

Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Renkli Görüntülerin Çok Seviyeli Eşiklenmesi ve Sınıflandırılması

Ahmet Selim KAHRAMAN [a,*], Taymaz Rahkar FARSHİ [a], Recep DEMİRCİ [a]

[a] *Bilgisayar Mühendisliği Bölümü, Teknoloji Fakültesi, Gazi Üniversitesi, Ankara, TÜRKİYE*
** Sorumlu yazarın e-posta adresi: ahmetselimkahraman@gmail.com*

ÖZET

Bu çalışmada renkli görüntüler için çok seviyeli eşikleme esaslı yeni sınıflandırma algoritması önerilmiştir. Öncelikle renkli görüntülerin her bir kanalının histogramı ve arı algoritması kullanılarak eşikler tespit edilmiştir. İkinci aşamada elde edilen eşik değerleri RGB renk uzayının bölümlenmesinde kullanılmıştır. Böylece ortaya çıkan alt küpler içerisinde kalan pikseller aynı sınıfa atanmış ve sonuçlar alınmıştır.

Anahtar Kelimeler: *Renkli görüntü, çok seviyeli eşikleme, sınıflandırma.*

Multilevel Color Image Thresholding and Classification

ABSTRACT

In this study, a new classification algorithm based on multilevel thresholding for color images has been proposed. Initially, thresholds for each channel of color images are determined by using histograms and bee algorithm. Then, the threshold values obtained are used for partition of RGB color space. Thus, pixels located in the relevant sub cubes were assigned in the same cluster and subsequently the results have been obtained.

Keywords: *Color image, multilevel thresholding, classification*

Geliş: 12/02/2018, Düzeltme: 15/05/2018, Kabul: 20/06/2018

I. GİRİŞ

Görüntü ayrıştırma, görüntü analizi ve örüntü tanıma işlemlerindeki ilk ve en önemli adımdır[1]. Ayrıştırma işleminin temel amacı, görüntü üzerinde farklı konumlardaki pikselleri gruplayarak anlamlı ve yararlı bilgi elde etmektir [2]. Söz konusu hedefe ulaşmak için günümüze kadar önerilen yöntemler ya kümeleme ya da eşikleme esaslı olmuştur. K-Ortalamalar[4] veya C-Ortalamalar[5] metotları görüntü ayrıştırmak için sıkça kullanılan iki algoritmadır. İlgili yaklaşımlar güçlü olmasına rağmen, hesaplama zamanları görüntü boyutuna bağlı olmakta ve gerçek zamanlı görüntü işlemede yetersiz kalmaktadırlar. Ayrıca, söz konusu yöntemlerin performansı küme merkezlerinin rastgele seçilmesi nedeni ile düşük kalmaktadır. İlave olarak ise küme sayısının kullanıcı tarafından seçilmesi en büyük dezavantajıdır.

Diğer taraftan Otsu[6] ve Kapur[7] algoritmaları gri ölçekli görüntüler için en çok tercih edilen eşikleme yöntemleri olup, iki veya çok seviyeli eşikleme süreçlerinde genişletilebilme imkânları mevcuttur. Ancak kullanıcı tarafından seçilen eşik sayısı artıkça, hesaplanma maliyeti logaritmik olarak artmaktadır. Bununla birlikte, optimizasyon algoritmalarının sürece dahil edilmesi ile hesaplamanın zaman maliyeti azaltılabilmektedir. Pal ve arkadaşları örümcek maymun optimizasyon algoritmasını söz konusu algoritmaların amaç fonksiyonlarına uygulayarak çok seviyeli eşikleme yapmışlardır[8]. Ayrıca Singh ve arkadaşları Otsu ve Kapur fonksiyonlarını ateş böceği ve sosyal örümcek algoritmalarıyla birleştirmişler ve gri ölçekli görüntülerde test etmişlerdir[9]. Son yıllarda, El-Aziz ve arkadaşları gri seviyeli görüntülerde çok seviyeli Otsu ve Kapur eşik değerlerini bulmak adına balina ve ateş böceği optimizasyon algoritmalarını önermişlerdir [10].

Görüldüğü üzere, çok seviyeli eşikleme algoritmaları sadece gri ölçekli görüntülerde etkili olmuştur ve birbirinden bağımsız üç bileşeni olan renkli görüntülerde yetersiz kalmıştır. Bununla birlikte, son yıllarda renkli görüntülerin sınıflandırılması için sadece tek eşik ve üç boyutlu renk uzayı yardımıyla yeni bir yaklaşım önerilmiştir[11]. İlgili çalışmada her bir renk kanalı için Otsu ve Kapur yöntemleri ile eşik değerleri bulunduktan sonra, üç boyutlu renk uzayı 8 adet alt prizmaya ayrıştırılmış ve her prizma içinde kalan pikseller aynı sınıfa dâhil edilmiştir.

Ancak söz konusu algoritmada oluşturulan alt prizma sayısı az ve hacmi büyük olduğundan, resimde elde edilen bölgelerin homojenliği düşük kalmıştır. Bu çalışmada Demirci ve arkadaşları[11] tarafından önerilen yöntem, yapay arı koloni algoritması ile çok değişkenli Otsu ve Kapur amaç fonksiyonlarıyla birleştirilmiştir. Böylece renk uzayındaki küplerin hacimleri küçültülmüş, oluşturulan sınıf sayısı ve sınıflandırma performansı artırılmıştır.

