

**T.C.
ISPARTA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ
MEKATRONİK ANABİLİM DALI**

**DERİN ÖĞRENME VE NESNELERİN İNTERNETİ
KULLANILARAK İNSAN YÜZ TİPİNE GÖRE MOBİL
DESTEKLİ GÖZLÜK ÖNERİ SİSTEMİNİN
GERÇEKLEŞTİRİLMESİ**

Hasan TEZCAN

**Danışman
Doç. Dr. Bekir AKSOY**

ISPARTA - 2024



© 2024 [Hasan TEZCAN]

ETİK BEYANI

Isparta Uygulamalı Bilimler Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kurallarına uygun olarak ve bilimsel ahlak ve geleneklere aykırı düşecek bir yol ve yardıma başvurmaksızın hazırladığım bu tez çalışmasında;

Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde ve ortaya çıkan sonuçlarda herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu, tezimle ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımı bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

27/06/2024

Hasan TEZCAN

.....

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	i
ÖZET.....	iii
ABSTRACT.....	iv
TEŞEKKÜR.....	v
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ	viii
SİMGELER VE KISALTMALAR DİZİNİ	ix
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	3
3. MATERYAL VE YÖNTEM	9
3.1. Materyal	9
3.1.1. Veri seti	9
3.1.2. Derin öğrenme.....	10
3.1.2.1. VGG derin öğrenme mimarisi	11
3.1.2.2. GoogLeNet derin öğrenme mimarisi	12
3.1.2.3. Residual Neural Network derin öğrenme mimarisi	13
3.1.2.4. Densely Connected Convolutional Network derin öğrenme mimarisi	15
3.1.2.5. MobileNet derin öğrenme mimarisi	16
3.1.3. Google colab platformu.....	18
3.1.4. Görüntü işleme	18
3.1.5. Android yapısı ve android işletim sistemi.....	18
3.1.6. Android studio editörü	19
3.1.6.1. Android activity	20
3.1.6.2. Android layout	21
3.1.6.3. Android servis.....	22
3.1.6.4. Intent yapısı.....	23
3.1.6.5. AndroidManifest.xml dosyası.....	23
3.1.6.6. Build.Gradle yapısı	24
3.1.6.7. Android application package dosyaları.....	25
3.1.7. Android cihaz	26
3.1.8. İnternet sunucu	26
3.1.9. Firebase veri tabanı	27
3.1.9.1. Yetkilendirme	28
3.1.9.2. Gerçek zamanlı veri tabanı	29
3.1.9.3. Bulut mesajlaşma servisi.....	30
3.1.10. JavaScript object notation veri yapısı.....	31
3.1.11. Flask framework yapısı	31
3.1.12. Nesnelerin interneti kavramı	32
3.1.13. Performans değerlendirme ölçütleri	33
3.2. Yöntem.....	34
3.2.1. Uygulamada kullanılan veri setleri	35
3.2.2. Çalışmada kullanılan derin öğrenme yöntemleri ve teknikleri	36
3.2.2.1. Veri ön işleme aşaması	37
3.2.2.2. Uygulama geliştirme ortamları	38
3.2.2.3. Derin Öğrenme modeli için kullanılan kütüphaneler.....	38
3.2.2.4. Derin öğrenme modeli için sınıflandırma klasörlerinin oluşturulması.....	38

3.2.2.5. Veri setinin rastgele ayrıştırılarak etiketlenmesi.....	39
3.2.2.6. Çalışmada kullanılan derin öğrenme mimarilerinin eğitilmesi.....	39
3.2.3. Mobil uygulama yazılımının geliştirilmesi	39
3.2.3.1. Mobil uygulama ana sayfa yapısı	40
3.2.3.2. Mobil uygulama ana sayfası arayüz tasarımı	41
3.2.3.3. Android cihazın donanımsal özelliklerini kullanabilmek için gerekli izinlerin alınması.....	42
3.2.3.4. Android cihazın kamera özelliğine ulaşılması ve kullanılması	43
3.2.3.5. Android cihazın galerisine ulaşılması ve kullanılması	45
3.2.3.6. Android cihaz ile elde edilen görüntünün düzenlenmesi	45
3.2.3.7. Android cihaz ile elde edilen görüntünün iyileştirilmesi ve görüntü ön işlemlerine tabi tutulması	48
3.2.3.8. Android ortamında iyileştirilen görüntünün FireBase veri tabanına kaydedilmesi	49
3.2.3.9. Android ortamında iyileştirilen görüntünün internet sunucuya gönderilmesi.....	50
3.2.3.10. Derin öğrenme modeli tarafından belirlenen yüz tipinin android ekranda gösterilmesi	51
3.2.4. İnternet sunucu yazılımının geliştirilmesi	52
4. BULGULAR	54
4.1. Derin Öğrenme Mimarilerinden Elde Edilen Eğitim Sonuçları.....	54
4.2. Derin Öğrenme Mimarilerin Karşılaştırılması.....	69
5. TARTIŞMA VE SONUÇ.....	75
KAYNAKLAR	78
ÖZGEÇMİŞ	84

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME VE NESNELERİN İNTERNETİ KULLANILARAK İNSAN YÜZ TİPİNE GÖRE MOBİL DESTEKLİ GÖZLÜK ÖNERİ SİSTEMİNİN GERÇEKLEŞTİRİLMESİ

Hasan TEZCAN

Isparta Uygulamalı Bilimler Üniversitesi
Lisansüstü Eğitim Enstitüsü
Mekatronik Anabilim Dalı

Danışman: Doç. Dr. Bekir AKSOY

Mobil cihazların işlevsel olarak hızlı büyümesi ve artan hesaplama gücü, mobil platformlarda derin öğrenme algoritmalarının uygulanması için önemli fırsatlar oluşturmuştur. Derin öğrenme, yapay zekâ alanında önemli bir alt disiplin alanı olup son yıllarda büyük ilerlemeler kaydedilmiştir. Bu ilerlemeler ile artan veri yığınlarını derin öğrenme yöntemleri kullanılarak, karmaşık hesaplamalar için, yüksek düzeyde soyutlama ve örüntü tanıma yetenekleri ile çözümler sunmuştur. Bu yetenekler, birçok uygulama alanında büyük avantajlar sağlaması sebebiyle, mobil cihazlar ile çevrimiçi çalışacak olan farklı sistemlerin koordinasyonu oldukça önemli bir hale gelmiştir. Son yıllarda kullanıcıların hem mobil cihazların kullanımındaki artış, hem de çevrimiçi sistemlerin birbirleri ile iletişimi, derin öğrenme yöntemlerinin mobil cihazlarda kullanımı için önemli uygulama alanlarından birisi olmuştur.

Tez çalışmasında, geliştirilmiş olan mobil yazılım ile android cihazın fotoğraf çekme özelliğinden faydalanılarak elde edilen fotoğraf veya android cihazın galerisinden seçilen bir görüntü, görüntü işleme teknikleri ile iyileştirilmiştir. İyileştirilen görüntü internet üzerinden çevrim içi çalışacak farklı bir cihaz ile derin öğrenme yöntemleriyle analiz edilerek, kişinin belirlenen yüz sınıf tipleri içerisinde hangi yüz tipine sahip olduğu tespit edilmiştir. Ayrıca belirlenen yüz tipine en uygun gözlük tiplerinin listesi tez kapsamında geliştirilmiş olan android yazılım ile kişiye gösterilmesi sağlanmıştır. Böylece, geliştirilen özgün yazılım ile kişiye gözlük tavsiyesinde bulunmasının yanında, kullanıcılara hem mekân açısından hem de zaman bakımından tasarruf sağlamıştır.

Anahtar Kelimeler: Derin öğrenme, Mobil uygulama, Gözlük, Yüz tipi, Nesnelere interneti

2024, 84 sayfa

ABSTRACT

Master's Thesis

IMPLEMENTATION OF A MOBILE SUPPORTED GLASSES RECOMMENDATION SYSTEM ACCORDING TO HUMAN FACE TYPE USING DEEP LEARNING AND INTERNET OF THINGS

Hasan TEZCAN

**Isparta University of Applied Sciences
The Institute of Graduate Education
Department of Mechatronics**

Supervisor: Assoc. Prof. Dr. Bekir AKSOY

The rapid functional growth of mobile devices and increasing computational power have created significant opportunities for the application of deep learning algorithms on mobile platforms. Deep learning is an important sub-discipline in the field of artificial intelligence and great progress has been made in recent years. With these advances, deep learning methods have been used to increase data volumes, providing solutions for complex calculations with high levels of abstraction and pattern recognition capabilities. Since these capabilities provide great advantages in many application areas, the coordination of different systems that will work online with mobile devices has become very important. In recent years, the increase in the use of mobile devices and the communication of online systems with each other have become one of the important application areas for the use of deep learning methods on mobile devices.

In the thesis study, a photo obtained by using the photo-taking feature of the Android device with the developed mobile software or an image selected from the gallery of the Android device was improved with image processing techniques. The improved image was analyzed using deep learning methods with a different device that works online over the internet, and it was determined which face type the person had within the determined face class types. In addition, the list of glasses types most suitable for the determined face type was displayed to the person with the android software developed within the scope of the thesis. Thus, in addition to recommending glasses to the person with the unique software developed, it saves users both in terms of space and time.

Key Words: Deep learning, Mobile application, Glasses, Face type, Internet of things

2024, 84 pages

TEŐEKKÜR

Tezimin yürütülmesinde desteęini ve emeęini hiçbir zaman esirgemeyen tez danışmanım sayın Doç. Dr. Bekir AKSOY'a alıőma süresince bana desteklerinden dolayı teőekkürlerimi sunarım.

Tez alıőması kapsamında gerçekleştirilen uygulamalar için 25.10.2023 tarih 166/05 nolu kararı ile etik kurul izni veren Isparta Uygulamalı Bilimler Üniversitesi Bilimsel Araőtırma ve Yayın Etięi Kuruluna teőekkür ederim.

Tüm bu süreç boyunca her konuda yardımcı ve destek olan kendi işlerini bırakıp benim için çabalayan eőim Ayőegül TEZCAN'a ve varlığı ile beni hayata bağlayan oęlum Eftal TEZCAN'a sonsuz teőekkürlerimi sunarım. Tez alıőması süresince bana manevi desteklerini hiç esirgemeyen babam Halil TEZCAN'a ve annem Havva TEZCAN'a teőekkür ederim. Uzaktan da olsa her şekilde bana yardım eden abilerim Zafer TEZCAN ve Hüseyin TEZCAN'a da teőekkürü borç bilirim. Gerek tez alıőması gerekse hayatımın her adımında ve her konuda yardımına koőan çok deęerli dostum, öęretmen arkadaşım Hamdi SAYIN'a teőekkür ederim. Hayatta olmasalar da hep benimle olan tüm aile üyelerime de manevi desteklerini hep hissettięim için teőekkür ederim. Ayrıca Isparta Borsa İstanbul Mesleki ve Teknik Anadolu Lisesi Biliőim Teknolojileri Alanı öęretmen arkadaşlarıma süreç boyunca bana verdikleri destek için de teőekkürlerimi sunmak isterim.

Hasan TEZCAN
ISPARTA, 2024

ŞEKİLLER DİZİNİ

Şekil 3.1. VGG16 derin öğrenme mimarisinin yapısı.....	12
Şekil 3.2. GoogLeNet (Inception) modülü.....	13
Şekil 3.3. ResNet derin öğrenme mimarisine ait artık blok yapısı.....	15
Şekil 3.4. DenseNet derin öğrenme mimarisi	16
Şekil 3.5. MobileNet derin öğrenme mimarisi.....	17
Şekil 3.6. Android işletim sistemi mimarisi.....	19
Şekil 3.7. Activity yaşam döngüsü.....	21
Şekil 3.8. Android Service sınıfı yaşam döngüsü	22
Şekil 3.9. Standart AndroidManifest.xml dosyasının genel görünümü	24
Şekil 3.10. Android Studio build.gradle dosyasının görünümü	25
Şekil 3.11. Raspberry Pi 3 geliştirme kartı görünümü.	27
Şekil 3.12. Firebase hizmetlerini gösteren yapı	28
Şekil 3.13. Firebase yetkilendirme görünümü	29
Şekil 3.14. Firebase gerçek zamanlı veri tabanında verilerin depolanma şekli	30
Şekil 3.15. Firebase bulut mesajlaşma şeması	31
Şekil 3.16. Tasarlanan sistemin iş-akış diyagramı	35
Şekil 3.17. Veri setinde kullanılan görüntülere ait örnek görsel.....	36
Şekil 3.18. Mobil uygulamanın ana sayfa görünümü	40
Şekil 3.19. Mobil uygulama ana sayfasının ara yüz tasarımı XML görünümü	42
Şekil 3.20. Android cihazın donanımsal özelliklerinin kullanılabilmesini sağlayan izinler için gerekli olan komut satırlarının görüntüsü	43
Şekil 3.21. Mobil uygulamanın fotoğraf çekme veya seçme sayfası görünümü	43
Şekil 3.22. Mobil uygulamanın fotoğraf çekme sayfası görünümü	44
Şekil 3.23. Mobil uygulamanın görüntü çekme işlemi sonrası onaylama veya iptal etme sayfasının görünümü.....	44
Şekil 3.24. Mobil Uygulamanın görüntü alma, odaklanma ve kırpma sayfası görünümü.....	46
Şekil 3.25. Mobil uygulamanın görüntü düzenleme sayfasında iken kişinin yüzüne odaklanıldığı anın görünümü	47
Şekil 3.26. Mobil uygulamanın ana sayfasına seçilen görüntünün yüklendiği anın görünümü.....	48
Şekil 3.27. Mobil uygulamada seçilen görüntünün, görüntü ön işlemlerine tabii tutulduğu anın görünümü	49
Şekil 3.28. Mobil uygulamada seçilen görüntünün FireBase veri tabanına kaydedildiği anın görünümü.....	50
Şekil 3.29. Mobil uygulamada, FireBase veri tabanına kaydedilen görüntünün raspberryPi internet sunucusuna gönderildiği anın görünümü	51
Şekil 3.30. Mobil uygulamada, belirlenen yüz tipi ve gözlük listesinin kullanıcıya gösterildiği anın görünümü.....	52
Şekil 4.1. Veri artırma öncesi VGG16 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri	54
Şekil 4.2. Veri artırma sonrası VGG16 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri	55
Şekil 4.3. VGG16 derin öğrenme mimarisine ait karmaşıklık matrisi.....	56
Şekil 4.4. Veri artırma öncesi ResNet50 derin öğrenme mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri	57

Şekil 4.5. Veri artırma sonrası ResNet50 derin öğrenme mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri	58
Şekil 4.6. ResNet50 derin öğrenme mimarisine ait karmaşıklık matrisi.....	59
Şekil 4.7. Veri artırma öncesi GoogLeNet derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri	60
Şekil 4.8. Veri artırma sonrası GoogLeNet derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri.....	61
Şekil 4.9. GoogLeNet derin öğrenme mimarisine ait karmaşıklık matrisi	62
Şekil 4.10. Veri artırma öncesi DenseNet201 derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri	63
Şekil 4.11. Veri artırma sonrası DenseNet201 derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri	64
Şekil 4.12. DenseNet201 derin öğrenme mimarisine ait karmaşıklık matrisleri	65
Şekil 4.13. Veri artırma öncesi MobileNet_V3 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri.....	66
Şekil 4.14. Veri artırma sonrası MobileNet_V3 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri	67
Şekil 4.15. MobileNet_V3 derin öğrenme mimarisine ait karmaşıklık matrisi	68
Şekil 4.16. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait doğruluk değerlerinin karşılaştırılması	69
Şekil 4.17. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait kesinlik değerlerinin karşılaştırılması	70
Şekil 4.18. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait duyarlılık değerlerinin karşılaştırılması	71
Şekil 4.19. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait F1 skor değerlerinin karşılaştırılması.....	72
Şekil 4.20. Tez çalışmasında kullanılan derin öğrenme mimarilerinin karşılaştırılması.....	72
Şekil 4.21. Tez çalışmasında kullanılan derin öğrenme mimarilerinin sınıflara göre F1 skorları.....	73

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1. Karmaşıklık matrisi.....	33
Çizelge 3.2. Veri Setinde bulunan sınıfların eğitim ve test veri setlerindeki dağılımları.....	37
Çizelge 4.1. VGG16 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları	56
Çizelge 4.2. ResNet50 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları	59
Çizelge 4.3. GoogLeNet derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları	62
Çizelge 4.4. DenseNet201 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları	65
Çizelge 4.5. MobileNet_V3 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor sonuçları	68

SİMGELER VE KISALTMALAR DİZİNİ

API	Uygulama Programlama Arayüzü (Application Programming Interface)
AUs	Eylem Birimleri (Action Units)
CE	Duyguların Bileşik Yüz İfadeleri (Compound Facial Expressions of Emotion)
CelebA	CelebFace Nitelikler Veri Kümesi (CelebFace Attributes Dataset)
CK	Cohn- Kanade Veri Kümesi (Cohn- Kanade Dataset)
CK+	Genişletilmiş Cohn-Kanade Veri Kümesi (Extended Cohn-Kanade Dataset)
CNN	Evrişimli Sinir Ağları (Convolutional Neural Networks)
CSV	Virgülle ayrılmış değerler (Comma Separated Values)
DN	Doğru Negatif (True Negative)
DNN	Derin Sinir Ağları (Deep Neural Networks)
DP	Doğru Pozitif (True Positive)
DWT	Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform)
FACS	Yüz Hareketi Kodlama Sistemi (Facial Actions Coding System)
FS-CNN	Yüze Duyarlı Evrişimli Sinir Ağı (Face-Sensitive Convolutional Neural Network)
GPIO	Genel Amaçlı Giriş/Çıkış (General-Purpose Input/Output)
GPU	Grafik İşleme Ünitesi (Graphics Processing Unit)
HMM	Saklı Markov Modeli (Hidden Markov Model)
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol
ID	Kimlik Tanılama (Identification)
IOT	Nesnelerin İnterneti (Internet Of Things)
IP	İnternet Protokolü (Internet Protocol)
JAFFE	Japon Kadın Yüz İfadeleri Veri Kümesi (Japanese Female Facial Expression Dataset)
JSON	JavaScript Object Notation
KNN	K-En Yakın Komşu Algoritması (K-Nearest Neighbors Algorithm)
LBP	Yerel İkili Örüntü (Local Binary Pattern)
LR	Lojistik Regresyon (Logistic Regression)
MVC	Model Görünüm Denetleyicisi (Model View Controller)
NoSQL	No Structured Query Language
OHA	Open Handset Alliance
PCA	Temel Bileşen Analizi (Principal Component Analysis)
RGB	Kırmızı-Yeşil-Mavi (Red-Green-Blue)
SSD	Tek Atış Çoklu Kutu Algılama (Single-Shot Multibox Detection)
SQL	Structured Query Language
SVM	Destek Vektör Makineleri (Support Vector Machine)
URI	Uniform Resource Identifier
WSGI	Web Sunucu Ağ Geçidi Arayüzü (Web Server Gateway Interface)
XML	Extensible Markup Language
YN	Yanlış Negatif (False Negative)
YP	Yanlış Pozitif (False Positive)

1. GİRİŞ

Son yıllarda, bilişim teknolojileri alanındaki gelişmeler göz önüne alındığında, özellikle elektronik ve bilgisayar alanında çalışan bilim insanları, insan gibi düşünen ve hareket edebilen yapay zekâ sistemlerine odaklanmıştır. Yapay zekâ, insan düşüncesi ve eylemlerine benzetmek için 1950'li yıllardan günümüze kadar sürekli olarak gelişmekte olup, nöroloji, psikoloji, eğitim, siber güvenlik ve mühendislik gibi farklı disiplinler arası çalışmayı kapsayan bir çalışma alanıdır (Altuntaş, 1998).

Derin öğrenme ise, karmaşık ve büyük veri yapılarını analiz etmek, örüntüleri tanımak ve öngörülerde bulunmak için kullanılan bir makine öğrenme yöntemi olup yapay sinir ağlarını temel alan bir yapıdır. Bununla birlikte mobil teknolojilerin hızla gelişmesi, mobil cihazlar için yapılacak uygulamalara derin öğrenme yöntemleri ile disiplinlerarası çalışmada önemli bir rol oynamaktadır (Güler, 2022).

Günümüzde makine öğrenmesi ve derin öğrenmedeki teknolojilerin gelişmesi, optik alanında da yapay zekânın kullanımını önemli ölçüde arttırmıştır (Ovatman, 2005). Gözlük almak isteyen bireyler, gözlük ile ilgili bilgilere genelde, internet üzerinden ya da gözlük satışı yapan optik firmaları üzerinden erişim sağlamaktadırlar. Bu durum, gözlük almak isteyen bireyler için, gözlük seçerken önemli ölçüde zaman kaybına neden olmaktadır. Bu probleme çözüm oluşturabilecek derin öğrenme yöntemlerini kullanan çevrimiçi çalışan bir sunucu cihaz ile, internet üzerinden bilgi alış-verişi yapabilen mobil cihazların (Cep Telefonu, Tablet vb.) koordineli bir şekilde kullanılması zaman ve ürün çeşitliliği anlamında optik firmalarına önemli kazançlar sağlayacağı önemli bir gerçektir.

Tez çalışması kapsamında, çevrimiçi sunucu olarak çalışacak cihaz için; Raspberry Pi kart; çevrimiçi görüntünün kaydedilerek istenildiği zaman kullanabilmesi için gerçek zamanlı çalışan veritabanı olarak; FireBase, çevrimiçi veri alış verişi ve Hyper Text Transfer Protokol (HTTP) işlemleri için; Flask framework, mobil cihaz için ise; android işletim sistemi kullanan ve görüntü çekme özelliği olan bir cep telefonu tercih edilmiştir. Cihazların hem birbirleri ile koordineli bir biçimde çalışabilmesi hem bilgi alışverişini yapabilmesi hem de çevrimiçi çalışabilen Python programlama diliyle yazılmış program ile, kullanıcıların zaman ve mekandan

bağımsız bir biçimde kullanabilmesi için Android Studio editörü yardımıyla Java programlama dili ile yazılmış çevrimiçi çalışan bir mobil uygulama geliştirilmiştir. Böylece geliştirilen yazılım ile bireyler sahip oldukları yüz tipleri belirlenmeye çalışılmıştır. Ayrıca belirlenen yüz tipine uygun gözlük seçiminin de bir listesi, geliştirilen mobil uygulama ile gerçek zamanlı olarak kullanıcılara gösterilmesi sağlanmıştır.

