

**T.C.**

**TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ**

**ÇORLU MÜHENDİSLİK FAKÜLTESİ**

**ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ**

**GÖRSEL SIKIŞTIRMA**

**SUNUM RAPORU**

**Hazırlayan**

**1210606022**

**Çağla ŞENTÜRK**

**1190606066**

**Emin Deniz BİLEN**

**1200606063**

**Ferhat Mert ELDENER**

**1190606036**

**Hulusi ŞİMŞEK**

**1200606046**

**Yiğit KARAMAN**

**Danışman**

**Doç. Dr. Ahmet SAYGILI**

**04/2024**

**Çorlu- Tekirdağ**

**AKILLI FİYAT**

**Tekirdağ Namık Kemal Üniversitesi**

**Çorlu Mühendislik Fakültesi**

**Endüstri Mühendisliği Bölümü**

**GÖRSEL SIKIŞTIRMA SUNUM ÖDEVİ**

**1210606022**

**Çağla ŞENTÜRK**

**1190606066**

**Emin Deniz BİLEN**

**1200606063**

**Ferhat Mert ELDENER**

**1190606036**

**Hulusi ŞİMŞEK**

**1200606046**

**Yiğit KARAMAN**

**Danışman: Doç. Dr. Ahmet SAYGILI**

**04/2024**

**Çorlu- Tekirdağ**

# **İÇİNDEKİLER**

**Sayfa**

[**LİSANS BİTİRME PROJESİ BEYANNAMESİ** iii](#_Toc155704437)

[**TEŞEKKÜR** iv](#_Toc155704438)

[**İÇİNDEKİLER** v](#_Toc155704439)

[**ŞEKİL LİSTESİ** viii](#_Toc155704440)

[**TABLO LİSTESİ** ix](#_Toc155704441)

[**SEMBOL LİSTESİ** x](#_Toc155704442)

[**KISALTMA LİSTESİ** xi](#_Toc155704443)

[**ÖZET** xii](#_Toc155704444)

[**ABSTRACT** xiii](#_Toc155704445)

[**1.** **Giriş** 1](#_Toc155704446)

[**1.1.** **Problem Tanımı** 1](#_Toc155704447)

[**1.2.** **Kullanılan Teknolojiler** 1](#_Toc155704448)

[**1.2.1.** **Java** 1](#_Toc155704449)

[**1.2.2.** **Android Studio** 1](#_Toc155704450)

[**1.2.3.** **Retrofit** 1](#_Toc155704451)

[**1.2.4.** **Picasso** 2](#_Toc155704452)

[**1.2.5.** **Jsoup** 2](#_Toc155704453)

[**1.2.6.** **Shared Preferences** 2](#_Toc155704454)

[**1.2.7.** **Google Play Developer Console** 2](#_Toc155704455)

[**1.2.8.** **Github** 2](#_Toc155704456)

[**1.2.9.** **Python** 2](#_Toc155704457)

[**1.2.10.** **SQLite** 2](#_Toc155704458)

[**1.3.** **Grup Oluşturma ve Toplantı** 3](#_Toc155704459)

[**1.4.** **Proje Konularının Belirlenmesi:** 3](#_Toc155704460)

[**1.5.** **Proje Odaklanma Kararı:** 3](#_Toc155704461)

[**1.6.** **Projenin Temel Amacı ve Özellikleri:** 3](#_Toc155704462)

[**1.6.1.** **Proje Amacı:** 3](#_Toc155704463)

[**1.6.2.** **Özellikler:** 4](#_Toc155704464)

[**2.** **Gelişme** 4](#_Toc155704465)

[**2.1.** **Mobil Uygulama Platformu Araştırması:** 4](#_Toc155704466)

[**2.2.** **Kotlin ve Java Kararı:** 4](#_Toc155704467)

[**2.3.** **Java ve Android Studio Kurulumu** 4](#_Toc155704468)

