

Görüntülerin Renk Uzayı Yardımıyla Ayırıştırılması ve Tekrarlı Uzunluk Kodlaması Tabanlı Görüntü Sıkıştırma

Mustafa Burak Kalkan^{*+}, Recep Demirci

Computer Engineering Department, Gazi University, Faculty of Technology, Ankara, Turkey

*Corresponding author: mburakkalkan@gmail.com

+Speaker: mburakkalkan@gmail.com

Presentation/Paper Type: Oral / Full Paper

Özet – Teknolojinin gelişimiyle birlikte her geçen gün daha kaliteli hale gelen ve boyutları büyümekte olan sayısal görüntüleri saklamak ve transfer etmek ciddi problem haline gelmektedir. Boyut ve kalite arttıkça görüntüyü saklamak için gereken bellek alanı artmakta ve aynı zamanda transfer süresi de uzamaktadır. Bu çalışmada görüntülerin renk uzayı yardımıyla ayırıştırılması ve tekrarlı uzunluk kodlaması (run-length encoding: RLE) kullanılarak yeni bir kayıplı sıkıştırma yöntemi geliştirilmiştir. Önerilen algoritmada öncelikle bölütleme süreci ile görüntüdeki renk sayısı indirgenmektedir. İkinci aşamada ise bölütlenmiş görüntülerin etiket bilgilerine RLE sıkıştırması uygulanarak dosya boyutu küçültülmektedir.

Anahtar Kelimeler – görüntü sıkıştırma, görüntü ayırıştırma, bölütleme, tekrarlı uzunluk kodlaması, renk uzayı

Abstract – With the development of technology, it becomes a serious problem to store and transfer digital images with high quality which has large sizes. When the size and quality of the image increase, the memory space required to store increases as well. Moreover, the transfer durations also increase. In this paper, a new lossy compression method has been developed by integrating run-length encoding (RLE) algorithm and image segmentation process with color space. In the proposed method, initially, the number of colors in image is reduced by the segmentation method. Then RLE compression strategy is applied to the label information of segmented image. Consequently, the file size of images has been reduced.

Keywords — image compression, image segmentation, run-length encoding, color space

I. GİRİŞ

Sayısal görüntü teknolojisinin gelişmesi, her geçen gün daha yüksek görsel kaliteye sahip görüntülerin elde edilebilmesini sağlamaktadır. Ancak görüntü boyutlarının büyümesi kaçınılmaz olmaktadır. Sayısal görüntülerin daha küçük boyutlarda saklanması ve hızlı transferi noktasında veri sıkıştırma yöntemlerine ihtiyaç duyulmaktadır. Bununla birlikte sıkıştırma algoritmaları ile verinin saklanması için gereken bellek alanı küçültülebilmektedir. Sıkıştırma yöntemleri ise kayıplı ve kayıpsız yaklaşımlar şeklinde tasarlanmaktadır. Kayıpsız algoritmalarda orijinal veri, sıkıştırılmış veri içerisinde hiçbir kayıp olmadan aynı şekliyle geri getirilebilir. Diğer taraftan, kayıplı yöntemlerde ise orijinal veri ile geri getirilen veri birebir aynı değildir [1, 2]. Bir bakıma bilgi kaybı kaçınılmazdır. Dolayısıyla görüntü sıkıştırma algoritmalarının performansı iki ayrı parametre ile değerlendirilir. Birinci metrik olan sıkıştırma oranı (compression ratio: CR), verinin sıkıştırılmamış haldeki orijinal boyutunun, sıkıştırılmış haldeki boyutuna oranı şeklinde tanımlanır [3]. Diğer metrik ise kaybolan bilgi miktarının veya kodlanan görüntünün orijinal görüntüden ne kadar farklılaştığının ölçümünde kullanılan PSNR'dir. Desibel ile ifade edilen PSNR değeri arttıkça, iki görüntü arasındaki benzerlik oranı da artacaktır [4]. PSNR kriteri gri ölçekli görüntülerin veya renkli görüntülerin her bir kanalının benzerliğini test etmek için kullanılmaktadır. Bununla birlikte

renkli görüntülerde üç kanalın ortalaması tercih edilen bir başka yaklaşım olmuştur [5].

Görüntü, ses, video vb. alanlarda kayıplı sıkıştırma yaklaşımı tercih edilerek daha yüksek sıkıştırma oranları elde edilebilmektedir. Görüntü sıkıştırmada ideal beklenti hem sıkıştırma oranının hem de PSNR benzerlik oranının aynı anda çok yüksek olmasıdır. Her iki kriteri birden sağlayan algoritmalar mevcut olmamakla birlikte, literatürde birçok görüntü sıkıştırma yöntemleri ve bunları kullanan dosya biçimleri mevcuttur. BMP, JPEG ve GIF biçimleri bunlardan bazılarıdır.