Tez çalışmasının ilk aşamasında ise geliştirilmiş olan android mobil uygulama ile android cihaz üzerinden kişinin görüntüsü elde edilerek, farklı görüntü ön işlemleri yöntemleri uygulanıp, çevrimiçi çalışan ve Python programlama dili ile derin öğrenme metotları kullanan cihaz üzerindeki yapay zeka modeline gönderilerek en yüksek doğruluk değerine sahip olan derin öğrenme yönteminden elde edilen yüz tipi android yazılımın sonuç ekranında gösterilmesi sağlanmıştır. Ayrıca, geliştirilen android uygulama ile android işletim sistemi üzerinden işlenen görüntü, eş zamanlı olarak çevrimiçi çalışan FireBase veritabanına kaydedilmiştir. Veritabanına kaydedilen görüntü, Python programlama dili ile geliştirilmiş olan Flask framework yardımıyla HTTP protokolleri üzerinden sunucuya gönderilmesi sağlanmıştır. Sunucuya ulaşan görüntü, sunucu üzerindeki yazılım yardımıyla yine sunucuda oluşturulan çalışma kapsamında kullanılan derin öğrenme modellerine girdi olarak gönderilmiştir. Çalışma kapsamında kullanılan derin öğrenme modelleri üzerinde gerekli eğitimler yapılarak: “Yuvarlak Yüz Tipi”, “Oval Yüz Tipi”, “Köşeli Yüz Tipi”, “Kare Yüz Tipi” veya “Kalp Yüz Tipi” sınıflarına ait sonuçlar kullanıcıya metinsel olarak gönderilmiştir. Elde edilen sonuç değeri sunucu tarafından cevap (İng. Response) mesajına dönüştürülmüştür. Bu mesaj, Flask framework yardımıyla HTTP protokolü üzerinden geliştirilen mobil uygulamaya gönderilerek mobil uygulama ekranında, geliştirilmiş olan android yazılım ile kullanıcıya sonuç olarak gösterilmesi sağlanmıştır. Aynı zamanda, çalışma kapsamında kullanılan derin öğrenme modellerinden elde edilen sonuca göre belirlenen yüz tipine uygun gözlüklerin bir listesi, geliştirilmiş olan android yazılım ile mobil uygulama ekranında gösterilmesi de sağlanmıştır. Çalışma kapsamında geliştirilen yazılımın mobil ortamda olması, hem kullanışlı hem de hızlı olması sağlanmıştır. Böylece gözlük almak isteyen bireylerin, hem zaman hem de mekândan bağımsız olarak tez çalışması kapsamında geliştirilmiş olan android mobil uygulama üzerinden yüz tipine uygun gözlüğün tespit edilmesi sağlanmıştır.

2. KAYNAK ÖZETLERİ

Gözlük kullanacak olan kişiler bilimsel olarak kendilerinin geometrik olarak hangi yüz tiplerine sahip olduklarını ve yüz tiplerine en uygun gözlüğün hangisi olduğuna karar vermekte zorluk çektikleri, günümüzün önemli sorunlarından bir tanesidir. Son yıllarda derin öğrenme algoritmalarının birçok disiplinler arası uygulamalarda kullanılmaya başlaması ile yüz tanıma sistemlerinde de derin öğrenme algoritmaları sıklıkla tercih edilmektedir.

Tez çalışmasında, yapay zekâ algoritmaları ile yüz tanıma sistemleri, bilimsel açıdan kişinin geometrik olarak hangi yüz tipine sahip olduğu ve insan yüz tipine en uygun gözlüğün hangisi olduğuna karar veren yapılar ve bu yapıları kullanan sistemler ile ilgili akademik literatürde yer alan çalışmalar incelenerek aşağıda detaylı bir biçimde ele alınmıştır.

Lamba ve Virmani (2021), çalışmalarında transfer öğrenme yöntemi kullanarak yüz tanıma ve yüz ifadesi belirleme üzerine ResNet50 modeli Cohn-Kanade (CK) veri setine uyarlamışlardır. Modeli özelleştirerek, verileri ön işleme tabi tutup görüntüleri hizalamışlar ve farklı aktivasyon fonksiyonları kullanarak eğitmişlerdir. Modelin eğitimi tamamlandıktan sonra test veri seti ile performans değerlendirmişler ve RELU aktivasyon fonksiyonunun en iyi sonuçları verdiğini ifade etmişlerdir.

Said ve Barr (2021), çalışmalarında yüksek çözünürlüklü görüntüler üzerinde ayrıntılı çalışmalar yapmışlar ve bellek ile hesaplama gereksinimlerini azaltmak için görüntüleri küçük parçalara ayırarak piramit tekniği kullanmışlardır. Bu yöntemi kullanarak, on beş evrişim katmanına sahip Yüze Duyarlı Evrişimli Sinir Ağı (İng. Face-Sensitive Convolutional Neural Network - FS-CNN) modelini geliştirmişlerdir. FS-CNN, Tek Atış Çoklu Kutu Algılama (İng. Single-Shot Multibox Detection -SSD) yöntemiyle yüz tespiti yapıp yüz işaretlerini kullanarak yedi temel duyguyu tanımlamışlardır. Oluşturdukları model, CelebFace Attributes (CelebA) veri setleriyle eğitilmiş ve %92.6 yüz tespiti, %94.6 duygu tanımlama doğruluğu elde etmişlerdir.

Arıř ve Turan (2020), alıřmalarında setikleri filmlere ait videolarda duygu analizi yaparak insan yz Őekillerini belirlemeye alıřmıřlardır. HaarCascade algoritmasını kullanarak grntlerdeki yzleri tespit ederek kesmiřlerdir. Kesilen yzler, filmlere gre gruplanıp kaydedilmiř ve Amazon Yz Tanımlama sistemi kullanılarak etiketlemiřlerdir. Yanlıř etiketlenen grntler ıkarıldıktan sonra yz Őekillerini %80 eęitim ve %20 test olarak ikiye ayırarak, yeni bir evriřimli sinir aęı modeli ile eęitmiřlerdir. Eęitimin sonunda modelde %60 doęruluk oranı elde etmiřlerdir.

Rao vd. (2020), yetiřkin ve ocuk yzlerini karřılařtırmak amacıyla Extended Cohn-Kanade Dataset (CK+) veri setlerini kullanarak yz sınıflandırması yapmıřlar ve performans deęerlendirme metrikleri ile analiz etmiřlerdir. Yz tespiti iin 68 anahtar noktayı kullanarak Derin Sinir Aęları (DNN) modeli geliřtirmiřler ve noktaların koordinatlarını eęitim iin .csv dosyasına kaydetmiřlerdir. alıřmalarının sonucunda CK+ veri setinde %87 doęruluk oranı ile yz Őeklini doęru tespit etmiřlerdir.

Wu ve Chen (2019), alıřmalarında kamera sistemlerinden almıř oldukları 1250 yz ifadelerini VGG16, VGG19, ResNet50, DenseNet121, DenseNet169, DNN, MobileNet ve MobileNetV2 derin ęrenme yntemleri kullanarak analiz etmiřlerdir. Toplam sekiz farklı derin ęrenme modellerinin eęitimi sonucunda en bařarılı model olarak %69.5 doęruluk oranı ile DNN modelinin olduęunu ifade etmiřlerdir.

Bayrakdar vd. (2016), alıřmalarında insan yz Őeklini ve yz ifadelerinin analizini yz tespiti, yz Őekli ve yz ifadesinin tanınması olmak zere  ana ařamada incelemiřlerdir. İlk ařamada, video dizilerinin ilk karesinden bařlayarak yz algılayıp son karesine kadar takip ederek yz tespitini gerekleřtirmiřlerdir. Yz tespit edildikten sonra, yz yapısındaki kas hareketlerine bakarak yz ifadesine ait zellikleri belirlemiřlerdir. Bu deęiřimleri lerek yz Őekillerini kategorik olarak tanımlamıřlar ve zelliklerini belirlemiřlerdir. Son ařamada, belirlenen yz ifadesi zelliklerine dayanarak yz ifadelerini tanımlayıp sınıflara ayırmıřlardır.

Chen ve Cheng (2015) alıřmalarında, ten rengi modelini kullanarak belirledikleri yzlerde, kenar algılama algoritmasıyla gzleri ve dudakları iřaretleyip kenar Őekil zelliklerini ıkarmıřlardır. Laplace, Robert, Canny ve Sobel yntemlerini kullanarak

bu özellikleri karşılaştırmışlardır. Geliştirdikleri sistemi, Japon Kadın Yüz İfadeleri (İng. Japanese Female Facial Expressions-JAFFE) veri tabanındaki yüz görüntüleri ile eğitmişler ve yeni görüntüler ile test etmişlerdir. Test sonuçlarında, üzüntü, mutluluk, normal ve şaşırma ifadelerini tanımada en yüksek başarıyı Canny yöntemi ile elde etmişlerdir. Yüz şeklinin tespiti ve yüz bulma doğruluğunun yüksek olması için iyi bir kenar algılama yönteminin seçiminin önemli olduğunu vurgulamışlardır.

Farahani vd. (2013) çalışmalarında, farklı yaş gruplarının göz ve ağız özelliklerinden yüz şeklini tespit etmek amacıyla, bulanık mantık tabanlı bir yöntem geliştirmişlerdir. Belirlenen yüzlerdeki göz ve ağız yapılarını farklı renk alanlarının birleşimi ile algılayıp, bulanık mantık ile analiz etmişlerdir. Gözün açık olması, ağzın kapalı olması, göz açıklık/genişlik oranı ve ağız genişliği gibi parametreleri kullanarak, Mamdani yöntemi ile yüz özelliklerini ve bunların hangi yüz şekline karşılık geldiğini kodlamışlardır. Bu yöntemle yuvarlak, kare ve kalp yüz şekline sahip kişileri %68 doğruluk oranında sınıflandırma işlemini gerçekleştirmişler; ancak oval ve köşeli yüz şekline sahip kişileri sınıflandırmada daha düşük doğruluk oranı elde etmişlerdir.

Kazmi vd. (2012), çalışmalarında yüz şekli ve yüz ifadelerinin otomatik tanınması için Ayrık Dalgacık Dönüşümü (İng. Discrete Wavelet Transform-DWT) özelliklerini kullanan bir yöntem geliştirmişlerdir. Yüz görüntülerinden ilgili bölgeleri tespit ederek özellikler çıkarmışlardır. Her bir görüntü için özellik seti hesaplanmış ve yedi ayrı Destek Vektör Makinesi (İng. Support Vector Machine - SVM), belirli yüz şekillerini ve ifadelerini tanımak üzere eğitmişler ve destek vektör makinelere çıkışları birleştirilerek çoklu sınıflandırma yapmışlardır. Çalışmanın sınıflandırma performansı JAFFE veri tabanı ile test ederek %74.5 doğruluk oranı elde etmişlerdir.

Oliveira vd. (2011), çalışmalarında insan yüz görüntülerinden özellik çıkarımı için İki Boyutlu Temel Bileşen Analizi (İng. Two-Dimensional Principal Component Analysis-2DPCA) yöntemini kullanmışlardır. Geleneksel Temel Bileşen Analizi'nden (İng. Principal Component Analysis-PCA) farklı olarak, 2DPCA yöntemi tek boyutlu vektörler yerine iki boyutlu matrisler kullanarak daha yüksek ayırt edici özellikler elde etmişlerdir. Tanıma oranını artırmak amacıyla, çok amaçlı

genetik algoritma tabanlı bir özellik seçme algoritması kullanmışlar ve K-en yakın komşular (İng. K-Nearest Neighbors-kNN) ile destek vektör makinesi sınıflandırıcılarını test etmişlerdir. Bu yöntemle ifadelerin sınıflandırılmasında %94 doğruluk oranı elde etmişlerdir.

Feng vd. (2005), çalışmalarında, yüz ifadelerini tanımlama sürecini yenilikçi bir yöntemle gerçekleştirmişlerdir. İlk olarak, yüz görüntülerinde Yerel İkili Örüntü (İng. Local Binary Pattern - LBP) operatörünü kullanarak özellikleri çıkarmışlar ve daha sonra yüz şekillerini sınıflandırmak için Lineer Programlama (İng. Linear Programming - LP) tekniğini adapte etmişlerdir. Bu şekilde ikili ağaç turnuva şeması kullanarak sınıflandırma işlemlerini gerçekleştirmişlerdir. Geliştirmiş oldukları yöntem ile yuvarlak, köşeli, kalp, kare ve oval olmak üzere beş farklı yüz şeklini sınıflandırmışlardır. Yöntemlerini JAFFE veri tabanında teste tabi tutmuşlar ve ortalama %93.8 doğruluk oranı elde etmişlerdir.

Silva ve Hui (2003), çalışmalarında yüz şeklini belirleme, yüz özelliklerini çıkarma ve duygu tanıma sisteminin gerçek zamanlı uygulamasını incelemişlerdir. Önerdikleri metodolojide, yüz özelliklerini belirlemek için kenar sayma ve görüntü korelasyon optik akış tekniklerini kullanmışlar ve insan yüz tipini belirlemek için sinir ağı tasarlamışlardır. Çalışmanın odak noktası, yüz ifadelerinin gerçek zamanlı tanınmasını sağlayarak insan yüz şeklini belirlemek olduğunu ifade etmişlerdir. Bu sebeple, geleneksel çevrimdışı hesaplama yöntemleri, gerçek zamanlı uygulamalara uygun hale getirmişlerdir.

Bush (2003) tarafından yapılan çalışmada, insanların beş farklı yüz tipine sahip olduğu vurgulanmıştır. Bu yüz tiplerinden kalp yüz tipinin geniş alınla başlayıp dar bir çeneye doğru sivrilen yapıya sahip olduğunu, oval yüz tipinin ise diğerlerine göre daha simetrik bir yapıya sahip olduğunu, kare yüz tipinin, kare şeklinde geniş alından başlayıp geniş ve köşeli bir çeneye kadar uzandığını ifade etmiştir. Köşeli yüz tipinin ise geniş alından daralan ve çene bölgesinde dar bir yapıya sahip olduğunu ve son olarak da yuvarlak yüz tipinin ise yumuşak hatlara ve geniş yanaklara sahip olduğunu belirtmiştir.

Zhang (1999), çalışmasında farklı katmanlara sahip bir algılayıcı mimari içinde, yüz şeklini ve ifadelerini tanıma sistemini geliştirmiştir. Yüz görüntülerinden elde edilen özelliklerin geometrik konumlarını çok ölçekli ve yönlü olan Gabor dalgacık katsayılarını kullanarak yüz şeklini ve ifadelerini tanımaya odaklanmıştır. Farklı özellik türlerini karşılaştırdığında, Gabor dalgacık katsayılarının geometrik konumlara göre daha güçlü olduğunu belirtmiştir.

Tsapatsoulis vd. (1998), çalışmalarında video görüntülerindeki hareket alanını istenilen yönlü filtrelemek ve optik akışın doğru yönde çıkarılmasını sağlamak amacıyla bir özellik vektör dizisi kullanmışlardır. Vektör dizisi zaman ölçeği değişmezliğini koruyan bir Saklı Markov Modeli (İng. Hidden Markov Model-HMM) ile yüz şeklini belirleme ve yüz ifadesi tanıma yöntemi geliştirmişlerdir. Ayrıca, bilinmeyen özellik vektör dizilerini sınıflandırmak için HMM'nin öğrenme yöntemini kullanmışlardır. Altı farklı yüz şeklini içeren birçok videoyu teste tabi tutmuşlar ve %82 doğruluk oranı elde etmişlerdir.

Yacoob ve Davis (1996), çalışmalarında, yüz görüntüsünün ana bölgelerini takip ederek yüzün eğim noktalarındaki yoğunluğunu hesaplayıp kişilerin yüz şeklini ve yüz ifadesini belirlemeye çalışmışlardır. Çalışmada, belirlenen yüzlerin özelliklerini içeren kare bölgeler için bir izleyici geliştirmiş ve her bir kare bölgeyi ayrı ayrı analiz ederek yüz şekillerini ve yüz ifadelerini tanımlamışlardır.

Ekman vd. (1986), çalışmalarında yüz ifadelerini algılayıp sınıflandırabilen bir analiz sistemi geliştirmişlerdir. Çalışmada, 1000'den fazla görüntü içeren bir veri seti oluşturmuşlar ve insan yüz şekilleri ile yüz ifadeleri sınıflandırmak için; gri-seviye görüntülerin mekânsal analizi, görüntü özelliklerinin ölçülmesi ve hareket içeriği ile kalıp şablon eşleştirme olmak üzere üç farklı yöntem uygulamışlardır. Gri-seviye görüntülerin mekânsal analizi %89, görüntü özelliklerinin ölçülmesi ile %57 ve hareket içeriği ile kalıp şablon eşleştirme ile %85 doğruluk oranları elde etmişlerdir. Üç yöntemin birleştirilmesiyle doğruluk oranını %92'ye kadar arttırmışlardır.

Ekman ve Friesen (1971), çalışmalarında her bir temel duygunun kendine özgü yüz ifadeleriyle tanımlanabileceğini belirtmişlerdir. Yüz ifadelerinin analizinde kullanılmak üzere yüz anatomisine dayanan eylem birimlerini (İng. Action Units-

AUs) kullanarak Yüz Eylemleri Kodlama Sistemi'ni (İng. Facial Actions Coding System - FACS) geliřtirmişlerdir. Bu sistemde, yüzün her bir bağımsız kas hareketine karşılık gelen kırk altı farklı eylem birimi tanımlamışlardır. Bu farklı birimlerini kategorilere ayırarak insan yüz şekillerini belirlemeye çalışmışlardır.



3. MATERYAL VE YÖNTEM

3.1. Materyal

Tez çalışması kapsamında tasarlanan sistem, yerel android işletim sistemine sahip android cihaz ile çevrimiçi çalışan internet sunucu ve bu cihazlar ile anlık veri alışverişi yapan uygulamalar ile geliştirilmiştir. Mobil uygulama geliştirme ortamı olarak; android studio, internet sunucu olarak; Raspberry Pi, çevrimiçi ve gerçek zamanlı çalışan veri tabanı için; Firebase, derin öğrenme mimarileri ile oluşturulan yapay zekâ modeli için Google Colab platformu ve farklı ortamlarda bulunan bu cihazların veri alışverişi için flask framework teknolojileri kullanılmıştır. Farklı yazılımlar kullanılarak geliştirilen uygulamalar gerçek cihazlar üzerinde test edilmiş olup çalışmayla ilgili literatür de geçen genel kavramlar ve detayları, gerekli materyaller ve kullanımları alt başlıklar halinde bu bölümde ele alınmıştır.

3.1.1. Veri seti

Veri seti, belirli bir amacı gerçekleştirmek için toplanmış veya oluşturulmuş veri koleksiyonu olarak tanımlanmaktadır. Bu veriler çoğunlukla belirli bir konu, sorun veya araştırma problemi ile ilgili olabilmektedir. Kullanıcılar tarafından bilgi edinmek, analiz yapmak, öğrenme algoritmalarını eğitmek veya farklı çalışmalarda kullanmak amacıyla toplanabilir, derlenebilir veya oluşturulabilirler (Yavuz, 2009).

Veri seti, sayılar, metinler, ses dosyaları, görüntüler, video görüntüleri gibi farklı türlerde olabilmektedir. Ayrıca veri seti genellikle belirli bir yapı veya formata sahiptir. Bundan dolayı veri analizi için makine öğrenmesi, derin öğrenme gibi birçok yöntemle kullanılmaktadır (Yeniad, 2006).

Veri seti, makine öğrenimi çalışmalarında kritik ve önemli bir rol oynamaktadır. Buradaki temel zorluk, projede kullanılacak olan en uygun makine öğrenme yöntemini seçmektir. Çünkü makine öğrenmesi modelinin performansı, modelin kendisi kadar eğitilen veri setine de bağlıdır. Veri seti yeterince farklı ve yeterli miktarda veri içermiyorsa, uygulamanın gerçek verilerle olan başarısı önemli ölçüde azalmaktadır. Ayrıca, veri setinde birbirine çok benzer verilerin bulunmaması da

oldukça önemlidir. Bu durum modeli ezberlemeye yönlendirir. Modelin ezber yapması modelin başarı oranı ne kadar yüksek olursa olsun gerçek verilerde yanlış sonuçlar üretme olasılığı oldukça yüksektir (Çetiner, 2020).

Veri setinizdeki verilerin sayısı yetersizse, verilerin sayısının artırılması gerekebilir. Resimlerden oluşan bir veri seti kullanılıyorsa, bazı kod parçacıkları eklenerek veri miktarını artırılması gerekir. Bu kodlar yardımıyla resimleri küçültme, büyütme ve döndürme gibi işlemlerle veri setini genişletilebilir. Ayrıca, veri setini eğitim ve test verileri olarak ayırmak oldukça önemlidir. Eğitim verisi her zaman test verisinden daha büyük olmalıdır, örneğin %80 eğitim ve %20 test veya %70 eğitim ve %30 test olarak ayrılabilir. Eğitim verilerindeki verilerin, test verisinde bulunmaması da gereklidir, böylece model test verisinde daha önce görmediği verilerle sınanabilir ve yüksek doğruluk oranlarına ulaşılabilir (Farboudi, 2009).

Çalışmada, internet sunucu içerisine Python programlama dili kullanılarak derin öğrenme algoritması yardımıyla oluşturulan modelin eğitimi ve doğruluk testi için, Kaggle açık erişimli internet sitesinde, farklı kişilere ait 5009 tane yüz görüntüsü olan veri seti kullanılmıştır (Anonim, 2024a).

3.1.2. Derin öğrenme

Derin öğrenme, çok katmanlı ve karmaşık yapay sinir ağları kullanarak veri temsillerini otomatik olarak öğrenen ve veriler üzerinde karmaşık modeller inşa eden bir yapay zeka yöntemidir. Bu yöntem, insan beyninin işleyişini taklit eden yapay sinir ağlarından meydana gelmektedir (Kurtboğan, 2023).

Derin öğrenme, görüntü tanıma, ses tanıma, oyun stratejileri ve öneri sistemleri başta olmak üzere pek çok araştırma ve uygulama alanında yaygın olarak kullanılmaktadır (Bal, 2021). Derin öğrenme teknikleri, genellikle büyük veri setleri üzerinde çalışarak verilerden karmaşık desenleri otomatik olarak öğrenmede sıklıkla kullanılmaktadır (Selamet, 2023).

Çalışmada oluşturulan görüntü veri setini eğitmede VGG, ResNet, GoogleNet, DenseNet ve MobileNet derin öğrenme mimarileri kullanılmıştır. Aşağıda tez çalışmasında kullanılan derin öğrenme mimarilerine ait detaylı bilgiler verilmiştir.

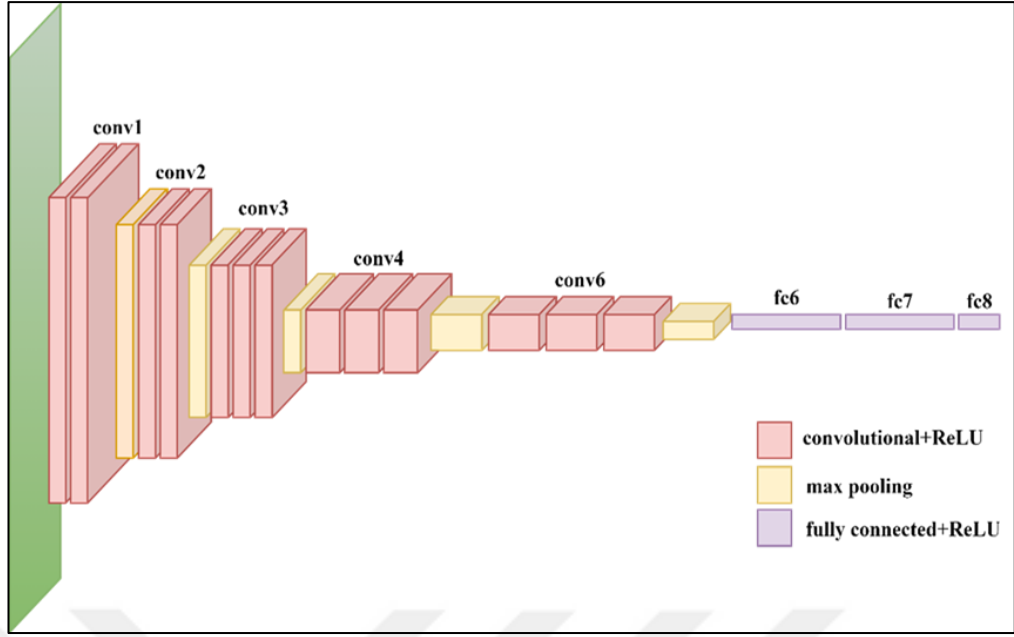
3.1.2.1. VGG derin öğrenme mimarisi

Visual Geometry Group (VGG) derin öğrenme mimarisi, bilgisayarlı görüntü üzerine çalışan Karen Simonyan ve Andrew Zisserman tarafından tasarlanan ve 2014'te tanıtılan bir evrişimli sinir ağı mimarisidir. VGG mimarisi, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) gibi geniş veri kümelerinde görüntü sınıflandırma görevlerinde başarı sağlamıştır.