II. EŞİKLEME YÖNTEMLERİ

Gri seviyeli bir görüntünün renk aralığı $\{0,\ 1,\ 2,\ldots,\ (L-1)\}$ şeklinde olup, i. gri seviyenin olasılığı

$$p_i = h_i / (M \times N) \tag{1}$$

şeklinde tanımlanabilir. Burada M ve N görüntünün boyutlarını, h_i ise i. seviyede griye karşılık gelen piksel değerinin tekrar sayısını temsil etmektedir $(0 \le i \le (L-1))$. Gri ölçekli görüntünün histogramı

kullanılarak, *m* adet eşik sayısı bulmak mümkündür ki böylece *m+1* adet sınıf oluşturulur. Bu hedefe uygun olarak tanımlanan *m* değişkenli amaç fonksiyonu maksimum yapılarak ilgili eşik değerleri tespit edilir. Literatürde en sık kullanılan eşikleme amaç fonksiyonları ise Kapur'un entropi yaklaşımı ve Otsu'nun sınıflar arası varyans tekniği olmuştur. Kapur yöntemine göre tek eşik bulmak için önerilen amaç fonksiyonu

$$J(t_1) = H_0 + H_1 \tag{2}$$

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0},\ \omega_0 = \sum_{i=0}^{t_1-1} p_i$$

ve

$$H_1 = -\sum_{i=t_1}^{L-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1},\ \omega_1 = \sum_{i=t_1}^{L-1} p_i$$

şeklindedir. Burada H_0 ve H_1 histogramın kısmi entropilerini temsil etmektedir. Eşt. (2)'de verilen t_1 değeri amaç fonksiyonunu maksimum yapan değerdir. Diğer taraftan Otsu'nun sınıflar arası varyans algoritmasında önerilen amaç fonksiyonu

$$J(t_1) = \sigma_0 + \sigma_1 \tag{3}$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_T)^2$$

ve

$$\sigma_1 = \omega_1(\mu_1 - \mu_T)^2$$

ve

$$\omega_0 = \sum_{i=0}^{t_1-1} p_i,\ \mu_0 = \sum_{i=0}^{t_1-1} \frac{i\,p_i}{\omega_0} \quad \omega_1 = \sum_{i=t_1}^{L-1} p_i,\ \mu_1 = \sum_{i=t_1}^{L-1} \frac{i\,p_i}{\omega_1}$$

ve $\mu_T = \sum_{i=0}^{L-1} i\,p_i$

şeklindedir. Burada μ_T görüntünün ortalama yoğunluğunu temsil etmektedir ve

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T \text{ ve } \omega_0 + \omega_1 = 1 \tag{4}$$

bağıntısı mevcuttur. Öte yandan, Kapur'un entropi kriteri çok seviyeli eşikleme için

$$J(t_1, t_1, ..., t_m) = H_0 + H_1 + H_2 + ... + H_m \tag{5}$$

ve

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0},\ \omega_0 = \sum_{i=0}^{t_1-1} p_i$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1},\ \omega_1 = \sum_{i=t_1}^{t_2-1} p_i$$

$$H_2 = -\sum_{i=t_2}^{t_3-1} \frac{p_i}{\omega_2} \ln \frac{p_i}{\omega_2},\ \omega_2 = \sum_{i=t_2}^{t_3-1} p_i$$

$$\vdots$$

$$H_m = -\sum_{i=t_m}^{L-1} \frac{p_i}{\omega_m} \ln \frac{p_i}{\omega_m},\ \omega_m = \sum_{i=t_m}^{L-1} p_i$$

şeklinde genişletilebilir. Böylece çok seviyeli eşikleme problemi, *m* boyutlu bir optimizasyon problemi olarak tanımlanabilir. Gri seviyeli görüntü için Eşt. (5)'i maksimum yapan [t_1, t_2, ..., t_m] değerleri optimal eşiklerdir[12]. Otsu yaklaşımında, herhangi bir gri seviyeli görüntüde çok seviyeli eşikler, Eşt. (3)'ün

$$J(t_1, t_2, ..., t_m) = \sigma_0 + \sigma_1 + \sigma_2 + ... + \sigma_m \tag{6}$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_T)^2$$

$$\sigma_1 = \omega_1(\mu_1 - \mu_T)^2$$

$$\sigma_2 = \omega_2(\mu_2 - \mu_T)^2$$

$$\sigma_m = \omega_m(\mu_m - \mu_T)^2$$

$$\omega_0 = \sum_{i=0}^{t_1-1} p_i,\ \mu_0 = \sum_{i=0}^{t_1-1} \frac{i\, p_i}{\omega_0} \qquad \omega_1 = \sum_{i=t_1}^{t_2-1} p_i,\ \mu_1 = \sum_{i=t_1}^{t_2-1} \frac{i\, p_i}{\omega_1}$$

$$\vdots$$

$$\omega_m = \sum_{i=t_m}^{L-1} p_i,\ \mu_m = \sum_{i=t_m}^{L-1} \frac{i\, p_i}{\omega_m}$$

şeklinde genişletilmesi ile elde edilir. Başka bir ifade ile sınıflar arası varyansı maksimum yapan eşik değerleri çözüm kümesidir. Ayrıca her iki durumda da, $t_1 < t_2 < t_3 < ... < t_m$ şartları sağlanmalıdır [12].