En çok kullanılan JPEG formatı, ayrık kosinüs dönüşümü (discrete cosine transform: DCT) tabanlı bir kayıplı sıkıştırma formatıdır. 24 bit RGB derinliğinde büyük boyutlu ve karmaşık görüntüleri sıkıştırmakta başarılı olduğu görülmüştür [1, 6, 7]. Önemli zayıf yönü ise her sıkıştırmada yeniden veri kaybının oluşması ve görüntü kalitesinin düşmesidir [6]. JPEG biçiminde çıkış görüntüsünün kalitesi 1 ile 100 arasında seçilebilen bir kalite faktörü ile ayarlanabilmektedir. Seçilen kalite faktörü yükseldikçe çıkış görüntüsünün orijinal görüntüye benzerliği artmaktadır fakat diğer taraftan sıkıştırma oranı düşmektedir [7].

GIF formatı, LZW (Lempel-Ziv-Welch) sıkıştırması algoritması kullanan ve 8 bit renk derinliğine sahip bir görüntü sıkıştırma biçimidir. Az sayıda renk içeren görüntülerde

başarıldır fakat düşük renk derinliği sebebiyle fotoğraflarda kalite başarımı düşmektedir [1].

Bu çalışmada Demirci ve arkadaşları [8] tarafından geliştirilen görüntülerin renk uzayı yardımıyla ayırıştırılması metodu ve kayıpsız bir sıkıştırma yöntemi olan tekrarlı uzunluk kodlaması (RLE) birleştirilerek yeni bir kayıplı görüntü sıkıştırma algoritması önerilmiştir. Geliştirilen algoritmanın, içerdiği renklerin uzaysal olarak homojen dağıldığı görüntülerde başarılı olduğu görülmüştür.

II. GÖRÜNTÜLERİN RENK UZAYI YARDIMIYLA AYIRIŞTIRILMASI

Görüntü bölütleme, görüntüyü birbiriyle çakışmayan anlamlı bölgelere ayırma işlemidir [9]. Bölütleme için literatürde birçok farklı yöntem ve yaklaşım bulunmaktadır. Görüntülerin renk uzayı yardımıyla ayırıştırılması yöntemi bunlardan birisidir [8]. Bölütleme sayesinde görüntü içerisindeki renk sayısı düşürülerek kayıp verilmektedir ancak görüntü sıkıştırmaya uygun hale getirilmektedir.

Renk uzayı tabanlı ayırıştırma yöntemi, gri seviyeli görüntülerde başarılı sonuçlar üreten Otsu ve Kapur algoritmaları temel alınarak geliştirilmiş bir yaklaşımdır. Üç boyutlu RGB renk uzayı esas alınmaktadır. Süreçte öncelikle her bir kanal için eşikler hesaplanmakta, sonra renk uzayı alt küpler şeklinde kesilmektedir. Her bir alt küp veya prizma içinde kalan pikseller aynı sınıfa atanmaktadır. Ayrıca aynı sınıfa atanan piksellerin ortalaması alınarak renk indirgemesi yapılmaktadır. Otsu ve Kapur algoritmaları histogram tabanlı eşiklemede yaygın olarak kullanılan tekniklerdendir [10, 11, 12]. Eşik değerinin seçiminde Kapur yöntemi her sınıfa ait yerel entropiler toplamını maksimum yapan değeri esas alırken, Otsu yöntemi ise sınıflar arası varyansı maksimum yapan değeri temel alınmaktadır. Kapur tarafından önerilen amaç fonksiyonu:

$$J(t) = H_0 + H_1 \quad (1)$$

şeklinde tanımlanmıştır. Denklem 1'deki H_0 ve H_1 değişkenleri her bir sınıfın kısmi entropilerini belirtmektedir. Söz konusu kısmi entropiler

$$H_0 = -\sum_{i=0}^{t-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \omega_0 = \sum_{i=0}^{t-1} p_i \quad (2)$$

$$H_1 = -\sum_{i=t}^{L-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \omega_1 = \sum_{i=t}^{L-1} p_i \quad (3)$$

ifadeleri ile hesaplanırlar. Denklem 2 ve Denklem 3 deki i parametresi gri seviye indisi olup, maksimum değeri (L) 255 tir. Herhangi bir pikselin i . seviyede bulunma olasılığı olan, p_i