VGG modeli, ardışık olarak evrişimli ve tam bağlantılı katmanlardan oluşmaktadır. VGG mimarisinin önemli özelliklerinden birisi de evrişimli sinir ağı yapılarına kıyasla daha derin bir yapıya sahip olmasıdır. Örneğin, VGG-16 modeli 16 katmandan, VGG-19 modeli ise 19 katmandan oluşmaktadır.

VGG derin öğrenme mimarisinin her bir katmanında, genellikle 3×3 (bazen de 1×1) boyutunda evrişim filtreleri kullanılarak takiben ReLU aktivasyon fonksiyonu uygulanmaktadır. Aktivasyon fonksiyonunu sonrasında ise genellikle 2×2 boyutunda maksimum havuzlama işlemi gerçekleştirilir. VGG yapısı modelin önceki katmanlarda öğrendiği özelliklerin daha karmaşık ve soyut özelliklere dönüştürülmesini sağlamaktadır.

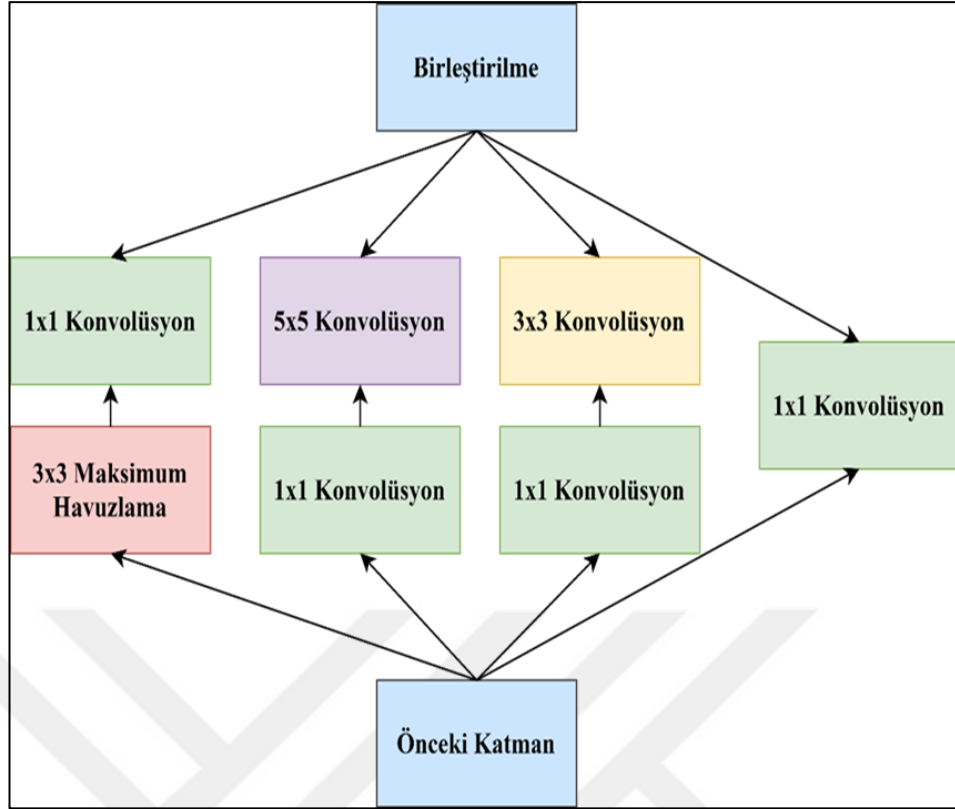
En son katmanlar ise genellikle tam bağlantılıdır ve çoğunlukla sınıflandırma amaçlı kullanılmaktadır. Bu katmanlarda softmax aktivasyon fonksiyonu kullanılarak sınıf olasılıkları hesaplanır. VGG16 derin öğrenme mimarisinin genel yapısı Şekil 3.1'de gösterilmiştir (Simonyan ve Zisserman, 2014).



Şekil 3.1. VGG16 derin öğrenme mimarisinin yapısı (Simonyan ve Zisserman, 2014)

3.1.2.2. GoogLeNet derin öğrenme mimarisi

GoogLeNet, 2015 yılında Szegedy ve ekibi tarafından, temel yapı taşları "Inception" denilen modüllerin birleştirilmesi ile oluşturulan bir derin öğrenme mimarisidir. GoogLeNet'in en önemli özelliği, "Inception" modüllerini kullanarak çoklu evrişim boyutlarını ve havuzlama boyutlarını aynı katmanda birleştirip kullanmasıdır. Bu yaklaşım, oluşturulan ağın daha derin olmasını sağlarken hesaplama maliyetini de azaltmaktadır. Inception modülleri 1×1 , 3×3 ve 5×5 boyutlarında evrişimlerin yanı sıra maksimum havuzlama işlemlerini içermesi ve işlemleri bir arada kullanması ile ağın farklı ölçeklerdeki özellikleri daha etkili bir şekilde öğrenmesini sağlamaktadır. Şekil 3.2 'de Inception modülü yapısı gösterilmiştir (Szegedy vd., 2015).



Şekil 3.2. GoogLeNet (Inception) modülü (Szegedy vd., 2015)

GoogLeNet, boyutsal azaltma için paralel 1x1 evrişim katmanlarını içeren yenilikçi teknikler de içermektedir. Bu yaklaşım, ağır derinliğini ve genişliğini artırırken hesaplama maliyetini düşürmektedir. Bu özellik GoogLeNet'i, diğer mimarilere göre çok büyük bir avantaj sağlamaktadır (Ballester ve Araujo, 2016).

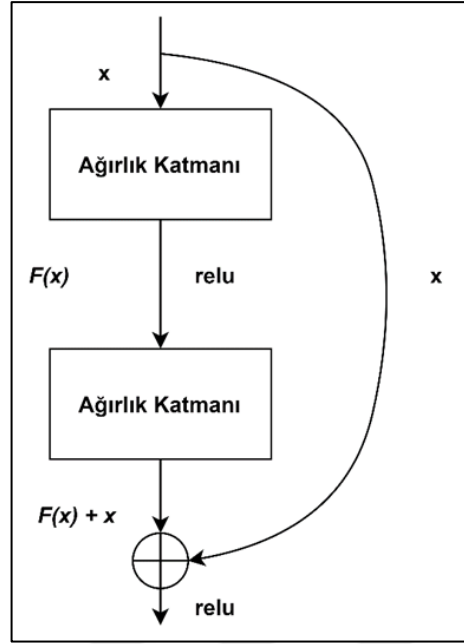
Diğer derin öğrenme mimarileriyle karşılaştırıldığında, GoogLeNet daha az parametre ve düşük hesaplama maliyetiyle daha etkili sonuçlar elde etmeyi hedeflemektedir. Bu nedenle, özellikle kaynağın sınırlı olduğu mobil cihazlar veya gömülü sistemlerde GoogLeNet derin öğrenme mimarisinin kullanımı daha uygundur (Ballester ve Araujo, 2016).

3.1.2.3. Residual Neural Network derin öğrenme mimarisi

Residual Neural Network (ResNet), Microsoft Research tarafından geliştirilen bir derin öğrenme mimarisi olup 2015 yılında tanıtılmıştır. ResNet, derin sinir ağlarının eğitilmesi sırasında ortaya çıkan "kaybolan gradyan" sorununu çözmek için özgün bir yaklaşım sunmaktadır. ResNet mimarisinin temel prensibi, her katmanın

çıktılarına "kalan bağlantılar" ekleyip tekrar çalıştırılarak ağıın öğrenme sürecini kolaylaşmaktadır. Böylece katmanlar, girişe göre değil, girişin artıklarına odaklanarak dönüşümü öğrenmeye çalışmaktadır. Bu yöntem, daha derin ağların eğitimini kolaylaştırarak artıkların geriye doğru yayılarak daha etkili bir biçimde çalışmasını sağlamaktadır. ResNet derin öğrenme mimarisi, ardışık evrişim (convolutional) katmanları ve artık bağlantılarla kurulu blok adı verilen yapılardan oluşmaktadır. Farklı derinliklere sahip olan ResNet mimarisinde farklı sayıda ve tipte bloklar oluşabilmektedir (Wu vd., 2019).

ResNet derin öğrenme mimarisinin bloklar arası derinliğinin artması, mimarinin öğrenme performansı yükseltmektedir. Bunun sebebi, oluşturulan ağa daha fazla katman eklenerek daha karmaşık özelliklerin öğrenilmesinin sağlanmasıdır. Ayrıca, bloklar arası kullanılan bağlantılar, geriye doğru gradyanların daha etkili bir şekilde yayılmasını sağladığı için oluşan ağıın daha verimli bir şekilde eğitilmesine olanak sağlamaktadır. Fakat model derinleştikçe gradyan sorunlarını meydana getirdiği için ağıın etkili çalışmasına engel olmaktadır (Liao ve Poggio, 2016). Engeli kaldırmak için farklı çözümler sunulmuştur. Özellikle, He ve arkadaşları tarafından 2016 yılında artık blok (İng. residual block) yapısı önerilmiştir (He vd., 2016). ResNet mimarisi ile oluşturulan yapılar, artık bloklarda atlama bağlantıları (İng. skip/shortcut connections) ile alt katmanların çıktılarının üst katmanlara bağlantı kurularak etkili çalışmasını sağlamaktadır (Al-Haija ve Adebajo, 2020). Bağlanan katmanlar boyut olarak birbirine eşit ise, bağlantı identity bağlantı ile gerçekleştirilmektedir. Ancak, bağlanan katmanlar arasında boyut farkı var ise, bağlantı 1×1 evrişim işlemi kullanılarak boyutlar eşitlenmektedir (Al-Haija ve Adebajo, 2020). Identity bağlantılı bir artık katman görüntüsü Şekil 3.3'te gösterilmiştir (He vd., 2016).

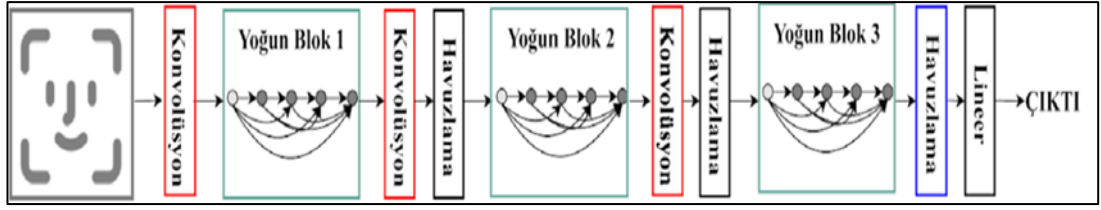


Şekil 3.3. ResNet derin öğrenme mimarisine ait artık blok yapısı (He vd., 2016)

ResNet mimarisinin en yaygın kullanılan versiyonları arasında ResNet50 ve ResNet101 bulunmaktadır. ResNet, geniş veri kümelerinde sınıflandırma işlemlerinde oldukça başarılı sonuçlar vermektedir ve özellikle, transfer öğrenme ve diğer görüntü işleme görevlerindeki başarılı sonuçlarından dolayı birçok farklı alanda kullanılmaktadır (Bhargavi ve Babu, 2021).

3.1.2.4. Densely Connected Convolutional Network derin öğrenme mimarisi

Densely Connected Convolutional Network (DenseNet), 2017 yılında tanıtılmış ve derin öğrenme alanında önemli bir yenilik getirmiş bir mimaridir (Hu vd., 2017). DenseNet'in temel özelliği, önceki katmanların çıktılarının mevcut katmana giriş olarak eklenmesiyle ağda yoğun bir bağlantı yapısının oluşturulmasıdır. ResNet'te olduğu gibi gradyan sorunları ve artan derinlikle azalan öğrenme oranı problemi için, ResNet'e benzer bir yapı olan DenseNet mimarisi geliştirilmiştir (Huang vd., 2017). ResNet yapısına benzer şekilde, DenseNet de atlama bağlantılarından oluşan ve yoğun bloklar (İng. dense blocks) olarak adlandırılan yapılar kullanılmaktadır. Şekil 3.4'te DenseNet mimarisinin temel yapısı gösterilmiştir (Huang vd., 2017).



Şekil 3.4. DenseNet derin öğrenme mimarisi (Huang vd., 2017)

DenseNet mimarisinin temel prensibi, her katmanın kendisinden önceki bütün katmanların çıktısını giriş olarak almasıdır. Bu sayede, her katman, ağı önceki katmanlarından gelen tüm bilgileri kullanabilmesine olanak sağlamaktadır. Böylece, oluşturulan ağı daha verimli bir şekilde öğrenmesini sağladığı için alt bölümdeki katmanlar önceki katmanların çıktısına doğrudan erişim sağlayarak ağıdaki her katman daha fazla bilgiye sahip olabildiğini sağlamaktadır (Huang vd., 2017).

DenseNet'in derin öğrenme mimarisinin sunduğu avantajlardan bir diğeri de daha az parametre ile daha derin ağların oluşturulabilmesidir. Ayrıca, sık ve güçlü bağlantılar sayesinde, gradyanların daha etkin bir şekilde yayılması ve bilginin daha verimli paylaşılması mümkün olmaktadır. Bu da ağı daha hızlı ve etkili bir şekilde öğrenmesini sağlamaktadır. Huang ve arkadaşları (2017), önerdikleri DenseNet yapısıyla gradyan problemlerini ortadan kaldıran, aşırı öğrenmeyi minimuma indiren, bütüncül çalışan ve daha az parametre gerektiren bir mimari geliştirmişlerdir. (Huang vd., 2017).

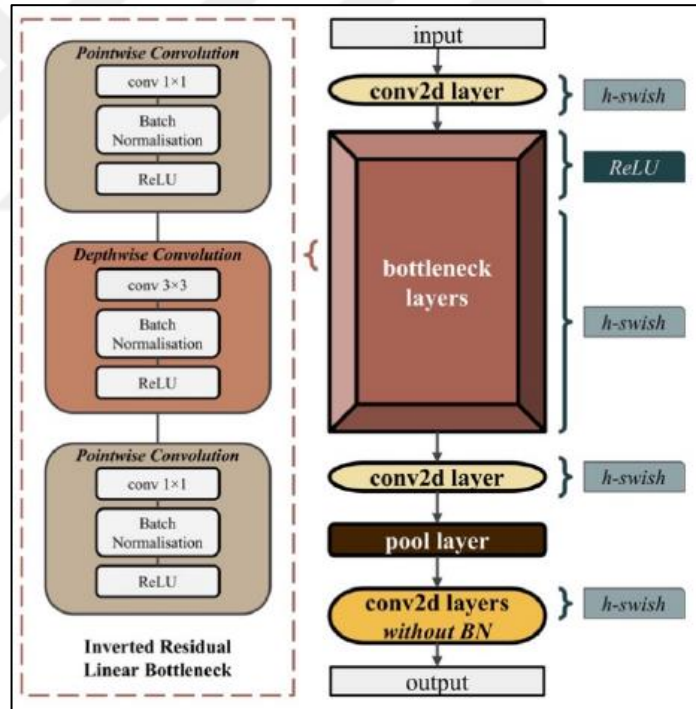
3.1.2.5. MobileNet derin öğrenme mimarisi

MobileNet, evrişimli sinir ağı tabanlı bir model olup görüntüleri kategorilere ayırarak sınıflandırma yapan yaygın bir derin öğrenme mimarisidir. MobileNet mimarisinin en büyük avantajı, geleneksel evrişimli sinir ağı modeline göre daha az hesaplama yaparak hızlı bir öğrenme yapısına sahip olmasıdır (Liu ve Zhu, 2018). Bunun sonucunda da düşük donanım özelliklere sahip bilgisayarlarda, mobil cihazlarda ve diğer sınırlı kaynak ortamlarda çalışmaya uygun bir derin öğrenme mimarisidir (Khasoggi vd., 2019).

MobileNet mimarisi, temel olarak evrişimsel katmanlardan oluşmaktadır. Ancak geleneksel evrişimli sinir ağı yapılarından farklı olarak, daha az parametre

kullanılarak karmaşıklık yapısı indirgenmiştir. Mimarının yapısı, genellikle "depthwise separable convolution" adı verilen bir evrişim katmanı türü kullanılarak sağlanmaktadır. Bu katman, standart bir evrişim katmanını üzerinde önce evrişim işlemi yapıp akabinde ise boyut azaltma işlemi gerçekleştirilmektedir. Böylece parametre sayısı ve hesaplama maliyetini düşürülürken ağın derinliğini muhafaza etmektedir (Khasoggi vd., 2019).

MobileNet mimarisi ayrıca, "depthwise convolution" ve "pointwise convolution" olarak adlandırılan iki tür evrişim katmanını bir araya getirmektedir. "Depthwise convolution", her kanal için bağımsız olarak evrişim işlemi yaparken, "pointwise convolution", boyut küçültme işlevini yerine getirmektedir. Bu iki katmanın birleştirilmesi, özellik çıkarımını daha verimli hale getirmektedir. Şekil 3.5'te MobileNet derin öğrenme mimarisinin yapısı gösterilmiştir (Qian vd., 2021).



Şekil 3.5. MobileNet derin öğrenme mimarisi (Qian vd., 2021)

Sonuç olarak, MobileNet mimarisi, mobil cihazlarda ve diğer sınırlı kaynak ortamlarda kullanılmak üzere tasarlanmış hafif, hızlı ve etkili bir derin öğrenme mimarisidir. Bu nedenle MobileNet mimarisi, görüntü sınıflandırma, nesne tespiti ve diğer farklı görüntü işleme görevlerinde geniş bir kullanım alanına sahiptir (Qian vd., 2021).

3.1.3. Google colab platformu

Google Colab platformu, makine öğrenimi ve eğitimi yaygınlaştırmak için tasarlanmış bir bulut hizmetidir. Derin öğrenme için tam donanımlı, gerçek zamanlı ve yüksek performanslı Graphics Processing Unit (GPU) kullanımına imkân tanıyan, açık kaynaklı ve ücretsiz erişim sunan bir platformdur. (Carneiro vd., 2018).

3.1.4. Görüntü işleme

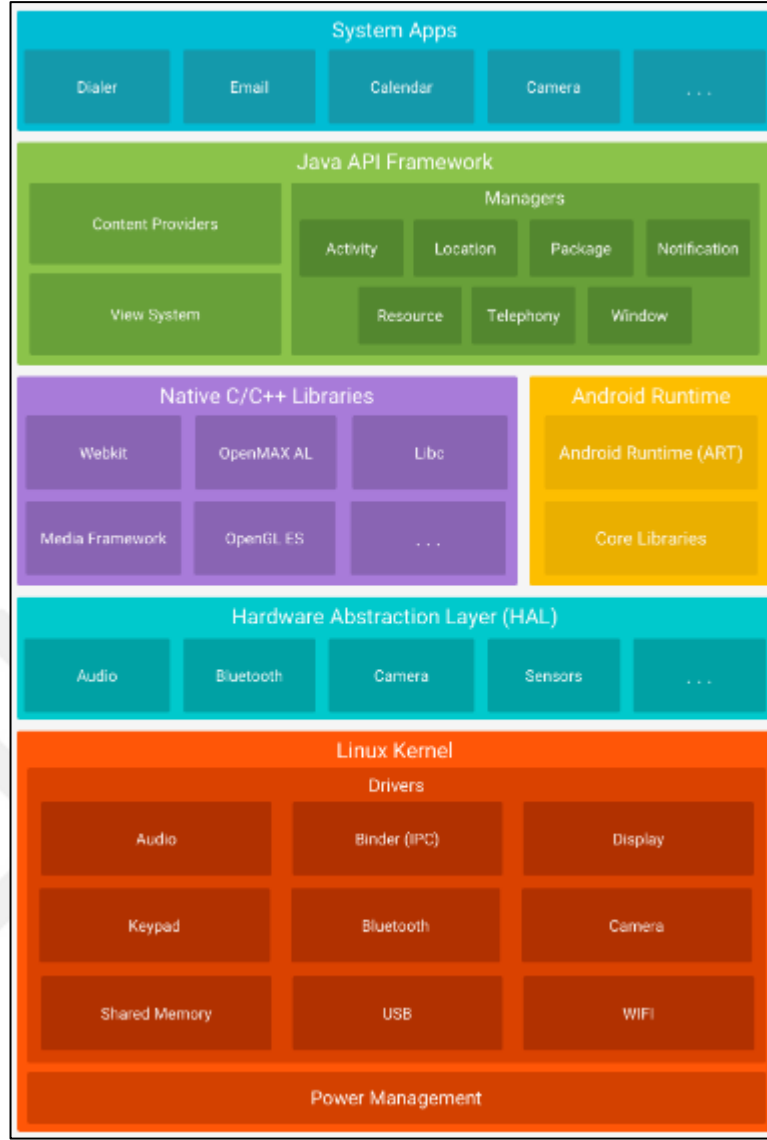
Görüntü işleme, dijital fotoğraflar veya resimler üzerinde, görüntü iyileştirme, görüntüyü analiz etme, görüntüyü istenen kriterlere göre yorumlama ve yapay zekâ uygulamalarında da hazırlama aşaması olarak uygulanan bir bilim alanıdır. Bu süreç, algoritmik ve matematiksel teknikler kullanılarak filtreleme, morfolojik işlemler, kenar tespiti, nesne tanıma ve desen tanıma gibi yöntemleri içermektedir (Eldem vd., 2017).

Görüntü işleme, güvenlik sistemleri, robotik, tıbbi görüntüleme, otomotiv, biyometrik tanıma, video analizi ve endüstriyel otomasyon gibi birçok farklı alanda sıklıkla kullanılmaktadır (Eldem vd., 2017). Bu nedenle görüntü işleme teknolojisi dijital görüntülerin anlamlı bilgiye dönüştürülmesinde sıklıkla kullanılan yöntemlerden birisidir.

3.1.5. Android yapısı ve android işletim sistemi

Open Handset Alliance (OHA) tarafından 2007 yılında kurulan Android, mobil cihazlar için Linux tabanlı ücretsiz, açık kaynak bir işletim sistemi ve mobil uygulamalar geliştirmek için kullanılan açık kaynak yazılımlarla çalışan cihazların birleşmesinden meydana gelmektedir (Meier, 2013).

Android işletim sistemi, mobil cihazlar için tasarlanmış Linux tabanlı açık kaynak kodlu bir işletim sistemidir (Narman, 2013). Android işletim sistemine ait resmî internet sitesinde geliştiriciler için sunduğu android işletim sisteminin mimarisi Şekil 3.6'da gösterilmiştir (Anonim, 2024b).