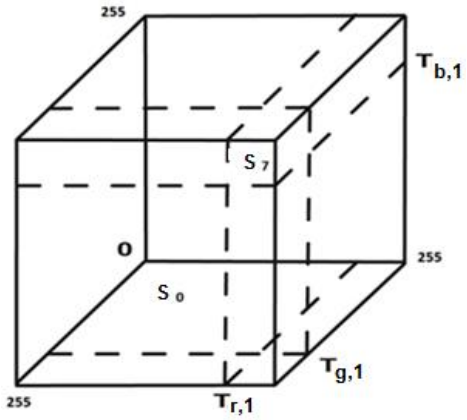
III. RENK UZAYININ BÖLÜMLENMESİ

Detayları daha önce verilen Otsu ve Kapur yöntemleriyle gri seviyeli görüntülerin ikili veya çoklu kümeleme işlemi yapılabilmektedir. Bununla birlikte, renkli görüntüler üç ayrı kanaldan meydana geldiğinden, her bir kanal kendi eşik değerine sahip olacaktır. Her bir kanal için eşikler elde edilmiş olsa bile, söz konusu bilgilerin anlamlı kümeler oluşturmak için kullanımı önemli bir sorundur. Diğer taraftan

Demirci ve arkadaşları [11] tarafından önerilen algoritmada, her kanaldan elde edilen tekli eşik bilgileri Şekil 1. de gösterildiği gibi birleştirilmiştir. Başka bir ifadeyle, renk küpü küçük küplere veya prizmalara bölünmüştür. Daha sonra alt küp veya prizmanın içerisinde yer alan her piksel aynı kümeye veya sınıfa dâhil edilmiştir. Şekil 1. incelediğinde her bir alt küp veya prizmaların hacimlerinin her kanalın eşiğine bağlı olduğu ve aynı olmadığı görülmektedir. Dolayısıyla, alt küplerin veya prizmaların hacimleri, görüntüdeki piksel yoğunluklarının dağılımıyla ilişkilidir. Büyük hacimlere sahip kümelerin homojenliğinin, küçük hacimli kümelere göre daha düşük olacağı açıktır. Eşik sayısı artırılarak her bir küpün hacmi azaltılır ise oluşacak sınıfların homojenliği de artacaktır. Böylece çok seviyeli eşikleme önemli bir çözüm olacaktır. Diğer taraftan, renkli resim için elde edilecek maksimum sınıf sayısı

$$C_{\mathrm{m}} = (m+1)^3 \tag{7}$$

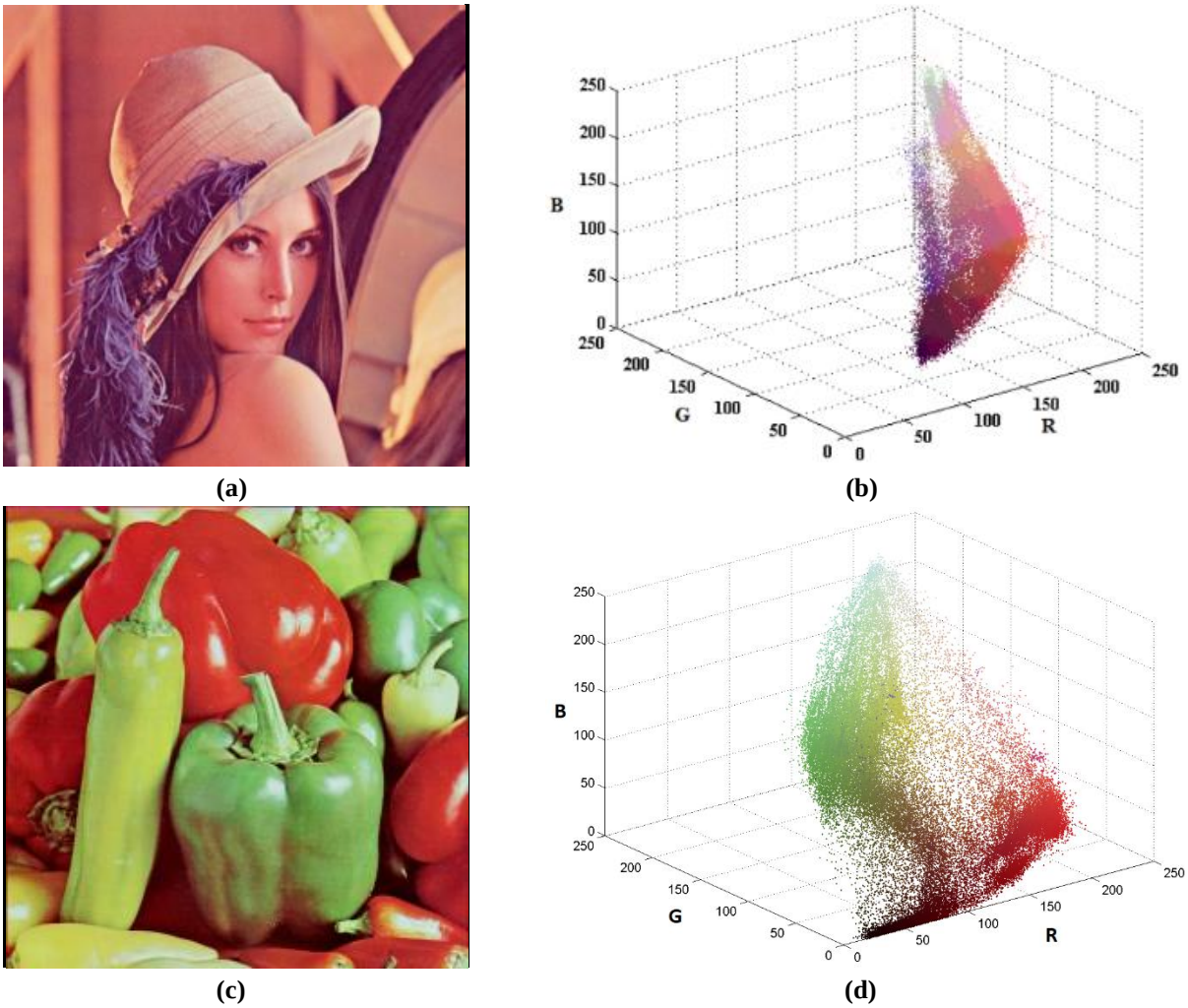
şeklinde olacaktır. Şekil 2(a)'da görüntü işleme alanında referans kabul edilen Lena ve Şekil 2(b)'de renk uzayındaki dağılımı verilmiştir. Ayrıca Şekil 2(c)'de Biber görüntüsü, Şekil 2(d)'de renk uzayındaki dağılımı verilmiştir. Görüntüdeki her bir piksel renk uzayında bir noktaya karşılık gelmektedir. Eşt. (7)'ye göre tek eşik kullanıldığından (*m=1*), sekiz alt küme oluşacaktır. Eşik sayısı, *m = 2* ise $C_{\mathrm{m}} = 27$ ve *m = 3* ise 64 tane alt küme meydana gelecektir. Renk uzayının tek eşikle bölümlenmesi Şekil.1'de gösterilmiş iken, aynı yaklaşım çok seviyeli eşikler için genişletilebilir. Tek eşikle renk uzayının bölümlenme kuralları ve oluşacak sınıf etiketleri Tablo 1 de gösterilmiştir.