$$p_i = \frac{n_i}{M \times N} \quad (4)$$

şeklinde hesaplanır. Denklem 4'teki n_i , i . seviyedeki piksel sayısını, $M \times N$ ise görüntünün toplam piksel sayısını temsil etmektedir. Diğer yandan Otsu yönteminin amaç fonksiyonu

$$J(t) = \sigma_0 + \sigma_1 \quad (5)$$

şeklinde tanımlanmıştır. Burada σ_0 ve σ_1 parametreleri sınıflar arası varyansı temsil etmekte olup,

$$\sigma_0 = \omega_0 (\mu_0 - \mu_T)^2 \text{ ve } \sigma_1 = \omega_1 (\mu_1 - \mu_T)^2 \quad (6)$$

olarak hesaplanır. Sınıfların Denklem 6'da istatistik bilgileri ise

$$\omega_0 = \sum_{i=0}^{t-1} p_i \text{ ve } \mu_0 = \sum_{i=0}^{t-1} \frac{ip_i}{\omega_0} \quad (7)$$

$$\omega_1 = \sum_{i=t}^{L-1} p_i \text{ ve } \mu_1 = \sum_{i=t}^{L-1} \frac{ip_i}{\omega_1} \quad (8)$$

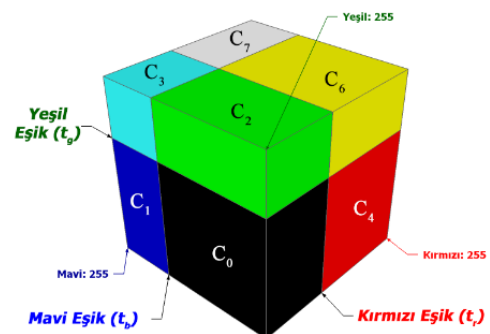
şeklinde olmaktadır. Ayrıca görüntünün gri seviye ortalaması μ_T ise

$$\mu_T = \sum_{i=0}^{L-1} ip_i \quad (9)$$

olarak hesaplanmaktadır. Demirci ve arkadaşları [8] tarafından geliştirilen ayırıştırma algoritmasında görüntünün kaç adet sınıfa (renge) indirgeneceği seçilen eşik sayısına bağlıdır. Başka bir ifade ile m adet eşik için $k = (m+1)^3$ adet bölge meydana gelmektedir. Örneğin Tablo 1'de $m = 1$ adet eşik için atama kuralları verilmiştir. Burada tek eşik seçilmiştir ve dolayısıyla 8 adet bölge oluşmaktadır. R, G ve B harfleri sırasıyla kırmızı, yeşil ve mavi kanalı, t_r , t_g ve t_b değerleri ise ilgili kanallar için hesaplanmış olan eşik değerlerini temsil etmektedir. Şekil 1'de ise Tablo 1'de verilen atama kurallarına göre tek eşikli bir bölütleme sonucu oluşan renk uzayı örneği verilmiştir. Buna göre renk indirgemesi yapılmış görüntüde en fazla 8 renk oluşacaktır. Fakat toplam renk sayısının 8'e indirgenmesi yüksek oranda bilgi kaybını da beraberinde getirmektedir.

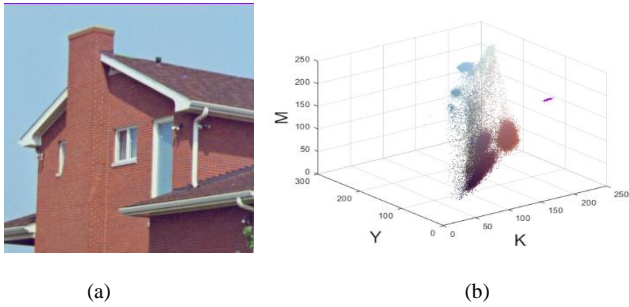
Tablo 1. Renk uzayını bölütleme kuralları ($m = 1$)

Etiket	Atama Kuralı	Atanan Renk	İkili kod
C ₀	$R \leq t_r \ \&\& \ G \leq t_g \ \&\& \ B \leq t_b$	0x000000	000
C ₁	$R \leq t_r \ \&\& \ G \leq t_g \ \&\& \ B \geq t_b$	0x0000FF	001
C ₂	$R \leq t_r \ \&\& \ G \geq t_g \ \&\& \ B \leq t_b$	0x00FF00	010
C ₃	$R \leq t_r \ \&\& \ G \geq t_g \ \&\& \ B \geq t_b$	0x00FFFF	011
C ₄	$R \geq t_r \ \&\& \ G \leq t_g \ \&\& \ B \leq t_b$	0xFF0000	100
C ₅	$R \geq t_r \ \&\& \ G \leq t_g \ \&\& \ B \geq t_b$	0xFF00FF	101
C ₆	$R \geq t_r \ \&\& \ G \geq t_g \ \&\& \ B \leq t_b$	0xFFFF00	110
C ₇	$R \geq t_r \ \&\& \ G \geq t_g \ \&\& \ B \geq t_b$	0xFFFFFF (Beyaz)	111