Şekil 3.6. Android işletim sistemi mimarisi (Anonim, 2024b)

3.1.6. Android studio editörü

Android Studio, Google firması tarafından geliştirilmiş, android uygulamaları oluşturma, geliştirme, test etme ve hata ayıklama gibi işlemler için farklı araçlar sunan, ücretsiz ve açık kaynak kodlu bir tümleşik yazılım geliştirme ortamıdır.

Android Studio, farklı android cihazlara uygulama geliştirmeyi hızlandıran birçok araçları bünyesinde bulundurmaktadır. İçerisine entegre edilmiş olan Gradle yapısı ile uygulama geliştirme sürecinde büyük esneklik sağlamak ve farklı ekran çözünürlüklerine sahip cihazlar için gerekli olan uygulamaları (.apk) üretebilmektedir. Ayrıca, Android Studio kod tamamlama, kod analizi ve kod

düzenleme gibi işlemleri de hızlı bir şekilde gerçekleştirerek hem zamandan tasarruf sağlamakta hem de yazılım standartlarını yükseltmemize olanak sağlamaktadır.

Tez çalışmasında android programlamada kullanılan sınıf ve kavramlara ait bilgiler aşağıda alt başlıklarla detaylı bir biçimde ele alınmıştır.

3.1.6.1. Android activity

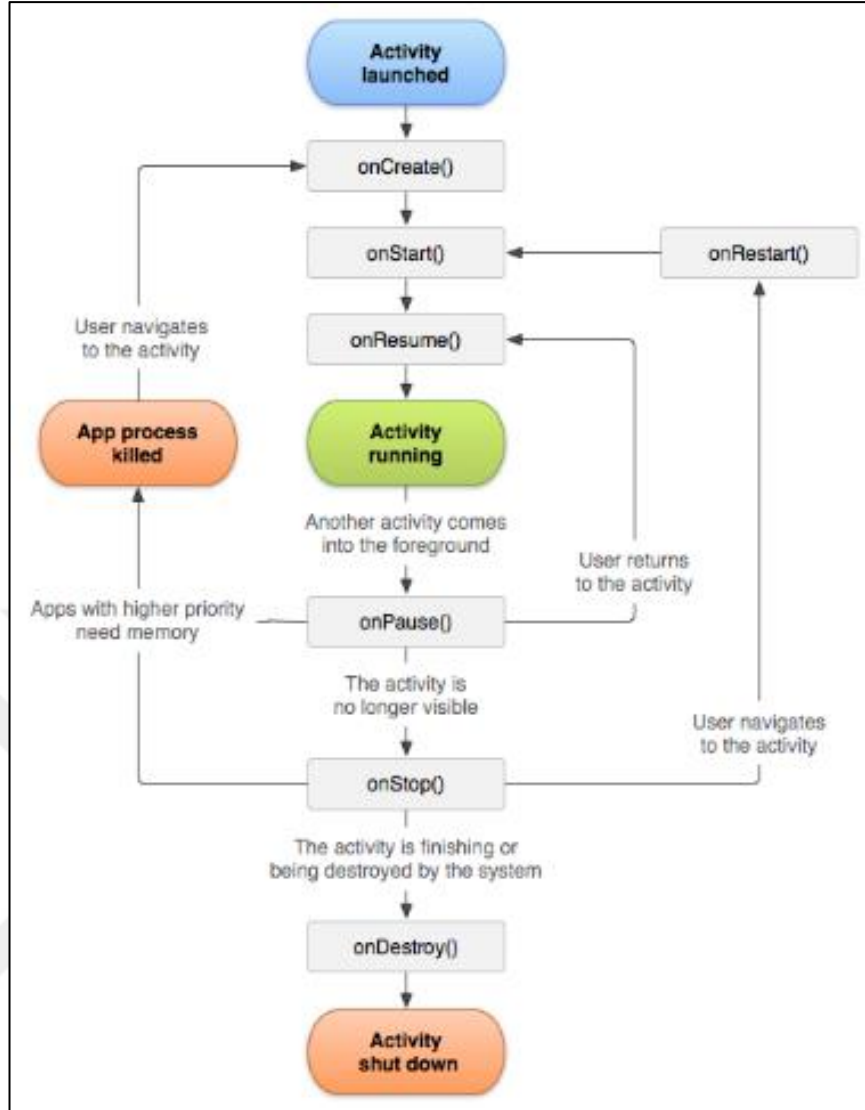
Android işletim sistemine sahip cihazlarda, kullanıcıyla etkileşime girilen ekranları temsil eden sınıflar activity sınıflarıdır. Bu sınıflar, geliştiricilerin tasarladığı ekranlarda yer alan bileşenlerin (TextView, DropDownList, EditText, Button, ListView gibi) işlevsel olarak kullanılmasını sağlayan yapılardır.

AndroidManifest.xml dosyasında tanımlanmamış bir activity sınıfını kullanmak mümkün değildir. Bu sınıfın dosyada belirtilmemesi durumunda, uygulama içinde kullanılamaz ve kullanıcı etkileşime geçtiğinde hatalar oluşabilir. Bu nedenle, activity sınıflarının doğru bir şekilde tanımlandığından emin olmak çok önemlidir.

Activity sınıfları, layout klasöründe yer alan .xml dosyalarıyla ilişkilendirilerek tasarlanır. Xml dosyalarında tanımlanan bileşenlerin kimlikleri (İng. id) kullanılarak bu bileşenlerin activity sınıfı içinde hangi işlevleri yerine getireceği belirlenir ve belirlenen kurallar çerçevesinde uygulanır.

Activity sınıflarının belirli bir yaşam döngüsü vardır. Kullanıcılara güvenilir bir uygulama sunabilmek için activity sınıflarının yaşam döngüsünü iyi anlamak ve içlerindeki tüm metotlara hâkim olmak oldukça önemlidir. Hangi metodun ne zaman ve hangi koşullarda kullanıldığını bilmek, uygulama geliştirme sürecinde büyük faydalar sağlamaktadır.

Android geliştiricilerin android activity'nin yaşam döngüsünün şematik gösterimi Şekil 3.7'de gösterilmiştir (Anonim, 2024c).



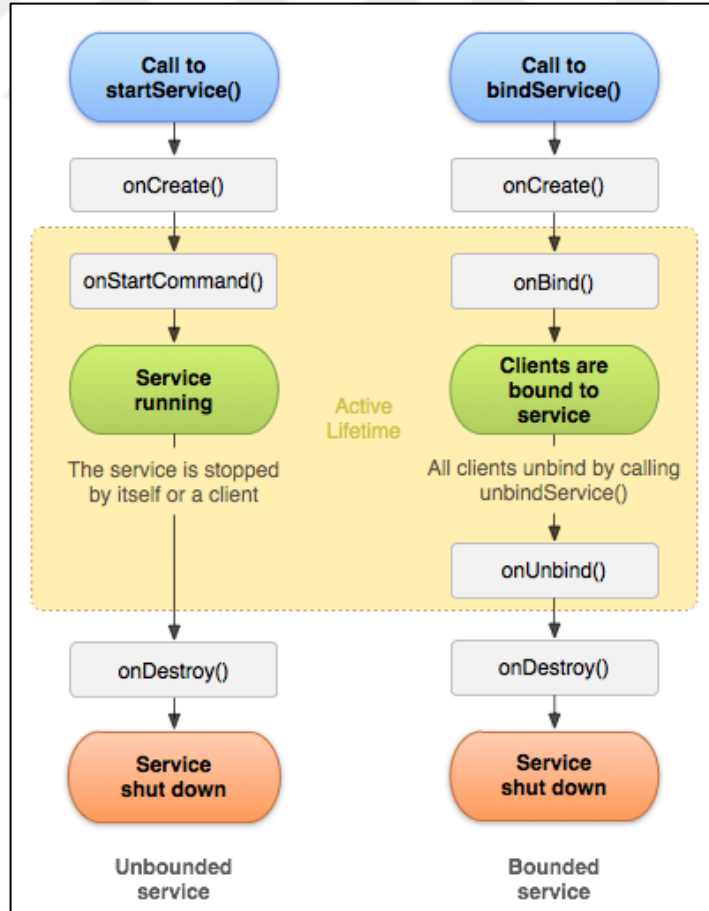
Şekil 3.7. Activity yaşam döngüsü (Anonim, 2024c).

3.1.6.2. Android layout

Android uygulama geliştirme sürecinde, kullanıcıya sunulacak ekran tasarımları res/layout klasöründeki .xml dosyalarıyla oluşturulmaktadır. Bu dosyalarda, uygulamada kullanılacak bileşenlerin nasıl görüneceği belirlenerek geliştiricilere LinearLayout, RelativeLayout, ConstraintLayout, FrameLayout gibi farklı yerleşim türleri sunulmaktadır. Uygulama içindeki tasarım dosyalarının işlevselliği, Activity sınıfları aracılığıyla sağlandığı için activity sınıfları ile .xml dosyaları arasında bir bağlantı kurulmaktadır. Mobil uygulama yazarken bu bağlantının doğru bir şekilde kurulması oldukça önemlidir.

3.1.6.3. Android servis

Mobil uygulamalarda, kullanıcıların mobil cihaz üzerinde kullandıkları uygulamaları kullanırken etkilemediği ancak arka planda belirli işlemleri gerçekleştiren sınıflara "Service" sınıfları adı verilmektedir (Caliskan ve Tasdemir, 2018). Android uygulamalarında, Service sınıfları herhangi bir .xml dosyasıyla ilişkilendirilemez. Bu sınıflar, genellikle ara yüzü olmayan, çoğunlukla veri tabanı işlemleri için kullanılan ve internet üzerinden gelen bildirimleri kontrol eden veya uygulama kapandığında bile arka planda belirli aralıklarla işlem yapılmasını sağlayan android işletim sisteminin önemli bileşenleri arasındadır. Bu işlemler, yazılım ile arka planda gerçekleştirilir ve android işletim sistemi içerisinde, service sınıfları, activity sınıflarına göre daha üst düzeyde önceliğe sahiptir. Ayrıca, android uygulaması yazılırken oluşturulan service sınıfı, AndroidManifest.xml dosyasında tanımlanmalı ve öncelik sırası belirlenmelidir. Oluşturulan Android Service sınıfının yaşam döngüsü Şekil 3.8’de gösterilmiştir (Anonim, 2024d).



Şekil 3.8. Android Service sınıfı yaşam döngüsü (Anonim, 2024d)

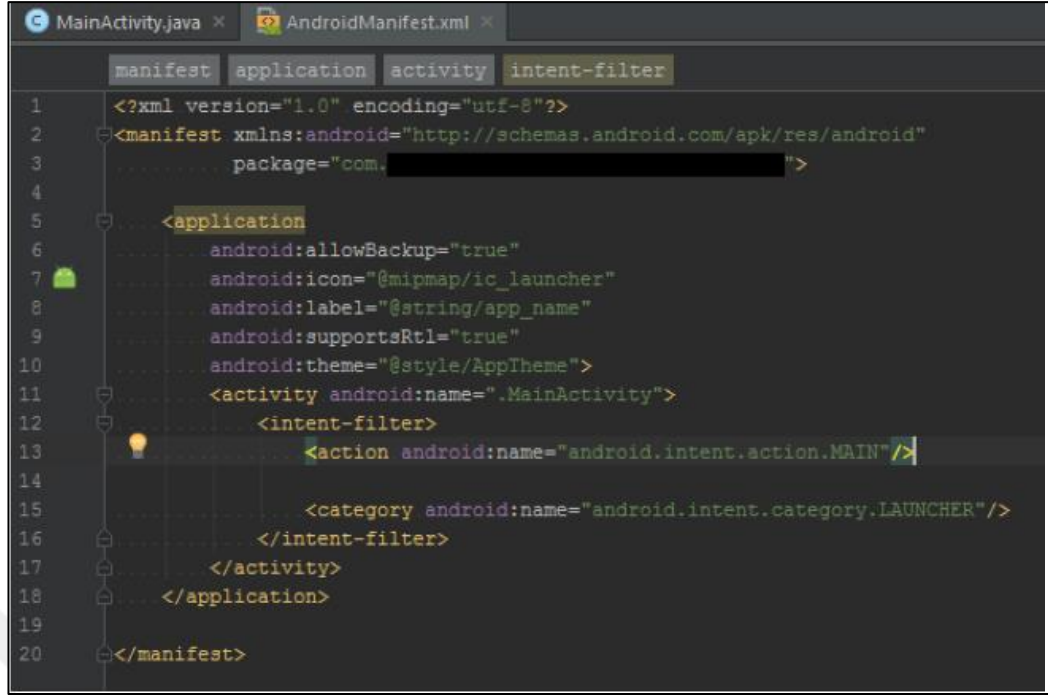
3.1.6.4. Intent yapısı

Geliştirilen android uygulamalarında, birden çok activity veya service, belirli kurallar dâhilinde aynı anda kullanılabilir. Intent yapısı ise, activityler arasında geçiş yapma, farklı bir uygulamayı görüntüleme veya servisleri başlatma ve durdurma gibi durumlarda kullanılan android işletim sistemine özgü özel bir yapıdır. Açık (İng. explicit) intent ve kapalı (İng. implicit) intent olmak üzere iki tür intent yapısı kullanılmaktadır. Açık intent, activityler arasında geçiş yapma gibi belirli bir bileşenin başlatılması için kullanılırken, kapalı intent ise görüntü alma, tarayıcı açma gibi belirli bir amaç için yeni bir bileşenin başlatılmasında kullanılmaktadır.

3.1.6.5. AndroidManifest.xml dosyası

Android uygulama geliştirme ortamında, uygulamanın ana dizininde yer alan ve uygulama için önemli bilgileri içeren, ayrıca uygulama içinde kullanılacak izinleri belirten zorunlu bir .xml dosyasıdır. Örneğin, uygulama içinde Android cihazın bluetooth, internet, dosya yazma ve okuma gibi özelliklerini kullanabilmek için gerekli izinler bu dosyada belirtilir. Ayrıca, activity ve service tanımlamaları gibi önemli android uygulama işlemleri de bu dosya içerisinde gerçekleştirilmektedir.

Android geliştirme ortamında oluşturulan standart bir AndroidManifest.xml dosyasının genel görünümü Şekil 3.9 'da gösterilmiştir (Çalışkan, 2019).



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.████████████████████" >
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="@string/app_name"
9         android:supportsRtl="true"
10        android:theme="@style/AppTheme">
11        <activity android:name=".MainActivity">
12            <intent-filter>
13                <action android:name="android.intent.action.MAIN">
14
15                <category android:name="android.intent.category.LAUNCHER"/>
16            </intent-filter>
17        </activity>
18    </application>
19
20 </manifest>
```

Şekil 3.9. Standart AndroidManifest.xml dosyasının genel görünümü (Çalışkan, 2019)

3.1.6.6. Build.Gradle yapısı

Build.Gradle yapısı android uygulama geliştirme süreçlerindeki karmaşıklığı düzenleyen bir yapı sistemidir. Build.Gradle yapısı kullanılarak birçok android projesinin oluşturulma süreçlerini daha iyi yönetilebilir hale getirilir ve kütüphane yönetimini geliştiricileri için daha kolay bir yapıya dönüştürülmektedir. Build.gradle dosyasındaki dependencies bölümünde belirtilen kütüphanelerin internet vasıtasıyla doğrudan indirilmesini sağlayarak, android uygulama geliştirme sürecindeki karmaşıklığı en alt seviyeye indirgeyerek, üçüncü parti kütüphanelerin hatasız şekilde indirilip yüklenmesi aşamasında geliştiriciye büyük avantajlar sağlamaktadır.

Tez kapsamında geliştirilen mobil uygulamaya ait build.gradle dosyasının görünümü Şekil 3.10'da gösterilmiştir.