Şekil 1. *Üç boyutlu renk uzayı ve atanan sınıflar: m=1* [11].

Tablo 1*. Renk uzayının bölümlenmesi*

Sınıf Etiketi	Kurallar	Atanan Renk
S_0	if (R<=T_r & G<= T_g & B<= T_b)	0x000000
S_1	if (R<=T_r & G<= T_g & B>= T_b)	0x0000FF
S_2	if (R<=T_r & G>= T_g & B<= T_b)	0x00FF00
S_3	if (R<=T_r & G>= T_g & B>= T_b)	0x00FFFF
S_4	if (R>=T_r & G<= T_g & B<= T_b)	0xFF0000
S_5	if (R>=T_r & G<= T_g & B>= T_b)	0xFF00FF
S_6	if (R>=T_r & G>= T_g & B<= T_b)	0xFFFF00
S_7	if (R>=T_r & G>= T_g & B>= T_b)	0xFFFFFF



Şekil 2. (a) *Lena* ***(b)*** *Lena: RGB uzayındaki renk dağılımı* ***(c)*** *Biber* ***(d)*** *Biber: RGB uzayındaki renk dağılımı*

IV. YAPAY ARI KOLONİSİ ALGORİTMASI

Yapay arı koloni algoritması[13] Karaboğa tarafından, doğadaki arıların yiyecek arama davranışlarından esinlenerek tasarladığı bir optimizasyon yaklaşımıdır. Karaboğa, esinlendiği modelde basitlik amacıyla, algoritma üzerinde bazı düzenlemeler yapmıştır. Her besin kaynağının konumu bir çözüme tekabül eder. İşçi arıların sayısı da gözcü arıların sayısına eşit olduğu varsayılmaktadır. Arı kolonisi, işçi arılar, gözcü arılar ve kâşif arılardan meydana gelmektedir. Algoritmanın süreçleri aşağıdaki gibidir.

A. BAŞLANGIÇ YİYECEK KAYNAĞI BÖLGELERİNİN ÜRETİLMESİ

İlk aşamada besin kaynaklarının konumu tespit edilerek süreç başlatılır. Arama uzayındaki çözümlere karşılık gelen rastgele yiyecek kaynaklarının oluşturulma süreci, Eşt. (8)'de belirtildiği gibi gerçekleşir [14].

$$X_{ij} = X_j^{\min} + rand(0,1)(X_j^{\max} - X_j^{\min}) \quad (8)$$

Burada *i*, yiyecek kaynağı sayısını, *j* parametre sayısını, *X* ise çözüm değerini ifade etmektedir. $X_j^{\min}$, *j*. parametrenin alt sınırını, $X_j^{\max}$ ise *j*. parametrenin üst sınırını temsil eder.