[**2.4.** **Emulator Kurulumu ve Telefon Bağlama:** 5](#_Toc155704469)

[**2.5.** **Android Studio Araştırmaları ve Kullanımı:** 5](#_Toc155704470)

[**2.6.** **Java Bilgi Güncelleme ve Yeni Öğrenimler:** 5](#_Toc155704471)

[**2.7.** **"Sync Android SDKs" Sorunu ve Kod Yönetimi:** 5](#_Toc155704472)

[**2.8.** **Online Toplantı ve Uygulama Özellikleri ile Tasarımın Temelleri:** 6](#_Toc155704473)

[**2.9.** **Logo Tasarımı ve Çözünürlük Problemi:** 6](#_Toc155704474)

[**2.10.** **Android Studio'da Uygulama Temelleri:** 6](#_Toc155704475)

[**2.11.** **Webscraping Denemesi ve Engeller:** 6](#_Toc155704476)

[**2.12.** **Logo Tasarımı ve Çözünürlük Problemi:** 6](#_Toc155704477)

[**2.13.** **Android Studio'da Uygulama Temelleri:** 6](#_Toc155704478)

[**2.14.** **Web Scraping Denemesi ve Engeller:** 6](#_Toc155704479)

[**2.15.** **Menulerin Oluşturulması:** 6](#_Toc155704480)

[**2.16.** **Arama Özellikleri Eklemeleri:** 6](#_Toc155704481)

[**2.17.** **Diğer Marketlerin Eklenmesi:** 7](#_Toc155704482)

[**2.18.** **Arama Sayfası ve RecyclerView Yapısı:** 7](#_Toc155704483)

[**2.19.** **Sesli Arama Özelliği:** 7](#_Toc155704484)

[**2.20.** **İndirimli Ürünler Bölümü ve Kaynaklar:** 7](#_Toc155704485)

[**2.21.** **Performans Sorunu ve Çözüm Stratejisi:** 7](#_Toc155704486)

[**2.22.** **Kod ve Görsel Düzeltmeler:** 7](#_Toc155704487)

[**2.23.** **Sepetim Sayfası ve Özelliklerin Eklenmesi:** 7](#_Toc155704488)

[**2.24.** **Kataloglar Sayfası ve Web Scrapping:** 8](#_Toc155704489)

[**2.25.** **Akıllı Sıralama ve Fiyata Göre Sıralama Eklenmesi:** 8](#_Toc155704490)

[**2.26.** **Marketlere Göre Filtreleme Özelliği ve Checkbox Kullanımı:** 8](#_Toc155704491)

[**2.27.** **Sepeti Paylaş Özelliği:** 8](#_Toc155704492)

[**2.28.** **Tema Renklerinin ve Görsel Değişikliklerin Güncellenmesi:** 8](#_Toc155704493)

[**3.** **Sonuç** 8](#_Toc155704494)

[**3.1.** **Google Play'e Yayınlama Hazırlıkları:** 9](#_Toc155704495)

[**4.** **Hafta İlerlemeleri:** 9](#_Toc155704496)

[**4.1.** **Google Play Hesabının Oluşturulması ve Onay Süreci:** 9](#_Toc155704497)

[**4.2.** **Uygulamanın Kapalı Beta Olarak Yüklenmesi:** 9](#_Toc155704498)

[**5.** **BİRİNCİ DERECE BÖLÜM BAŞLIĞI** 12](#_Toc155704499)

[**5.1.** **İkinci Derece Altbölüm Başlığı** 13](#_Toc155704500)

[**5.1.1.** **Üçüncü derece altbölüm başlığı** 13](#_Toc155704501)

[**5.1.2.** **Üçüncü derece altbölüm başlığı** 13](#_Toc155704502)

[**5.2.** **İkinci Derece Altbölüm Başlığı** 13](#_Toc155704503)

[**5.3.** **İkinci Derece Altbölüm Başlığı** 13](#_Toc155704504)