```

1  plugins {
2      id 'com.android.application'
3  }
4
5  android {
6      namespace 'com.example.myapplication'
7      compileSdk 34
8
9      defaultConfig {
10         applicationId "com.example.myapplication"
11         minSdk 27
12         targetSdk 34
13         versionCode 1
14         versionName "1.0"
15
16         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
17     }
18
19     buildTypes {
20         release {
21             minifyEnabled false
22             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
23         }
24     }
25
26     compileOptions {
27         sourceCompatibility JavaVersion.VERSION_1_8
28         targetCompatibility JavaVersion.VERSION_1_8
29     }
30 }
31
32 dependencies {
33     implementation 'androidx.appcompat:appcompat:1.6.1'
34     implementation 'com.google.android.material:material:1.11.0'
35     implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
36     testImplementation 'junit:junit:4.13.2'
37     androidTestImplementation 'androidx.test.ext:junit:1.1.5'
38     androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
39
40     implementation 'com.google.firebase:firebase-core:23.0.8'
41     implementation 'com.google.firebase:firebase-analytics:21.3.0'
42     implementation 'com.google.firebase:firebase-database-collection:18.0.1'
43     implementation 'com.google.firebase:firebase-datatransport:18.2.1'
44     implementation 'com.google.firebase:firebase-abt:21.1.2'
45     implementation 'com.google.firebase:firebase-crash-plugin:1.1.5'
46     implementation 'com.google.firebase:firebase-auth:22.3.1'
47     implementation 'com.google.firebase:firebase-messaging:23.4.1'
48 }

```

Şekil 3.10. Android Studio build.gradle dosyasının görünümü

3.1.6.7. Android application package dosyaları

Android Application Package (APK) dosyaları, android işletim sistemine sahip cihazlar için hazırlanan uygulama dosyalarıdır. Geliştirilen uygulamaların .apk uzantılı çıktıları, mobil cihazlara yüklenerek uygulamaların kurulumunu gerçekleştirilmektedir. Böylece geliştirilen uygulamalar, kullanıcılar tarafından mobil cihazlarda etkin bir şekilde kullanılabilir.

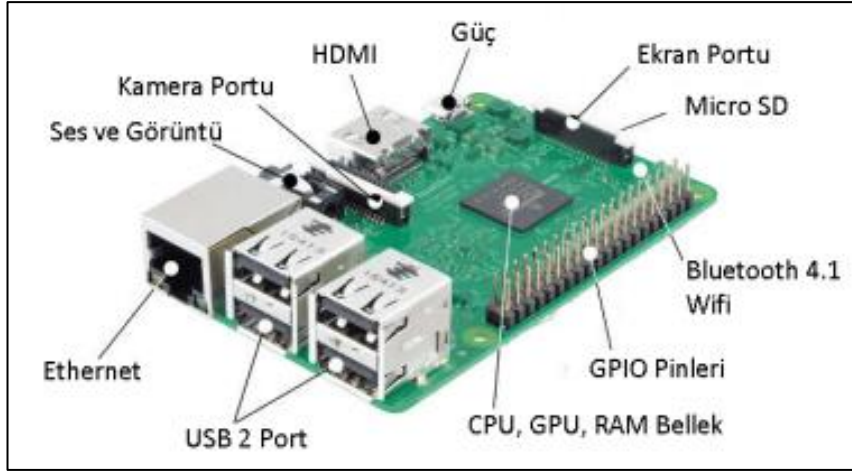
3.1.7. Android cihaz

Android cihazlar, genellikle dokunmatik ekranlarla donatılmış ve Google firmasının geliştirdiği dünya genelinde yaygın olarak kullanılan mobil işletim sistemine sahip elektronik cihazlardır. Android cihazlar, çeşitli uygulamaların indirilip kullanılmasına olanak tanıyan elektronik devrelerin birleşimiyle oluşturulmuştur. Android işletim sistemi, Linux tabanlı açık kaynak kodlu bir yapıya sahiptir ve bu nedenle birçok üretici tarafından farklı cihazlarda kullanılabilir. Özellikle cep telefonları ve tabletler, android cihazlar arasında kullanıcılar tarafından en çok bilinenler arasındadır.

3.1.8. İnternet sunucu

İnternet sunucusu, internet siteleri ve uygulamalarının internet üzerinde erişilebilir olmasını sağlayan donanımsal cihazlar olarak tanımlanmaktadır. İnternet sunucular, farklı uygulamalar arasındaki iletişimi ve veri alışverişini belirli bir veri formatı aracılığıyla sağlayan bir hizmettir ve kullanıcılara bağımsız bir yapı sunmaktadır. İnternet sunucular, kullanıcıların internet tarayıcıları aracılığıyla yaptıkları isteklere yanıt verirler ve html dosyaları, mesajlar, görüntüler ve videolar gibi internet içeriklerini kullanıcılara iletmekte de kullanılmaktadır.

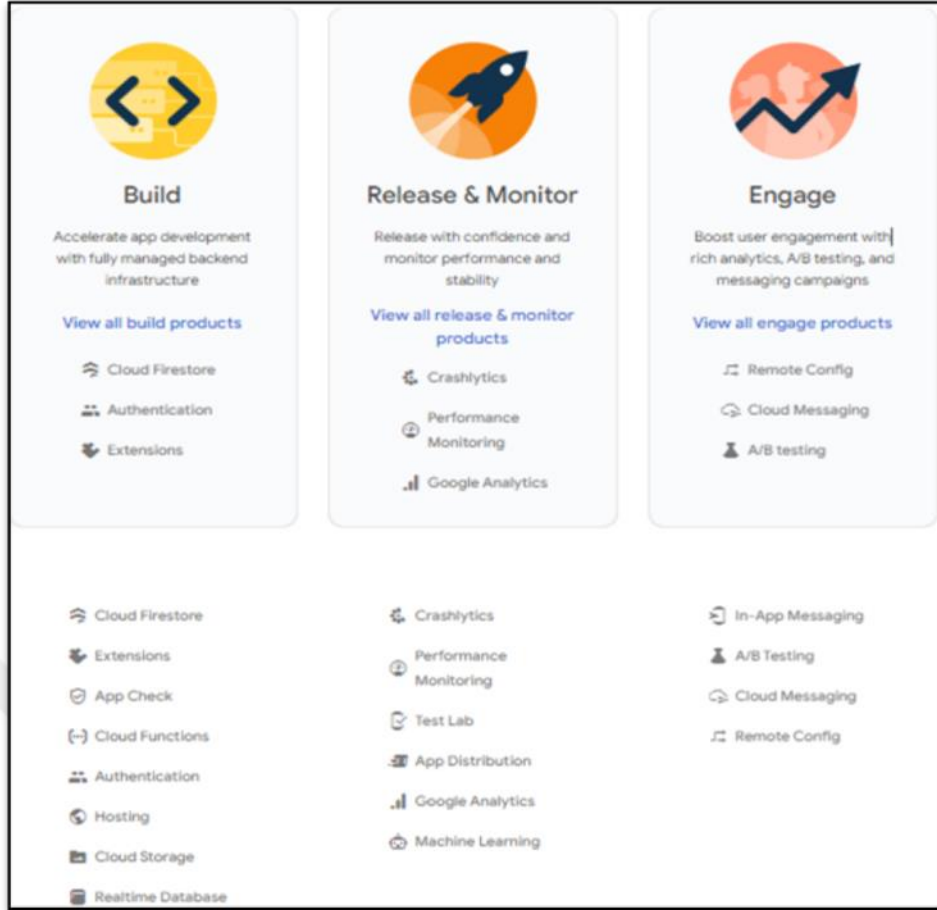
Raspberry Pi, 2012 yılında Raspberry Pi Vakfı tarafından piyasaya sürülen ve açık kaynaklı Raspbian işletim sistemi ile çalışan bir geliştirme kartıdır. Üzerinde bulunan General-Purpose Input/Output (GPIO) pinleri sayesinde çevresel sensörlerle etkileşim kurabilmektedir. Aynı zamanda, beraberinde gelen Raspbian işletim sistemi ile bir bilgisayar gibi kullanılabilmesinin yanı sıra internet sunucu olarak da kullanılabilir. Raspberry Pi 3 geliştirme kartı Şekil 3.11'de gösterilmiştir (Vujović ve Maksimović, 2015). Gerçekleştirilen tez çalışması kapsamında internet sunucu olarak Raspberry Pi 3 kartı kullanılmıştır.



Şekil 3.11. Raspberry Pi 3 geliştirme kartı görünümü (Vujović ve Maksimović, 2015).

3.1.9. Firebase veri tabanı

Firestore, başlangıçta bağımsız bir girişim olarak kurulmuş ancak daha sonrasında ise Google tarafından satın alınarak geliştirilmiş, mobil ve taşınabilir uygulamalara yönelik anlık veri tabanı işlemlerini gerçekleştirmeyi sağlayan ücretsiz bulut tabanlı bir platformdur. Firestore platformu, uygulama geliştiricilerin sunucu tarafı işlemlerini kolaylaştırırken, çevrim içi olarak çalışan ve farklı lokasyonlarda bulunan elektronik cihazlar arasında da veri alışverişini sağlamaktadır. Firestore platformunun sunduğu özellikler Şekil 3.12’de gösterilmiştir (Anonim, 2024e).

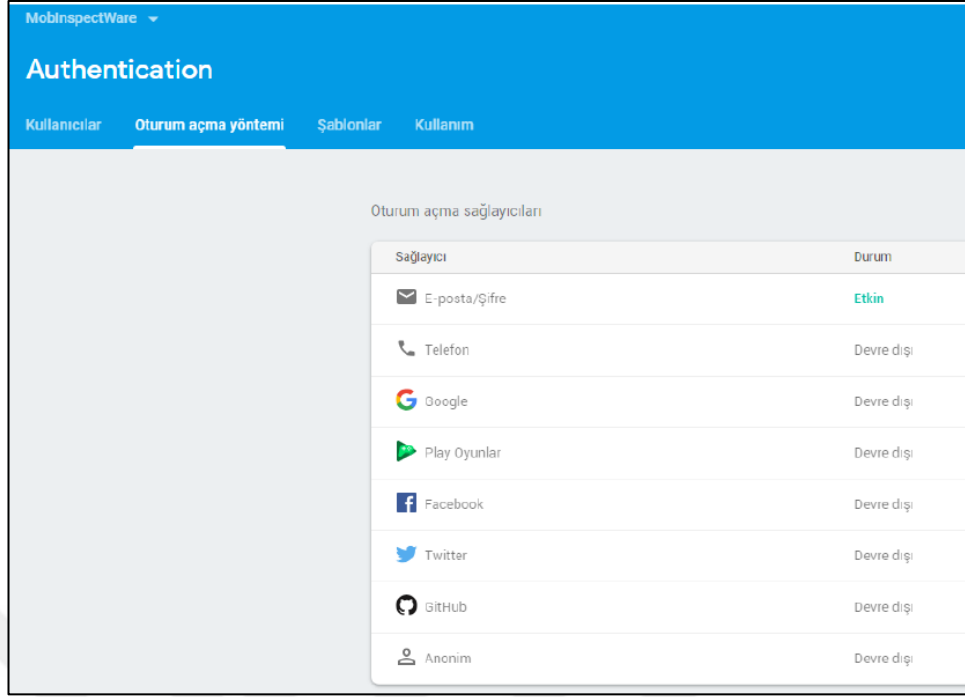


Şekil 3.12. Firebase hizmetlerini gösteren yapı (Anonim, 2024e)

Tez çalışması kapsamında firebase veri tabanı platformunda; yetkilendirme (İng. authentication), gerçek zamanlı veri tabanı (İng. real time database) ve bulut mesajlaşma (İng. cloud messaging) özellikleri kullanılmıştır.

3.1.9.1. Yetkilendirme

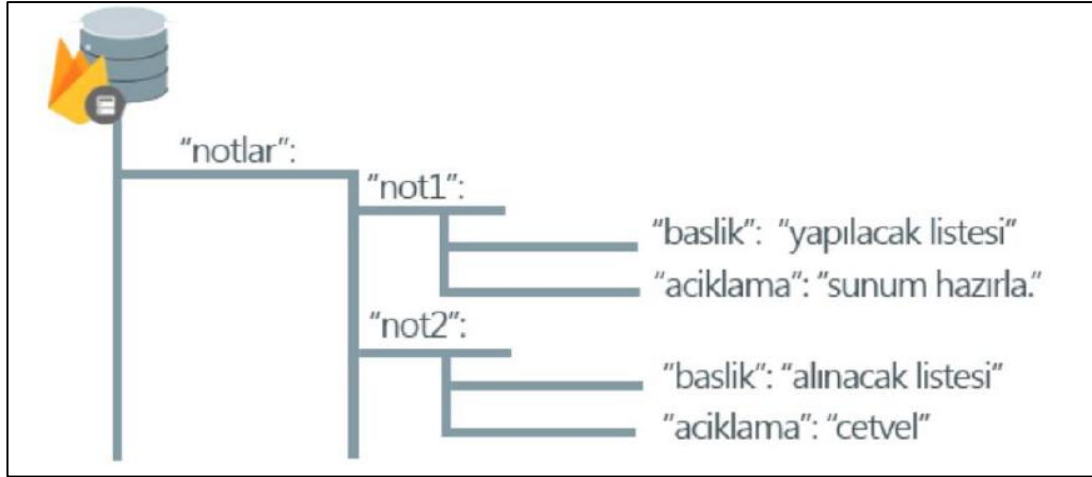
Firebase yetkilendirme, kullanıcıların kimlik doğrulama ve yetkilendirme işlemlerinin Firebase hizmetleri aracılığıyla yönetilmesini ifade etmektedir. Yetkilendirme hizmeti, geliştiricilere, kullanıcıların Google, facebook, X gibi platformlara e-posta ve şifre, telefon numarası veya sosyal kimlik sağlayıcıları kullanarak giriş yapmalarına olanak tanımaktadır. Yetkilendirme süreci, kullanıcıların veri tabanında yeni bilgi ekleme, okuma, güncelleme ve silme gibi yetkilere sahip olmasını sağlamaktadır. Şekil 3.13'te yetkilendirme ekrana ait örnek görsel verilmiştir (Anonim, 2024f).



Şekil 3.13. Firebase yetkilendirme görünümü (Anonim, 2024f)

3.1.9.2. Gerçek zamanlı veri tabanı

Çevrimiçi çalışan Firebase veri tabanı, JavaScript Object Notation (JSON) yapısını kullanarak verileri saklamaktadır. Bu nedenle JSON verileri gerçek zamanlı olarak depolanabilmekte ve eş zamanlı olarak da veri tabanı kullanımına olanak sağlamaktadır. Böylece android uygulamasında yapılan herhangi bir değişiklik anında güncellenmesi ile uygulama geliştiricileri için önemli faydalar sağlamaktadır. Aynı zamanda firebase veri tabanının, No Structured Query Language (NoSQL) veri tabanı özelliği taşıması, JSON formatında depolanan ve kontrol edilen veriler için Structured Query Language (SQL) sorgularına ihtiyaç duyulmaması nedeniyle uygulama geliştiriciler tarafından sıklıkla tercih edilmektedir. Firebase gerçek zamanlı veri tabanında bilgiler JSON objesi olarak tutulmaktadır. Firebase veri tabanında verilerin saklanma biçimine ait örnek görsel Şekil 3.14'te gösterilmiştir (Al-Kababji vd., 2019).

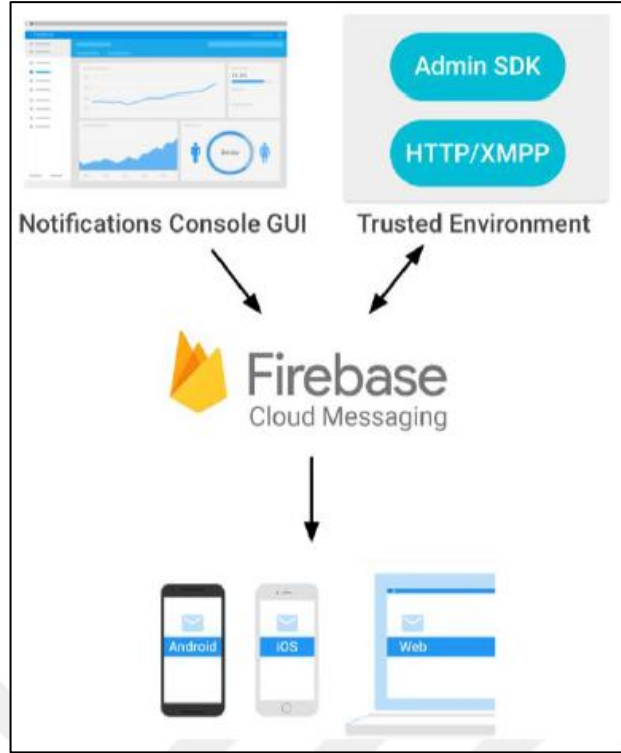


Şekil 3.14. Firebase gerçek zamanlı veri tabanında verilerin depolanma şekli (Al-Kababji vd., 2019)

Firestore veri tabanının gerçek zamanlı veri tabanı panelinde, kullanıcıların okuma ve yazma yetkilerini belirlemek için belirli kurallar bölümü bulunmaktadır. Güvenlik açısından, kuralların dikkatli bir şekilde düzenlenmesi oldukça önemlidir. Aksi takdirde, depolanan verilere herkesin erişebilmesi ve özel bilgilerin siber saldırılar yoluyla başkalarının eline geçmesi gibi büyük riskler oluşabilmektedir. Bu riski kaldırmak için yetkilendirme kurallarındaki talimatlara göre veri tabanı güvenliği alınması oldukça önemlidir.

3.1.9.3. Bulut mesajlaşma servisi

Google Cloud Messaging servisinin güncel sürümü olan Firebase Cloud Messaging (FCM), anlık mesaj gönderimi için kullanılan özel bir mesajlaşma hizmetidir. Bu hizmet, JSON formatındaki verilerin kullanıcılara anında iletilmesini sağlamaktadır. JSON verilerine ait mesajlar, sunucu ile kullanıcı veya kullanıcılar arasında iletişim kurularak veri transferleri daha verimli bir biçimde gerçekleştirilebilmektedir. Firestore veri tabanının mesajlaşma şeması Şekil 3.15'te gösterilmiştir (Anonim, 2024g).



Şekil 3.15. Firebase bulut mesajlaşma şeması (Anonim, 2024g)

3.1.10. JavaScript object notation veri yapısı

JavaScript Nesne Gösteriminin (İng. JavaScript Object Notation) kısaltması olan JSON, JavaScript tabanlı bilgileri düzenli ve kolay erişilebilir bir biçimde saklama yöntemlerinden olup programcılar tarafından okunabilir, değiştirilebilir ve mantıklı bir veri koleksiyonu sunmasından dolayı son yıllarda sıklıkla tercih edilmektedir. JSON verileri, herhangi bir programlama diliyle oluşturulabilir, okunabilir ve değiştirilebilmektedir. JSON, xml programlama diline alternatif olarak geliştirilmiş ve xml diline kıyasla daha basit bir yapıya sahip olmasının yanı sıra boyut olarak da daha az yer kaplaması ve daha hızlı çalışması nedeniyle sıklıkla tercih edilen bir veri yapısı olmuştur. JSON formatındaki veriler, anahtar (İng. key) ve değer (İng. value) olmak üzere iki bileşenden oluşmaktadır. Anahtar, ilgili değişkene verilen ismi, değer ise bu değişkenin içeriğini ifade etmektedir (Peng vd., 2011).

3.1.11. Flask framework yapısı

Flask framework, Armin Ronacher tarafından 2010 yılında geliştirilen bir Python mikro framework yapısıdır. "Mikro" kelimesi, Flask framework yazılımcıların temel

ihtiyalarını karřılayan ve gerektiğinde geniřletilebilen bir yapıya sahip olmasını ifade etmektedir (Singh vd., 2019). Flask framework, birok kütüphaneyi standart olarak içermemekle birlikte, bunun yerine, en gerekli özellikleri yazılımcılara sunup ek paketlerle geniřletme imkânı sağlamaktadır. Böylece daha karmařık yapılar yerine hafif ve esnek bir yazılım geliřtirme süreci sağlanmaktadır.

Flask framework, Web Server Gateway Interface (WSGI) protokolünü destekleyen Werkzeug ile bir sunucu gibi davranması nedeniyle ek bir sunucu kurma ihtiyacını ortadan kaldırmaktadır. Flask framework kullanılarak yapılan her istek Python yönlendiricisi aracılıęıyla ilgili metot ismine yönlendirilir ve bu metot altında gerekli işlemler yapılarak yanıt döndürmektedir. Flask framework, Python diline özgü ve basit bir yapısının olması, farklı donanım desteęi sunan kütüphaneleri içermesi, Model View Controller (MVC) yapısıyla uyumlu alışarak kod karmařıklıęını azaltması ve internet sunucularıyla etkin iletiřim kurarak hızlı işlemler yapabilmesi nedeniyle çevrimii uygulamalarda sıklıkla tercih edilmektedir (Mufid vd., 2019).

3.1.12. Nesnelerin interneti kavramı

Günümüzde internet teknolojinin hızla gelişmesi ile birlikte, gerek dünyada kullanılan nesnelerin akıllı sanal nesneler ile kontrol edilmesi kaçınılmaz bir gerek olarak nesnelerin interneti kavramını (İng. Internet of Things - IoT) ortaya ıkarmıştır. İnternet, dünya apında milyonlarca kullanıcıya hizmet veren ve birbirine baęlı bilgisayar aęlarından oluřan küresel bir aę sistemidir. Bu sistem, farklı elektronik, kablosuz ve optik aę teknolojileriyle birbirine baęlı özel, kamu, akademik, işleme ve devlet aęlarını içermektedir (Nunberg, 2012).

Nesnelerin interneti, çevresel durumlara ve deęişikliklere karřı otomatik olarak kendini geliřtirebilen bilgi, veri ve kaynakları paylařma, tepki verme ve harekete geme kapasitesine sahip, akıllı nesnelere oluřan geniř bir aę olarak tanımlanabilir. Son yıllarda nesnelerin interneti, aęa baęlı fiziksel nesnelerin küresel bir altyapısını oluřturma vizyonuyla dikkat ekmiř ve her zaman, her yerde bulunan nesnelerin birbirleri ile haberleřmesine olanak sağlamıştır (Kosmatos vd., 2011).

3.1.13. Performans değerlendirme ölçütleri

Tez çalışmasında kullanılan derin öğrenme mimarilerinin eğitimden elde edilen sonuçları değerlendirmede doğruluk, hassasiyet, duyarlık, hata oranı ve F1-değeri performans değerlendirme metrikleri kullanılarak bu sonuçlara ait karmaşıklık matrisleri oluşturulmuştur. Karmaşıklık matrisi, doğru sınıflandırılmış pozitif (DP), yanlış negatif (YN), yanlış pozitif (YP) ve doğru negatif (DN) sınıflandırma değerlerini göstermektedir. Karmaşıklık matrisinin kullanım şekline ait detaylı bilgiler Çizelge 3.1'de verilmiştir (Aksoy, 2021).

Çizelge 3.1. Karmaşıklık matrisi (Aksoy, 2021)

<i>KARMAŞIKLIK MATRİSİ</i>	Öngörülen Sınıf		
		Sınıf 1 (Pozitif)	Sınıf 2 (Negatif)
Gerçek Sınıf	Sınıf 1 (Pozitif)	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Sınıf 2 (Negatif)	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Karmaşıklık matrisi kullanılarak hesaplanan doğruluk ölçümü, doğru sınıflandırılan örneklerin toplam örneklere oranını ifade eder ve matematiksel olarak denklem (3.1) verilen matematiksel eşitlik ile hesaplanmaktadır. Hata oranı ise, yanlış sınıflandırılan örneklerin toplam örneklere oranı olup, denklem (3.2) verilen matematiksel eşitlik ile hesaplanmaktadır. Ayrıca, hata oranı ise doğruluk değerinin 1'den çıkarılmasıyla da elde edilebilmektedir (Šimundić, 2009).

$$\text{Doğruluk} = \frac{DP+DN}{DP+YN+YP+DN} \quad (3.1)$$

$$\text{Hata Oranı} = \frac{YP+YN}{DP+YN+YP+DN} \quad (3.2)$$

Karmaşıklık matrisi kullanılarak hesaplanan diğer önemli ölçümler arasında duyarlılık ve hassasiyet bulunmaktadır. Duyarlılık, doğru sınıflandırılan pozitif örneklerin (DP), yanlış sınıflandırılan negatif örnekler (YN) ile toplamına oranı olup,

denklem (3.3)'de verilen matematiksel eşitlik ile hesaplanmaktadır. Hassasiyet ise, doğru sınıflandırılan pozitif örneklerin (DP), toplam pozitif örnek sayısına oranı olarak hesaplanır ve bu oranda denklem (3.4)'de verilen matematiksel eşitlik ile hesaplanmaktadır. (Eusebi, 2013).

$$Duyarlılık = \frac{DP}{DP+YN} \quad (3.3)$$

$$Hassasiyet = \frac{DP}{DP+YP} \quad (3.4)$$

Duyarlılık ve hassasiyet ölçümlerinin harmonik ortalamasının alınmasıyla elde edilen bir performans ölçüm kriteri olan F1-değeri ölçümü (F1- skor) değeri olup denklem (3.5) 'teki matematiksel ifade ile hesaplanmaktadır (Eusebi, 2013).

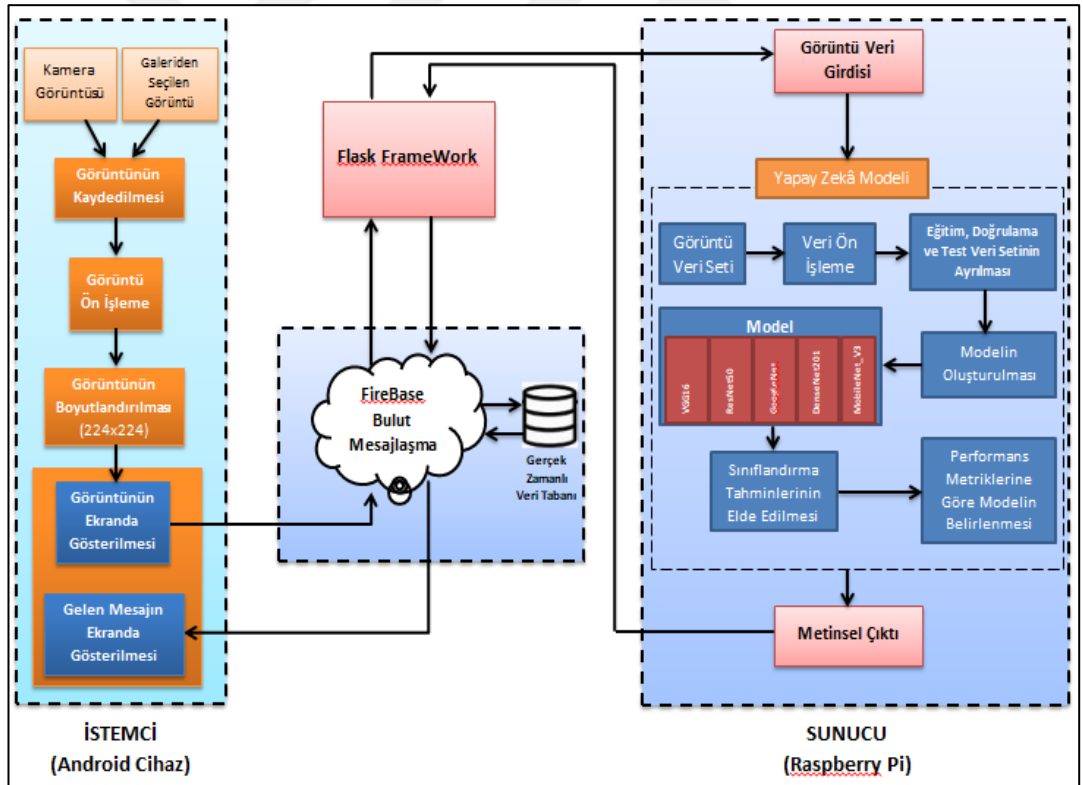
$$F - \text{değeri} = \frac{2DP}{2DP+YP+YN} \quad (3.5)$$

3.2. Yöntem

Tez çalışmasında oluşturulan yazılım için gerçekleştirilen işlemler detaylı bir biçimde çalışmanın yöntem kısmında ele alınmıştır. Tez çalışmasının tüm aşamalarını gösteren iş akış diyagramı Şekil 3.16'da gösterilmiştir. İlk olarak açık erişimli internet sitelerinden farklı insan yüzlerinin olduğu hazır veri setleri araştırılmış ve çalışmaya uygun olan veri seti belirlenmiştir. Bu veri seti yardımı ile internet sunucu olarak seçilen Raspberry Pi üzerinde Python programlama dili kullanılarak görüntü sınıflandırmalarında kullanılan farklı derin öğrenme mimarileri ile modeller oluşturulmuştur. Oluşturulan modeller belirlenen veri seti ile eğitilmiş ve en doğru sonucu veren model, internet sunucusuna yüklenmiştir.

Tez çalışmasında, geliştirilmiş olan android mobil uygulama ile android cihaz üzerinden kişinin görüntüsü alınarak ya da android cihazın galerisinden seçilecek olan görüntünün farklı görüntü ön işlemleri yöntemleri uygulanıp iyileştirildikten sonra, android yazılım ile JSON yapısı kullanılarak FireBase veri tabanına kaydedilmesi sağlanmıştır. Ayrıca Python programlama dili kullanılarak http

işlemleri için geliştirilmiş olan Flask framework ile FireBase veri tabanı üzerinde kayıtlı olan analiz edilmiş görüntü, http protokolü üzerinden Raspberry Pi 3 kartı ile internet sunucusuna gönderilmesi sağlanmıştır. Bu görüntü, Raspberry Pi 3 kartındaki internet sunucusu üzerinde yüklü olan derin öğrenme modeline giriş parametresi olarak verilmiştir. Modele girdi olarak gelen görüntü, derin öğrenme modelinde gerekli analiz ve değerlendirmeler yapılarak çıktı üretilmesi sağlanmıştır. Modelin çıktısı, internet sunucusu tarafından metin olarak cevap mesajı ile Flask framework vasıtasıyla http protokolü üzerinden android mobil uygulamasına gönderilerek uygulama ekranında, bireylerin geometrik olarak hangi yüz tipine sahip olduğu sonuç olarak gösterilmesi sağlanmıştır. Elde edilen sonuca göre, mobil uygulamadaki yazılım ile kullanıcının yüz tipine uygun, bilimsel olarak hangi gözlüklerin kişiye uygun olacağını gösteren bir liste de mobil uygulama ekranında gösterilmesi sağlanmıştır.



Şekil 3.16. Tasarlanan sistemin iş-akış diyagramı

3.2.1. Uygulamada kullanılan veri setleri

Çalışmada, Raspberry Pi 3 kartı üzerinde Python programlama dili kullanılarak derin öğrenme algoritması yardımıyla oluşturulan modelin eğitimi ve doğruluk testi için,

Kaggle açık erişimli internet sitesinde, farklı kişilere ait 5009 tane yüz görüntüsü kullanılmıştır (Anonim, 2024h). Açık erişimli Kaggle internet sitesinde çok farklı boyutta ve kategorilere ayrılmış, çok fazla sayıda veri seti bulunmaktadır. Bu veri setleri arasından tez çalışmasında kullanılmak üzere, kendi içinde “Heart, Oblong, Square, Round ve Oval” şeklinde kategorilere ayrılmış toplamda 5009 adet yüz görüntüsü içeren veri seti kullanılmıştır. Şekil 3.17’de Kaggle açık erişimli internet sitesinden alınmış olan görüntülere ait örnek görsel gösterilmiştir (Anonim, 2024h).



Şekil 3.17. Veri setinde kullanılan görüntülere ait örnek görsel (Anonim, 2024h)

Tez çalışması kapsamında, Isparta Uygulamalı Bilimler Üniversitesi Bilimsel Araştırma ve Yayın Etiği Kurulu’nun 25.10.2023 tarih ve 166/05 nolu kararı ile 5 kişiye ait yüz görüntüsü, çalışmada kullanılan test veri setine eklenmiştir.

Çalışmada, belirlenen veri setindeki görüntüler, derin öğrenme modelinin eğitimi ve doğruluk testi için “kaggle.json” eklentisi yardımı tez kapsamında geliştirilmiş olan yazılımdaki uygulamaya dahil edilmiştir.

3.2.2. Çalışmada kullanılan derin öğrenme yöntemleri ve teknikleri

Çalışmada, nesne tanıma, görüntü işleme, örüntü tanıma ve görüntü işlemlerine bağlı çoklu sınıflandırma problemlerinde başarılı sonuçlar veren, transfer learning

kullanılarak ImageNet kütüphanesi ile oluşturulmuş, VGG16, ResNet50, GoogleNet, DenseNet201 ve MobileNet_V3 derin öğrenme mimarileri tercih edilmiştir. Çalışma kapsamında yukarıda belirtilen derin öğrenme mimarileri ile eğitimler gerçekleştirilmiş ve en başarılı derin öğrenme mimarisi tez çalışmasında kullanılmıştır.

3.2.2.1. Veri ön işleme aşaması

Çalışmada kullanılacak olan derin öğrenme modellerinin eğitiminden önce görüntüler; veri ön işleme aşamasından geçirilmiştir. Öncelikli olarak veri setinde olan tüm görüntülerdeki yüzler, HaarCascade algoritması kullanılarak görüntü yaklaştırma ve kırpma işlemleri ile iyileştirilmiştir. Bu işlemlerden sonra görüntü boyutu 224x224 piksel olacak şekilde yeniden boyutlandırılmıştır. Görüntülerdeki yüzler net bir şekilde belirlendikten sonra veri seti %80 eğitim, %20 test veri seti olmak üzere iki bölüme ayrılmıştır. Tüm sınıflara ait eğitim ve test veri seti dağılımları Çizelge 3.2’de verilmiştir.

Çizelge 3.2. Veri Setinde bulunan sınıfların eğitim ve test veri setlerindeki dağılımları

Sınıf adı	Eğitim Veri Seti	Test Veri Seti	Toplam Veri
Oblong	788	195	983
Square	778	191	969
Oval	814	209	1 023
Round	802	201	1 003
Heart	821	210	1 031
Toplam	4 003	1 006	5 009

Derin öğrenme mimarileri ile elde edilen modeller eğitildikten sonra, test veri seti ile değerlendirilmiştir. Çalışmada kullanılan mimarilerin doğruluk değerini arttırmak için rastgele döndürme, yatay çevirme, düşük oranda yatay ve dikey kaydırma işlemleri uygulanarak veri seti üzerinde veri arttırımı gerçekleştirilmiştir. Veri arttırma işlemi ile Oblong sınıfı için 153, Square sınıfı için 144, Oval sınıfı için 165, Round sınıfı için 148 ve Heart sınıfı için 169 adet veri görüntüsü olmak üzere toplamda 729 adet görüntü veri setine eklenmiştir.

Gerçekleştirilen bu ön işleme adımlarının tamamlanmasından sonra Python programlama dilinin tensorflow platformu altında çalışan keras uygulama arayüzü (İng. Application Programming Interface/API) içinde, her modele özgü ön işleme adımları gerçekleştirilmiştir.

3.2.2.2. Uygulama geliştirme ortamları

Çalışmada, oluşturulan derin öğrenme modelinin daha hızlı eğitilebilmesi ve eğitim sonuçlarının hızlı bir şekilde görülebilmesi için Google'ın desteklemiş olduğu Colab çevrimiçi çalışma platformu kullanılmıştır. Bu platform için programlama dili olarak Python programlama dili tercih edilmiştir. Kullanıcıların mobil platformunda sistemi kullanabilmesi için Android Studio editörü ve bu editörün içerisinde yer alan Java programlama dili kullanılmıştır.

3.2.2.3. Derin Öğrenme modeli için kullanılan kütüphaneler

Çalışma kapsamında geliştirilmiş olan yazılımın önemli bölümlerinden görüntü tanıma ve görüntü sınıflandırma işlemleri için kullanılan derin öğrenme mimarilerinin eğitimde kullanılan kütüphanelerdir. Birçok yazılımcı tarafından açık kaynak kod tabanlı olan Python programlama dili içerisinde yer alan Google ve bağımsız geliştiriciler tarafından geliştirilen Tensorflow ve onun alt kütüphanesi olan Keras kütüphanesi kullanılmıştır. Python programlama dili yardımı ile tensorflow, keras, numpy, os, shutil, matplotlib, zipfile, random, tensorflow lite ve keras kütüphanesinden türetilmiş Optimizers, Image, Model, Preprocessing, Dense, GlobalAveragePooling2D kütüphaneleri çalışmaya dâhil edilmiştir. Geliştirilen yazılımın mobil platformda yayınlanması ve çalışması için de android işletim sistemi tercih edilmiştir.

3.2.2.4. Derin öğrenme modeli için sınıflandırma klasörlerinin oluşturulması

Çalışmanın bu aşamasında ise açık erişimli bir internet sitesi olan Kaggle hazır veri seti Python programlama dilindeki os kütüphanesi kullanılarak farklı yüz tiplerine sahip tüm görüntüler, çalışmaya dâhil edilmiştir. Bu işlemin ardından çalışmanın ana

klasörünün altına “test” ve “train” klasörleri oluşturulmuştur. Daha sonraki aşamada ise tez çalışmasında kullanılan “Heart”, ”Oblong”, “Oval”, “Round”, “Square” sınıflarına ait alt klasörler oluşturulmuştur.

3.2.2.5. Veri setinin rastgele ayrıştırılarak etiketlenmesi

Python programlama dili ile yazılan “veri_ayirma” fonksiyonu ile her bir sınıfa ait görüntü verileri %80 eğitim ve %20 test verisi olacak şekilde ikiye ayrılmıştır. Bir sonraki aşamada ise daha önceden oluşturulmuş olan “Heart”, ”Oblong”, “Oval”, “Round” ve “Square” klasörlerine, shutil kütüphanesinin copyfile ve random fonksiyonları kullanarak, görüntülerin rastgele belirtilen klasörlere aktarılması sağlanmıştır. Aynı zaman da görüntülerin aktarılması ile etiketleme işlemleri de Python programlama dili ile gerçekleştirilmiştir.

3.2.2.6. Çalışmada kullanılan derin öğrenme mimarilerinin eğitilmesi

Çalışmada, nesne tanıma, görüntü işleme, örüntü tanıma ve görüntü işlemlerine bağlı çoklu sınıflandırma işlemleri için seçilmiş olan, VGG16, ResNet50, GoogleNet, DenseNet201 ve MobileNet_V3 derin öğrenme mimarileri, seçilen hazır veri setiyle eğitime başlamadan önce, aşırı öğrenmeyi azaltmak için 0.7 seyreltme katmanı eklenerek optimize edilmiştir. Bu düzenlemeyle oluşturulan modeller, doğrulama veri setinin kayıp değerine bağlı olarak 50 devir boyunca eğitilmiştir. Test veri seti üzerinde en iyi performansı gösteren modeller, diğer modellerle karşılaştırma yapmak amacıyla belirlenmiştir.

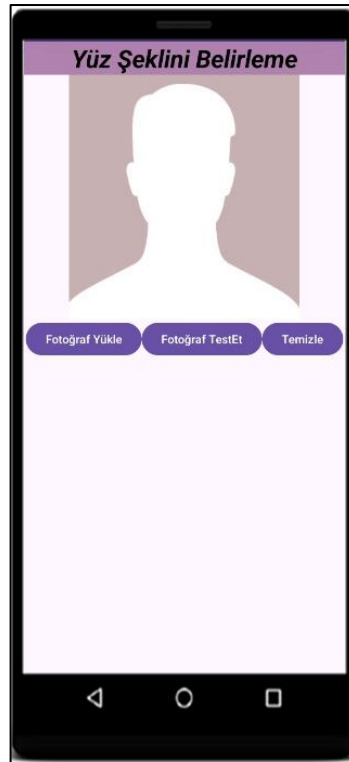
3.2.3. Mobil uygulama yazılımının geliştirilmesi

Çalışma kapsamında geliştirilen mobil uygulama, Android Studio editörü yardımıyla Java programlama dili kullanılarak yazılmıştır. Çalışmanın mobil uygulama kısmı üç ana bölümden oluşmaktadır. İlk bölüm android işletim sistemine sahip cihazın kamera özelliği kullanılarak kişinin görüntüsünün alındığı ya da android cihazın galerisinde bulunan herhangi bir görüntünün seçildiği bölümdür. İkinci bölüm ise android cihaz tarafından alınan veya android cihazın galerisinden seçilen görüntünün kırılarak boyutlarının derin öğrenme modeli için tekrar ayarlandığı bölümdür. Son

bölüm ise mobil uygulamanın ikinci kısmında; kırılmış ve yeniden boyutlandırılmış olan kişi görüntüsünün hangi yüz tipe sahip olduğunun belirlendiği ve belirlenen yüz tipine en uygun gözlüklerin bir listesinin görüntülediği bölümdür. Çalışmaya özgü geliştirilen mobil yazılımda kullanılan yapılar detaylı biçimde açıklanmıştır.

3.2.3.1. Mobil uygulama ana sayfa yapısı

Mobil uygulama açıldığında “MainActivity.java” ilk çalışan dosyadır ve “activitymain.xml” tasarım dosyası ile tümleşik çalışmaktadır. “MainActivity.java” dosyası ilk olarak bünyesinde bulunan kurucu metodu (İng. Constructor) olan “onCreate” metodunu çalıştırır. Bu metot içerisinde “activitymain.xml” dosyasının tasarımında kullanılan bütün nesnelerin tanımlanma işlemleri gerçekleştirilmiştir. Ayrıca tasarımda kullanılan butonların tıklama, sürükle bırak gibi işlevleri de MainActivity.java yapısı ile gerçekleştirilmektedir. Mobil uygulamanın ana sayfasında “Fotoğraf Yükle”, “Fotoğraf Test Et” ve “Temizle” butonları oluşturulmuştur. Mobil uygulama açıldığında ilk ana sayfa görünümü Şekil 3.18’de gösterilmiştir.



Şekil 3.18. Mobil uygulamanın ana sayfa görünümü

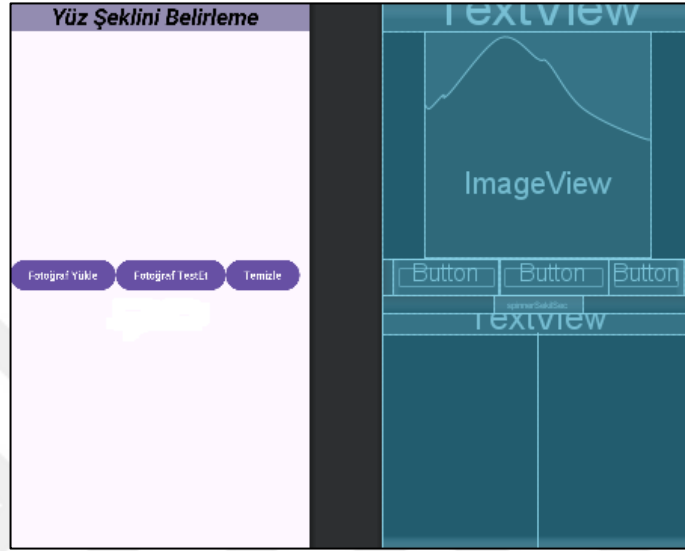
Kullanıcı, mobil uygulamanın anasayfasında yer alan “Fotoğraf Yükle” butonuna tıkladığında, “goruntuYukle.java” dosyası, Java programlama dilindeki “Intent” yapısını kullanarak çağrılmaktadır. Burada kullanıcı, android cihazın fotoğraf çekme ya da galerisinde bulunan bir görüntüyü seçme, seçilen görüntüyü kırpma ve düzenleme gibi işlemlerden sonra, mobil uygulama ana sayfasına tekrar yönlendirilmekte ve son olarak elde edilen görüntü “MainActivity.java” dosyasında bulunan “imageView” nesnesinin içerisine programlama komutları ile yüklenmektedir.

Kullanıcı, mobil uygulamanın ana sayfasında “Temizle” butonuna tıklanıldığında ise, kayıtlı olan ve ekranda görülen bütün görüntü ve mesajların silinmesi sağlanarak uygulamanın ilk açıldığı andaki görünümüne dönülmesi sağlanmıştır.

3.2.3.2. Mobil uygulama ana sayfası arayüz tasarımı

Geliştirilen mobil uygulama yazılımında, uygulama ekranı boyutlarının ayarlanması, ekran görünüm tasarımı için kullanılacak nesnelere eklenmesi ve eklenen nesnelere ekranın istenen konumlara yerleştirilmesi gibi işlemler; “activitymain.xml” dosyasında “XML” komutları kullanılarak gerçekleştirilmiştir. Mobil uygulamaya eklenecek olan nesnelere için “LinearLayout” container yapısı kullanılmıştır. “LinearLayout” yapısının tercih edilme sebebi ise, farklı boyutlarda ekran yapısına sahip android cihazlarda görüntü kayması veya görüntü kayıplarının önüne geçip uygulamanın daha efektif bir şekilde çalışmasını sağlamaktır. Arayüz tasarım sayfasında 3 adet buton, 2 adet imageView ve 4 adet textView nesnesi kullanılmıştır. Birinci buton olan “Fotoğraf Yükle” butonuna tıklanıldığında, android cihazın fotoğraf çekme bölümüne ya da android cihazın galerisine bağlanarak istenilen görüntüyü seçip, imageView nesnesinin içine aktarılması için kullanılmıştır. İkinci buton olan “Fotoğraf TestEt” butonu ise, imageView nesnesi içerisine yüklenmiş görüntüyü, firebase veri tabanına kaydetme ve internet sunucusuna gönderme işlemleri yapacak nesne olarak belirlenmiştir. Mobil uygulamanın herhangi bir anında, kullanıcı uygulamanın başlangıç konumuna dönebilmesini sağlayan “Temizle” butonu da tasarıma eklenmiştir. Tasarımda kullanılan textView nesnelere ise kullanıcıya yüz tipi, yüz tipine ait özellikler ve belirlenen yüz tipine uygun gözlüklerin listesinin görüntülenmesi için kullanılmıştır. İkinci imageView nesnesi

ise belirlenen gözlüklerin kullanıcı tarafından görülebilmesi için bir ön izlemesi nesnesi olarak kullanılmıştır. Tasarımda yatay olarak kullanılan olan bütün nesnelere arasında 20 dp, dikey nesnelere arasında ise 30 dp boşluk olacak şekilde xml komutları ile ayarlanmıştır. Mobil uygulamanın “activitymain.xml” dosyasına ait tasarım görünümü Şekil 3.19’da gösterilmiştir.



Şekil 3.19. Mobil uygulama ana sayfasının ara yüz tasarımı XML görünümü

3.2.3.3. Android cihazın donanımsal özelliklerini kullanabilmek için gerekli izinlerin alınması

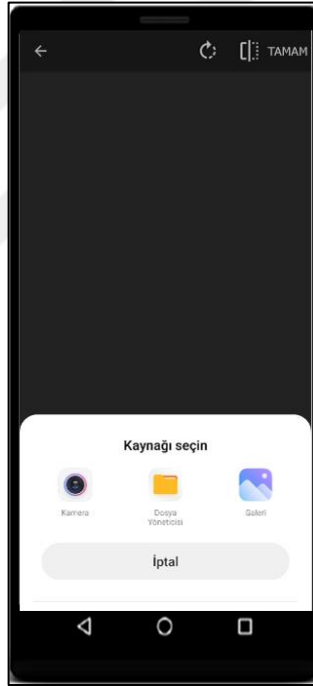
Tez çalışması kapsamında geliştirilen mobil uygulama yazılımında android cihaza ait donanımsal özellikleri kullanabilmek için kullanıcı tarafından izinler verilmesi gerekmektedir. Mobil uygulama için gerekli olan kamera özelliğine erişim ve kamera özelliğini kullanarak fotoğraf çekme işlemi için kamera izinlerinin alınması gerekmektedir. Aynı zamanda çekilen görüntünün android cihazın fotoğraf galerisine kaydetmek veya fotoğraf galerisinin içerisinde olan bir görüntüyü kullanabilmek için de okuma ve yazma izinlerinin de alınması gerekmektedir. Son olarak da mobil yazılımın sistemin çevrimiçi çalışmasından dolayı mobil yazılımın kullanıldığı android cihazın internet kullanımı için izin alınması gerekmektedir. Android Studio editörü yardımıyla manifests klasörünün içindeki AndroidManifest.xml dosyasının içerisine yukarıda belirtilen izin komutları eklenerek alınan izinlere ait görsel Şekil 3.20’de gösterilmiştir.

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-feature android:name="android.hardware.camera"/>
```

