

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**JPEG DOSYALARININ HAFIZA BOYUTLARININ
DÜŞÜRÜLMESİ**
Yüksek Lisans Tezi

Tezi Hazırlayan:
Süleyman ÜNLÜ

İstanbul, 2019

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**JPEG DOSYALARININ HAFIZA BOYUTLARININ
DÜŞÜRÜLMESİ**
Yüksek Lisans Tezi

Tezi Hazırlayan:
Süleyman ÜNLÜ

Öğrenci No:
160820059

Danışman:
Dr. Öğr. Üyesi Ediz ŞAYKOL

İstanbul, 2019

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “**JPEG Dosyalarının Hafıza Boyutlarının Düşürülmesi**” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım.

17/05/2019

Süleyman ÜNLÜ

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi.....no'luin tarihinde yapılan tez savunma sınavı¹ sonucunda... dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle, **KABUL** kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : **Bilgisayar Mühendisliği Anabilim Dalı**
Programı : **Bilgisayar Mühendisliği**
Tez Başlığı³ : **JPEG Dosyalarının Hafıza Boyutlarının Belirlenmesi**

Tez Sınav Jürisi

Öğretim Üyesi

Danışman

: **Doç.Dr. Üyesi Ediz SAKAL**

Üye

: **Prof. Dr. Gökhan SİHAHTAROĞLU**

Üye

: **Doç.Dr. Üyesi Atıf YILMAZ**

İmza

¹ Jüri üyeleri, söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez sınavına alır. Tez savunma sınav süresi en az 45, en çok 90 dakikadır. Jüri üyeleri, sınav öncesi yapılacak toplantıda, kendi aralarından danışman dışında bir üyeyi başkan seçer. Tez sınavı, tez çalışmasının sunulması ve bunu izleyen soru-cevap bölümünden oluşur. Tez sınavı, öğretim elemanları, lisansüstü öğrenciler ve alanın uzmanlarından oluşan dinleyicilerin katılımına açık ortamlarda gerçekleştirilir. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda, jüri en geç on beş gün içinde toplanarak adayı tez savunma sınavına alır. (05 Ağustos 2017 tarihli 30145 sayılı Resmi Gazetede Yayınlanan Değişiklik-Madde 29-3)

² Tez sınavının tamamlanmasından sonra jüri, tez hakkında salt çoğunlukla “kabul”, “düzeltme” veya “ret” kararı verir. Jüri başkanı, jüri üyelerince imzalanmış karar tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve birinci fıkradaki usule göre tezini aynı jüri önünde yeniden savunur. Süresi içerisinde “düzeltme” savunmasına girmeyen öğrencinin enstitü ile ilişkisi kesilir. (Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği-Madde 29-4)

³ İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

Adı ve Soyadı : Süleyman ÜNLÜ
Danışman : Dr. Öğr. Üyesi Ediz ŞAYKOL
Türü ve Tarihi : Yüksek Lisans Tezi/2019
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Görüntü İşleme, jpeg, jpg, jpeg dosya boyutu küçültme

ÖZ

JPEG DOSYALARININ HAFIZA BOYUTLARININ DÜŞÜRÜLMESİ

Bu çalışma ile, JPEG dosyalarının bilgisayar hafızasında kapladıkları boyutun küçültülmesi amaçlanmıştır.

Çalışma süresince dijital görüntü işleme ve tekniklerine ilişkin incelemeler yapılmış ve mevcut bazı yöntemler günümüz örnekleri ile açıklanmaya çalışılmıştır.

Çalışmanın detaylandırıldığı, tez önerisinin sunulduğu bölümde ise, hali hazırda bu tez konusunu belirli yöntemlerle gerçekleştiren diğer çalışmalara yer verilmiş ve yeni yaklaşım ortaya atılmıştır.

Name and Surname : Süleyman ÜNLÜ
Supervisor : Dr. Lecturer Ediz ŞAYKOL
Type and Year : Master Thesis/2019
Major : Computer Engineering
Keywords : Digital Image Processing, jpeg, jpg, jpeg file size reduction

ABSTRACT

REDUCING MEMORY SIZES OF JPEG FILES

In this study, it is aimed to reduce the size of JPEG files occupied in computer memory.

During the study, digital image processing and techniques have been examined and some of the available methods have been tried to be explained with today's examples.

In the section where the study is elaborated and the thesis proposal is presented, other studies that have already carried out this thesis subject with certain methods have been included and a new approach has been proposed.

İÇİNDEKİLER

	Sayfa No.
ÖZ.....	ii
ABSTRACT.....	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
GİRİŞ.....	1

BİRİNCİ BÖLÜM

DİJİTAL GÖRÜNTÜ İŞLEME VE TEKNİKLERİ

1. DİJİTAL GÖRÜNTÜ İŞLEME.....	3
1.1. Dijital Görüntü İşleme Tarihçesi	4
2. DİJİTAL GÖRÜNTÜ İŞLEME TEKNİKLERİ	6
2.1. Ölçeklendirme.....	7
2.2. Kontrast Geliştirme.....	8
2.3. Parlaklık Ayarı.....	9
2.4. Çözünürlük.....	10
2.5. Filtreleme.....	11
2.5.1. Ortalama Filtresi.....	11
2.5.2. Gauss Filtresi.....	13

İKİNCİ BÖLÜM

GÖRÜNTÜ FORMATLARI VE JPEG ALGORİTMASI

1. GÖRÜNTÜ FORMATLARI.....	14
1.1. TIFF.....	15
1.2. BITMAP	15
1.3. GIF.....	16
1.4. PNG (Portable Network Graphics).....	16

1.5. EPS (Encapsulated PostScript file).....	17
1.6. RAW Image Files.....	18
1.7. JPEG (Joint Photographic Experts Groups).....	18
1.7.1. JPEG Algoritması.....	20
1.7.1.1. DCT Formülü.....	21
1.7.1.2. Niceleme (Quantization).....	23
1.7.1.3. Huffman Kodlaması.....	24

ÜÇÜNCÜ BÖLÜM

JPEG DOSYALARININ HAFIZA BOYUTLARININ DÜŞÜRÜLMESİ

1. HUFFMAN KODLAMASI.....	25
2. JPEG DOSYALARININ HAFIZA BOYUTLARININ DÜŞÜRÜLMESİ.....	27

DÖRDÜNCÜ BÖLÜM

İNCELEME

1. İNCELEME.....	35
SONUÇ.....	36
KAYNAKÇA.....	37
ÖZGEÇMİŞ.....	39

TABLÖLAR LİSTESİ

	Sayfa No.
Tablo 1. Standart Parlaklık Niceleme Tablosu.....	23
Tablo 2. ASCII Tablosu.....	26



ŞEKİLLER LİSTESİ

	Sayfa No.
Şekil 1. Dijital Resim.....	3
Şekil 2. İlk Görüntü İşleme Tekniği Uygulanan Resim.....	4
Şekil 3. Atlantik Okyanusu'nu Delikli Bant ile Geçen Resim.....	4
Şekil 4. NASA – Mars'a Yakın İlk Resim.....	5
Şekil 5. Ölçeklendirme Örnek - 1.....	7
Şekil 6. Ölçeklendirme Örnek - 2.....	7
Şekil 7. Kontrast Geliştirme Örnek - 1.....	8
Şekil 8. Kontrast Geliştirme Örnek - 2.....	8
Şekil 9. Parlaklık.....	9
Şekil 10. Çözünürlük.....	10
Şekil 11. Ortalama Filtresi Orijinal Görüntü Kesiti.....	12
Şekil 12. Ortalama Filtresi Ortalama Filtresi Uygulanmış Görüntü Kesiti.....	12
Şekil 13. Gauss Filtresi.....	13
Şekil 14. EPS Formatı.....	17
Şekil 15. PNG Görüntü.....	19
Şekil 16. JPEG Görüntü.....	19
Şekil 17. 255 Değerli Kırmızı Rengi.....	27
Şekil 18. 254 Değerli Kırmızı Rengi.....	27
Şekil 19. Orijinal resim.....	28
Şekil 20. Ara resim.....	29

Şekil 21. Sonuç resim.....	30
Şekil 22. Orijinal Resim Kırmızı Piksel Dağılımı.....	31
Şekil 23. Ara Resim Kırmızı Piksel Dağılımı.....	31
Şekil 24. Sonuç Resim Kırmızı Piksel Dağılımı.....	31
Şekil 25. Orijinal Resim Yeşil Piksel Dağılımı.....	32
Şekil 26. Ara Resim Yeşil Piksel Dağılımı.....	32
Şekil 27. Sonuç Resim Yeşil Piksel Dağılımı.....	32
Şekil 28. Orijinal Resim Kırmızı Piksel Dağılımı.....	33
Şekil 29. Ara Resim Yeşil Piksel Dağılımı.....	33
Şekil 30. Sonuç Resim Yeşil Piksel Dağılımı.....	33
Şekil 31. JPEG Dosya Küçültücü Algoritma.....	34

GİRİŞ

Cep telefonu ve fotoğraf makinası gibi cihazlar ile resim ve video çekmek, çekilen resim ve videoları sosyal platformlarda yayınlamak günümüz dünyasında oldukça popüler hale gelmiştir. Hemen hemen dünyanın her yerinde milyonlarca belki milyarlarca insan, Instagram, Snapchat, WhatsApp vb. uygulamalar üzerinden çektikleri video ve resimleri yayınlamakta, paylaşmaktadırlar. Milyonlarca insanın yapmış oldukları bu paylaşımlar neticesinde, belki de sonsuz büyüklükte veri yığınları oluşmaktadır. Bu veri yığınlarına bir de devletlerin ve güvenlik şirketlerinin sokaklara, alışveriş merkezlerine, işletme binalarına, yollara vs. güvenlik amaçlı koymuş oldukları kameralarda saklanan görüntüler de eklendiğinde, video ve görüntüler açısından ortaya çıkan veri yığını büyüklüğü düşünülemez, hesaplanamaz bir hal almaktadır.

Dijital platformlarda yer alan ve gitgide bir çığ gibi büyüyen resimler ve videoları saklamak için gerekli, yine sınırsız sayıda bellek gerekmektedir. Bu belleğin bir kısmı, resmi veya videoyu çeken kişiler tarafından kullanmış oldukları teknolojik argüman hafızasında saklanırken büyük çoğunluğu paylaşım yapılan platformlarda yer almaktadırlar.

Bugün, Google, Microsoft, IBM vb. uluslararası kullanıma açık onlarca teknolojik araç ve dijital iletişim kanalları barındıran şirketler, bu verileri saklayabilmek için tahmin edilemeyecek büyüklükte sistemlere sahip olmalıdırlar.

Günümüz dijital dünyasında bilgileri saklayabilmek için gerekli hard diskleri, sunucuları ve bunların maliyetleri düşünüldüğünde, milyonlarca ve belki milyarlarca doların, her yıl sadece görüntü ve video saklamak için harcandığını tahmin etmek zor olmamaktadır.