[**6.** **BİRİNCİ DERECE BÖLÜM BAŞLIĞI** 14](#_Toc155704505)

[**7.** **BİRİNCİ DERECE BÖLÜM BAŞLIĞI** 15](#_Toc155704506)

[**8.** **BİRİNCİ DERECE BÖLÜM BAŞLIĞI** 16](#_Toc155704507)

[**9.** **KAYNAKLAR** 17](#_Toc155704508)

[**10.** **EKLER** 18](#_Toc155704509)

[**Ek-A** 18](#_Toc155704510)

[**ÖZGEÇMİŞ** 19](#_Toc155704511)

# **ŞEKİL LİSTESİ**

**Sayfa**

[**Şekil 1** Kurulum Hatası 16](#_Toc155713137)

[Şekil 2 Menu Taslağı 17](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713138)

[Şekil 3 Arama Sayfası Taslak Hali 18](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713139)

[Şekil 4 İndirimli Ürünler Taslak Hali 18](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713140)

[Şekil 5 Kataloglar Sayfası 19](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713141)

[Şekil 6 Uygulamanın Son Hali 20](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713142)

[Şekil 7Arama Sayfası Son hali 20](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713143)

[Şekil 8 İndirimli Ürünler Son Hali 20](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713144)

[Şekil 9 Kataloglar Son Hali 21](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713145)

[Şekil 10 Sepetim Son Hali 21](file:///C:\Users\Hulusi\Desktop\raporv2.docx#_Toc155713146)

# **KISALTMA LİSTESİ**

|  |  |
| --- | --- |
| **XML** | : Extensible Markup Language- Genişletilebilir İşaretleme Dili |
| **PNG** | : Portable Network Graphic- Taşınabilir Ağ Grafiği |
| **SVG** | : Scalable Vector Graphics- Ölçeklendirilebilir Vektör Grafiği |
| **SDK** | : Software Development Kit- Yazılım Geliştirme Kiti |
| **Sync** | : Eş zamanlama |
| **RAM** | : Random Access Memory- Rastgele Erişimli Hafıza |
| **Restful** | : Temsili Durum Transferi |
| **QR** | : Quick Response, Çabuk Tepki |
| **SQL** | : Structured Query Language, Yapılandırılmış Sorgu Dili |
| **HTML** | : HyperText Markup Language , Hiper Metin İşaretleme Dili |
| **JSON** | : JavaScript Object Notation, JavaScript Nesnesi Gösterimi |
| **IDE** | : Integrated Development Environment, Tümleşik geliştirme ortamı |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **ÖZET**

Lisans Bitirme Projesi

AKILLI FİYAT

**Çağla ŞENTÜRK**

**Emin Deniz BİLEN**

**Ferhat Mert ELDENER**

**Hulusi ŞİMŞEK**

**Yiğit KARAMAN**

Danışman: **Doç. Dr. Ahmet SAYGILI**

Bu tez, bir mobil uygulama geliştirme sürecini ve bu uygulamanın Google Play Store'da yayınlanma sürecini detaylı bir şekilde ele almaktadır. Araştırma ve geliştirme ekibi, bir grup halinde bir araya gelerek uygulama fikrini oluşturmuş ve belirli konseptler üzerinde kararlar almıştır. Proje, fikir oluşturma aşamasından başlayarak, kod yazma, test etme ve son olarak Google Play Store'a yükleme aşamasına kadar olan süreci kapsar.

Geliştirme ekibi, uygulamanın genel işlevselliğini, kullanıcı dostu arayüzünü ve pratik özelliklerini geliştirmek adına yoğun çalışmalar yürüttü. Özellikle, barkodlu arama yapma, en ucuz marketi bulma, akıllı arama sıralama özelliği ve fiyata göre sıralama gibi kullanıcı deneyimini artıran özellikler üzerinde duruldu. Ayrıca, sepeti paylaşma ve farklı marketlere göre filtreleme gibi özelliklerin uygulamaya entegre edilmesi, kullanıcıların pratiklik ve kişiselleştirme açısından önemliydi.

