluğunu ve performansını karşılaştırmak amacıyla literatürdeki HAA ve YAA ile karşılaştırılmıştır [30,31]. Tablo 6 incelendiğinde YA optimizasyon algoritmasının literatürdeki diğer algoritmalara göre işlem süresi ve amaç fonksiyonunun maksimum değeri bakımından iyi sonuçlar vermiştir. YA optimizasyon tekniğinin güçlü özelliklerinden biri de BG histogramı üzerinde eşikler uygun dağılmıştır, özellikle 230 değerlerinden sonra da eşik ataması yapıldığı görülmektedir.

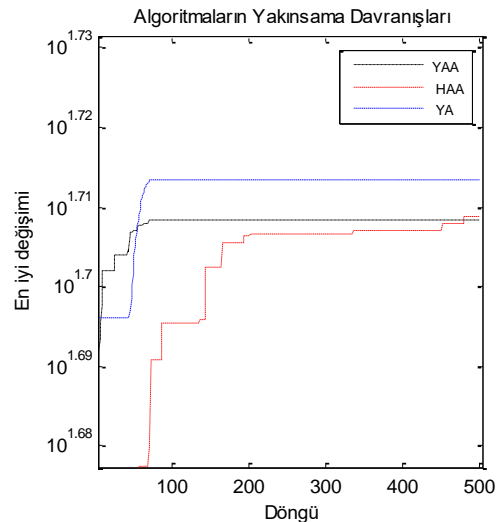
Şekil 9'da YA sonuçlarının HAA ve YAA ile yakınsama hızı karşılaştırmasının görsel sonuçlarına yer verilmiştir. Ev ve su görüntülerinde YA, HAA'ya göre erken sonuca ulaşmıştır. Aynı şekilde YAA ve HAA ya göre en iyi sonuç değeri belirgin farklar ile YA optimizasyon tekniğine aittir.

**Tablo 6.** YA Bölütleme sonuçlarının literatür karşılaştırılması

Görüntü	YAA			HAA			YA			
	Çoklu Eşikler	Zaman	En iyi	Çoklu Eşikler	Zaman	En iyi	Çoklu Eşikler	Zaman	En iyi	
Ev	8	20 50 87 109 138 169 196	2,27	37,33	22 50 104 127 165 180 228	2,68	37,07	4 13 50 75 154 154 205 255	2,29	37,63
	10	29 48 79 101 128 149 169 192 220	2,56	43,96	12 36 64 83 118 152 162 195 203	2,75	41,37	38 65 92 113 164 170 201 215 241 255	2,47	44,22
	12	17 35 50 73 99 118 141 162 183 206 222	4,46	50,32	11 24 48 52 90 113 142 152 178 203 221	2,60	47,23	45 62 71 85 111 136 150 158 172 207 213 243	2,73	50,39
Lena	8	26 51 86 115 145 175 206	3,52	35,04	28 48 82 118 160 172 217	2,78	37,28	27 102 108 149 184 201 216 238	2,24	38,24
	10	21 51 78 105 126 147 165 192 214	3,74	43,52	20 50 76 113 144 161 184 208 224	2,83	44,11	1 45 89 115 120 122 181 240 253 255	2,51	45,91
	12	16 34 51 76 100 120 141 162 179 199 218	2,98	49,68	14 24 50 74 88 117 140 155 181 220 230	2,16	48,91	75 84 117 137 138 151 157 189 197 229 237 255	2,75	49,88
Su	8	43 83 111 140 174 205 227	2,40	38,68	38 77 97 104 158 189 216	2,59	37,31	52 103 130 151 172 214 229 245	2,24	39,05
	10	19 44 81 108 136 160 180 201 219	2,64	44,19	27 40 86 107 123 137 158 202 212	2,83	43,86	17 26 60 120 152 163 178 213 255 255	2,43	44,24
	12	17 44 64 84 106 127 148 166 187 205 225	2,84	51,52	23 40 71 83 109 136 188 208 234 239 245	2,95	48,35	43 52 60 71 77 110 119 164 199 219 238 255	2,72	53,08



(a) Ev;  $D=32$ , Eşik=12, Kapur



(b) Su;  $D=32$ , Eşik=12, Kapur

**Şekil 9.** YA'nın diğer optimizasyon teknikleri ile yakınsama hızı karşılaştırması



## IV. SONUÇ

Renkli görüntüden iki boyutlu bağıntı matrisine dayalı BG elde edildikten sonra, bu görüntü üzerinde en uygun eşik değeri için amaç fonksiyonu olarak Kapur ve Otsu'nun metotları kullanılmıştır. YA optimizasyonu çoklu eşik bölütleme metoduna uygulanarak, sınıflara ayrılmış görüntü elde edilmiştir. Algoritmanın performansını test etmek için bölütleme kalitesini elde etmek de kullanılan TSO ve SP tercih edilmiştir. Uygulama sonuçları kendi içinde, görsel kendi içinde ve bilinen literatürler arasında olmak üzere üç farklı yaklaşım ile test edilmiştir.