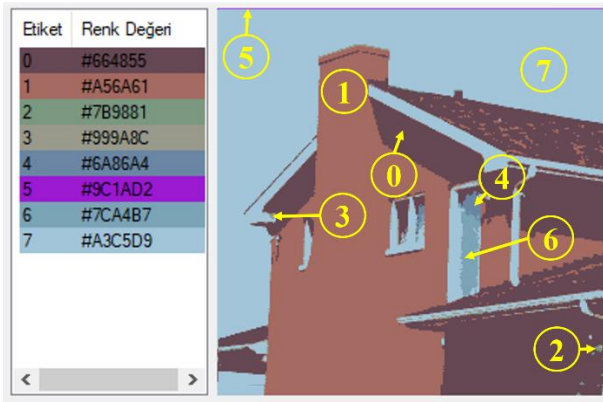


Şekil 1. Renk uzayının bölütlenmesi ve oluşan sınıflar (C₀ - C₇)

Ev görüntüsünün Şekil 2(a)'da orijinali ve Şekil 2(b)'de ise RGB uzayındaki renk dağılımı verilmiştir. Şekil 3'te görüntünün tek eşikli bölütlenmesi sonucu oluşan bölgeler ve bölgelere atanan renk etiketleri verilmiştir. Renk indirgemesi yapıldığından oluşan bölgelerin homojen olduğu açıktır.



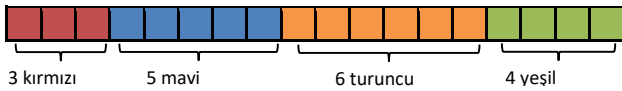
Şekil 2. Ev a) orijinal b) Renk dağılımı



Şekil 3. Ev: bölütleme ve atanan renk etiketleri

III. TEKRARLI UZUNLUK KODLAMASI (RLE)

Tekrarlı uzunluk kodlaması (RLE), birbirini takip eden aynı sembollerin hepsinin saklanması yerine, sadece sembol ve tekrar sayılarının saklanması mantığına dayalı bir kayıpsız sıkıştırma algoritmasıdır [13]. RLE algoritmasının görüntü sıkıştırmada kullanımına Şekil 4'teki yapı örnek verilebilir. Şekil 4'te verilen resmin 24 bit RGB renk modunda, bir satırının 1x18 boyutunda olan bir görüntü olduğu varsayılırsa, bu boyuttaki bir ham görüntü için gereken bellek alanı $3 \times 18 = 54$ byte'tır.



Şekil 4: 1x18 boyutunda 24 bit RGB modunda görüntü örneği

Görüldüğü üzere resimdeki pikseller yan yana tekrarlar içermektedir. Her renk için bir etiket belirlenirse, renkli görüntü RLE algoritması yardımıyla etiket-tekrar ikilileri şeklinde temsil edilebilir. Başka bir ifade ile daha küçük boyutlara sıkıştırılmış olur. Örneğin, K, M, T ve Y etiketlerinin sırasıyla kırmızı, mavi, turuncu ve yeşili temsil ettiği varsayılırsa, sıkıştırılmış görüntü "K3 M5 T6 Y4" şeklinde 8 byte olarak saklanabilir. Böylece sıkıştırma oranı (CR) $54/8 = 6.75$ olacaktır. Böylece sıralı renk tekrarlarının olduğu homojen renk dağılımlı görüntülerde RLE algoritması başarılı bir sıkıştırma yapabilecektir. Diğer taraftan tekrarların az olduğu veya renklerin homojen bir dağılım göstermediği görüntülerde algoritmanın başarısı düşük kalabilmektedir. Şekil 5(a) yüksek başarıma bir örnek iken, Şekil 5(b) ise düşük başarıma bir örnek teşkil etmektedir.