Bu çalışma ile birlikte, hafıza ve maddi avantajları nedeniyle dijital dünyada en çok kullanılan JPEG formatlı resimlerin hafızada kapladıkları alanın azaltılması ve bu işlemi gerçekleştirirken resim kalitesinde gözle görülebilir veya minimal düzeyde bir değişiklik olması amaçlanmaktadır.

Çalışmanın birinci bölümünde, resimler üzerinde değişiklik yapabilmek için gerekli dijital görüntü işleme ve teknikleri anlatılacaktır. Dijital görüntü işleme tekniklerinden bahsedilmesi ile birlikte günümüzde bu teknikleri kullanarak, kullanıcılara görüntü işleme imkanı veren bilinen bazı uygulamalara değinilecektir.

Çalışmanın ikinci bölümünde, dijital görüntü işleme yapılabilen görüntü formatları örnekleri ile birlikte verilecek, tezin konusu olan JPEG formatının algoritması incelenecektir.

Çalışmanın üçüncü bölümünde JPEG resimlerinin hafızada kapladıkları alanı azaltmak için yeni bir tez ortaya atılacaktır.

Çalışmanın son bölümü olan dördüncü bölümünde, ortaya atılan tezin derinlemesine analiz ve sonuçları verilmeye çalışılacaktır.



BİRİNCİ BÖLÜM

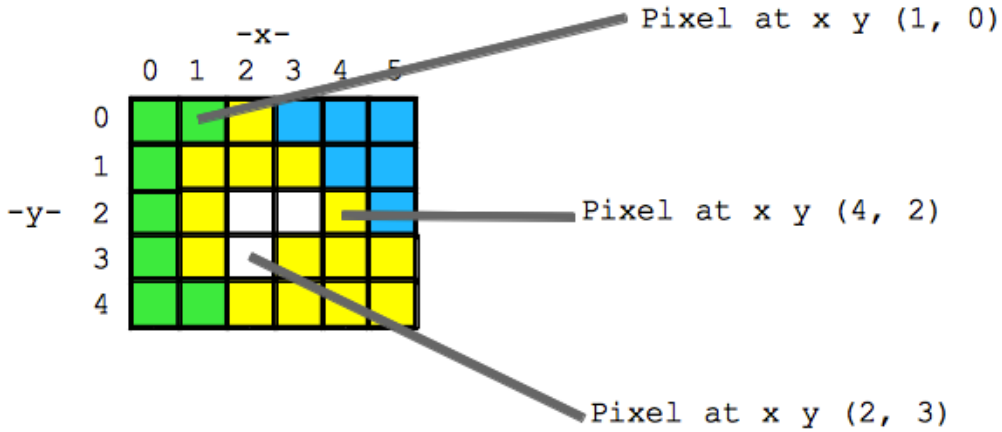
DİJİTAL GÖRÜNTÜ İŞLEME VE TEKNİKLERİ

1. DİJİTAL GÖRÜNTÜ İŞLEME

Dijital görüntü işleme, dijital form haline getirilmiş görüntüler üzerinde birtakım işlemler yapma bilimi olarak geçmektedir.¹ Dijital form durumuna getirilmiş görüntü ile, bilgisayar ortamında, fotoğraf makinalarında vb. cihazlarda saklanabilen, x ve y koordinatları bulunan piksellerin birleşimi neticesinde meydana çıkan 2 boyutlu resim kastedilmektedir. Diğer bir deyişle dijital görüntü işleme, görüntü kaydediciler tarafından elde edilen ve dijital form haline dönüştürülmüş görüntü üzerinde, çeşitli teknikler ile oynama yapılması sonucu, yeni bir görüntü ortaya çıkarma, görüntü üzerinde anlamlı veriler elde etme veya görüntü üzerinde ölçeklendirme, çözünürlük, parlaklık değişimi gibi sonuçların ortaya çıkmasına neden olan yöntemlerin tamamı olarak adlandırılabilir.

Şekil 1 ile dijital resim gösterilmeye çalışılmıştır.²

Şekil 1. Dijital Resim



Dijital resim, Şekil 1’de görüleceği üzere, x – genişlik ve y – yükseklik koordinatları bulunan, herhangi (x,y) koordinatının piksel adını aldığı ve bu piksellerin de renkleri sembolize ettiği, bilgisayar, çeşitli sunucular vb. dijital kaynaklarda bulunan resimlerdir.

¹ Doğu Akdeniz Üniversitesi, “Digital Image Processing”, <http://faraday.ee.emu.edu.tr/ee583/Lectures/EE%20583-Lecture01.pdf>, 10/05/2019

² Stanford University, “Introduction to Digital Images”, <https://web.stanford.edu/class/cs101/image-1-introduction.html>, 10/05/2019

1.1. Dijital Görüntü İşleme Tarihçesi

Bilinen ilk görüntü işleme uygulaması 1920'li yıllarda Londra'dan New York'a gönderilen bir gazete resmi içindi.³ Gazetede basılacak olan resim, Bartlane kablo ile resim taşıma sistemi ile, Atlantik üzerinden kablolar ile 1 haftada taşınabileceken 3 saatte taşınabilmiştir. Bartlane kablosu ile gerçekleştirilen görüntü işleme tekniğine göre, gönderilecek olan resim gönderilmeden önce, resimler için özelleştirilmiş teçhizatlar ile kodlanmış ve aktarım bittikten sonra görüntü, aynı teçhizatlar ile tekrar oluşturulmuştur. 1920 yılında gönderilmiş resim Şekil 2'de verilmiştir.

Şekil 2. İlk Görüntü İşleme Tekniği Uygulanan Resim



Görüntü işleme süreçleri ile oluşturulan ve 1922 yılında Atlantik Okyanusu'nu delikli bantlar ile 2 defa geçen resim ise Şekil 3'te gösterilmiştir.

Şekil 3. Atlantik Okyanusu'nu Delikli Bant ile Geçen Resim

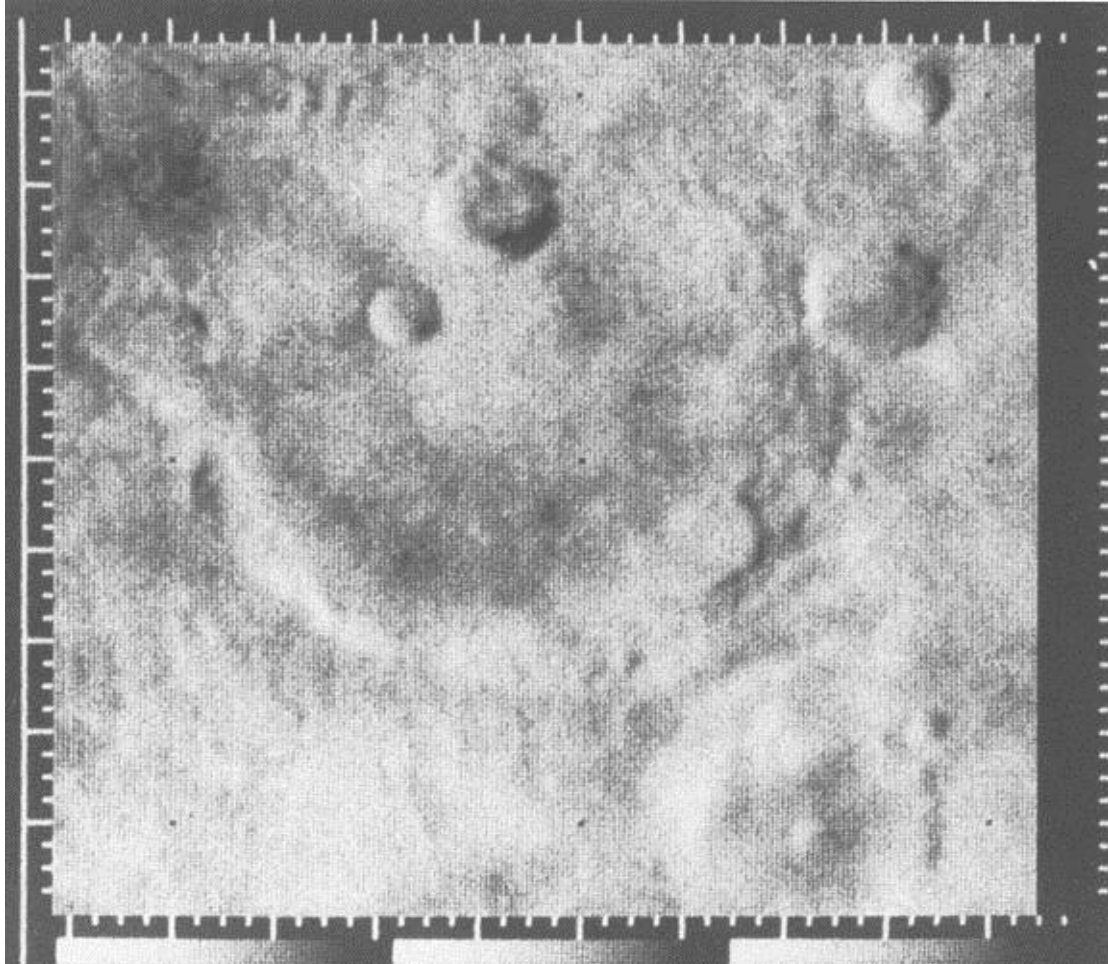


³ Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing", https://www.academia.edu/37539199/Digital_Image_Processing_-_3rd_Edition_Pearson, 10/05/2019

Belirtilen resimler ve yılları incelendiğinde, bahsedilen teknikleri, görüntü işlemenin ilk aşamaları olarak adlandırmak mümkün olmak ile birlikte, dijital görüntü işleme tekniği olarak adlandırmak mümkün olmamaktadır. Çünkü burada bahsedilen görüntülerin işlendiği ortamın dijital olarak adlandırılması gerekmektedir. Dijital ortam ile bilgisayarlar, laptoplar, notebooklar, internet dünyası vb. kastedilmektedir. Bu sebeptendir ki dijital görüntü işleme tekniklerinin ilk uygulanmaya başlandığı tarihler olarak 1960'lı yıllar olarak gösterilebilir.^{4,5}

NASA kaynakçasında belirtilen ve üzerinde dijital görüntü işleme teknikleri uygulanmış Mars'a yakın olarak çekilmiş ilk resim olan Şekil 4'teki resimdir.

Şekil 4. NASA – Mars'a Yakın İlk Resim



⁴ NASA, "Computers in Spaceflight: The NASA Experience", <https://history.nasa.gov/computers/Ch9-3.html>, 10/05/2019

⁵ IOSR Journal of Engineering, "The Origins of Digital Image Processing & Application areas in Digital ImageProcessing Medical Images", https://www.academia.edu/2324136/The_Origins_of_Digital_Image_Processing_and_Application_areas_in_Digital_Image_Processing_Medical_Images, 10/05/2019

2. DİJİTAL GÖRÜNTÜ İŞLEME TEKNİKLERİ

Dijital dünyanın özellikle 2000'li yılların başından itibaren hızla insan hayatına girmesi ve yaygın olarak kullanılması ile birlikte, her gün hızla artan resim paylaşımı neticesinde, dijital görüntü işleme teknikleri de resmin görünüşünü değiştirmek, görüntü kalitesini arttırmak gibi ihtiyaçlar doğrultusunda hızla gelişmiş ve gelişmeye devam etmektedir. Günümüzde resim ve video gibi multimedya araçları ile uğraşan her profesyonel, muhakkak görüntü işleme tekniklerinden faydalanmak suretiyle istedikleri resmi elde edebilmek için çeşitli uygulamalar kullanmaktadırlar. Görüntü işleme tekniklerinden yararlanan, kullanımı çok yaygın ve bilinen bazı uygulamalar aşağıda verilmiştir.