Üstel fonksiyonda gürültü oranı az bir BG görüntüsü elde edildiği için kenar belirleme ve bölütleme görsel sonuçlarına olumlu yansımıştır. Kapur yönteminde TSO değerlerinin yüksek çıkması orijinal görüntüye yakınlığın daha fazla olduğu, standart sapma değerlerinin küçük olması ise sonuçların birbirine daha yakın olduğunu ve benzer sonuçlar ürettiğini göstermektedir. Bundan dolayı, Kapur yönteminde elde edilen eşiklerin histogram üzerinde uygun yayılımı tespit edilmiştir.

YA'nın uygunluğunu ve performansını karşılaştırmak amacıyla literatürdeki HAA ve YAA ile karşılaştırılmıştır. YA, diğer algoritmalara göre işlem süresi ve amaç fonksiyonunun maksimum değeri bakımından iyi sonuçlar vermiştir. YA optimizasyon tekniğinin güçlü özelliklerinden biri de BG histogramı üzerinde eşikler uygun dağılmıştır. YA yakınsama hızı, doğruluk ve performansı ile sonuçların doğruluğunu kanıtlamıştır.

## V. KAYNAKLAR

- [1] E. Navon, O. Miller, O. A. Averbuch *Image and vision computing* **23(1)** (2005) 69-85.
- [2] P. Y. Yin *Applied mathematics and computation* **184(2)** (2007) 503-513.
- [3] E. Cuevas, D. Zaldivar, M. Pérez-Cisneros *Expert Systems with Applications* **37(7)** (2010) 5265-5271.
- [4] B. Akay *Applied Soft Computing* **13(6)** (2013) 3066-3091.
- [5] J. Zhang, H. Li, Z. Tang, Q. Lu, X. Zheng, J. Zhou *Mathematical Problems in Engineering* **2014(1)** (2014) 1-12.
- [6] I. Brajevic, M. Tuba *Springer International Publishing* **516(1)** (2014) 115-119.
- [7] S. A. Underwood, J. K. Aggarwal *Computer Graphics and Image Processing* **6(1)** (1977) 1-24.
- [8] H. D. Cheng, ve Y. Sun *IEEE Transactions on Image Processing* **9(12)** (2000) 2071-2082.
- [9] T. Kurban, P. Civicioglu, R. Kurban, E. Besdok *Applied Soft Computing* **23(1)** (2014) 128-143.
- [10] S. Sarkar, S. Das, S. S. Chaudhuri *Pattern Recognition Letters* **54(1)** (2015) 27-35.
- [11] S. Patra, R. Gautam, A. Singla *Applied Soft Computing* **23(1)** (2014) 122-127.

- [12] B. Green, [http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html) (*Erişim tarihi: 15<sup>th</sup> of January, 2017*).
- [13] R. C. Gonzalez, R. E. Woods, R. E., *Digital image processing*, Beijing, (2011).
- [14] D. Marr, E. Hildreth *Proceedings of the Royal Society of London B: Biological Sciences* **207(1167)** (1980) 187-217.
- [15] J. Canny *IEEE Transactions on pattern analysis and machine intelligence* **6(1)** (1986) 679-698.
- [16] Y. Yu, J. Lu, J. Zheng *Journal of Tsinghua* **45(10)** (2005) 1139-1149.
- [17] S. Di Zeno *Computer vision, graphics, and image processing* **33(1)** (1986) 116-125.
- [18] A. N. Evans, X. U. Liu *IEEE Transactions on Image Processing* **15(6)** (2006) 1454-1463.
- [19] M. Bastan, S. S. Bukhari, T. M. Breuel *arXiv preprint arXiv* **1609(3415)** (2016) 1-14.
- [20] R. Shahverdi, M. Tavana, A. Ebrahimnejad, K. Zahedi, H. Omranpour *Cybernetics and Systems* **47(3)** (2016) 161-179.
- [21] F. Katircioglu, *Renkli görüntülerin bağıntı matrisine dayalı ayrıştırılması ve kenar algılama*, Yüksek Lisans Tezi, Abant İzzet Baysal Üniversitesi Fen Bilimleri Enstitüsü, (2007).
- [22] R. Demirci, F. Katircioglu, *Segmentation of color images based on relation matrix*, In 2007 IEEE 15th Signal Processing and Communications Applications, Eskişehir-Türkiye, (2007).
- [23] R. Demirci *AEU-International Journal of Electronics and Communications* **60(6)** (2006) 435-442.
- [24] D. Oliva, E. Cuevas, G. Pajares, D. Zaldivar, D. M. Perez-Cisneros *Journal of Applied Mathematics* **2013(1)** (2013) 1-24.
- [25] N. Otsu *IEEE Transactions on Systems, Man, Cybernetics* **9(1)** (1979) 62–66.
- [26] J. N. Kapur, P. K. Sahoo, A. K. C. Wong *Computer Vision, Graphics, & Image Processing* **29(3)** (1985) 273–285.
- [27] C. W. Reynolds *ACM SIGGRAPH computer graphics* **21(4)** (1987) 25-34.
- [28] S. Mirjalili *Neural Computing and Applications* **27(4)** (2016) 1053-1073.
- [29] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi *Information sciences* **179(13)** (2009) 2232-2248.
- [30] Z. W. Geem, J. H. Kim, G. V. Loganathan *Simulation* **76(2)** (2001) 60-68.
- [31] X. S. Yang, L. Press, *Nature-Inspired Metaheuristic Algorithms Second Edition*, Luniver, United Kingdom, (2010).