B. İŞÇİ ARILARIN YİYECEK KAYNAĞI BÖLGELERİNE GÖNDERİLMESİ

Bir sonraki aşamada, konumu tespit edilen yiyecek kaynaklarının kalitesinin değerlendirilmesi gerekmektedir. Elde edilen çözüm hafızada tutulur ve yiyecek kaynağının komşu bölgelerindeki kaynağın kalitesi belirlenir. Komşu konumdaki kaynağın kalitesi daha iyi ise, yeni kaynak hafızaya alınır. $\vec{X}_i$ ile gösterilen her bir kaynak için bu kaynağın yani çözümün tek bir parametresi *j* değiştirilerek, Eşt. (9)'da belirtildiği üzere $\vec{X}_i$ komşuluğundaki $\vec{V}_i$ kaynağı bulunur. Eşitlikte θ_{ij}, [-1,1] aralığında rastgele değer alır.

$$V_{ij} = V_{ij} + \theta_{ij}(X_{ij} - X_{kj}) \tag{9}$$

V_{ij} değerinin bulunmasının ardından, alt ve üst sınır değerlerinin aşılması durumunda öteleme işlemi, *j*. parametre sınırlarına göre yapılır.

$$V_{ij} = \begin{cases} X_j^{\min}, \mathrm{V}_{ij} < X_j^{\min} \\ V_{ij}, X_j^{\min} \le \mathrm{V}_{ij} \le X_j^{\max} \\ X_j^{\max}, \mathrm{V}_{ij} > X_j^{\max} \end{cases}$$

C. GÖZCÜ ARI SAFHASI

Gözcü arı olasılığa bağlı olarak bir gıda kaynağı seçer. Bu besin kaynağı ile ilişkili değer olan p_i Eşt. (10)'da belirtildiği gibi hesaplanır.

$$p_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} \tag{10}$$

Burada $fitness_i$, i. kaynağın kalitesini ifade ederken, *SN* görevli arı sayısını temsil etmektedir.

ABC Algoritmasının Temel Adımları:
Adım 1.Başlangıç yiyecek kaynağı bölgelerinin üretilmesi,
repeat
Adım 2.İşçi arıların yiyecek kaynağına gönderilmesi ve nektar miktarının hesaplanması,
Adım 3.Olasılık değerlerinin görevli arılardan gelen bilgiye göre hesaplanması,

Adım 4. Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmeleri,
Adım 5. Bırakılacak kaynakların bırakılışı ve kâşif arı süreci,
until(Çevrim Sayısı = Maksimum Çevrim Sayısı)

V. DENEYSEL SONUÇLAR VE YORUMLAR

Önerilen strateji, C# programlama dili kullanılarak kodlanmıştır. İlk aşamada yapay arı koloni algoritması kullanılarak renkli görüntülerin eşik değerlerinin hesaplanması amaçlanmıştır. Eşik değerleri, Otsu ve Kapur amaç fonksiyonlarının yapay arı koloni algoritması ile optimize edilmesine dayanmaktadır. Her bir renk kanalı için Otsu ve Kapur amaç fonksiyonunu ayrı ayrı maksimum yapan uygun eşik değerleri hesaplandıktan sonra, Demirci ve arkadaşları[11] tarafından önerilen yöntem ile birleştirilmiştir. İlgili çalışmada arı sayısı ve maksimum çevrim sayısı 350, limit parametresi ise 20 olarak tanımlanmıştır. Görüntü, arı kolonisindeki yiyecek kaynağı ile eşleştirilmiş olup, en küçük değer 0 en yüksek değer ise 255 olarak ifade edilmiştir. Problemin boyut değişkeni ise hesaplanmak istenen eşik sayısı türünden verilmektedir. Dolayısıyla, 2 eşiğin bulunması istenen durumda problemin boyutu 2, 3 eşiğin bulunması istenen durumda problem boyutu 3 olacaktır. Geliştirilen algoritma, görüntü işleme alanında referans kabul edilen Lena ve Biber görüntüleri ile 1, 2 ve 3 eşik kullanılarak test edilmiştir. Elde edilen eşik değerleri ve fonksiyon tipleri, Lena için Tablo 2'de, Biber için Tablo 3'de verilmiştir.

Lena görüntüsünün Şekil 2(b)'de gösterilen renk uzayındaki dağılımı daha yoğun iken, Biber görüntüsünün Şekil 2(d)'de gösterilen renk dağılımının seyrek olduğu açıktır. Yoğun renk dağılımında eşiklerin birbirlerine yakın olması muhtemeldir. Diğer taraftan boş ya da hiçbir pikselin atanmadığı kümelerin oluşma ihtimaline karşılık maksimum küme sayısı Eşt.(7) ile sınırlıdır. Bu açıdan bakıldığında Biber görüntüsünde boş küme oluşma ihtimali daha düşüktür. Biber görüntüsünde herhangi bir kümeye atanan piksel, sayıca düşük olacaktır ki görüntüde birkaç pikselden oluşan küçük bölgelerin ortaya çıkması kaçınılmazdır. Lena görüntüsü ile yapılan deneylerde Otsu amaç fonksiyonu ile sırasıyla 1,2 ve 3 eşik için elde edilen sonuçlar Şekil 3(a), Şekil 3(b) ve Şekil 3(c)'de gösterilmiştir. Şekil 3'de sunulan sonuçlar incelediğinde, eşik sayısı arttıkça resmin sağ alt kısmında oluşan bölgelerin arttığı görülmektedir. Böylece eşik sayısı arttıkça küme sayısı artmakta ve daha fazla detaylar yakalanabilmektedir. Örneğin, Lena'nın yüz hatları, şapkası ve şapka aksesuarındaki detaylar daha net bir biçimde ortaya çıkmaktadır.