- Photoshop
- Paint
- GIMP
- Sketchapp
- Affinity Photo

Bu ve bunlara benzer uygulamalar ile resim üzerinde çeşitli matematiksel ve vektörel işlemler yapabilmek mümkündür.

Resim üzerinde piksel seviyesinde yapılan değişiklikler görüntü işleme teknikleri olarak adlandırılmaktadır. Bu teknikler ile, resmin parlaklığı, çözünürlüğü, ölçeklendirilmesi, boyutunun sıkıştırılması, kontrast ayarı, bulanıklaştırılması, filtreleme, kesit alma, görüntüyü yan ve ters çevirme gibi akla gelebilecek her özelliği değiştirilebilmektedir.

Yaygın olarak kullanılan görüntü işleme teknikleri aşağıdaki gibidir.

- Ölçeklendirme
- Kontrast Geliştirme
- Parlaklık Ayarı
- Çözünürlük
- Filtreleme

Yukarıda sayılan teknikler dışında birçok görüntü işleme tekniği de bulunmaktadır. Bu tekniklerin tamamından bahsetmek mümkün olmamaktadır.

2.1. Ölçeklendirme

Ölçeklendirme resmin boyutlarını büyültme veya küçültme işlemleridir. Ölçeklendirme tekniklerinin resim üzerinde uygulanması sonrasında Şekil 5 ve Şekil 6'ya benzer biçimde resimler elde edilebilir.

Şekil 5. Ölçeklendirme Örnek – 1



Şekil 6. Ölçeklendirme Örnek - 2



2.2. Kontrast Geliştirme

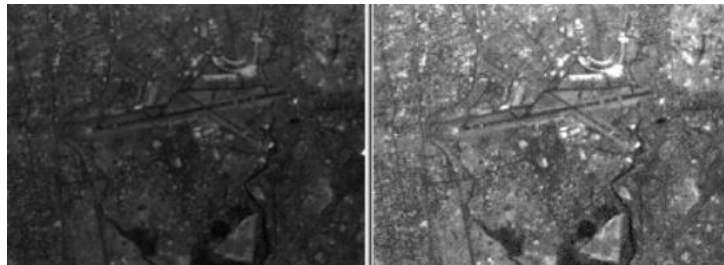
Kontrast ayarı görüntünün ayırt edilebilirliğini ifade eder. ⁶ Kontrast geliştirme, insan gözü ile ayırt edilemeyen detayları belirlemek veya bazı detayları gizlemek için kullanılmaktadır. Örneğin, bir resmin her pikseli aynı seviye gri ton ile kaplandığında resimdeki detayları fark edebilmek oldukça güçleşebilir.

Kontrast geliştirme, resmi oluşturan gri tonların dağılımına ilişkin geliştirilmiştir. Resim içerisinde yer alan her bir nesnenin, kendi belirgin özelliklerine sahip iken, resmin parlaklık veya gri tonların dağılımı neticesinde, sahip oldukları özellikler fark edilememektedir. Kontrast geliştirme teknikleri ile bu durumun ortadan kaldırılması amaçlanmaktadır. Bu durum Şekil 7 ve Şekil 8 ile gösterilmiştir.

Şekil 7. Kontrast Geliştirme Örnek - 1



Şekil 8. Kontrast Geliştirme Örnek - 2



⁶ Nihat Dipova, “Görüntü Analizi Yöntemlerinin Geoteknik Mühendisliğinde Kullanımı”, Akdeniz Üniversitesi, 2018

2.3. Parlaklık Ayarı

Parlaklık, her bir resim pikselinin temsil ettiği rengin ne ölçüde gri veya parlak gözükeceği ile ilgilidir. Örneğin, bir fotoğraf makinası ile çekilen resimde, fotoğraf çeken kişinin, güneş ışığına göre pozisyon belirlemesi gerekmektedir. Fotoğrafçı, güneş ışığını arkasına alıp çekimi yaptığında daha parlak görüntüler elde edebilir çünkü fotoğrafı çekilecek yüzey üzerine daha fazla ışın düşecek ve parlaklığın şiddeti artacaktır.

Parlaklık, pikselin temsil ettiği rengin görüntü şiddetidir. Görüntüyü oluşturan her piksel, her biri [0,255] aralığında değer alan kırmızı, yeşil ve mavi renkleri karışımı sonucu oluşan rengi temsil eder. Bu değerler kullanılarak parlaklık değeri hesaplanır ve piksel bu parlaklık değeri neticesinde görülebilir.

$$\text{Parlaklık} = \text{Kırmızı} \times 0.299 + \text{Yeşil} \times 0.587 + \text{Mavi} \times 0.114$$

olarak hesap edildiğinde, pikseli oluşturan renklerin her biri 255 değeri aldığıda yalnızca beyaz (tam parlaklık), her biri 0 değeri aldığıda parlaklığın olmadığı yalnızca siyah görüntü ortaya çıkmış olur.

Parlaklık değerini çeşitli görüntü işleme teknikleri ile değiştirmek mümkündür. Bu durum Şekil 9 ile gösterilmiştir.

Şekil 9. Parlaklık



2.4. Çözünürlük

Çözünürlük, bir resmin yatay ve dikey gerçek ölçütlerine göre ne ölçüde gösterildiği ile ilgilidir. Dijital ortamda genişliği 1024 ve yüksekliği 1024 piksel olan resmin, yeniden ölçeklendirilerek genişliği 768 ve yüksekliği 768 piksel durumuna getirilmesi, gerçek resminde yer alan ayrıntıların azalmasına, yani çözünürlüğün düşmesine neden olur çünkü bu işlem, gerçek görüntü üzerinden daha az sayıda örneklem alınarak yapılmaktadır. Bu kapsamda çözünürlük, gerçek görüntüden alınan örneklem sayısı olarak adlandırılabilir.

Dijital görüntü işleme tekniklerinin en çok kullanıldığı yöntemlerden bir tanesi çözünürlüğü artırma ve azaltma teknikleridir. Bilgisayar ekranlarında yer alan çözünürlük ayarları, televizyonlar ve kameralar, kameralı telefonlar vb. gibi görüntüyü gösteren her cihazda çözünürlük ayarı bulunmaktadır.

Görüntü işleme teknikleri ile çözünürlük seviyesi değiştirilmiş bir resmin örnek görüntüleri Şekil 10'da gösterilmiştir.

Şekil 10. Çözünürlük⁷



⁷ İbrahim Çayiroğlu, "Görüntü İşleme Ders Notları", http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-1.Hafta.pdf, 10/05/2019

2.5. Filtreleme

Dijital görüntü işleme teknikleri arasında filtreleme teknikleri önemli bir yer tutar. Filtreleme, adı üstünde, görüntüyü oluşturan pikseller üzerinde bir takım işlemler yapılması neticesinde, bazı piksellerin yok edilmesi ve değişikliğe uğraması işlemidir.

Filtreleme teknikleri ile bir dizi işlemler sonucu, görüntü üzerinde aşağıdaki durumlar meydana getirilebilir.

- Görüntü bulanıklaştırma
- Gürültüyü azaltma
- Görüntü kenarlarını belirleme

En çok kullanılan filtreleme teknikleri:

- Ortalama Filtresi (Mean Filter)
- Gauss Filtresi (Gaussian Filter)
- Sobel Filtresi (Sobel Filter)

olarak söylenebilir.

2.5.1. Ortalama Filtresi

Ortalama filtresi (mean filter), görüntüyü yumuşatma için kullanılmaktadır. Burada, yumuşatma ile, pikseller arasındaki değişim miktarının azaltılması kastedilmektedir.

Ortalama filtresi, piksel ve piksele en yakın komşu piksellerin ortalamaları alınarak, pikselin kendisine bu değer yazılması mantığı ile çalışır. Resimde ortalama hesabı yapabilmek için, ihtiyaca göre $n \times n$ boyutlu bir şablon seçilir. Bu şablon ile görüntü pikselleri taranması neticesinde ortalamalar ilgili piksellere yazılır.

Ortalama filtresinin çalışma mantığı Şekil 11 ile gösterilmiştir.

Şekil 11. Ortalama Filtresi Orijinal Görüntü Kesiti

10	80	70	2	3	4
20	90	60	3	3	5
30	40	50	2	4	1
1	2	2	4	4	2
1	2	4	5	5	3
9	8	7	6	5	2

Şekil 11 ile verilen örnek bir resim kesintisi üzerine 3 x 3 boyutunda ortalama filtresi uygulandığında, koyu renkle yazılmış 1 değerli piksel yerine gelecek olan yeni değeri,

$$(10 + 20 + 30 + 40 + 50 + 60 + 70 + 80 + 90) / 9 = 50$$

olarak hesaplandığında ortaya çıkacak yeni görüntü Şekil 12'deki gibi olacaktır.

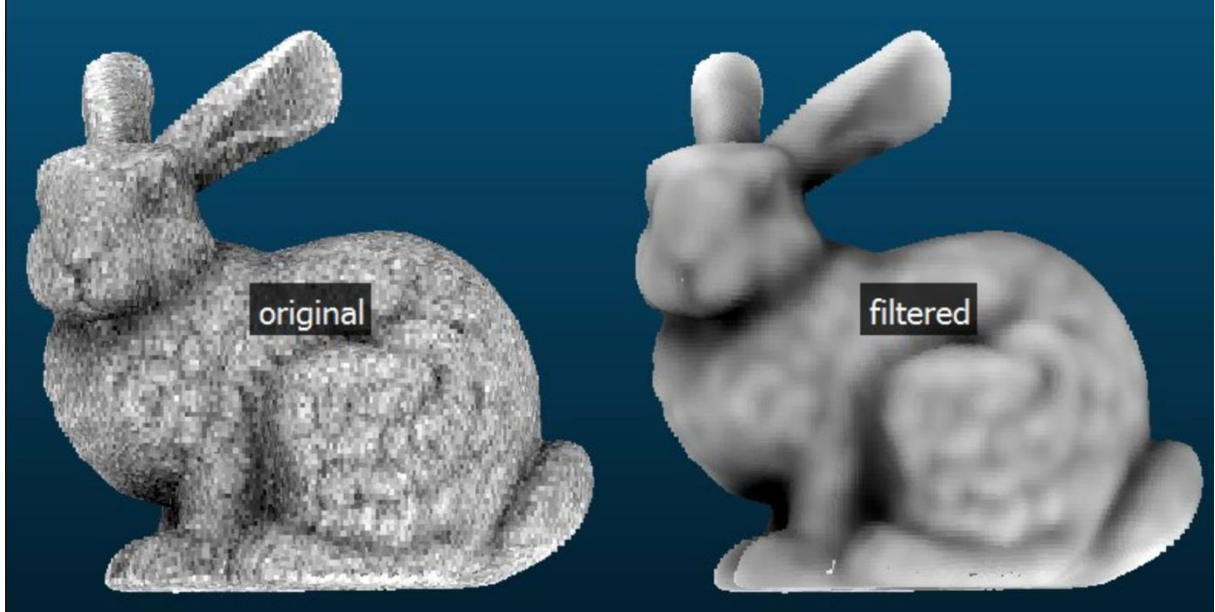
Şekil 12. Ortalama Filtresi Ortalama Filtresi Uygulanmış Görüntü Kesiti

10	80	70	2	3	4
20	50	60	3	3	5
30	40	50	2	4	1
1	2	2	4	4	2
1	2	4	5	5	3
9	8	7	6	5	2

2.5.2. Gauss Filtresi

Gauss filtresi, yüksek sıklığa sahip pikselleri yok ederek, resim bulanıklığını arttırma yoluyla gürültüleri temizlemek için kullanılmaktadır.^{8, 9} Burada, gürültü ile, resimde istenmeyen noktalar kastedilmektedir.