Şekil 3.20. Android cihazın donanımsal özelliklerinin kullanılabilmesini sağlayan izinler için gerekli olan komut satırlarının görüntüsü

3.2.3.4. Android cihazın kamera özelliğine ulaşılması ve kullanılması

Geliştirilen mobil uygulama yazılımının ana sayfasındaki “Fotoğraf Yükle” butonuna tıklanıldığında, “goruntuYukle.java” dosyası çağrılmaktadır. Dosya içerisine eklenen fotoğraf çekme, galeri ve dosyalar simgelerine ait görüntüler Şekil 3.21’de gösterilmiştir.



Şekil 3.21. Mobil uygulamanın fotoğraf çekme veya seçme sayfası görünümü

Mobil yazılım uygulamasında fotoğraf çekme simgesine tıklanıldığında, android studio editöründe Java programlama dili ile yazılmış olan “KameraAc” adındaki fonksiyon çağrılmıştır. Şekil 3.22’de de görüldüğü gibi bu fonksiyon içerisinden “getCameraInstance” sınıfı ve bu sınıfa ait komutlar aracılığı ile android cihazın kamerasına erişim sağlanarak “FrameLayout” özelliği yardımı ile kullanıcının görüntü çekme işlemi için hazır hale getirilmiştir.



Şekil 3.22. Mobil uygulamanın fotoğraf çekme sayfası görünümü

Mobil yazılım uygulamasında kullanıcı görüntü çekme butonuna tıkladığında, Şekil 3.23'te de görüldüğü gibi çekilen görüntünün bir ön izlemesi kullanıcıya gösterilerek aynı ekran görüntüsü üzerinde “Tamam” ve “İptal” butonları programatik olarak oluşturulmuştur.



Şekil 3.23. Mobil uygulamanın görüntü çekme işlemi sonrası onaylama veya iptal etme sayfasının görünümü

Mobil yazılım uygulamasında elde edilen görüntünde kullanıcı “Tamam” butonu ile onaylama yaptığında görüntü android cihazın “cache” klasörüne “resim.jpg” şeklinde kaydedilerek dosyanın konumu bir parametre olarak saklanmıştır. Onaylama işlemi ile otomatik olarak “goruntuAl.java” dosyası çağrılıp parametre olarak saklanan dosya yolu metinsel olarak dosyaya aktarılmıştır. Eğer kullanıcı “İptal” butonuna tıklar ise, uygulama ekranında görüntü çekme bölümüne tekrar geri dönülmesi sağlanmıştır.

3.2.3.5. Android cihazın galerisine ulaşılması ve kullanılması

Mobil uygulama yazılımının ana sayfasındaki “Fotoğraf Yükle” butonuna tıklanıldığında, “goruntuYukle.java” dosyası çağrılmıştır. Bu sayfa yüklendiğinde görüntü çekme, galeri ve dosyalar isminde simgeleri olan butonlar görüntülenmektedir. Kullanıcı bu sayfada, “galeri” butonunu seçtiğinde ise “galeriAc” fonksiyonu çalışır ve Java programlama dilindeki “Intent” yardımı ile “Intent.ACTION_PICK” parametresi kullanılarak android cihazın resim galerisi için yeni bir pencerede açılması sağlanmıştır. Kullanıcı açılan galeride istediği bir görüntüyü seçer ve onaylama işlemini gerçekleştirmesi sağlanır. Onaylama işleminin ardından seçilen görüntü “cache” klasörüne “resim.jpg” şeklinde kaydedilir ve kaydedilen dosyanın konumu parametre olacak şekilde saklanır. Onaylama işlemi ile otomatik olarak “goruntuAl.java” dosyası çağrılıp parametre olarak saklanan dosya konumu metinsel olarak bu dosyaya gönderilmiştir.

3.2.3.6. Android cihaz ile elde edilen görüntünün düzenlenmesi

Mobil uygulama yazılım ekranında, kullanıcı android cihazın görüntü çekme bölümünde çektiği bir görüntü ya da android cihazın galerisinde seçtiği bir görüntüyü onayladıktan sonra, “goruntuAl.java” dosyası çağrılmıştır. Şekil 3.24’te de görüldüğü gibi android cihazın “cache” klasörüne kaydedilen “resim.jpg” dosyasına ait dosya konumunu parametre olarak alan ve görüntü ile ilgili kırpma, odaklanma, yaklaşma, uzaklaşma, döndürme ve boyutlandırma gibi işlemleri yapabilen hazır kütüphaneleri içinde bulunduran “CropImageActivity.java” sayfası çalıştırılmış ve

mobil uygulama ekranında “frameLayout” yapısı yardımı ile yeni pencere şeklinde gösterimi sağlanmıştır.



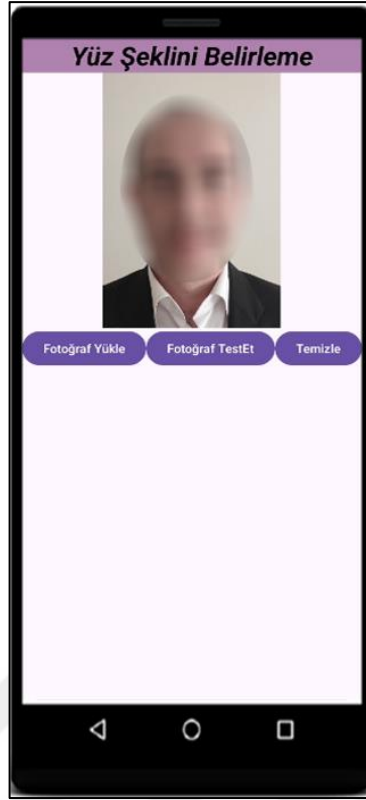
Şekil 3.24. Mobil Uygulamanın görüntü alma, odaklanma ve kırpma sayfası görünümü

Mobil uygulama yazılımında kullanıcı bu bölümde görüntü ile ilgili görüntüye yaklaşma ve uzaklaşma, görüntüye odaklanma, görüntüyü kırpma, görüntüyü istenilen açıda döndürme gibi görüntü düzenleme işlemlerini, görüntü araçlarını kullanarak gerçekleştirilmesi sağlanmıştır. Şekil 3.25’te de görüldüğü gibi kişi görüntüsünde, yüz görüntüsüne odaklanılması bu aşamada gerçekleştirilmiştir. Kullanıcı, bu işlemin ardından sayfanın sağ üst kısmında bulunan “Tamam” butonuna tıklayarak onaylama işlemini yapmaktadır.