Uygulamanın Google Play Store'a yüklenmesi aşamasında, hesap oluşturma, gereken belgelerin hazırlanması ve özellikle kapalı beta olarak yükleme süreci detaylı bir şekilde incelendi. Bu aşamada, şifreleme teknikleriyle uygulamanın kapalı beta olarak yüklenmesi adımları atıldı.

Sonuç olarak, bu tez, bir mobil uygulama geliştirme sürecinin adımlarını, karşılaşılan zorlukları ve çözüm yollarını, ayrıca Google Play Store'da bir uygulamanın yayınlanma sürecini ayrıntılı bir şekilde ele alarak, bu sürecin başarıyla tamamlanmasını sağlayan adımları içermektedir.

**Anahtar Kelimeler:** Akıllı, Fiyat, ucuz

**2024, 10 Sayfa**

# **ABSTRACT**

Bachelor Graduation Project

AKILLI FİYAT

**Emin Deniz BİLEN**

**Ferhat Mert ELDENER**

**Hulusi ŞİMŞEK**

**Yiğit KARAMAN**

Supervisor: Dr. Halil Nusret BULUŞ

This thesis thoroughly discusses the development process of a mobile application and its deployment on the Google Play Store. The research and development team collectively brainstormed the application idea and made decisions on specific concepts. The project encompasses the entire process from idea generation to coding, testing, and finally uploading to the Google Play Store.

The development team worked extensively to enhance the overall functionality, user-friendly interface, and practical features of the application. Emphasis was placed on features that enhance user experience, such as the smart search ranking and sorting by price. Additionally, integrating features like sharing the cart and filtering by different markets were deemed crucial for user convenience and customization.

During the stage of uploading the application to the Google Play Store, the thesis delves into account creation, preparation of necessary documentation, and particularly examines the detailed process of uploading the application as a closed beta. Encryption techniques were utilized for the closed beta upload phase.

In conclusion, this thesis covers the steps of a mobile application development process, the challenges encountered, their solutions, and extensively details the process of publishing an application on the Google Play Store, outlining the steps that led to a successful deployment.

**Keywords:** Smart, Price, Cheap

**2024, 10 Pages**

# **Görüntü Sıkıştırma Nedir?**

## **Problem Tanımı**

## Görüntü sıkıştırma, günümüzde geniş bir uygulama alanına sahip olan önemli bir bilgi işlem konusudur. Görüntü sıkıştırma, dijital görüntü verilerini daha az miktarda depolama alanı veya iletim bant genişliği kullanarak temsil etmenin bir yoludur.

## **Neden Görüntü Sıkıştırma Gereklidir?**

Görüntü sıkıştırma, birçok alanda önemli avantajlar sağlar:

Depolama Alanı Tasarrufu: Görüntülerin sıkıştırılması, dosya boyutlarını önemli ölçüde azaltarak depolama alanından tasarruf sağlar.

İletim Hızı ve Bant Genişliği: Sıkıştırılmış görüntülerin iletilmesi, iletim hızını artırır ve internet üzerinden daha hızlı aktarım sağlar.

Görüntü sıkıştırma, birçok alanda önemli avantajlar sağlar:

Mobil ve Web Uygulamaları: Mobil cihazlar ve web uygulamaları gibi kaynak sınırlı platformlarda, küçük dosya boyutları önemlidir.

Görüntü Kalitesi: İyi tasarlanmış sıkıştırma yöntemleri, görüntü kalitesini korurken dosya boyutunu azaltabilir.