Lena görüntüsü ile yapılan ikinci testte, Kapur amaç fonksiyonu yardımıyla 1,2 ve 3 eşik için elde edilen sonuçlar Şekil 4(a), Şekil 4(b) ve Şekil 4(c)'de gösterilmiştir. Şekil 4'de sunulan sonuçlar incelendiğinde, tek eşik ile elde edilen görüntüde Lena'nın yüz hattının, saç ve şapka aksesuarındaki detayların ortaya çıkmadığı ve arka plandaki nesnelerin iç içe girdiği görülmektedir. Eşik sayısının artışı birlikte söz konusu detaylar daha anlaşılır hale gelmektedir. Ancak üç eşik seçilen durumda da arka planda oluşan bazı piksellerin farklı kümelere dâhil olması beklenirken aynı kümeye dâhil olduğu görülmektedir.

Biber görüntüsünün, Otsu ile yapılan ilk deneyinde sırasıyla 1,2 ve 3 eşik için elde edilen sonuçları Şekil 5(a), Şekil 5(b) ve Şekil 5(c)'de gösterilmiştir. Biber görüntüsü sonuçları ele alındığında, tek eşikte özellikle ortadaki Biberin arka plan ile iç içe girmesinin problem teşkil ettiği açıktır. İki eşikte, Biberin kenarları keskinleşmekte, üç eşikte ise ortadaki Biberin hatları daha algılanabilir bir şekilde ayırt edilmiş

ve sınırlar belirgin hale gelmiştir. Diğer taraftan görüntü üzerindeki cisimlerin renk tonları daha net bir biçimde ortaya çıkmıştır.

Kapur amaç fonksiyonu kullanılarak Biber görüntüsü üzerindeki elde edilen sonuçlar, Şekil 6(a), Şekil 6(b) ve Şekil 6(c)'de sunulmuştur. Tek eşik ile yapılan testler incelediğinde, özellikle arka planda yer alan Biberlerin ve uzun Biberin sol tarafında yatay şekilde yer alan kırmızı Biberin iç içe geçtiği ve kenar hatlarının ortaya çıkmadığı görülmektedir. Eşik sayısı arttıkça, Biberler üzerindeki gölgelendirme ve hatlar ortaya çıkmıştır. Böylece daha fazla detayın yakalanabildiği anlaşılmaktadır.

Tablo 2. ve Tablo 3. incelendiğinde Kapur ile elde edilen eşik değerlerinin, Otsu ile elde edilen eşik değerlerine göre daha büyük sonuçlar ürettiği açıktır. Lena görüntüsü üzerinde Otsu ile elde edilen bazı bölgelerin Kapur ile oluşmadığı gözlenmiştir. Bununla birlikte, arka plan detaylarının Otsu ile daha net bir içimde ortaya çıktığı görülmektedir. Biber görüntüsü ele alındığında, Otsu yöntemi ile elde edilen sonuçlarda, ortadaki Biber üzerinde oluşan küme sayısının daha fazla olduğu ve uzun Biberin yanında yer alan Biberin daha iyi kümelendiği görülmektedir. Dolayısıyla, Otsu yönteminin Kapur yönteminden daha başarılı sonuç verdiği söylenebilir.