Şekil 13. Gauss Filtresi



Gauss filtresi uygulaması ortanca filtresine benzer şekilde, değişime uğrayacak piksel değeri, piksel ve bu piksele komşu piksellerin belirli bir konvolüsyon (evrişim) sürecinden geçmesi sonrası belirlenmektedir. Konvolüsyon işlemi için belirlenecek $n \times n$ boyutlu çekirdek değerleri Gauss dağılımından yararlanılarak elde edilmektedir.

⁸ Cornell Üniversitesi, “Computer Vision”,
https://www.cs.cornell.edu/courses/cs6670/2011sp/lectures/lec02_filter.pdf, 10/05/2019

⁹ Auckland Üniversitesi, “Gaussian Filtering”,
https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf,
10/05/2019

İKİNCİ BÖLÜM

GÖRÜNTÜ FORMATLARI VE JPEG ALGORİTMASI

1. GÖRÜNTÜ FORMATLARI

Dijital görüntü işleme teknikleri, bilinen her dijital resim formatlarına uygulanabilmektedir. Dijital görüntü dosyaları, kamera, fotoğraf makinası vb. cihazlar ile çekilen resimlerin, bilgisayar, hard disk, telefon vb. dijital ortamlarda saklanabilecek duruma getirilmiş dosyalardır.

Fotoğrafın çekilmesi sonucu elde edilen ham resim, istenilen görüntü kalitesi veya kullanılacağı ortama göre ihtiyaç doğrultusunda, belirli formatlara dönüştürülerek dijital ortamında saklanabilir. Bahsedilen formatlar, görüntü işleme teknikleri sonucunda ortaya çıkan resim, orijinal resimde yer alan bilgilerin, çeşitli algoritmalar ile sıkıştırılmış olup olmadığına, kullanılan renk aralıklarına, alfa kanalına sahip olup olmadıklarına ve mevcut piksel değerlerinin korunup korunmadıklarına göre değişiklik göstermektedirler.

Resim formatları, yıllar içerisinde çeşitli firmalar tarafından ortaya atılmış ve dijital ortamlarda kullanılabilen duruma getirilecek biçimde geliştirilmiş formatlardır. Bilinen bazı resim formatları¹⁰

- TIFF (Tagged Image File Format)
- Bitmap
- GIF (Graphics Interchange Format)
- PNG (Portable Network Graphics)
- EPS (Encapsulated PostScript file)
- RAW Image Files
- JPEG (Joint Photographic Experts Groups)

olarak verilebilmektedir.

¹⁰ Michigan Üniversitesi, "Image File Formats", <https://guides.lib.umich.edu/c.php?g=282942&p=1885348>, 10/05/2019

1.1. TIFF

TIFF (Tagged Image File Format), 1986 yılında şunda Adobe şirketinin bir parçası olan Aldus şirketi tarafından ortaya atılmış ve geliştirilmiştir.¹¹

TIFF dosyaları, dijital ortamda .tiff veya .tif dosya uzantıları olarak tutulan dosyalardır. Bu tür dosyalar, orijinal resmin yüksek kaliteli çözünürlüğü, sıkıştırılmaması ve orijinal resim şeffaflık değerlerinin de tutulabileceği biçimde elde edilen dosyalardır. Her pikseli 24 bit değer alabilir. Yüksek kalitede olmaları nedeniyle tutulduğu hafızada diğer dosya biçimlerine oranla daha fazla yer tutarlar.

TIFF dosyaları, genellikle yüksek kalite yazıcı çıktıları, resmin her ince detayının gösterilmesi gerektiği yayınlar, kısaca yüksek renk derinliğine ihtiyaç duyulan durumlarda gösterilmektedirler.

1.2. BITMAP

Bitmap kelimesi, bit ve map İngilizce kelimelerin yan yana gelmesi ile oluşmuş ve haritalanmış bit anlamı taşımaktadır. Dijital görüntü piksellerinde yer alan renkler bitmap haritalama yöntemi ile tutulmaktadır. Diğer bir deyişle bitmap dijital resimlerinde her piksel, ilgili işletim sistemi hafızasında indekslenerek saklanmaktadır.

Bitmap dosya formatı, bitmap dijital resimlerin özellikle Microsoft ve OS/2 işletim sistemlerinde tutulabilmesi için geliştirilmiştir. Bitmap, günümüzde halen Microsoft firması tarafından geliştirilmektedir.

Bitmap dosyaları dijital ortamda .bmp dosya uzantıları olarak tutulan dosyalardır. Bu dosyalar için şu özellikler söylenebilir:

- Yüksek kalite görüntü ile saklanır.
- Sıkıştırılmamış orijinal resim
- Her pikseli 32 bite kadar veri tutabilir.
- Hafızada diğer dosya biçimlerine göre oldukça fazla yer tutar.
- Yüksek kalite taramalar için kullanılır.

¹¹ Indiana Üniversitesi, “What is the TIFF graphics file format?”, <https://kb.iu.edu/d/afjn>, 10/05/2019

1.3. GIF

GIF (Graphics Interchange Format), 1987 yılında CompuServe Information Service şirketi tarafından ortaya atılmış ve geliştirilmiş bir resim formatıdır.¹²

GIF dosyaları yalnızca 256 değişik renk barındırabilirler. Az sayıda renk barındırmalarından dolayı özellikle hafıza az yer tutacakları için web tabanlı uygulamalarda sıklıkla kullanılmaktadırlar. GIF dosyaları ayrıca resimlerin şeffaflık düzeylerini de destekleyebilmektedirler.

GIF dosyaları sıkıştırılmamış dosyalar olup, resimlerde animasyon gösterimi de sağlayabilmeleri açısından, görsel hareketlilik de sağlayabilmektedirler. Bu tarz dosyalar dijital ortamda .gif uzantısı ile tutulmaktadır.

1.4. PNG (Portable Network Graphics)

PNG dosya formatı GIF dosya formatının bazı kısıtlarını ortadan kaldırmak ve geliştirmek için 1995 yılında ortaya atılmış dosya formatıdır.¹³ GIFF formatını oluşturan algoritma aynı zamanda lisans patentine sahip olması nedeniyle, patent durumunun ortadan kaldırılması da yine bu dosya formatının oluşturulma nedenleri arasındadır.

Literatürde, Taşınabilir Ağ Grafikleri olarak geçen bu dosya formatı, GIFF formatlı dosyaların 256 değişik rengine karşı 16 milyona kadar renk barındırabilmektedir.

PNG dosyaları sıkıştırılmamış dosyalardır. Her bir pikseli 24 bit RGB ve alfa kanalı ile birlikte 32 bit RGBA yani şeffaflık ve opaklığı destekleyecek biçimde geliştirilmiştir. Dijital ortamda .png uzantısı ile gösterilen bu dosyalar, sıkıştırılmamış olmaları ve her bir pikseli 32 bite kadar yer kaplayabilmesi nedeniyle hafızada önemli bir yer tutmaktadırlar.

PNG dosyaları, GIFF dosyaları gibi internet tabanlı uygulamalarda tutuldukları gibi bilindiği üzere Mac OS işletim sisteminde ekran görüntüsü alınan resimlerinde varsayılan görüntü biçimidir.

¹² World Wide Web Consortium, "GIFF", <https://www.w3.org/Graphics/GIF/spec-gif87.txt>, 10/05/2019

¹³ Greg Roelofs, "PNG Site Map", <http://www.libpng.org/pub/png/#history>, 10/05/2019

1.5. EPS (Encapsulated PostScript file)

EPS (Encapsulated PostScript file) formatı, genellikle, görüntü bilgilerini vektörel olarak tutmak ve platform bağımsızlığı sağlaması neticesinde geliştirilmiştir.

EPS dosyaları, 2 boyutlu vektörel grafikleri, bitmap ve metin dosya bilgilerini tutabilmek ile birlikte günümüzde Adobe Photoshop, CorelDRAW ve FlexiSign gibi birçok resim düzenleyici program tarafından desteklenmektedir ve genellikle ikon yapımı sırasında kullanılmaktadır. Örnek EPS dosyası içeriği Şekil 14 ile gösterilmiştir.

Şekil 14. EPS Formatı

```
%!PS-Adobe-3.0 EPSF-3.0
%%Document-Fonts: Times-Roman
%%Title: circle.eps
%%Creator: PS_Write.F
%%CreationDate: 02-Aug-99
%%Pages: 1
%%BoundingBox: 36 36 576 756
%%LanguageLevel: 1
%%EndComments
%%BeginProlog
%%EndProlog
/inch {72 mul} def
/Palatino-Roman findfont
1.00 inch scalefont
setfont
0.0000 0.0000 0.0000 setrgbcolor
%% Page: 1 1
save
  63 153 moveto
  newpath
  63 153 moveto
  549 153 lineto
  stroke
  newpath
  549 153 moveto
  549 639 lineto
  stroke
  newpath
  549 639 moveto
  63 639 lineto
  stroke
  newpath
  63 639 moveto
  63 153 lineto
  stroke
  newpath
  360 261 108 0 360 arc
  closepath stroke
  newpath
  361 357 moveto
  358 358 lineto
  353 356 lineto
  348 353 lineto
  278 237 lineto
  stroke
  171 261 moveto
  0.0000 0.0000 0.0000 setrgbcolor
/Palatino-Roman findfont
  0.250 inch scalefont
setfont
(This is "circle.plt".) show
  171 342 moveto
/Palatino-Roman findfont
  0.125 inch scalefont
setfont
(This is small print.) show
restore showpage
%%Trailer
```

1.6. RAW Image Files

Raw dosyaları, çeşitli kamera, fotoğraf makinası vb. dijital araçlar ile çekilen görüntülerin, görüntü işleme teknikleri ile henüz işlenmemiş (ham veri) bilgilerini tutan dosyalardır. Dijital ortamda .raw, .cr2, .nef, .orf, .sr2 gibi dosya uzantıları ile tutulan bu dosyaların, görsellik kazanabilmeleri için işlenmeleri gerekmektedir.