# **Kayıpsız Görüntü Sıkıştırma**

## **Huffman Kodlaması**

## 

## Huffman Kodlama çok sık kullanılan bir veri sıkıştırma yöntemidir.

## PC’ler üzerinde birçok popüler programın tabanını oluşturur.

## Shannon – Fano yöntemine benzemektedir.

## Genelde daha iyi sonuçlar vermektedir.

## En iyi kodu karakter olasılıklarının 2’nin negatif kuvveti olduğu durumlarda üretir.

## En belirgin farkları Shannon – Fano kodu yukarıdan aşağıya oluştururken, Huffman kodlamada ağacının aşağıdan yukarı oluşturmasıdır.

## 1952 yılında D. Huffman tarafından bulunmuştur**.**

Yöntem sembollerin olasılıklarına göre azalan sırada sıralanmasıyla başlamaktadır.

Aşağıdan yukarıya her yaprakta bir sembol olacak şekilde bir ağaç oluşturulur.

Her adımda en düşük olasılıklı iki sembol seçilir ve kısmi ağacın en tepesine eklenir, listeden silinir ve her iki sembolü de ifade eden bir sembolle yer değiştirir.

Liste tek bir sembol kalana kadar devam eder.

Sembollerin kodlarını elde etmek için ağaç bir uçtan diğerine dolaşılır.

Görüntü sıkıştırmasında Huffman kodlaması, bir görüntünün orijinal boyutunu azaltarak saklama veya iletim maliyetlerini düşürür. Bu kodlama yöntemi, görüntü piksellerinin frekans dağılımını temel alarak sık kullanılan pikseller için kısa kodlar ve daha az kullanılan pikseller için daha uzun kodlar oluşturarak verinin sıkıştırılmış halin i oluşturur



## **LZ77 ve LZ78 Algoritmaları**

LZ77 ve LZ78, Abraham Lempel ve Jacob Ziv'in 1977 ve 1978'de makalelerinde yayınlanan iki kayıpsız veri sıkıştırma algoritmasıdır.

Bunlar ayrıca sırasıyla LZ1 ve LZ2 olarak da bilinir. Bu iki algoritma, LZW, LZSS, LZMA ve diğerleri dahil olmak üzere birçok varyasyonun temelini oluşturur.

Akademik etkilerinin yanı sıra bu algoritmalar, PNG ve ZIP'te kullanılan GIF ve DEFLATE algoritması da dahil olmak üzere her yerde bulunan çeşitli sıkıştırma şemalarının temelini oluşturdu.

Her ikisi de teorik olarak sözlük kodlayıcılardır.

Görüntü sıkıştırma, büyük miktarda veriyi daha küçük boyutlarda saklamak ve iletmek için kullanılan bir dizi algoritma ve yöntem içerir.

Görüntü Sıkıştırmada LZ77'nin Kullanımı:

Görüntü verilerinde tekrar eden desenler oldukça yaygındır. Örneğin, arka plan renkleri, düz renk alanları, tekrar eden desenler veya benzer yapıdaki nesneler gibi.

LZ77, bu tekrar eden desenleri tanımlayarak ve sıkıştırarak görüntü verilerini önemli ölçüde azaltabilir.

Özellikle, BMP gibi kayıpsız sıkıştırmaya elverişli görüntü formatlarında LZ77 kullanımı oldukça etkilidir.

Örnek: "sir\_sid\_eastman\_easily\_t"

Adımlar;

Başlangıçta, sıkıştırma penceresi ve tampon boştur.

Her karakter sırasıyla okunur ve sıkıştırma penceresine eklenir.

Her karakter için, sıkıştırma penceresindeki mevcut en uzun eşleşme aranır ve bu eşleşme için uzaklık ve uzunluk belirlenir.

Eşleşme bulunamadığında, sıkıştırma penceresine bir sonraki karakter eklenir.

Özetle, "sir\_sid\_eastman\_easily\_t" cümlesi için LZ77 algoritması minimal sıkıştırma sağlar çünkü cümlede çok fazla tekrar eden veri yoktur. Bu nedenle, sıkıştırılmış veri neredeyse orijinal cümle ile aynı olacaktır.