Şekil 3.25. Mobil uygulamanın görüntü düzenleme sayfasında iken kişinin yüzüne odaklanıldığı anın görünümü

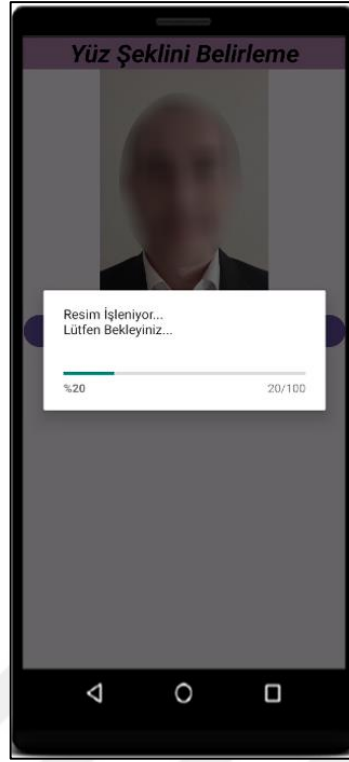
Mobil uygulama yazılımında kullanıcı onaylama işlemini gerçekleştirdikten sonra, elde edilen düzenlenmiş görüntü “goruntuAl.java” dosyasında bulunan “onActivityResult” fonksiyonu yardımı ile tekrar android cihazın “cache” klasörüne boyutları 224x224 olacak şekilde kaydedilmiştir. Şekil 3.26’da da görüldüğü gibi kaydedilen görüntü “imageView” nesnesinin içerisine yüklenerek kullanıcı tarafından görüntülenmesi sağlanmıştır.



Şekil 3.26. Mobil uygulamanın ana sayfasına seçilen görüntünün yüklendiği anın görünümü

3.2.3.7. Android cihaz ile elde edilen görüntünün iyileştirilmesi ve görüntü ön işlemlerine tabi tutulması

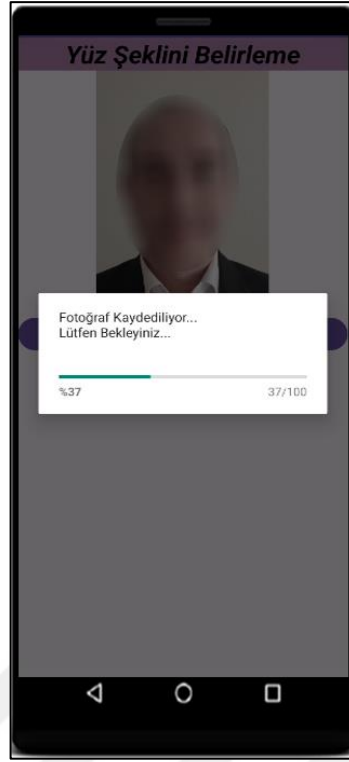
Mobil uygulama yazılımında uygulamanın ana sayfasına istenen görüntü yüklendikten sonra, “Fotoğraf Test Et” butonuna tıklanıldığında, yüklenen görüntünün hemen bitmap değeri hesaplanmış ve hesaplanan değerler otomatik Kırmızı-Yeşil-Mavi (İng. Red-Green-Blue - RGB) renk kanal değerlerine dönüştürülmüştür. Şekil 3.27’de de görüldüğü gibi dönüştürülen RGB değerleri temel alınarak, görüntünün yüz hatları belirgin olacak şekilde, görüntü sıkılaştırma ve görüntü iyileştirme teknikleri kullanılarak görüntü ön işleme işlemleri gerçekleştirilerek android cihazın ekranında mesaj gönderilmiştir.



Şekil 3.27. Mobil uygulamada seçilen görüntünün, görüntü ön işlemlerine tabii tutulduğu anın görünümü

3.2.3.8. Android ortamında iyileştirilen görüntünün FireBase veri tabanına kaydedilmesi

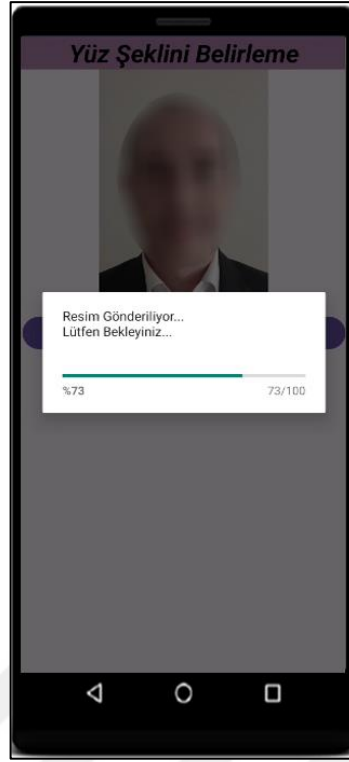
Mobil uygulama yazılımının içerisinde iyileştirilen görüntü otomatik olarak android cihazın “cache” klasörüne tekrar kaydedildikten sonra Java programlama dili ile yazılmış “goruntuKaydetYolla.java” dosyası çağrılmıştır. Şekil 3.28’de de görüldüğü gibi dosya içerisinde bulunan “goruntuKaydet” metodu yardımıyla, görüntü gerçek zamanlı ve çevrimiçi çalışan FireBase veri tabanına kaydedilmiş ve mobil uygulamanın ana ekranından kullanıcıya mesaj olarak gönderilmiştir.



Şekil 3.28. Mobil uygulamada seçilen görüntünün FireBase veri tabanına kaydedildiği anın görünümü

3.2.3.9. Android ortamında iyileştirilen görüntünün internet sunucuya gönderilmesi

Mobil uygulama yazılımında FireBase veri tabanına kaydedilen görüntü, flask framework yardımıyla http protokolleri kullanılarak, <http://localhost:5000/> adresine gönderilmiştir. Çevrimiçi ve gerçek zamanlı çalışan Raspberry Pi kartının internet sunucusu içerisinde Python programlama dili ile yazılmış ve <http://localhost:5000/> adresini her saniye dinleyen fonksiyon yardımı ile bu görüntünün, internet sunucusuna aktarılması sağlanmıştır. Şekil 3.29'da kullanıcıya gerçekleştirilen işlemler mobil yazılım uygulaması üzerinden mesaj olarak gönderilmesine ait görsel gösterilmiştir.



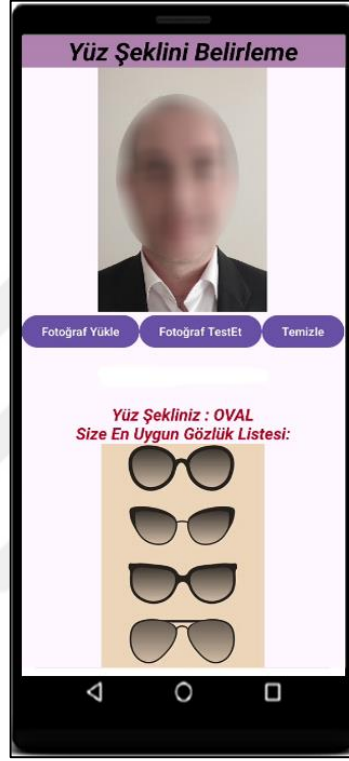
Şekil 3.29. Mobil uygulamada, FireBase veri tabanına kaydedilen görüntünün raspberryPi internet sunucusuna gönderildiği anın görünümü

3.2.3.10. Derin öğrenme modeli tarafından belirlenen yüz tipinin android ekranda gösterilmesi

Mobil yazılım uygulamasında internet sunucusuna gelen görüntü, internet sunucu içerisinde yüklü olan yapay zekâ modeline girdi olarak verilmiştir. Girdi olarak gelen görüntü yapay zekâ modelinde eğitimler yapılarak değerlendirilmiştir. Değerlendirme işlemi sonucunda, yapay zekâ modeli “Yuvarlak Yüz Tipi”, “Oval Yüz Tipi”, “Köşeli Yüz Tipi”, “Kare Yüz Tipi” veya “Kalp Yüz Tipi” şeklindeki metinsel seçeneklerden bir tanesini modelin çıktısı olarak belirlenmiştir. Belirlenen çıktı, sunucu tarafından response mesajına dönüştürülerek, Flask framework yardımıyla http protokolleri üzerinden mobil yazılım uygulamasına gönderilmiştir.

İnternet sunucusu tarafından response mesajı olarak android cihaza gelen bilgi, Java komutları kullanılarak string formatına dönüştürülmüştür. Dönüştürülen bu string mesajı, daha önceden assets klasörünün altında yüz tiplerinin listesi olan “yuzTipleri.txt” dosyasındaki veriler bir döngü yardımı ile tek tek karşılaştırılarak kontrol edilmiştir. Şekil 3.30’da tahmin edilen yüz tipi ile okunan yüz tipinin aynı

olması durumunda mobil uygulama ekranında belirlenen yüz tipinin textView nesnesi ile gösterilmesine ait görsel görülmektedir. Belirlenen yüz tipine göre, assets klasörü altında bulunan “gozluk.txt” dosyasındaki veriler, döngü yardımı ile karşılaştırılır ve belirlenen yüz tipine uygun gözlüklerin bir listesi de geliştirilmiş android yazılım ile imageView nesnesi aracılığı ile kullanıcıya gösterilmesi sağlanmıştır.



Şekil 3.30. Mobil uygulamada, belirlenen yüz tipi ve gözlük listesinin kullanıcıya gösterildiği anın görünümü

3.2.4. İnternet sunucu yazılımının geliştirilmesi

Mobil uygulama yazılımında flask framework yardımıyla <http://localhost:5000/> adresine gönderilen görüntü, çevrimiçi ve gerçek zamanlı çalışan raspberry Pi kartının içerisinde yer alan internet sunucusu içerisinde Python programlama dili ile yazılmış ve <http://localhost:5000/> adresinden her saniye dinleyen “resimKontrolEt” fonksiyonu yardımı ile internet sunucusuna aktarılmıştır.

İnternet sunucusuna gelen görüntü, internet sunucu içerisine yüklenen derin öğrenme modeline girdi olarak verilmiştir. Derin öğrenme modelinde gerekli eğitimler

yapılarak modellerin doğruluđu deęerlendirilmiřtir. Deęerlendirme sonucu olarak derin öğrenme modelinden elde edilen “Yuvarlak Yüz Tipi”, “Oval Yüz Tipi”, “Köřeli Yüz Tipi”, “Kare Yüz Tipi” veya “Kalp Yüz Tipi” olarak belirlenen sınıflar içerisinden doęru sınıflandırılan yüz sınıfına ait metinsel ifade tanesini modelin çıktıısı olarak belirlenmiřtir. Belirlenen çıktı deęeri, internet sunucu tarafından response mesajına dönüřtürölmüřtür. Dönüřtürölen response mesajı, flask framework yardımıyla http protokolleri kullanarak mobil yazılım uygulamasına gönderilmiřtir.

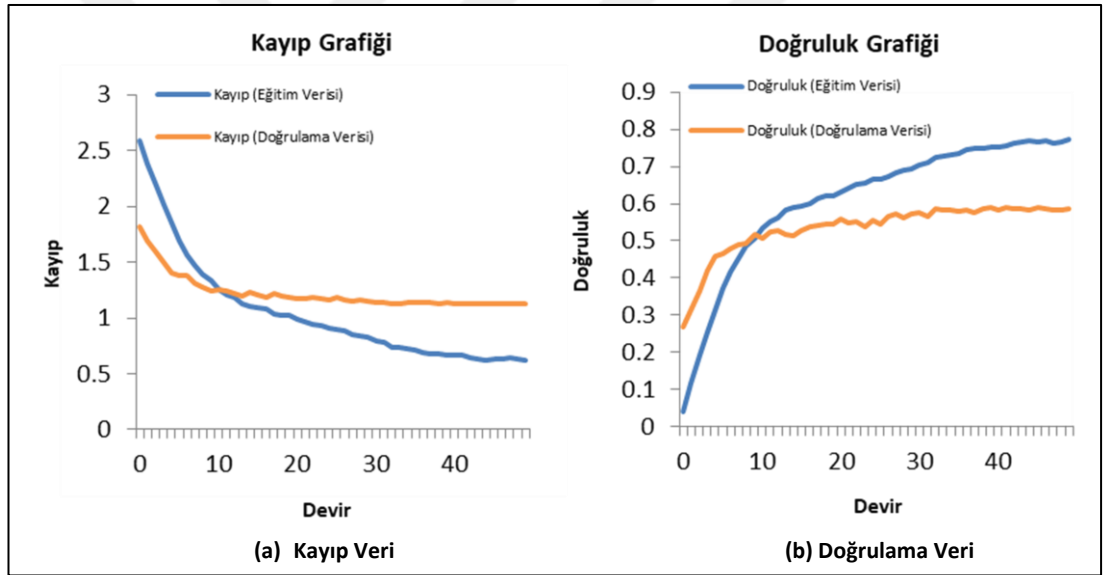


4. BULGULAR

Çalışma kapsamında kullanılan derin öğrenme modelleri ile eğitimi gerçekleştirilen modellerin test veri seti üzerinden doğruluk, duyarlılık, kesinlik ve F1 skor değeri performans değerlendirme metrikleri ile değerlendirilerek elde edilen sonuçlar aşağıdaki alt başlıklarda detaylı bir biçimde ele alınmıştır.

4.1. Derin Öğrenme Mimarilerinden Elde Edilen Eğitim Sonuçları

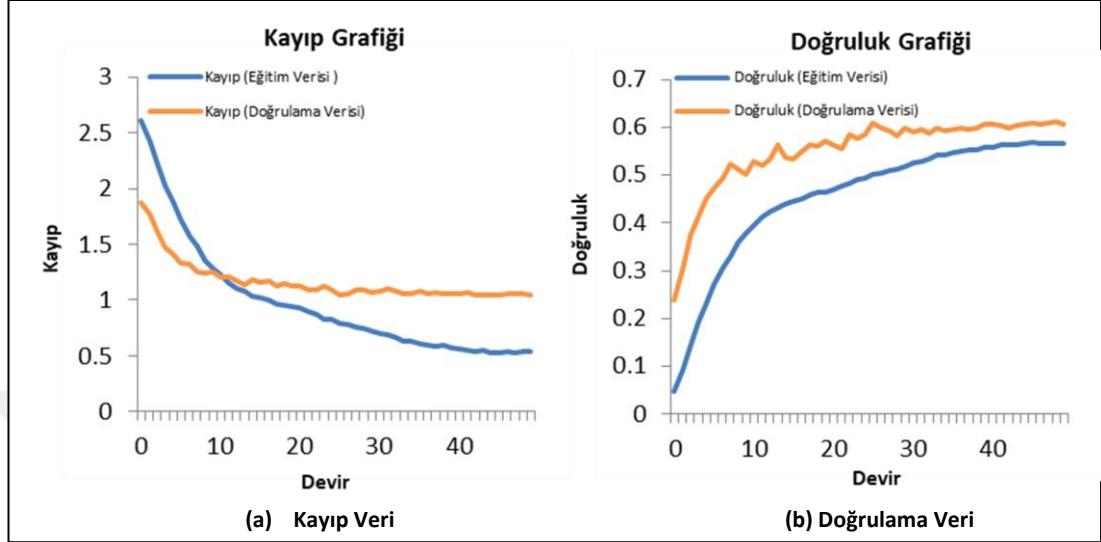
VGG16 derin öğrenme mimarisinin veri artırma öncesinde, eğitim veri seti ile 50 devir (İng. epoch) boyunca eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Veri artırma öncesi VGG16 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri

Şekil 4.1’deki grafikler incelendiğinde VGG16 derin öğrenme ile gerçekleştirilen eğitim esnasında en düşük kayıp değeri olarak 0.7222 (0.5607 doğruluk değeri ile), en yüksek doğruluk değeri olarak ise 0.5891 (1.1381 kayıp değeri ile) elde edilmiştir. VGG16 derin öğrenme modelinin 50 devirlik eğitimin sonunda 1.1268 kayıp değeri ve 0.5817 doğruluk değeri elde edilmiştir.

Veri ön işleme yöntemiyle veri artırma işlemi yapılarak, VGG16 derin öğrenme mimarisinin eğitim veri seti ile 50 devir boyunca eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.2’de gösterilmiştir.



Şekil 4.2. Veri artırma sonrası VGG16 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri

Şekil 4.2’deki grafikler incelendiğinde VGG16 derin öğrenme ile gerçekleştirilen eğitim esnasında en düşük kayıp değeri olarak 1.0412 (0.5694 doğruluk değeri ile), en yüksek doğruluk değeri olarak ise 0.6262 (1.0886 kayıp değeri ile) elde edilmiştir. VGG16 derin öğrenme modelinin 50 devirlik eğitimin sonunda 0.5221 kayıp değeri ve 0.5779 doğruluk değeri elde edilmiştir. En düşük kayıp değerine ve en yüksek doğruluk oranına sahip modeller eğitim sonunda kaydedilerek test veri seti üzerinde değerlendirilmiştir. Test veri seti üzerinde yapılan değerlendirme sonucunda en iyi sonuç, eğitim sırasında en yüksek doğruluk değerini vermiş olan model ile elde edilmiştir.

VGG16 derin öğrenme mimarisi ile test veri seti üzerinden elde edilen karmaşıklık matrisi Şekil 4.3’te gösterilmiştir.

		Karmaşıklık Matrisi				
Gerçek Etiketler	Kalp	112	27	20	20	21
	Köşeli	23	131	18	12	16
	Oval	28	22	88	37	24
	Yuvarlak	11	9	18	144	17
	Kare	3	21	13	18	145
		Tahmin Edilen Etiketler				
		Kalp	Köşeli	Oval	Yuvarlak	Kare

Şekil 4.3. VGG16 derin öğrenme mimarisine ait karmaşıklık matrisi

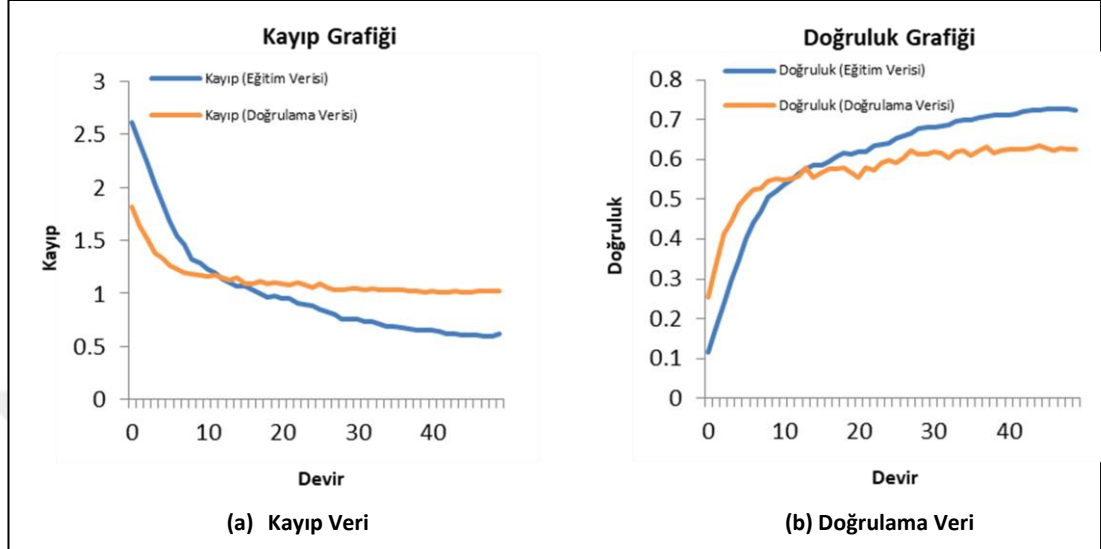
Şekil 4.3'te verilen karmaşıklık matrisi incelendiğinde VGG16 derin öğrenme mimarisinin en başarılı olarak Square (Kare yüz tipi) sınıfını, en başarısız olarak ise Oval (Oval yüz tipi) sınıfını sınıflandırdığı görülmektedir. Modelin test veri seti üzerinde gerçekleştirmiş olduğu tahminler kullanılarak mimariyi değerlendirmek için doğruluk, kesinlik, duyarlılık ve F1 skor performans değerlendirme metriklerine göre değerlendirilerek elde edilen sonuçlar Çizelge 4.1'de verilmiştir.

Çizelge 4.1. VGG16 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları

VGG16	Doğruluk (Test)	Kesinlik	Duyarlılık	F1 skor
	0.621	0.712	0.665	0.689

Çizelge 4.1 incelendiğinde VGG16 derin öğrenme mimarisinin test veri seti üzerinde en yüksek değerinin 0.621 olduğu tespit edilmiştir. VGG16 derin öğrenme mimarisinin test veri seti üzerinde 0.621 doğruluk değeri, 0.712 kesinlik, 0.665 duyarlılık ve 0.689 F1 skor değerleri elde edilmiştir.

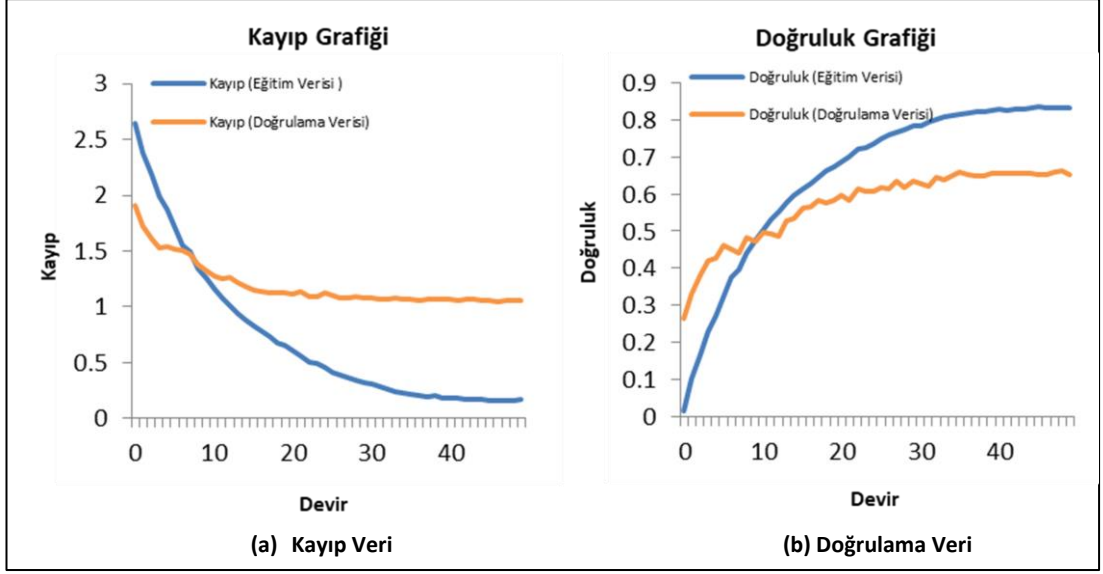
Çalışma kapsamında ResNet50 derin öğrenme mimarisi, veri artırma öncesinde eğitim veri seti ile 50 devir boyunca eğitilerek elde edilen kayıp ve doğruluk grafikleri Şekil 4.4'te gösterilmiştir.