1.7. JPEG (Joint Photographic Experts Groups)

Günümüzde kullanımı en yaygın dosya formatı olan JPEG, adını, geliştiricileri olan Joint Photographic Experts Group'tan almıştır.¹⁴

JPEG dosyaları, dijital görüntü yakalayıcı araçlar tarafından çekilen ham veri içeren görüntü dosyalarının, JPEG algoritması kullanılarak sıkıştırılmış biçimidir. Bu sıkıştırma yöntemi ile, dosyalar dijital ortama, orijinal resmin hafızada kaplayacağı yer azaltılırken görüntü piksellerinde insan gözü ile fark edilemeyecek veya minimum fark edilebilen biçimde kaydedilmektedirler.

JPEG dosyalarında her piksel, 24 bit renk ile temsil edilebilmektedir. JPEG dosyaları alfa kanalını desteklememektedirler. Bu dosyalar dijital ortamda .jpg veya .jpeg uzantısı ile tutulmaktadır.

JPEG dosyaları, hafızada az yer tuttukları için, özellikle internet dünyasında sıkça kullanılmaktadır. Bu durum kullanıcıların internet hızlarından kaynaklanmaktadır. Bir internet sayfası ziyaretinde, sayfa içerisinde yer alan resimlerin hafıza boyutları ne kadar az ise, sayfada yer alan bilgiler kullanıcı tarayıcısı tarafından o derece hızlı yorumlarını ve bu durum sayfanın hızlı açılmasını sağlar. Sayfanın hızlı açılması kullanıcı deneyimi ve sitede bulunma süresi açısından oldukça önemlidir.

Örnek bir PNG dosyası ve bu dosyanın JPEG durumuna getirilmiş hali Şekil 15 ve Şekil 16 ile gösterilmiştir. Bu gösterimi yapabilmek için Mac OS işletim sisteminden ekran görüntüsü alınmış ve PNG uzantılı bu görüntü, resim dönüştürücüler kullanılarak JPEG dosyasına çevrilmiştir.

¹⁴ JPEG, "About JPEG", <https://jpeg.org/about.html>, 10/05/2019

Şekil 15. PNG Görüntü



Şekil 16. JPEG Görüntü



Şekil 15'te yer alan PNG dosyasının büyüklüğü 662 KB iken Şekil 16'da yer alan dönüştürülmüş JPEG dosyası 63 KB olarak gözlemlenmiştir.

1.7.1. JPEG Algoritması

JPEG Algoritması 4 ana adımdan oluşmaktadır.¹⁵

- Renk dönüşümü
- Ayrık Kosinüs Dönüşümü (DCT)
- Niceleme
- Huffman Kodlaması

Renk dönüşümü adımı, görüntüyü oluşturan renk uzayı olan RGB renk uzayının, YCbCr renk uzayına dönüştürülmesi işlemidir. Burada Y harfi, görüntü parlaklığını, Cb ve Cr simgeleri ise mavi ve kırmızı farklılıklarını temsil etmektedirler.

RGB renk uzayının YCbCr renk uzayına dönüşüm formülü aşağıda verilmiştir.¹⁶

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.1689 & -0.3316 & 0.50 \\ 0.4998 & -0.4185 & -0.0812 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Buradan,

$$Y = 0.299 \times R + 0.587 \times G + 0.144 \times B$$

$$Cb = -0.1689 \times R - 0.3316 \times G + 0.5 \times B$$

$$Cr = 0.4998 \times R - 0.4185 \times G - 0.0812 \times B$$

olarak elde edilir.

Ayrık kosinüs dönüşüm adımı öncesinde resim örneklendirilmektedir. Bu örnekleme, insan gözünün aydınlığa duyarlı fakat fazla renkliliğe karşı duyarsız olması neticesinde yapılmaktadır. Örnekleme sayesinde pikseller ayrık kosinüs dönüşümü için de hazır durumda bulunacaklardır.

¹⁵ Cornell Üniversitesi, "JPEG: Image compression algorithm", <http://pi.math.cornell.edu/~web6140/TopTenAlgorithms/JPEG.html>, 10/05/2019

¹⁶ Microsoft, "Color Conversion (RGB to YCbCr)", https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdprfx/b550d1b5-f7d9-4a0c-9141-b3dca9d7f525, 10/05/2019

Örnekleme işlemi her piksel değerinden 128 sayısının çıkartılması ile yapılmaktadır. Bunun nedeni [0, 255] aralığında değer alan piksellerin [-128, 127] aralığında değerler alması sonucunda verileri 0 etrafında merkezileştirmektedir.

Ayrık kosinüs dönüşüm adımında, örneklenmiş ve YCbCr renk uzayına dönüştürülmüş olan resim, tekniğin uygulanması için 8 x 8 bloklara ayrılır ve blokların her biri 2 boyutlu DCT dönüşümüne tabii tutulur. Bu dönüşüm ile birlikte görüntü pikselleri farklı kosinüs dalgaları ile temsil edilebilecek ve böylelikle insan gözünün algılayamadığı yüksek sıklığa sahip renkler ortadan kaldırılabilir.

1.7.1.1. DCT Formülü

Bir resmin belirli bir noktadaki pikseline ait DCT değeri, aşağıdaki denklem ile verilmiştir.¹⁷

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases}$$

Burada p(x,y), resmin (x,y) noktasına ilişkin değeri, i ve j koordinatları ve N ise DCT bloğunun boyutu olarak belirtilir. JPEG algoritması 8 x 8 bloğu kullandığından N = 8 alındığında denklem aşağıdaki gibi olmaktadır.

$$D(i,j) = \frac{1}{4} C(i)C(j) \sum_{x=0}^7 \sum_{y=0}^7 p(x,y) \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right]$$

¹⁷ Hong Kong Çin Üniversitesi, "Image Compression and the Discrete Cosine Transform", <https://www.math.cuhk.edu.hk/~lmlui/dct.pdf>, 10/05/2019

DCT matrisini elde edebilmek için DCT denkleminin matris formunu elde etmek gerekmektedir. DCT denkleminin matris formu elde edildikten sonra $N \times N$ boyutlu bir bloğa DCT uygulanabilir. Bu uygulama formülü,

$$D = T \times M \times T'$$

olarak verilir ki, burada T, DCT denkleminin matris formu, M, DCT uygulanacak, resim üzerinde seçilmiş $N \times N$ boyutlu piksellerin her birinin değerinden 128 çıkarılmış, merkezileştirilmiş matris ve T' ise T matrisinin transpozu olarak yer almaktadırlar.

Örnek 1: 4 x 4 boyutlu DCT matrisinin elde edilişi;

DCT denkleminin matris formu:

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{cases}$$

için, $N = 4$ olduğunda, oluşacak nokta çiftleri

$$\begin{bmatrix} (0,0) & (0,1) & (0,2) & (0,3) \\ (1,0) & (1,1) & (1,2) & (1,3) \\ (2,0) & (2,1) & (2,2) & (2,3) \\ (3,0) & (3,1) & (3,2) & (3,3) \end{bmatrix}$$

olarak belirtilir. Bu noktalar formül i, j için yerine konulduklarında elde edilecek T matrisi için;

$$i = 0, j = 0 : T_{0,0} = \frac{1}{\sqrt{4}} = 0.5$$

$$i = 0, j = 1 : T_{0,1} = \frac{1}{\sqrt{4}} = 0.5$$

$$i = 0, j = 2 : T_{0,2} = \frac{1}{\sqrt{4}} = 0.5$$

$$i = 0, j = 3 : T_{0,3} = \frac{1}{\sqrt{4}} = 0.5$$

$$i = 1, j = 0 : T_{1,0} = \sqrt{\frac{2}{4}} \cos(\pi/8) = 0.6532$$

$$i = 1, j = 1 : T_{1,1} = \sqrt{\frac{2}{4}} \cos(3\pi/8) = 0.2705$$

gibi işlemler sonucunda T matrisi elde edilir. Elde edilen T matrisi sonucunda

$$D = T \times M \times T'$$

denkleminde DCT matrisi N x N bloğa uygulanmış olur.

1.7.1.2. Niceleme (Quantization)

Niceleme adımı temel amaç, değerleri 0 etrafında toplamaya çalışmaktır. Bunu yapabilmek için elde edilen DCT matrislerinin her biri standart niceleme tablosu matrisine bölünür. Bu matris Tablo 1 ile verilmiştir.

Tablo 1. Standart Parlaklık Niceleme Tablosu¹⁸

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

¹⁸ Stanford Üniversitesi, "JPEG standard", <https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>, 10/05/2019

1.7.1.3. Huffman Kodlaması

JPEG algoritmasının son adımında elde edilen veriler, Huffman kodlaması yöntemi ile sıkıştırılırlar.

JPEG algoritmasının 2 önemli amacı bulunmaktadır:

- Orijinal resmin hafızada kapladığı yeri azaltmak
- Kaybolan veya değişen piksel değerlerinin insan gözü ile ayırt edilememesini sağlamak



ÜÇÜNCÜ BÖLÜM

JPEG DOSYALARININ HAFIZA BOYUTLARININ DÜŞÜRÜLMESİ

1. HUFFMAN KODLAMASI

Huffman kodlaması, diğer adı ile Huffman Coding, JPEG resimlerinin elde edilişi sırasında, sıkıştırma aşamasında kullanılan sıkıştırma algoritmasıdır.

Huffman kodlaması, bir nesnede yer alan verilerin en çok tekrar eden kısmını daha az bit ile göstermek için geliştirilmiştir.¹⁹

Huffman algoritması aşağıdaki örnek ile açıklanabilir.

Örnek 1: Tablo 2’de yer alan Ascii tablosuna göre A harfi, 65 değeri ile, B harfi, 66 ve L harfi ise 74 değeri ile sembolize edilmektedir.

65 değeri, 8 bit ve ikili gösterim olarak 01000001, 66 değeri 01000010 ve 74 değeri ise 01001010 ile ifade edilmektedir. Buna göre ABLA kelimesi hafızada 32 bitlik bir alanda saklanabilecektir. Yani;

ABLA=01000001010000100100101001000001

Burada A harfi 0, B harfi 10 ve L harfi 11 ile temsil edilseydi bu kelimeyi 6 bit ile göstermek mümkün olabilecektir. Yani;

ABLA=010110

Huffman kodlamasındaki amaç, verilecek temsili bitler arasında ilgili nesneyi, olabilecek en düşük bit ile yazmaktır. Örnekte yer alan ABLA kelimesi için her harfi 4 bit ile de göstermek mümkün iken olası en iyi gösterim 6 bitlik olarak hesaplanmıştır.

Huffman kodlaması çözülrken bit serisi soldan sağa doğru okunur. Temsili verilen bitlerden ilgili bit gelinceye kadar devam eder ve bit gelince okumaya yeni bit arayışı ile devam edilir. Yani 010110 soldan sağa okunurken önce 0, sonra 10, daha sonra 11 ve en son 0 elde edilmesi ve bunlara karşılık gelen ABLA elde edilir.