* LZ77'nin Avantajları: Basit ve etkilidir.
* Kayıpsız veri sıkıştırma sağlar.
* Tekrar eden desenleri iyi tanımlar.
* LZ77’nin Dezavantajları
* Büyük sözlük boyutları gerektirebilir.
* Sıkıştırma oranı, verinin özelliklerine bağlı olarak değişebilir.

## **LZW (Lempel Ziv Welch)**

İlk olarak, Abraham Lempel ve Jacob Ziv tarafından geliştirilmiş, daha sonra Terry Welch tarafından iyileştirilmiştir.

Kayıpsız veri sıkıştırma için kullanılan bir algoritmadır.

En önemli özelliği çıktının ikinci parametresini de elemine etmesidir.

Piksel değerlerinin sıkıştırılması, görüntü dosyasının temel bileşenlerinden biridir. Bir pikselin değeri, genellikle kırmızı (R), yeşil (G), ve mavi (B) bileşenlerin kombinasyonuyla belirlenir. Örneğin, 8 bitlik renk derinliği olan bir görüntüde, her bir R, G ve B bileşeni 0 ile 255 arasında değer alabilir.

LZW algoritması gibi sıkıştırma algoritmaları, ardışık piksel değerlerini tanımlayarak ve bu değerleri daha kısa sembollerle temsil ederek piksel değerlerini sıkıştırır. Özellikle, aynı veya benzer piksel değerlerinin tekrarlanması durumunda, bu değerlerin kısaltılmış sembollerle temsil edilmesiyle sıkıştırma sağlanır.

Örnek: “sir\_sid\_eastman”

Adımlar;

Sözlüğün 0-255 girişlerini 8 bitlik karakterlere ata.

İlk sembol s (115) okunur ve sözlükte bulunur. Sıradaki karakter i okunur ancak si katarı sözlükte yer almaz. Kodlayıcı şunları yapar: 115 çıktı olarak üretilir, si katarını sözlüğün uygun bir yerine ekler (256 girişi) ve sözlük işaretçisini i sembolü yapar.

Sıradaki r karakteri okunur. “ir” sözlükte yer almaz. Kodlayıcı çıktı olarak 105 (i) üretir ve ir 257 indeksinde saklanır. Sözlük işaretçisi i sembolü yapılır.

Tüm metin işlendiğinde en sonda eof ile çıktı biter.

LZW'nin Avantajları

* Yüksek Sıkıştırma Oranı
* Kayıpsız Sıkıştırma
* Hızlı Sıkıştırma ve Genişleme İşlemi

LZW’nin Dezavantajları

Yüksek Bellek ve İşlemci Kullanımı

Sıkıştırma Tablosunun Boyutu

## **RLE (Run Length Encoding):**

Her tür veri için kullanılabilir bir algoritma olsa da, aynı sembolün ardışık olarak çok defa tekrar etmesi durumunda iyi bir sıkıştırma oranı sağladığı için, genellikle görüntü sıkıştırmada kullanılır.

RLE’nin temel mantığı, bir değerin ardışık olarak çok sayıda tekrar etmesi durumunda, o değeri bir defa kodlayıp, ardından tekrar sayısını kodlamaktır.

Örneğin 1-bit renk derinliğine sahip bir görüntünün (2 renge sahip: sadece siyah ve beyaz) bir satırındaki 640 pikselin renkleri sırayla; önce 200 adet beyaz, sonra 240 adet siyah ve sonra yine 200 adet beyaz şeklinde ise, o satır RLE ile (200, 240, 200) şeklinde kodlanabilir.

Tekrar sayıları çok fazla farklı sayıda olasılığa sahip olabilecekleri için, 64’ten büyük olan tekrar sayıları önce 64’e bölünür, sonra bölüm ve kalan farklı Huffman Tabloları kullanılarak kodlanır. Bu kodlama modified Huffman (MH) ya da Group 3 One-Dimensional (G31D) olarak bilinir.