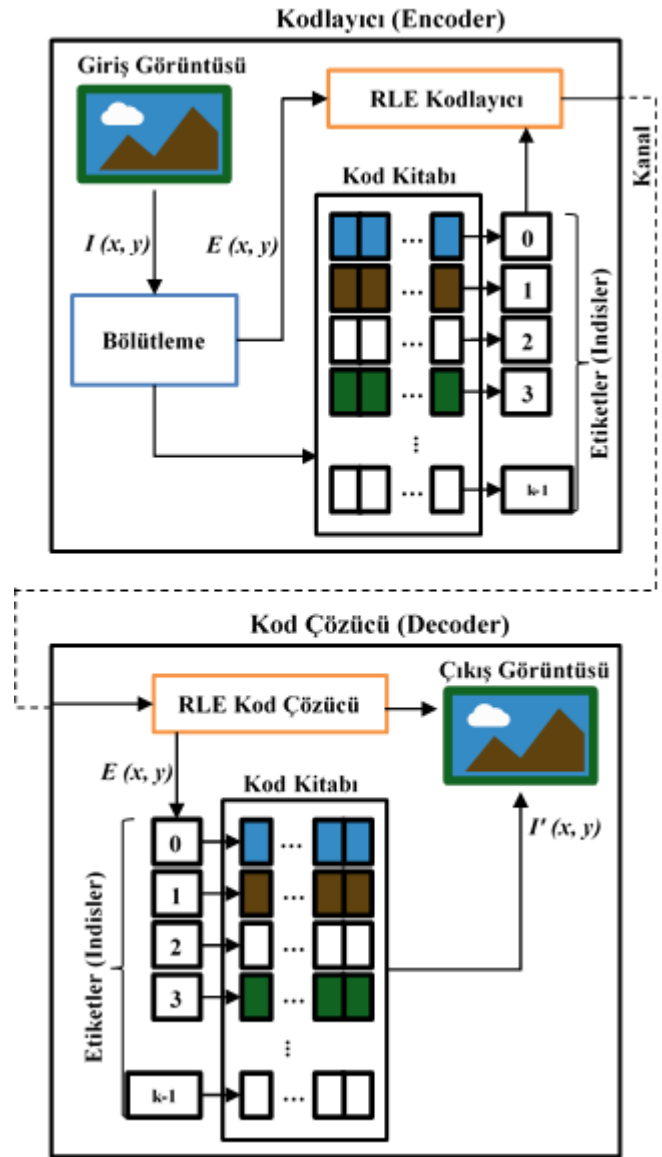
a) K3 Y2 S2 M3 C5 V4 F1 D5 (CR: $75/16 = 4.69$)

b) K1 M1 K1 S1 Y1 S1 C1 M1 F1 M1 C1 S1 C1 Y1 C1 V1 M1 V1 K1 F1 D1 Y1 D1 S1 D1 (CR: $75/50 = 1.5$)

Şekil 5: Tekrarlı renkler ve etiketleri (K:kırmızı, Y:açık yeşil, S:sarı, M:mavi, C:mor, G:kahverengi, F:siyah, D:su yeşili) a) Yüksek başarıma b) düşük başarıma

IV. AYRIŞTIRMA VE RLE TABANLI SIKIŞTIRMA

Önerilen algoritma Şekil 6'da gösterildiği gibi kodlayıcı ve kod çözücü şeklinde iki aşamadan oluşmaktadır. Birinci aşamada görüntülerin öncelikle renk uzayı yardımıyla ayrıştırılması yapılmaktadır. Ayrıştırma işlemi sonucunda elde edilen bölge etiketleri, $E(x,y)$ RLE algoritması ile sıkıştırılmaktadır. Başka bir ifade ile piksellerin renk bilgileri değil etiket bilgileri sıkıştırılmıştır.



Şekil 6: Kodlayıcı ve kod çözücü işlem adımları

Hedef dosyaya ise sıkıştırılmış etiketler ve kod kitabındaki (renk paleti) renk bilgileri kaydedilmiştir. Kaydedilen dosya içeriğinde kod çözücü için gerekli olan genişlik-yükseklik bilgisi, kod kitabı ve sıkıştırılmış etiket bloğu olmak üzere üç kısım bulunmaktadır. Buna göre dosya biçimi yapısal olarak Şekil 7'deki şablona benzetilmektedir:

```

5 5 → Resmin piksel cinsinden genişlik-yükseklik bilgisi
8 → Kod kitabı (renk paleti) uzunluğu
FF0000, FFFF00, 92D050, 00B0F0, 7030A0, B95B22,
000000, 578279 → Kod Kitabı: Sırasıyla
etiketlere atanan renk değerler
0*3, 2*2, 1*2, 3*3, 4*5, 5*4, 6*1, 7*5 ... →
Etiket * tekrar ikilileri şeklinde
sıkıştırılmış veri
    
```

Şekil 7: Kaydedilen dosya içeriği şablonu

Kaydedilen dosya okunmak istendiğinde ise kod çözücü devreye girmektedir. Kod çözücü, kaydedilen dosya verisini RLE çözücü ile açarak görüntünün piksel cinsinden boyutlarını, kod kitabını ve $E(x,y)$ etiket matrisini oluşturmaktadır. Sonrasında ise kod kitabındaki etiketlere karşılık gelen renk değerlerine göre $I(x,y)$ bitmap görüntüsünü çıktı olarak üretilmektedir.