¹⁹ Cardiff Üniversitesi, “Huffman Coding”, <https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node210.html>, 10/05/2019

Tablo 2. ASCII Tablosu²⁰

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
\$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
;	59	0073	0x3b	[91	0133	0x5b	{	123	0173	0x7b
<	60	0074	0x3c	\	92	0134	0x5c		124	0174	0x7c
=	61	0075	0x3d]	93	0135	0x5d	}	125	0175	0x7d
>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
?	63	0077	0x3f	_	95	0137	0x5f				

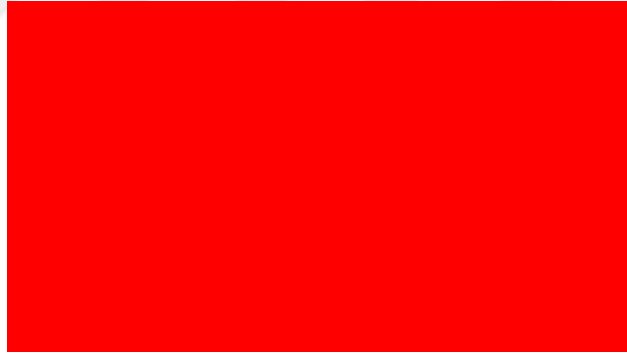
²⁰ IBM, "ASCII table", https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ioaq100/ascii_table_appendix.htm, 10/05/2019

2. JPEG DOSYALARININ HAFIZA BOYUTLARININ DÜŞÜRÜLMESİ

JPEG dosyalarını oluşturan algoritma ikinci bölümde verilmişti. Bu algoritmaya göre, son adım olan sıkıştırma adımı, Huffman kodlaması üzerinde olduğundan söylenebilir ki, Huffman kodlaması yöntemine gelindiğinde, ne kadar çok tekrar eden veri var ise o kadar iyi bir sıkıştırma ve düşük dosya boyutu elde edilebilir. Örneğin Huffman algoritması ile sıkıştırılacak bir metin için, **aaaabbbbccccddddeeee** metninin sıkıştırılması ile **abcdefghijklmnoöprst** sıkıştırılması farklı olacaktır. Bu durum resimde yer alan ve [0-255] arasında değer alan pikseller için de aynıdır. Her piksel aynı renkte olsaydı JPEG algoritması ile tutulduğunda hafıza hiç yer kaplamayacak biçimde olabileceği gibi her pikselin farklı olduğu durumda ise, algoritma daha fazla bit ile sıkıştırma işlemine çalışacak ve dosya boyutu da o ölçüde fazla olacaktır.

İnsan gözünün, tüm pikselleri 255 kırmızı değerini taşıyan ile 254 kırmızı değerini taşıyan arasındaki farkı algılamasının mümkün olmadığı Şekil 17 ve Şekil 18 da gösterilmiştir.

Şekil 17. 255 Değerli Kırmızı Rengi



Şekil 18. 254 Değerli Kırmızı Rengi



Verilen bilgiler ışında, yeni bir yaklaşıma göre sıkıştırılmış bir JPEG dosyasındaki her bir elemanın seçilmiş bir N sayısına bölünüp kaydedilmesi işlemi sonucunda, veriler, [0,255] aralığından [0,127] aralığına çekilecekleri için Huffman kodlaması bu veriler üzerinde daha çok sıkıştırmaya neden olacaktır. Sıkıştırma işlemi sonrasında, elde edilen ara resim üzerinde tersine bir işlem olarak, her pikselin aynı N sayısı ile çarpılması neticesinde veriler [0-255] aralığına tekrar geri çekilecek ve orijinal resimde değişiklik olmayacağı gibi dosya boyutu da çoğu durumda azalacaktır. Dosya boyutunun azalmadığı durumlar ayrıca incelenecektir.

Ara resmin her bir pikselinin N sayısı ile çarpılması sonucu ortaya çıkan sonuç resminin orijinal resme göre hafızada daha az yer kaplamasının nedeni, ara resimdeki piksel değerlerinin orijinal resimdeki piksel değerlerine oranla birbirlerine daha yakın değerler almalarıdır.

Çalışmaya konu olan örnek resim, Şekil 19 ile, her bir pikselin 2 sayısına bölümü ile elde edilen ara resim, Şekil 20 ve ara resimdeki her bir pikselin 2 sayısına çarpımı ile elde edilen son resim Şekil 21 ile gösterilmiştir.

Şekil 19. Orijinal resim



Şekil 20. Ara resim



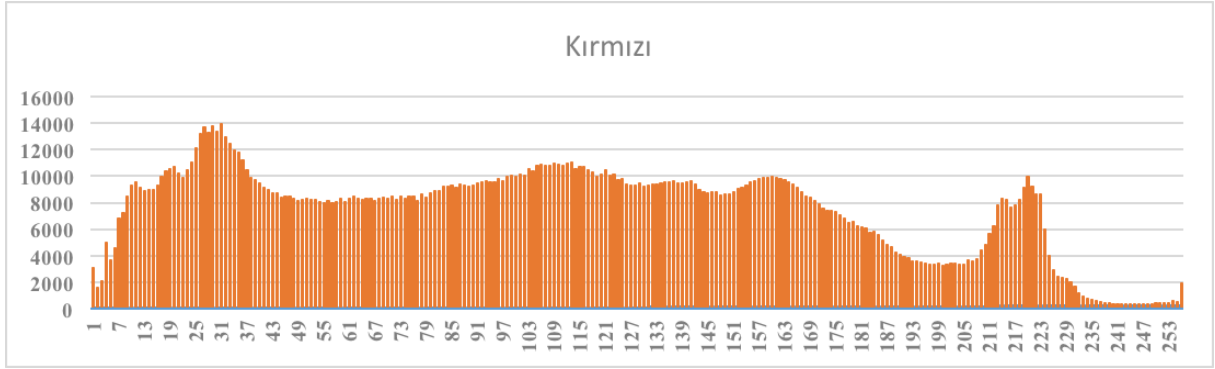
Şekil 21. Sonuç resim



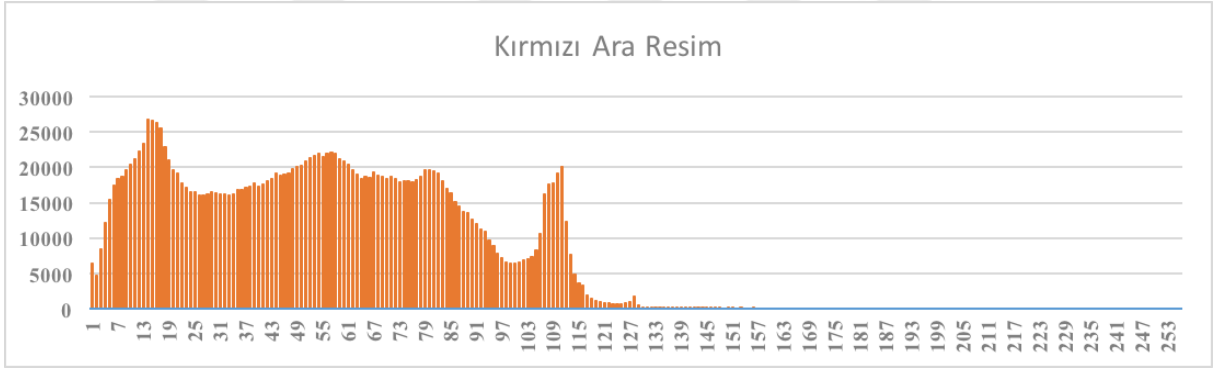
Burada, orijinal resim 276 KB, ara resim 152 KB ve sonuç resmi ise 208 KB hafıza yer tutmaktadırlar.

Yeni algoritmaya göre N değerinin seçimi oldukça önemlidir. Her bir pikselin bölüneceği ve çarpılacağı N değeri 2 olarak seçildiğinde yeni görüntü ile orijinal görüntü arasında insan gözü ile görülemeyecek değişimlerin olması beklenmekte iken N değeri arttıkça bu beklenti azalmaktadır. Çünkü $N = 2$ iken resimde yer alan her piksel değeri en fazla 1 değer azalarak dönüşüme tabii tutulurken $N = 4$ iken piksellerde gözle görülebilir biçimde değişimler yaşanabilmektedir. Orijinal, ara ve sonuç resimlerine ilişkin histogram grafikleri aşağıdaki şekillerde verilmiştir.

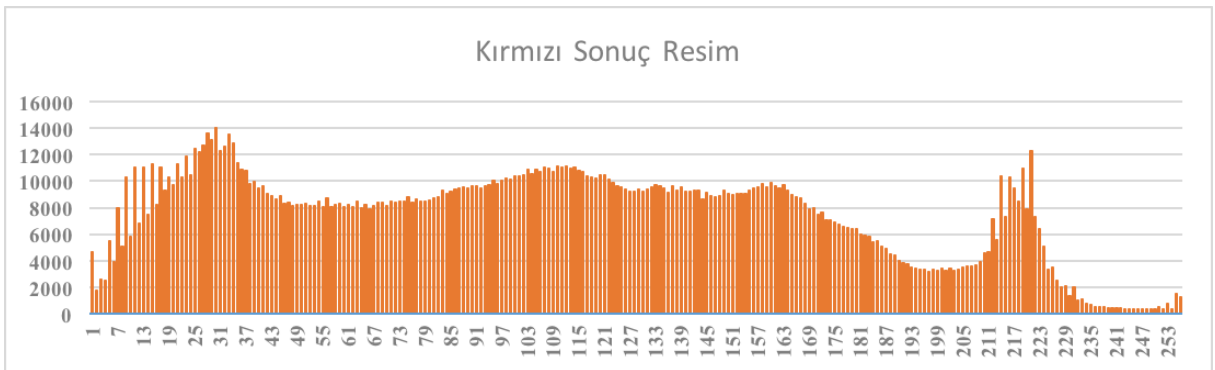
Şekil 22. Orijinal Resim Kırmızı Piksel Dağılımı



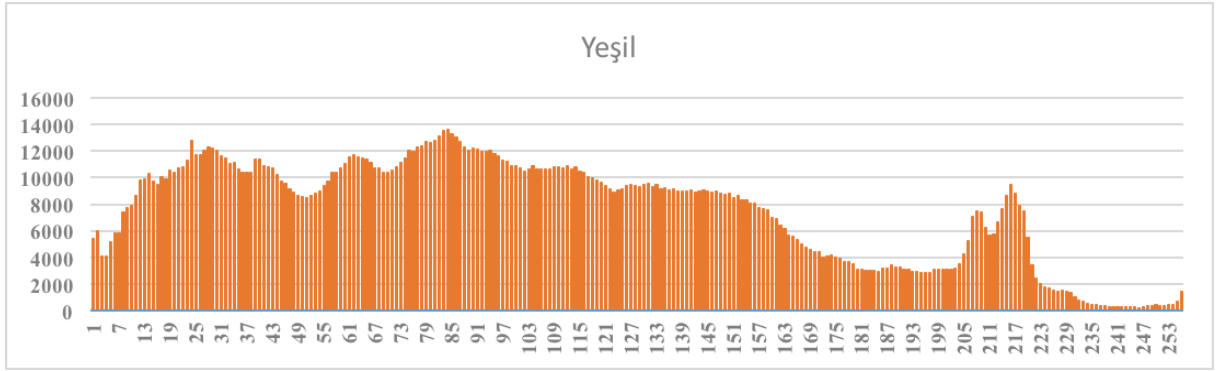
Şekil 23. Ara Resim Kırmızı Piksel Dağılımı