RLE Avantajları

* Basitlik
* Yüksek Sıkıştırma Oranı
* Kayıpsız Sıkıştırma
* Hızlı İşlem

RLE Avantajları

* Veri Yapısına Bağımlılık
* Tekrarlama Sınırı
* Bellek Kullanımı
* Veri Duyarlılığı

## **Kayıpsız Görüntü Sıkıltırma: PNG**

PNG, kayıpsız sıkıştırma için tasarlanmış bir görüntü formatıdır. JPEG'e kıyasla, PNG sıkıştırması kalite kaybına neden olmaz. İşte PNG'nin ana özellikleri:

* Zorunlu Filtreler: PNG, görüntüyü sıkıştırmadan önce bazı filtrelerin uygulanmasını gerektirir. Bu filtreler, dosya boyutunu küçültmek için görüntüdeki tekrar eden desenleri tanımlar ve sıkıştırma için veri tekrarını azaltır.
* Deflate Sıkıştırması: PNG'nin temel sıkıştırma algoritması, LZ77 ve Huffman kodlamasını birleştiren Deflate algoritmasıdır. Bu algoritma, veri tekrarlarını tanımlayarak ve sıkıştırma için Huffman kodlaması kullanarak dosya boyutunu küçültür.
* İyi Alfa Kanal Desteği: PNG, alfa kanalı olarak adlandırılan şeffaflık bilgisini destekler. Bu, görüntülerin arka planlarından bağımsız olarak yerleştirilmesini sağlar.
* Karmaşık Renk Paleti Desteği: PNG, karmaşık renk paletlerini destekler, bu da yüksek kaliteli ve renkli görüntülerin sıkıştırılmasını sağlar.
* PNG, özellikle web grafiklerinin ve şeffaf arka planların saklanması için tercih edilir. Ancak, JPEG'e kıyasla daha büyük dosya boyutlarına sahip olabilir. Bu nedenle, kullanım senaryonuza ve kalite gereksinimlerinize bağlı olarak doğru formatı seçmek önemlidir.

# **Kayıplı Görüntü Sıkıştırma**

Kayıplı görüntü sıkıştırma, görüntü verilerini sıkıştırmak için veri kaybını kabul eden bir yaklaşımdır. Bu yöntem, sıkıştırılmış görüntüyü orijinaline tam olarak geri döndürmek yerine, bazı bilgi kayıplarını tolere eder. Bu, dosya boyutunu daha da küçültmek için kullanılırken, göreceli olarak küçük bir kalite kaybına neden olabilir. Kayıplı sıkıştırma genellikle fotoğraf ve görüntü tabanlı web grafiklerinde kullanılır.

## **Kayıplı Görüntü Sıkıltırma: JPEG**

JPEG, Joint Photographic Experts Group'ün kısaltmasıdır ve az ya da hiç fark edilmeyen bir görüntü kalitesi ve çözünürlüğü etkilemeksizin önemli ölçüde daha küçük dosya boyutları sağlayan kayıplı bir sıkıştırma algoritmasıdır. JPEG ile sıkıştırılmış bir resim, orijinalinden on kat daha küçük olabilir. JPEG, insan gözünün kolayca algılayamadığı bilgileri kaldırarak, insan gözünün algılamakta iyi olduğu bilgileri koruyarak çalışır.

Renk Alanı Dönüşümü

Renk uzayı, renklerin belirli bir düzenlenmesidir. RGB (Kırmızı, Yeşil ve Mavi) ve CMYK (Cyan, Magenta, Sarı ve Anahtar/Siyah) renk uzayı örnekleridir. Ayrıca, bir resimdeki her pikselin kendi renk uzayı değerleri vardır.