V. DENEYSEL SONUÇLAR VE YORUMLAR

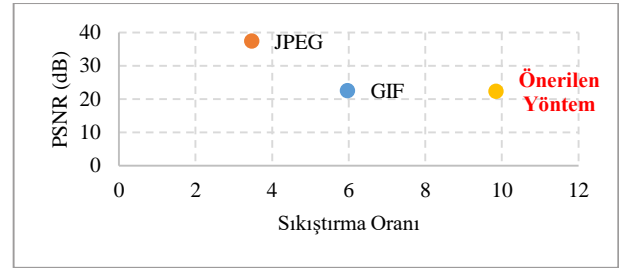
Önerilen yöntemi test etmek için C# ortamında bir arayüz geliştirilmiştir. Test amaçlı seçilen orijinal görüntüler 24 bit 'lik BMP formatındadır. Test edilen renkli görüntüler öncelikle Otsu algoritması yardımıyla her bir kanal için elde edilen eşikler ile bölütlenmiştir. Akabinde sıkıştırma yapılarak dosya oluşturulmuştur. Elde edilen dosya boyutları ve ortalama PSNR değerleri hesaplanmıştır. Ayrıca BMP formatındaki orijinal görüntüler JPEG ve GIF biçimleriyle kaydedilerek karşılaştırma yapılmıştır. JPEG için 100 kalite faktörü seçilmiştir. GIF formatı için ise 8 bit renk derinliği bulunmaktadır.

Şekil 8(a)'da Lena görüntüsü, Şekil 8(b)'de ise önerilen yöntem ile sıkıştırılmış hali verilmiştir. Şekil 9 farklı formatlar için PSNR- sıkıştırma oranı değişimi göstermektedir. Şekil 9 incelendiğinde önerilen yöntemin Lena görüntüsü için PSNR bazında GIF biçimine göre neredeyse aynı başarıyı elde ettiği fakat JPEG biçimine göre yeterli başarıya ulaşamadığı görülmektedir. Sıkıştırma oranı bazında ise GIF ve JPEG sıkıştırmalarına göre öne çıktığı açıktır.

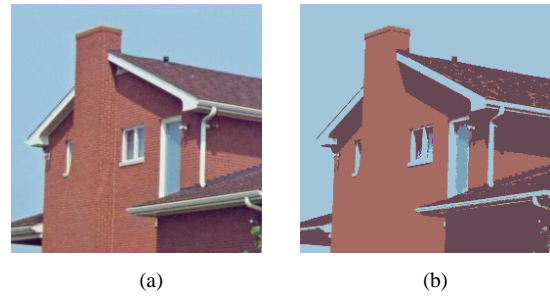
Lena için yapılan deneyler Şekil 10(a)'da orijinali verilen Ev görüntüsü için de tekrarlanmıştır. Şekil 10(b)'de önerilen yöntem ile sıkıştırılmış hali verilmiştir. Diğer taraftan Şekil 11 incelendiğinde önerilen yöntemin Ev görüntüsü için PSNR bazında GIF biçimine göre daha iyi başarı elde ettiği fakat JPEG biçimine göre yeterli başarıya ulaşamadığı görülmektedir. Sıkıştırma oranı bazında ise GIF ve JPEG sıkıştırmalarına göre yüksek farkla öne çıktığı görülmektedir. Sıkıştırma oranındaki bu yüksek başarımın nedeni olarak bölütleme sonucu elde edilen görüntüdeki renk dağılımının RLE sıkıştırmasına elverişli bir şekilde uzun sıralı tekrarlar içeren homojen bir yapıda olması söylenebilir.