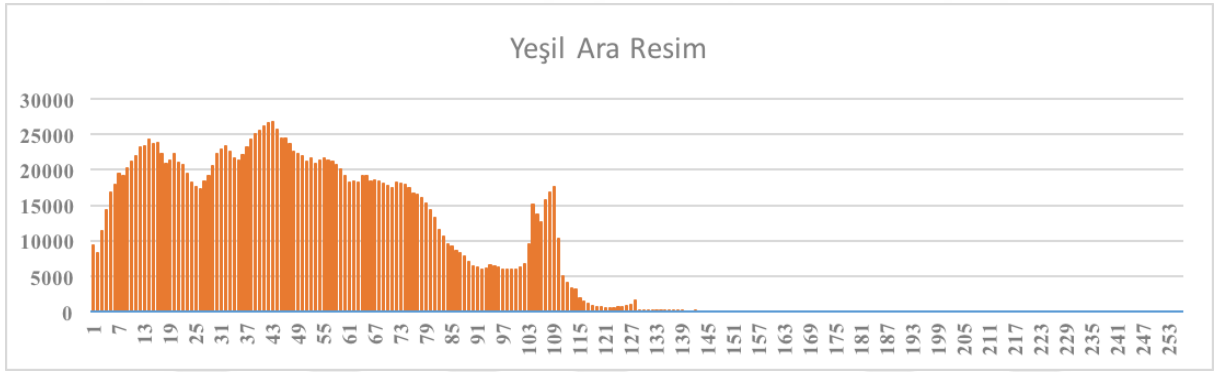
Şekil 24. Sonuç Resim Kırmızı Piksel Dağılımı



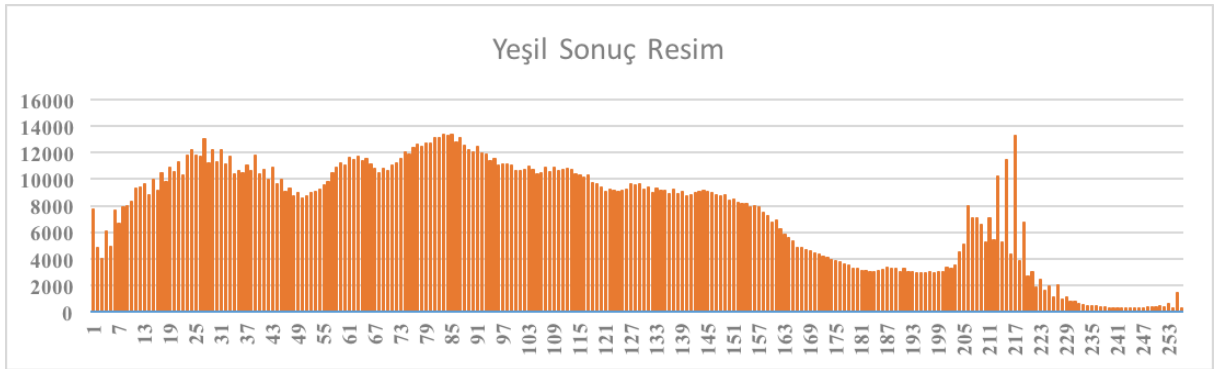
Şekil 25. Orijinal Resim Yeşil Piksel Dağılımı



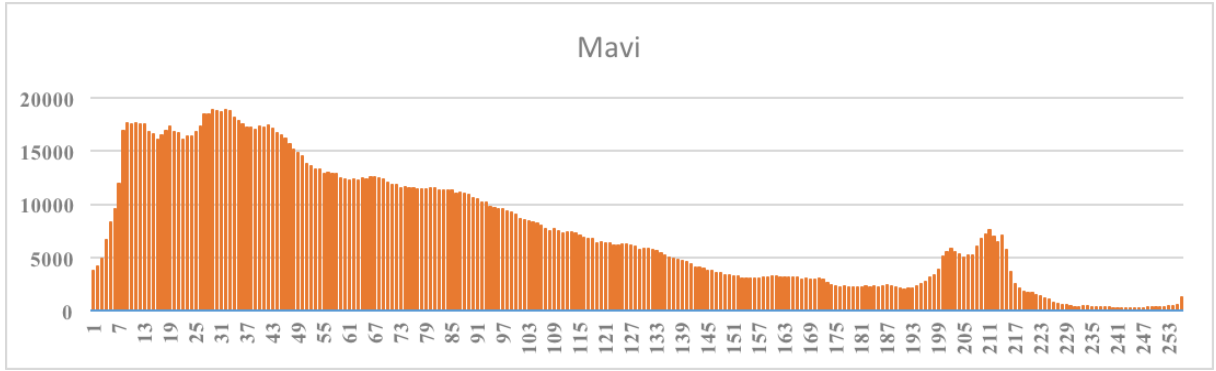
Şekil 26. Ara Resim Yeşil Piksel Dağılımı



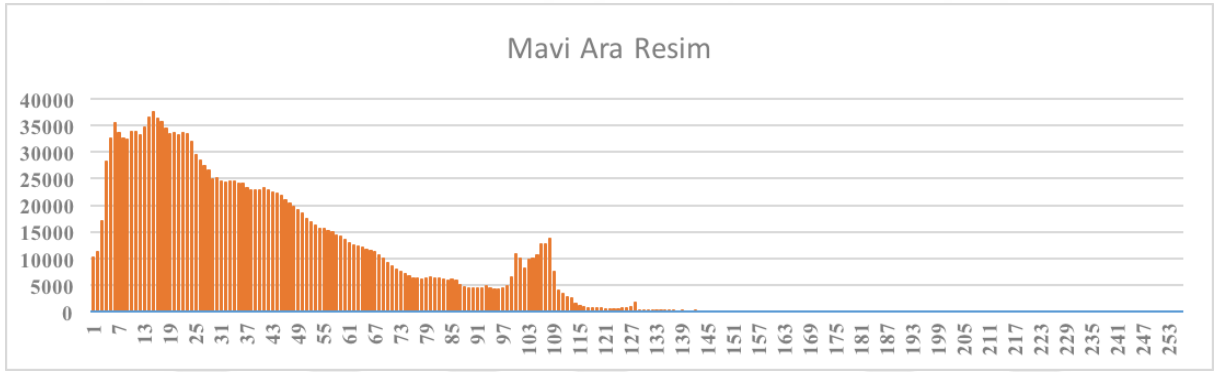
Şekil 27. Sonuç Resim Yeşil Piksel Dağılımı



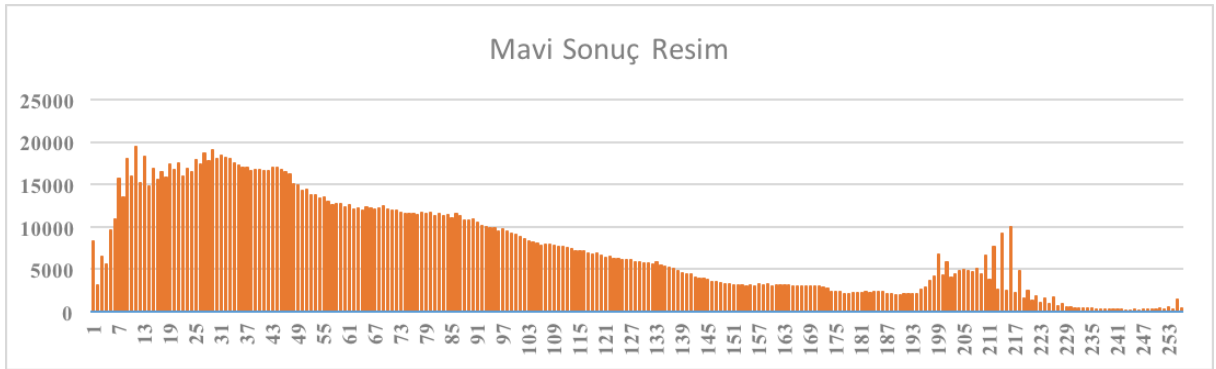
Şekil 28. Orijinal Resim Mavi Piksel Dağılımı



Şekil 29. Ara Resim Mavi Piksel Dağılımı



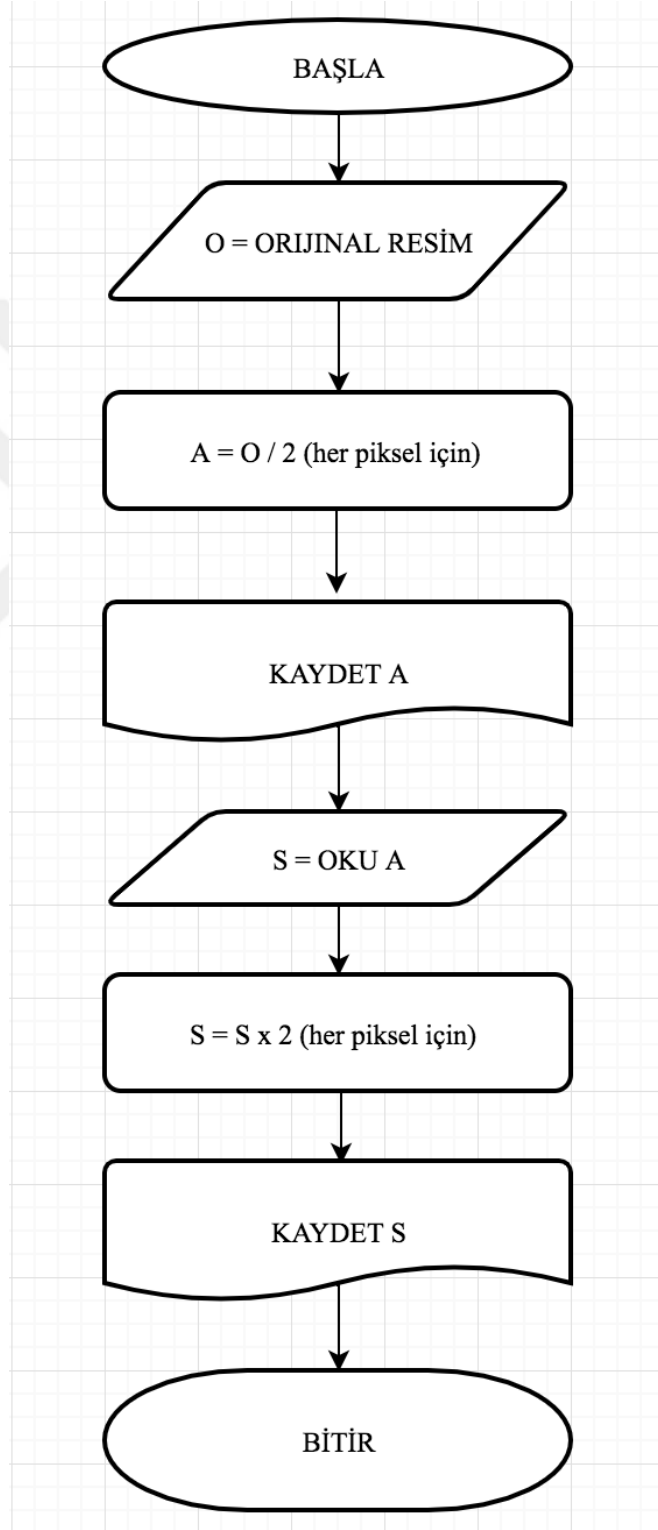
Şekil 30. Sonuç Resim Mavi Piksel Dağılımı



Histogram deęerlerinden de anlaşılacağı üzere orijinal resim ile sonuç resmi arasında gözle görülebilecek ayırım yoktur.