Örnekleme Azaltma

Renk bilgisini parlaklık bilgisinden ayırdıktan sonra, JPEG krominans kanallarını orijinal boyutlarının dörtte birine örnekleme azaltır. Her 4 piksel bloğu, 4 pikselin tümü için tek bir renk değerine ortalanır. Sonuç olarak, bazı bilgiler kaybolur ve resmin boyutu yarıya düşer, ancak insan gözü renge çok duyarlı olmadığından, değişiklikler kolayca ayırt edilemez.

8x8 Piksel Bloklarına Bölme

Örnekleme azaltmadan sonra, her kanalın piksel verisi 64 pikselin 8x8 bloklarına bölünür. Bundan sonra, algoritma her piksel bloğunu bağımsız olarak işler.

İleri DCT (Discrete Cosine Transform)

İlk olarak, her kanal için her piksel değeri 128 çıkarılarak değer aralığı -128 ile +127 arasına getirilir.

DCT kullanılarak, her kanal için 64 pikselin her bloğu, karşılık gelen ağırlık değerlerini çarpan bir sabit set taban görüntüleri ile yeniden oluşturulabilir ve ardından bunları toplar.

Entropi Kodlama

Her matrisin içinde bulunan değerleri listelemek, çok sayıda sıfırımız olduğundan RLE (Run Length Encoding) çalıştırmak ve sonra verileri depolamadan önce Huffman Kodlama algoritmasını çalıştırmaktır.

DCT(Discrete Cosine Transform)

**Temel İlke:** DCT, sinyalin zaman veya uzay bileşenlerini frekans bileşenlerine dönüştüren bir matematiksel işlemdir.

**Katsayıları Sıralama ve Kuantalama:** DCT, bir görüntü bloğunu frekans bileşenlerine dönüştürdüğünde, bu frekans katsayıları çıkarılır.

**Kesme ve Katsayı Kullanımı:** DCT'nin en güçlü yönlerinden biri, yüksek frekanslı katsayıları sıfıra kesme yeteneğidir.

**JPEG Sıkıştırma:** JPEG formatı, DCT'nin sıkıştırma işlemi için yaygın olarak kullanıldığı bir örnektir.

**Kalite ve Kayıp Oranı Dengesi:** DCT tabanlı sıkıştırma yöntemlerinde, kalite ve sıkıştırma oranı arasında bir denge sağlamak önemlidir.

**Ters Dönüşüm:** Sıkıştırılmış bir görüntüyü geri yüklemek için, sıkıştırma sırasında yapılan adımların tersini yapmak gerekir.

## **Google Play Hesabının Oluşturulması ve Onay Süreci:**

Google Play'e uygulamayı yüklemek için bir hesap açıldı ve hesabın onaylanması için çaba harcandı. Uygulamanın yüklenmesi için belirli şartlar gerekiyordu. Bu şartlardan biri uygulamanın kapalı beta olarak yüklenmesiydi.

Şekil 9 Kataloglar Son Hali

Şekil 10 Sepetim Son Hali

## **Uygulamanın Kapalı Beta Olarak Yüklenmesi:**

Uygulamanın Google Play'e yüklenmesi için belirtilen şartlar arasında kapalı beta olarak yükleme talebi vardı. Bu talebi karşılamak için uygulama şifreleme anahtarıyla şifrelendi ve ardından kapalı beta olarak Google Play'e yüklendi.

# **KAYNAKLAR**

<https://chat.openai.com/>

https://stackoverflow.com/

https://www.udemy.com/course/android-mobil-uygulama-gelistirme-egitimi-java/learn/lecture/28225870?start=375#overview

Johnson, M. (2020). "Mobile Application Development: Best Practices." Publisher.

Smith, A. (2018). "Introduction to Android Programming." Journal of Mobile Development, 5(2), 45-58.

Jackson, K., & Garcia, E. (2019). "Effective User Interface Design for Mobile Applications." Proceedings of Mobile Technologies Conference, 20-25.

Android Developers. (n.d.). "Android Studio Documentation." https://developer.android.com/studio/documentation

SQLite. (n.d.). "Official Documentation." https://www.sqlite.org/