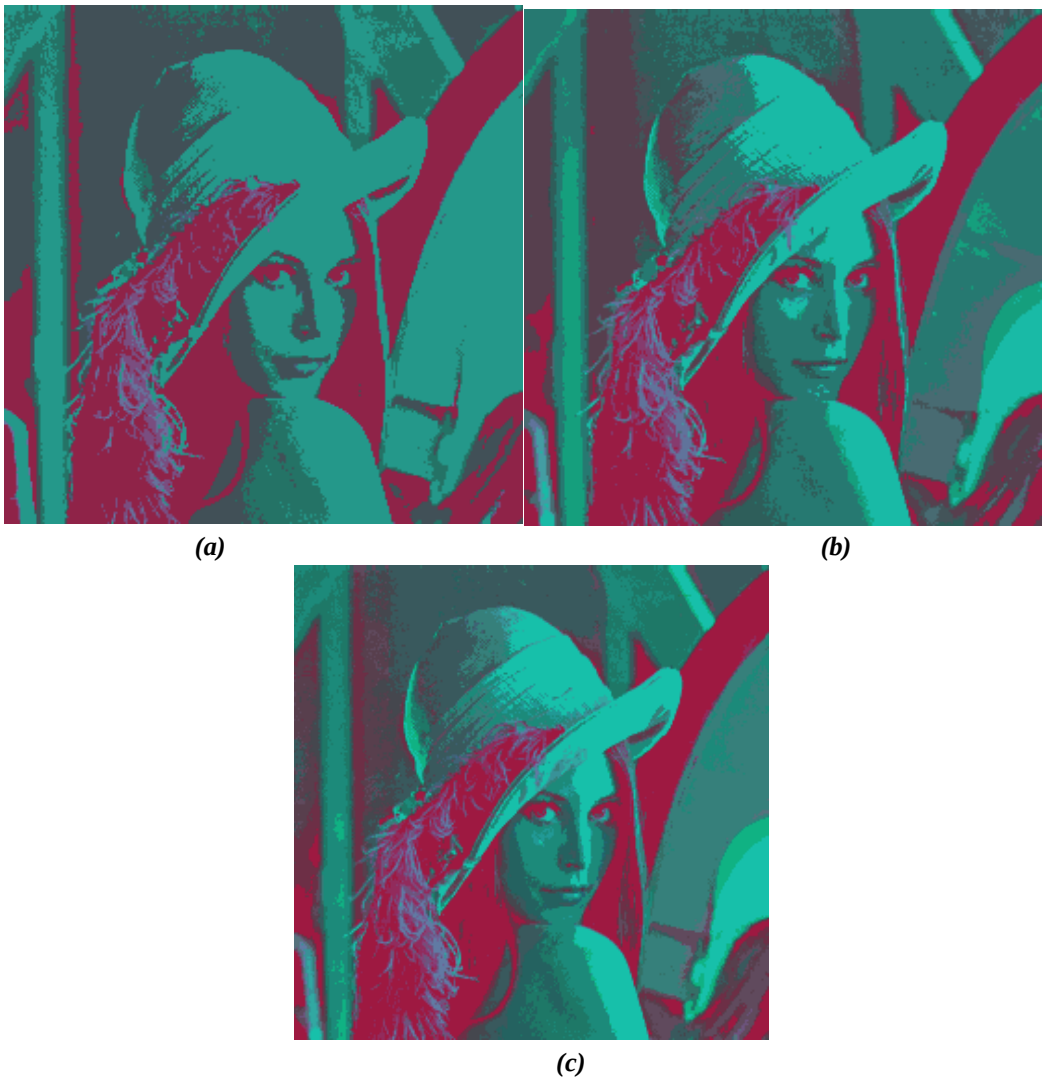
Deney sonuçları genel olarak değerlendirildiğinde, geliştirilen yöntemin görüntü ayrıştırmanın temel amacı olan nesneleri ayırt etme, farklı konumdaki pikselleri gruplayarak anlamlı bilgiler elde etme ve benzer pikselleri aynı sınıf altında tasnif etme hedefleriyle büyük oranda örtüştüğü görülmektedir. İlave olarak, insan algısına yakın sonuçlar verdiği ve metodun renkli görüntüler için uygulanabilir olduğu söylenebilir. Geliştirilen yöntemin en önemli avantajlarından bir diğeri ise, küme sayısının kullanıcı bağımlı bir parametre olmamasıdır. Bununla birlikte, sunulan yöntemin geliştirilmeye ihtiyaç duyduğu konular vardır. Biber görüntüsünün sonuçları ele alındığında ortadaki büyük Biber ve bitişiğinde yer alan uzun Biberin sap kısımlarının etrafındaki bölge incelendiğinde, kırmızı noktalarla oluşturulmuş kümeler dikkat çekmektedir. Dolayısıyla, görüntü üzerinde gereksiz sınıfların meydana geldiği anlaşılmaktadır. Üç eşik seçilmesi halinde en fazla 64 adet küme meydana gelecektir ki görüntü üzerinde 64 farklı nesne olmadığı durumda, sınıflandırma işleminde bu ve buna benzer sıkıntıların meydana gelmesi kaçınılmazdır. Benzer açıdan bakıldığında algoritmanın bahsedilen sıkıntıları giderecek şekilde geliştirilmesi gereklidir.

Tablo 2*. Lena: Eşik Değerleri*

Amaç Fonksiyonu	**Kapur**			**Otsu**		
Tek Eşik	**t_1**			**t_1**		
R	167			165		
G	139			102		
B	129			113		
2 Eşik	**t_1**	**t_2**		**t_1**	**t_2**	
R	134	191		138	196	
G	82	151		78	146	
B	98	144		96	141	
3 Eşik	**t_1**	**t_2**	**t_3**	**t_1**	**t_2**	**t_3**
R	115	155	209	130	180	215
G	59	109	159	55	105	160
B	91	130	170	82	109	146

Tablo 3. *Biber: Eşik Değerleri*

Amaç Fonksiyonu	**Kapur**			**Otsu**		
Tek Eşik	**t_1**			**t_1**		
R	150			149		
G	113			115		
B	171			73		
2 Eşik	**t_1**	**t_2**		**t_1**	**t_2**	
R	98	160		102	161	
G	79	157		78	157	
B	59	128		58	126	
3 Eşik	**t_1**	**t_2**	**t_3**	**t_1**	**t_2**	**t_3**
R	85	136	176	86	135	176
G	38	101	168	39	101	167
B	30	69	130	27	69	130



(a) ***(b)*** ***(c)***

Şekil 3. *Lena: Otsu* ***(a)*** *1 eşik* ***(b)*** *2 eşik* ***(c)*** *3 eşik*



(d) ***(e)***



(f)