Yeni yaklaşım algoritması Şekil 31 ile verilmiştir.

Şekil 31. JPEG Dosya Küçültücü Algoritma



DÖRDÜNCÜ BÖLÜM

İNCELEME

1. İNCELEME

Yeni algoritma ile JPEG ile sıkıştırılmış dosyaların hafıza boyutlarını küçültmek mümkün olabileceği gibi, bazı durumlarda daha fazla sıkıştırma yapılamamaktadır.

1.1. DEĞİŞMEYEN DOSYA BOYUTLARI

Teorik ve fiziksel olarak herhangi bir uzayda herhangi bir nesneyi sonsuza kadar küçültmek demek onu yok etmek demektir. Dolayısıyla ne kadar matematiksel işlem yapılırsa yapılsın JPEG dosyalarının, hafızada kapladıkları boyutun da bir noktadan sonra sabit kalması veya daha da küçülüyor ise piksellerin yok olması beklenmektedir.

Sıkıştırma algoritmalarını, ilgili veriler üzerinde aynı algoritma ile N defa tekrar etmek, verilerin N defa sıkışacağı anlamını taşımamaktadır. Çünkü, sıkıştırma uygulandıktan sonra ortaya çıkan sonuç, elde edilmek istenen optimum noktaya ulaşılmış, mümkün olan en düşük bit ile gösterilmiş olmalıdır. Diğer bir deyişle, hali hazırda en düşük bit ile temsil edilen bir durum aynı algoritma ile daha düşük bit ile temsil edilemez.

Bu tez ile ortaya atılan sav, aslında JPEG algoritmasını geliştirmeye yöneliktir. Dijital fotoğraf üreten araçların üretmiş oldukları ham data üzerinde uygulanan JPEG algoritması, ham data üzerinden görüntüyü sıkıştırarak elde etmek üzere geliştirilmiştir. Bu tez ile, sıkıştırma işlemi sürecine yalnızca bir adım daha eklemektedir: JPEG algoritması ile oluşan ilk resmin de sıkıştırılabilir olması durumu! Diğer bir deyişle ham data sıkıştırılarak JPEG dosyası, JPEG dosyası sıkıştırılarak yeni bir dosya oluşturulmuştur. Bu dosyanın da sıkıştırılabilir olup olmayacağı gelecekteki çalışmalara bağlıdır.

Bu durum, JPEG sıkıştırma yöntemi son adımı olan Huffman kodlaması ile örneklendirilebilir.

Örneğin ABLA kelimesi Huffman kodlaması sonucu 010110 ile ifade edilebiliyorken, aynı Huffman algoritması ile 010110 ifadesi sıkıştırıldığında ortaya yine aynı durum çıkar. Yani Huffman ile sıkıştırılan 010110 = 010110'dir. Başka bir sonuç ortaya çıksaydı bu ABLA kelimesinin değil 010110'ın sıkıştırılmış hali olacaktı.

SONUÇ

JPEG resimlerinin boyutları, ham verilere göre oldukça küçük oldukları ve ham görüntü dosyalarında yer alan resimler ile karşılaştırıldığında insan gözü ile pek farklı olmadığından JPEG dosyalarının kullanım alanları oldukça yaygındır.

Görüntü dosyalarında JPEG kullanımının yaygın olmasının bir diğer nedeni zaman ve maliyettir. Sıkıştırılmamış dosyaların hafızada JPEG dosyalara göre daha fazla yer tutmaları nedeniyle, özellikle tüm işleri görüntü olan kullanıcılar normal internet kullanıcılarına göre daha yüksek boyutlu hard disklere yani daha bütçeli cihazlara sahip olmak durumundadırlar. Bu duruma bir diğer örnek, sürekli kayıt yapmak durumunda olan güvenlik şirketleri, devlet kuruluşları ve NASA gibi daha bir çok kuruluşun bünyelerinde barındırdıkları, görüntü frame'lerinden oluşan video ve multimedya dosyalarını saklamaları için çok ciddi maliyet içerecek serverlar tutmalarıdır. Hal böyleyken JPEG kullanmak bazı kuruluşlar için yüksek gider kalemini minimuma indirgeyebilecektir.

JPEG dosyaları, internet sayfalarının yüklenebilme hızını da etkilemektedir. Sitesinde çok fazla sayıda görüntü bulunduran bir e-ticaret sitesi ziyareti sırasında, kullanıcıların sayfaları açılış hızları daha fazla alışveriş yapabilme olasılıklarını arttıracak iken, sitesinde yüzlerce megabayt veri bulunduran sitenin de açılış hızı çok düşük olacağından bu durum kullanıcı rahatsızlığı ve siteden ayrılma nedenine yol açabilmektedir.

Bu tez ile birlikte JPEG dosyaları biraz daha sıkıştırılabilmekte ve desteklenmesi ile birlikte, zaman ve maliyet açısından uluslararası bir etki yaratabilir. Bunun bir örneği e-bebek web sitesidir. Şuanda hali hazırda çekilen yüksek çözünürlüklü resimler tez de bahsedilen algoritma ile minimize edilmeye çalışılmakta ve aktif olarak kullanılmaktadır.

KAYNAKÇA

Auckland Üniversitesi, “Gaussian Filtering”,

https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf, (erişim tarihi: 10/05/2019)

Cardiff Üniversitesi, “Huffman Coding”,

<https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node210.html>, (erişim tarihi: 10/05/2019)

Cornell Üniversitesi, “Computer Vision”,

https://www.cs.cornell.edu/courses/cs6670/2011sp/lectures/lec02_filter.pdf, (erişim tarihi: 10/05/2019)

Cornell Üniversitesi, “JPEG: Image compression algorithm”,

<http://pi.math.cornell.edu/~web6140/TopTenAlgorithms/JPEG.html>, (erişim tarihi: 10/05/2019)

Doğu Akdeniz Üniversitesi, “Digital Image Processing”,

<http://faraday.ee.emu.edu.tr/ee583/Lectures/EE%20583-Lecture01.pdf>, (erişim tarihi: 10/05/2019)

Greg Roelofs, “PNG Site Map”, <http://www.libpng.org/pub/png/#history>, (erişim tarihi:

10/05/2019)

Hong Kong Çin Üniversitesi, “Image Compression and the Discrete Cosine Transform”,

<https://www.math.cuhk.edu.hk/~lmlui/dct.pdf>, (erişim tarihi: 10/05/2019)

IBM, “ASCII table”,

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ioaq100/ascii_table_appendix.htm, (erişim tarihi: 10/05/2019)

Indiana Üniversitesi, “What is the TIFF graphics file format?”, <https://kb.iu.edu/d/afjn>,

(erişim tarihi: 10/05/2019)

IOSR Journal of Engineering, “The Origins of Digital Image Processing & Application areas in Digital ImageProcessing Medical Images”,

https://www.academia.edu/2324136/The_Origins_of_Digital_Image_Processing_and

[Application areas in Digital Image Processing Medical Images](#) (erişim tarihi: 10/05/2019)

İbrahim Çayıroğlu, “Görüntü İşleme Ders Notları”,

http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-1.Hafta.pdf, (erişim tarihi: 10/05/2019)

JPEG, “About JPEG”, <https://jpeg.org/about.html>, (erişim tarihi: 10/05/2019)

Michigan Üniversitesi, “Image File Formats”,

<https://guides.lib.umich.edu/c.php?g=282942&p=1885348>, (erişim tarihi: 10/05/2019)

Microsoft, “Color Conversion (RGB to YCbCr)”, [https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdprfx/b550d1b5-f7d9-4a0c-9141-](https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdprfx/b550d1b5-f7d9-4a0c-9141-b3dca9d7f525)

[b3dca9d7f525](https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdprfx/b550d1b5-f7d9-4a0c-9141-b3dca9d7f525), (erişim tarihi: 10/05/2019)

NASA, “Computers in Spaceflight: The NASA Experience”,

<https://history.nasa.gov/computers/Ch9-3.html>, (erişim tarihi: 10/05/2019)

Nihat Dipova, “Görüntü Analizi Yöntemlerinin Geoteknik Mühendisliğinde Kullanımı”, Akdeniz Üniversitesi, 2018

Rafael C. Gonzales, Richard E. Woods, “Digital Image Processing”,

https://www.academia.edu/37539199/Digital_Image_Processing_-_3rd_Edition_Pearson, (erişim tarihi: 10/05/2019)

Stanford University, “Introduction to Digital Images”,

<https://web.stanford.edu/class/cs101/image-1-introduction.html>, (erişim tarihi: 10/05/2019)

Stanford Üniversitesi, “JPEG standard”,

<https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>, (erişim tarihi: 10/05/2019)

World Wide Web Consortium, “GIF”, <https://www.w3.org/Graphics/GIF/spec-gif87.txt>,

(erişim tarihi: 10/05/2019)

ÖZGEÇMİŞ

3 Haziran 1985 tarihi, İstanbul ili Bakırköy ilçesi doğumluyum. İlkokul, ortaokul ve liseyi İstanbul'da tamamladıktan sonra bir müddet iş hayatına atıldım. Sonrasında, 2008 yılında Karadeniz Teknik Üniversitesi, Fen ve Edebiyat Fakültesi, İstatistik ve Bilgisayar Bilimleri Bölümü'nde lisans eğitimine kaydoldum. 2012 yılında lisans eğitimini tamandıktan sonra tekrar iş hayatına atıldım. Sırasıyla Akbank, HSBC Bankası, Anadolu Sigorta, e-bebek şirketlerinde uygulama geliştirici olarak çalıştım. 2016 yılında yüksek lisans kararı ile Beykent Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı'na kaydımı yaptırdım.

Detaysoft şirketinde uygulama geliştirme takım lideri olarak çalışma hayatımı sürdürmekteyim. Evli ve 2 çocuk babasıyım.

Süleyman ÜNLÜ