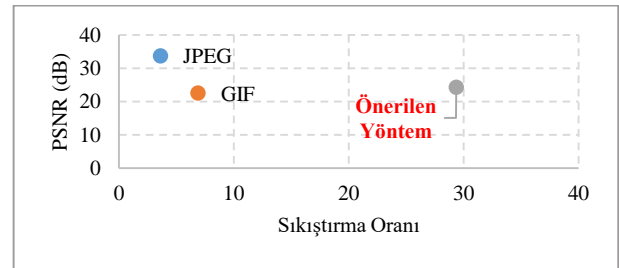
Şekil 8. Lena (512x512) a) Orijinal b) Önerilen, PSNR: 22.38, CR: 9.85



Şekil 9. Lena: PSNR- sıkıştırma oranı değişimi



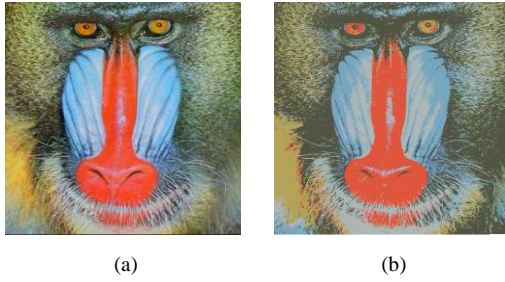
Şekil 10. Ev (256x256) a) Orijinal (b) Önerilen, PSNR: 24.36, CR: 29.37



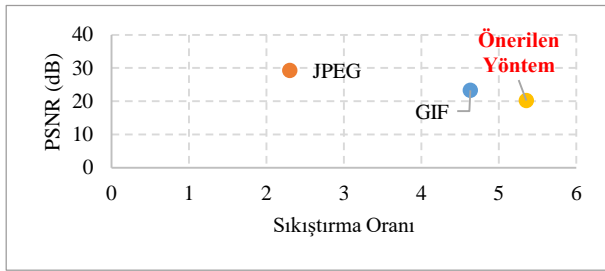
Şekil 11. Ev: PSNR- sıkıştırma oranı değişimi

Şekil 12(a)'da Babun görüntüsü, Şekil 12(b)'de ise önerilen yöntem ile sıkıştırılmış hali verilmiştir. Şekil 13 incelendiğinde ise önerilen yöntemin PSNR bazında GIF ve JPEG biçimlerine göre yeterli başarıya ulaşamadığı, sıkıştırma oranı bazında ise yine öne çıktığı görülebilmektedir. Fakat Ev ve Lena görüntülerinde elde edilen sıkıştırma oranı seviyelerine çıkılamamıştır. Bu durumun nedeni olarak ise bölütleme sonucu elde edilen görüntüdeki renk dağılımının sık değişkenlik göstermesi, dolayısıyla RLE sıkıştırmasının bu durumlarda zayıf kaldığı söylenebilir.

Şekil 14(a)'da Biberler görüntüsü, Şekil 14(b)'de ise önerilen yöntem ile sıkıştırılmış hali verilmiştir. Şekil 15'te verilen sonuçlar incelendiğinde ise önerilen yöntemin PSNR bazında GIF ve JPEG biçimlerine göre yeterli başarıya ulaşamadığı, sıkıştırma oranı bazında ise öne çıktığı görülebilmektedir.



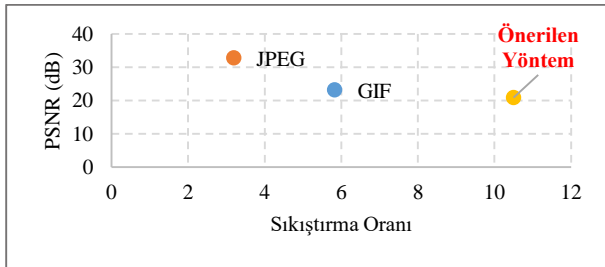
Şekil 12. Babun (512x512) a) Orijinal b) Önerilen, PSNR: 20.23, CR: 5.36



Şekil 13. Babun: PSNR - sıkıştırma oranı değişimi



Şekil 14. Biberler (512x512) a) Orijinal b) Önerilen, PSNR: 20.85, CR: 10.5



Şekil 15. Biberler PSNR - sıkıştırma oranı değişimi

Test görüntülerine ait sıkıştırma oranı – PSNR grafiklerindeki değerler Tablo 2'de ayrıntılı olarak da gösterilmiştir.