Şekil 4. *Lena: Kapur* ***(a)*** *1 eşik* ***(b)*** *2 eşik* ***(c)*** *3 eşik*



(a) ***(b)***

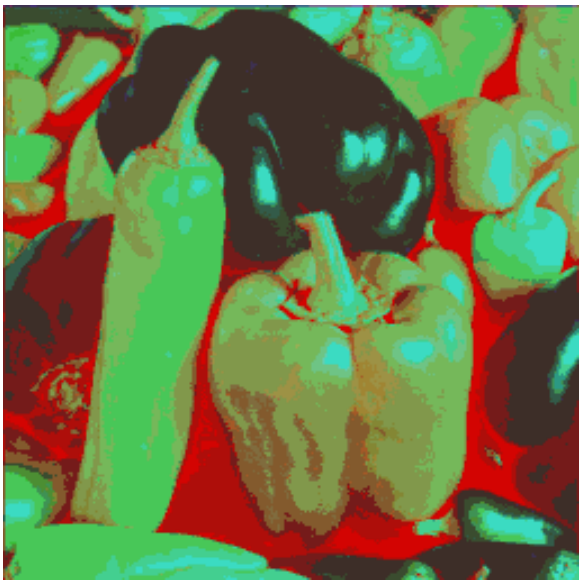


(c)

Şekil 5. *Biber: Otsu* ***(a)*** *1 eşik* ***(b)*** *2 eşik* ***(c)*** *3 eşik*



(a) ***(b)***



(c)

Şekil 6. *Biber: Kapur* ***(a)*** *1 eşik* ***(b)*** *2 eşik* ***(c)*** *3 eşik*

VI. SONUÇ

Eşikleme yöntemleri gri ölçekli resimlerde sıklıkla kullanılırken, renkli görüntülerde birden fazla kanal ve her kanalın farklı bir histograma sahip olmasından ötürü bu yöntemler yetersiz kalmaktadır. Dolayısıyla, her renk kanalından elde edilen farklı eşik değerlerinin bir araya getirilmesi temel problemlerden biridir. Önerilen çalışmada renkli görüntülerin çok seviyeli eşiklenme ve sınıflandırılmasına dayalı yeni bir metot geliştirilmiştir. Yapay arı koloni optimizasyon algoritması kullanılarak Otsu ve Kapur yöntemleri ile her bir kanalın eşik değerleri hesaplanmış ve elde edilen eşik değerleri kullanılarak RGB renk uzayı bölümlenmiştir. Alt küp veya prizmanın içerisinde yer alan her piksel aynı sınıfa dâhil edilmiştir. Eşik sayısı arttıkça renk uzayında oluşturulan küplerin hacimlerinin küçüldüğü ve oluşturulan sınıfların homojenliğinin arttığı görülmüştür.

VII. KAYNAKLAR

[1] F. Kurugollu, B. Sankur and A. E. Harmanci, "Color image segmentation using histogram multithresholding and fusion," *Image and vision computing*, vol. 19, no. 13, pp. 915-928, 2001.

[2] Rezaee, M. R., Lelieveldt, B. P. and Reiber, J. H, "A new cluster validity index for the fuzzy c-mean," *Pattern recognition letters*, vol.19, no. 3, 237-246,1998.

[3] Pare, S., Kumar, A., Bajaj, V. and Singh, G. K., "A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve", *Applied Soft Computing*, vol.47, pp. 76-102, 2016.

[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations", *In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, pp. 281-297, 1967.

[5] Bezdek, J. C., Ehrlich, R. and Full, W,FCM: "The fuzzy c-means clustering algorithm", *Computers & Geosciences*, vol. 10 no.2-3,pp. 191-203, 1984.

[6] N. Otsu, "A threshold selection method from gray-level histograms*", IEEE transactions on systems, man, and cybernetics*, vol. 9 no. 1, pp. 62-66, 1979.

[7] Kapur, J. N., Sahoo, P. K. and Wong, A. K, "A new method for gray-level picture thresholding using the entropy of the histogram", *Computer vision, graphics, and image processing,* vol. 29, no:3, pp. 273-285, 1985.

[8] Pal, S. S., Kumar, S., Kashyap, M., Choudhary, Y. And Bhattacharya, "M.,Multi-level thresholding segmentation approach based on spider monkey optimization algorithm", *In Proceedings of the Second International Conference on Computer and Communication Technologies*, Springer, New Delhi, 2016, p. 273-287.

[9] Singh, R., Agarwal, P., Kashyap, M. and Bhattacharya, "M,Kapur's and Otsu's based optimal multilevel image thresholding using social spider and firefly algorithm", *In Communication and Signal Processing (ICCSP), International Conference on*, IEEE, 2016, pp. 2220-2224.

[10] El Aziz, M. A., Ewees, A. A. and Hassanien, A. E.,"Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation", *Expert Systems with Applications*, vol. 83, pp. 242-256, 2017.

[11] R. Demirci ,U. Güvenç Kahraman, H.T. "Görüntülerin Renk Uzayı Yardımıyla Ayrıştırılması", *İleri Teknoloji Bilimleri Dergisi*, c. 3, s. 1, ss. 1-8, 2014.

[12] Sathya, P. D. and Kayalvizhi, R. "Amended bacterial foraging algorithm for multilevel thresholding of magnetic resonance brain images", *Measurement*, vol. 44 no. 10, pp. 1828-1848, 2011.

[13] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Erciyes University Engineering Faculty Computer Engineering Department, Technical Report TR06, 2005.

[14] D. Karaboğa, *Yapay Zekâ Optimizasyon Algoritmaları*, 4. baskı, Ankara, Türkiye: Nobel Yayın Dağıtım, 2011, böl. 9, ss. 207- 202.