Tablo 2. Test görüntüleri ve performans değerleri

BMP Görüntü	24 bit Ham Boyut (kB)	Sıkıştırma Yöntemi	Sıkıştırma Oranı:CR	Ortalama PSNR	Sıkıştırılmış Boyut (kB)
Lena (512x512)	768.00	GIF	5.96	22.62	128.76
		JPEG	3.47	37.48	221.65
		Önerilen	9.85	22.38	77.98
Ev (256x256)	192.00	GIF	6.89	22.6	29.7
		JPEG	3.68	38.02	53.1
		Önerilen	29.37	24.36	6.69
Babun (512x512)	768.00	GIF	4.63	23.33	165.88
		JPEG	2.3	29.29	333.64
		Önerilen	5.36	20.23	143.41
Biberler (512x512)	768.00	GIF	5.82	23.16	131.87
		JPEG	3.19	32.8	240.55
		Önerilen	10.50	20.85	73.15

Deneysel sonuçlardan da anlaşıldığı üzere önerilen yöntem, renk geçiş ve değişimlerinin az olduğu, renk dağılımının homojenlik gösterdiği resimlerde iyi bir başarımla göstermektedir. Fakat karmaşık yapıları resimlerde yetersiz kalabilmektedir. Deneysel bölütleme aşamasında tek eşikli bölütleme kullanılmış olması nedeniyle resimlerdeki toplam renk sayısı $(1+1)^3 = 8$ 'e indirgenmektedir. Bu durum da düşük PSNR değerlerine sebep olmaktadır. Eşik sayıları artırılarak daha yüksek PSNR seviyelerine ulaşılabilir. Sıkıştırma aşamasındaki başarımın artırılması için RLE ve dosyalama biçimi üzerinde geliştirmeler yapılabilir.

VI. SONUÇ

Yapılan çalışma sonucunda görüntü bölütleme ve tekrarlı uzunluk kodlaması harmanlanarak yeni bir kayıplı sıkıştırma yöntemi geliştirilmiştir. Yöntemin görüntü kalitesi açısından eksikleri bulursa da sıkıştırma oranı tarafından başarılıdır. Başka bir ifade ile kayıpların fazla önemsenmediği kullanım alanları için tercih edilebilir bir alternatif olarak önerilmektedir. Bununla birlikte eşik sayıları artırılarak PSNR oranlarının yükseltilmesi ileriki çalışma konusudur.

VII. KAYNAKLAR

- [1] A. Mesut, "Veri Sıkıştırmada Yeni Yöntemler", Edirne, 2006.
- [2] K. Sayood, Introduction to Data Compression, Newnes, 2012.
- [3] S. Grgic, M. Grgic ve B. Zovko-Cihlar, "Performance Analysis of Image Compression Using Wavelets", *IEEE Transactions on Industrial Electronics*, cilt 48, no. 3, pp. 682-695, 2001.
- [4] P. Kaushik ve Y. Sharma, "Comparison of Different Image Enhancement Techniques based upon PSNR & MSE", *International Journal of Applied Engineering Research*, cilt 7, no. 11, 2012.
- [5] P. Gupta, P. Srivastava, S. Bhardwaj ve V. Bhateja, "A Modified PSNR Metric based on HVS for Quality Assessment of Color Images", *2011 International Conference on Communication and Industrial Application*, Kolkata, West Bengal, 2011.
- [6] R. H. Wiggins, H. Davidson, H. Harnsberger, J. R. Lauman ve P. A. Goede, "Image File Formats: Past, Present, and Future", *Radiographics*, cilt 21, no. 3, pp. 789-798, 2001.
- [7] R. Pandit, N. Khosla, G. Singh ve H. Sharma, "Image Compression and Quality Factor in case of JPEG Image Format", *International Journal of Advanced Research in Computer and Communication Engineering*, cilt 2, no. 7, pp. 2578-2581, July, 2013.
- [8] R. Demirci, U. Güven ve H. T. Kahraman, "Görüntülerin Renk Uzayı Yardımıyla Ayırıştırılması", *İleri Teknoloji Bilimleri Dergisi*, cilt 3, no. 1, 2014.
- [9] U. Güneç, Ç. Elmas ve R. Demirci, "Renkli Görüntülerin Otomatik Ayırıştırılması", *Politeknik Dergisi*, cilt 11, no. 1, pp. 9-12, 2008.
- [10] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE transactions on systems, man, and cybernetics*, cilt 9, no. 1, pp. 62-66, 1979.
- [11] J. N. Kapur, P. K. Sahoo ve A. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram", *Computer Vision, Graphics, and Image Processing*, cilt 29, no. 3, pp. 273-285, 1985.
- [12] M. Kılıçarslan, U. Tanyeri ve R. Demirci, "Renkli Görüntüler İçin Tek Boyutlu Histogram", *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, cilt 6, no. 4, pp. 1094-1107, 2018.
- [13] A. Mesut ve A. Carus, "Kayıpsız Görüntü Sıkıştırma Yöntemlerinin Karşılaştırılması", *II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi*, İstanbul, 2005.