Şekil 4.4. Veri artırma öncesi ResNet50 derin öğrenme mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri

Şekil 4.4'teki grafikler incelendiğinde ResNet50 derin öğrenme mimarisi ile veri artırma öncesi gerçekleştirilen eğitim esnasında en düşük kayıp değeri olarak 1.0114 (0.6242 doğruluk değeri ile), en yüksek doğruluk değeri olarak ise 0.6332 (1.0171 kayıp değeri ile) elde edilmiştir. ResNet50 derin öğrenme modelinin 50 devirlik eğitimin sonunda 1.0209 kayıp değeri ve 0.6239 doğruluk değeri elde edilmiştir.

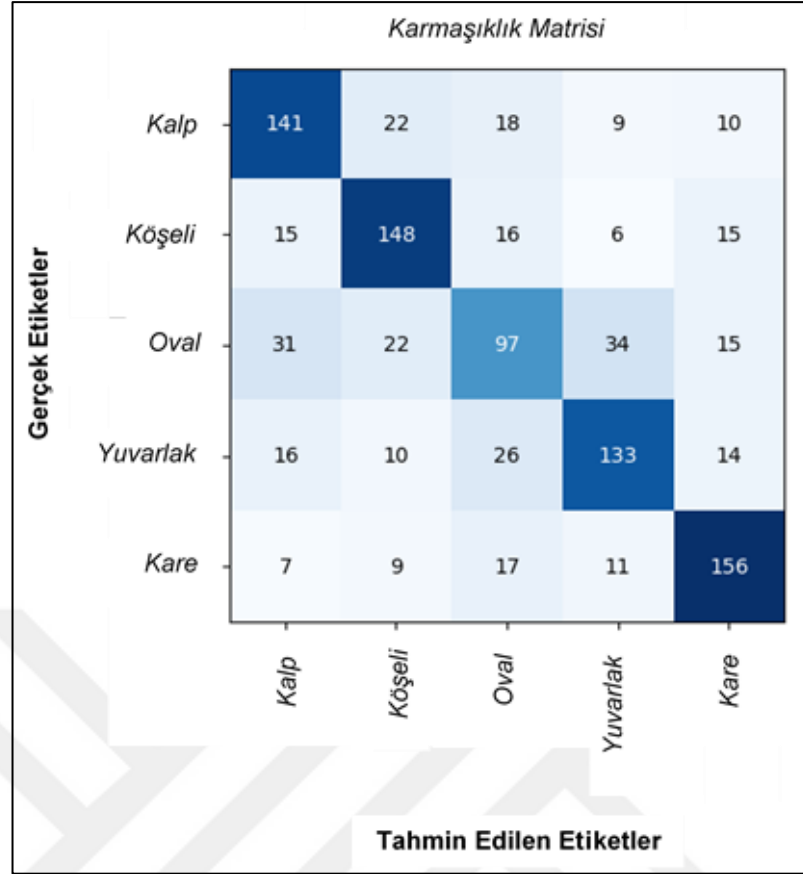
Veri ön işleme yöntemiyle veri artırma işlemi sonunda elde edilen eğitim veri seti ile ResNet50 derin öğrenme mimarisinin 50 devir boyunca eğitime tabi tutulmuş ve eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.5'te gösterilmiştir.



Şekil 4.5. Veri artırma sonrası ResNet50 derin öğrenme mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri

Şekil 4.5 incelendiğinde veri artırma sonunda ResNet50 derin öğrenme modelinin eğitimi esnasında en düşük kayıp değeri olarak 1.0508 (0.6533 doğruluk oranı ile) en yüksek doğruluk değeri ise 0.6893 (1.0567 kayıp değeri ile) olduğu tespit edilmiştir. ResNet50 derin öğrenme modeli 50 devirlik eğitimin sonunda 1.0679 kayıp değeri ile 0.6641 doğruluk değerleri elde edilmiştir. ResNet50 derin öğrenme modelinin eğitiminden elde edilen sonuçlar test veri seti üzerinde yapılan değerlendirme sonucunda en iyi sonuç ve eğitim sırasında en yüksek doğruluk değerini veren model tez çalışmasında kullanılmıştır.

ResNet50 mimarisi ile test veri seti üzerinden elde edilen karmaşıklık matrisi Şekil 4.6'da verilmiştir.



Şekil 4.6. ResNet50 derin öğrenme mimarisine ait karmaşıklık matrisi

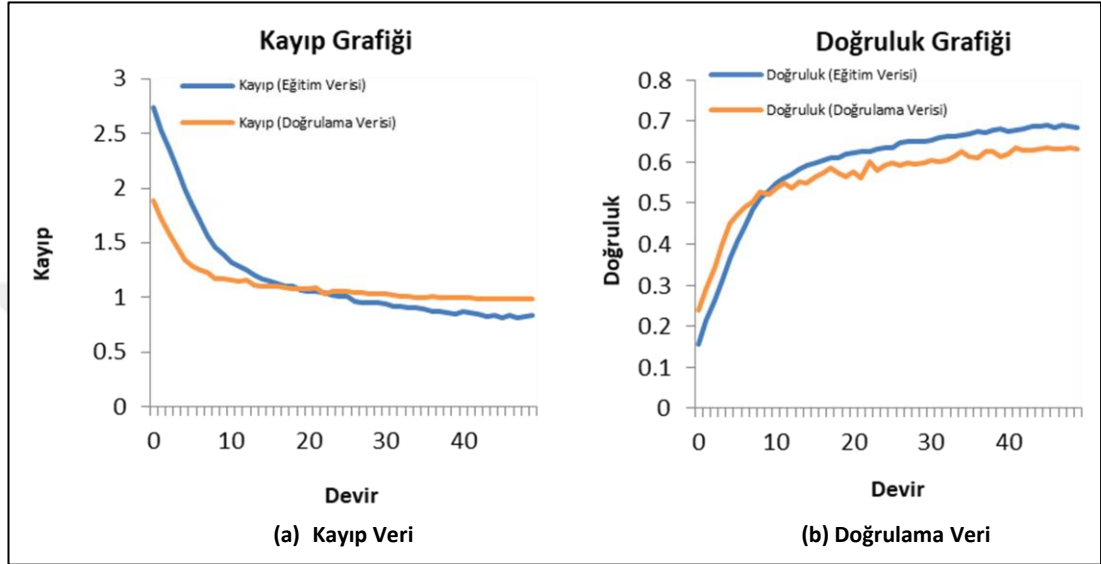
Şekil 4.6’da verilen karmaşıklık matrisi incelendiğinde ResNet50 derin öğrenme mimarisinin en başarılı yüz modeli olarak Square (Kare Yüz Tipi) sınıfını, en başarısız yüz modelini ise Oval (Oval Yüz Tipi) sınıfını sınıflandırdığı görülmektedir. Modelin test veri seti üzerinde gerçekleştirmiş olduğu tahminler kullanılarak mimariyi değerlendirmek için doğruluk, kesinlik, duyarlılık ve F1 skor performans değerlendirme metriklerine göre değerlendirilerek elde edilen sonuçlar Çizelge 4.2’de verilmiştir

Çizelge 4.2. ResNet50 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları

ResNet50	Doğruluk (Test)	Kesinlik	Duyarlılık	F1 skor
	0.676	0.731	0.737	0.734

Çizelge 4.2 incelendiğinde ResNet50 derin öğrenme mimarisinin test veri seti üzerinde en yüksek değerinin 0.676 olduğu tespit edilmiştir. ResNet50 derin öğrenme mimarisinin test veri seti üzerinde ise 0.676 doğruluk değeri, 0.731 kesinlik, 0.737 duyarlılık ve 0.734 F1 skor değerleri elde edilmiştir.

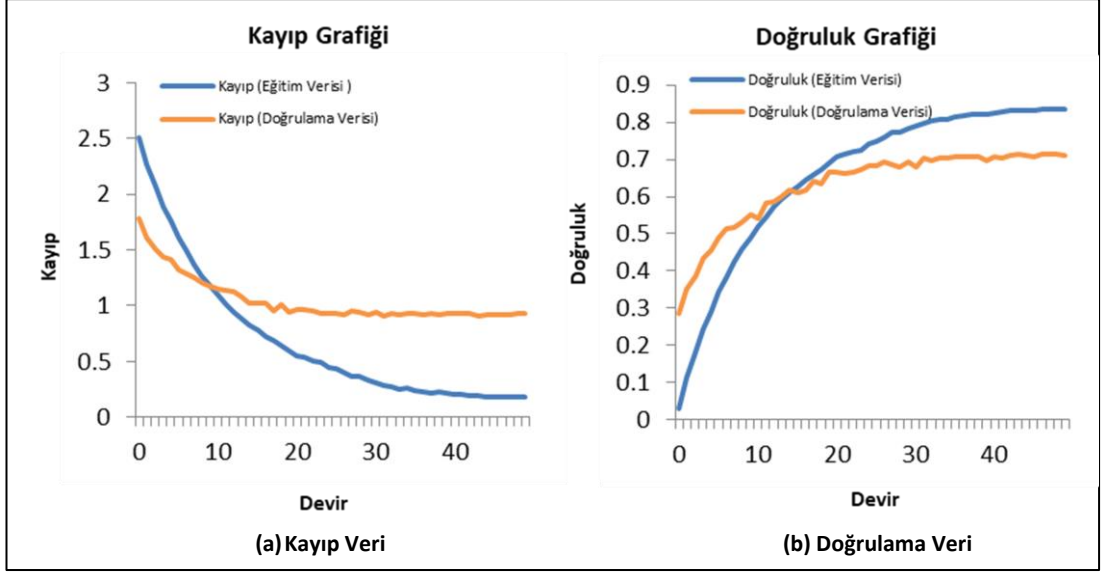
Tez çalışmasında üçüncü derin öğrenme mimarisi olarak kullanılan GoogLeNet derin öğrenme mimarisi, veri artırma işlemi öncesi eğitim veri seti ile 50 devir boyunca eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.7’de gösterilmiştir.



Şekil 4.7. Veri artırma öncesi GoogLeNet derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri

Şekil 4.7’deki grafikler incelendiğinde GoogLeNet derin öğrenme mimarisi ile veri artırma öncesi gerçekleştirilen eğitim esnasında en düşük kayıp değeri olarak 0.9882 (0.6322 doğruluk değeri ile), en yüksek doğruluk değeri olarak ise 0.6437 (0.9943 kayıp değeri ile) elde edilmiştir. GoogLeNet derin öğrenme modelinin 50 devirlik eğitimin sonunda 0.9901 kayıp değeri ve 0.6239 doğruluk değeri elde edilmiştir.

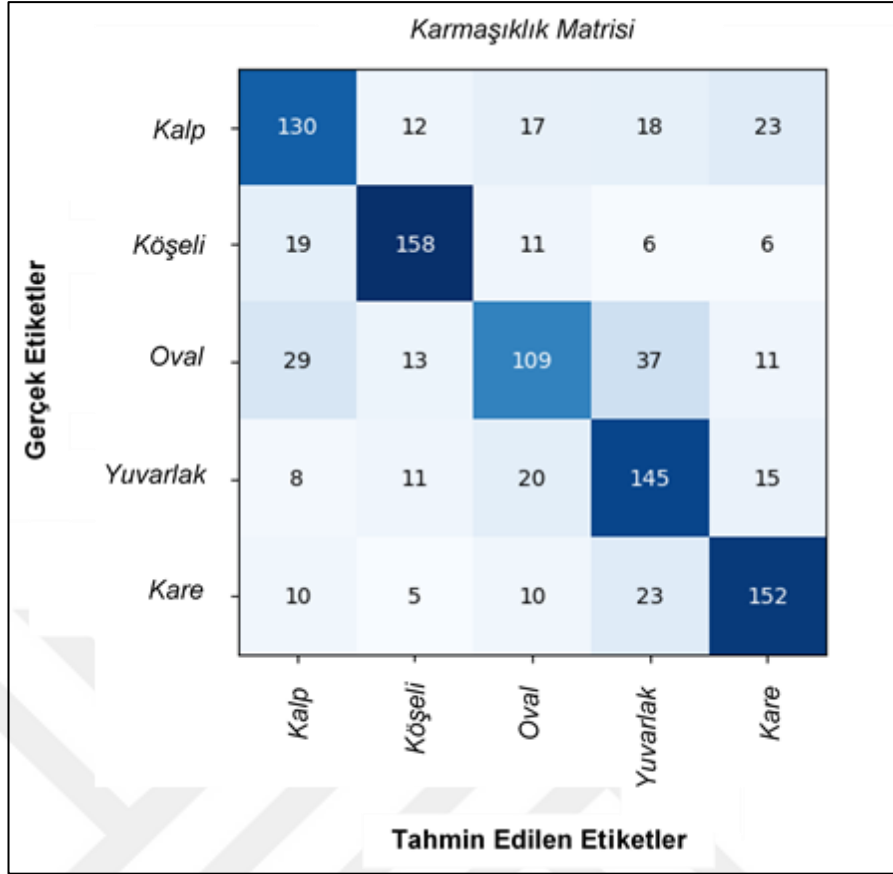
Veri ön işleme ile veri artırma işlemi yapılarak edilen eğitim veri seti 50 devir boyunca, GoogLeNet derin öğrenme mimarisinin eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.8’de gösterilmiştir.



Şekil 4.8. Veri artırma sonrası GoogLeNet derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri

Şekil 4.8 incelendiğinde veri artırma işlemi sonucunda, GoogLeNet derin öğrenme modelinin eğitimi esnasında en düşük kayıp değeri olarak 0.9106 (0.7024 doğruluk oranı ile) en yüksek doğruluk değeri olarak 0.7154 (0.9146 kayıp değeri ile) tespit edilmiştir. GoogLeNet derin öğrenme modeli 50 devirlik eğitimin sonunda 0.9257 kayıp değeri ile 0.7094 doğruluk değeri elde edilmiştir. GoogLeNet derin öğrenme modelinin eğitiminden elde edilen sonuçlar test veri seti üzerinde yapılan değerlendirme sonucunda en iyi sonuç ve eğitim sırasında en yüksek doğruluk değerini veren model tez çalışmasında kullanılmıştır.

GoogLeNet derin öğrenme mimarisi ile test veri seti üzerinden elde edilen karmaşıklık matrisi Şekil 4.9'da verilmiştir.



Şekil 4.9. GoogLeNet derin öğrenme mimarisine ait karmaşıklık matrisi

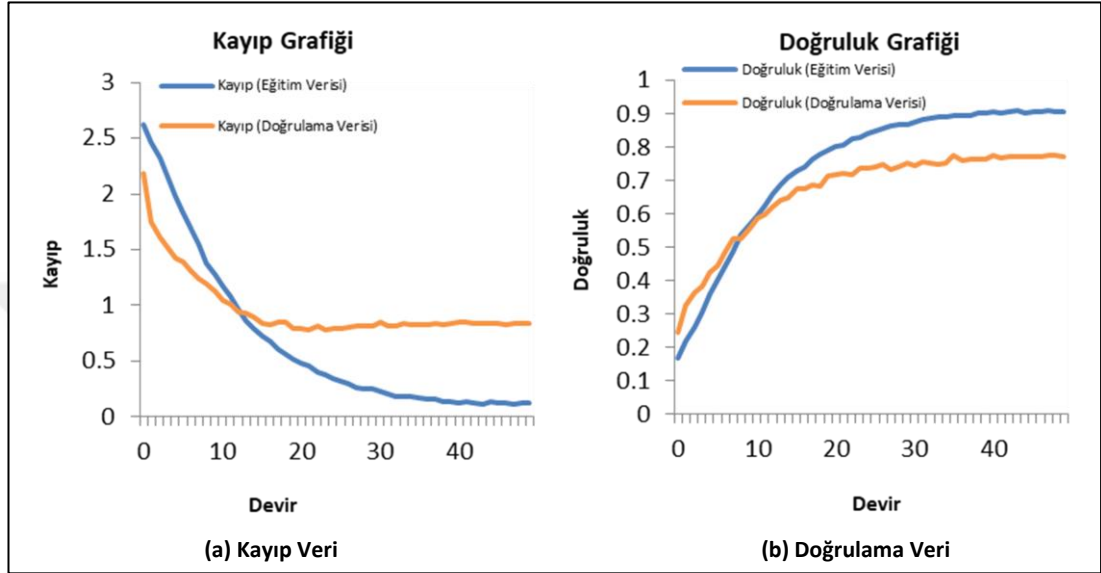
Şekil 4.9’da verilen karmaşıklık matrisi incelendiğinde GoogLeNet derin öğrenme mimarisinin en başarılı yüz modeli olarak Oblong (Köşeli yüz tipi) sınıfını, en başarısız yüz modeli olarak ise Oval (Oval Yüz Tipi) sınıfını sınıflandırdığı görülmektedir. Modelin test veri seti üzerinde gerçekleştirmiş olduğu tahminler kullanılarak derin öğrenme mimarisinin doğruluk, kesinlik, duyarlılık ve F1 skor performans değerlendirme metriklerine göre değerlendirilerek elde edilen sonuçlar Çizelge 4.3’te verilmiştir.

Çizelge 4.3. GoogLeNet derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları

GoogLeNet	Doğruluk (Test)	Kesinlik	Duyarlılık	F1 skor
	0.695	0.754	0.745	0.749

Çizelge 4.3 incelendiğinde GoogLeNet derin öğrenme mimarisinin test veri seti üzerinde en yüksek değer 0.695 olduğu tespit edilmiştir. GoogLeNet derin öğrenme mimarisinin test veri seti üzerinde ise 0.695 doğruluk değeri, 0.754 kesinlik, 0.745 duyarlılık ve 0.749 F1 skor değerleri elde edilmiştir.

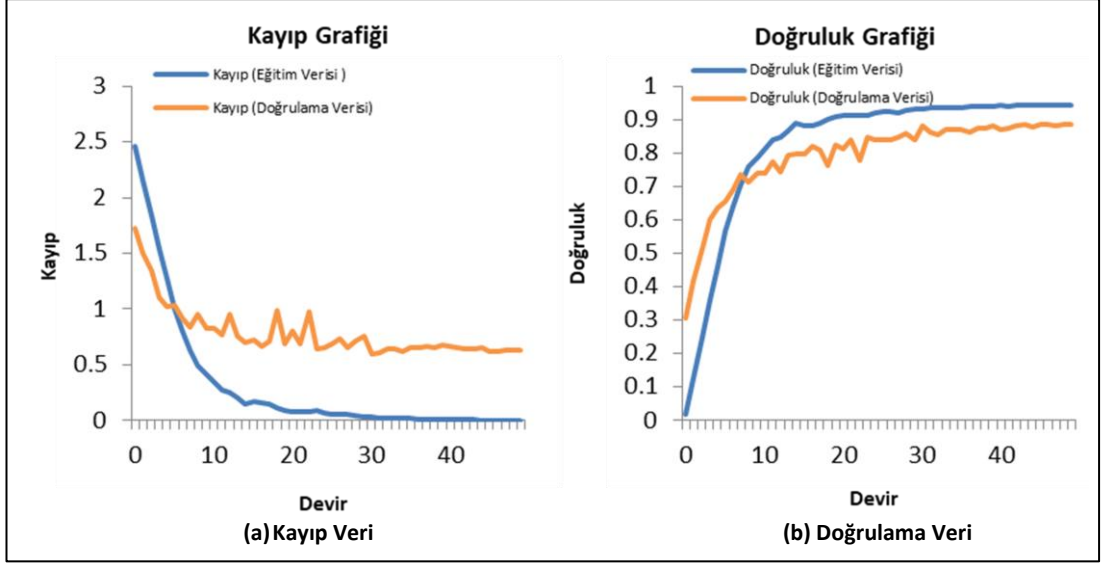
Tez çalışmasında dördüncü derin öğrenme mimarisi olarak DenseNet201 derin öğrenme mimarisi kullanılmıştır. Veri artırma öncesi DenseNet201 derin öğrenme mimarisi eğitim veri seti ile 50 devir boyunca eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.10'da verilmiştir.



Şekil 4.10. Veri artırma öncesi DenseNet201 derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri

Şekil 4.10 incelendiğinde DenseNet201 derin öğrenme modelinin veri artırma işlemi öncesi, eğitimi esnasında en düşük kayıp değeri olarak 0.7838 (0.7241 doğruluk değeri ile), en yüksek doğruluk değeri olarak ise 0.7745 (0.8365 kayıp değeri ile) değeri tespit edilmiştir. DenseNet201 derin öğrenme modeli 50 devirlik eğitimin sonunda 0.8233 kayıp değeri ile 0.7692 doğruluk değeri elde edilmiştir.

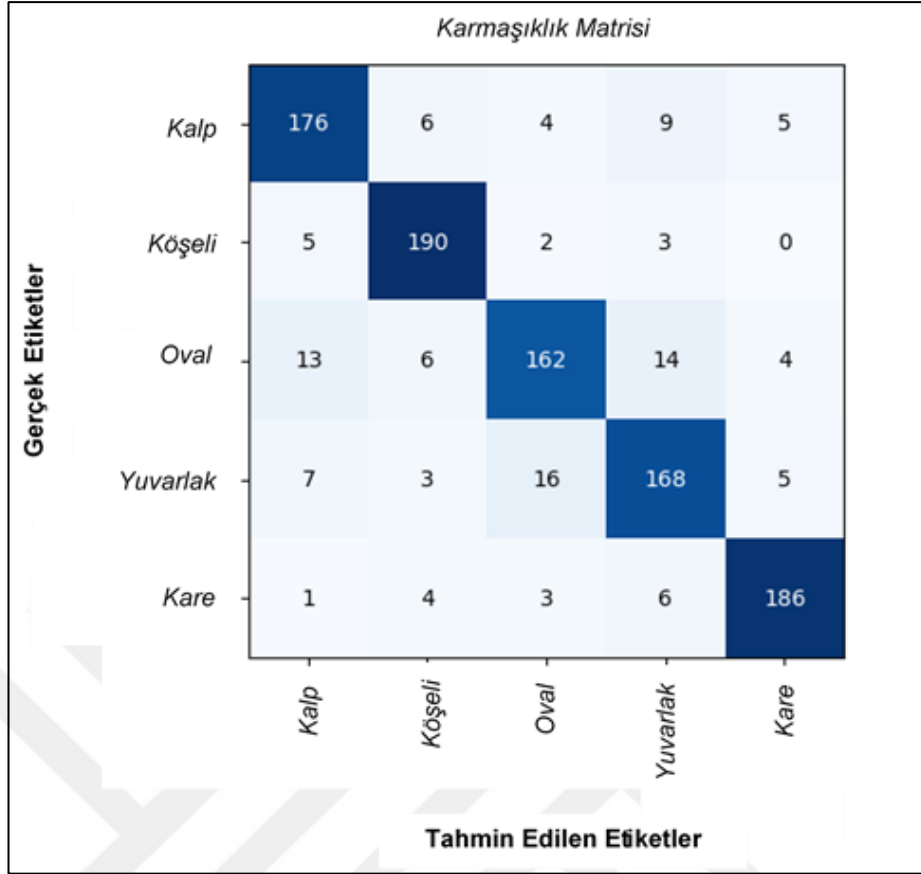
DenseNet201 derin öğrenme mimarisi, veri ön işleme yöntemiyle veri artırma işlemi yapılarak elde edilen eğitim veri seti ile 50 devir boyunca eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.11'de verilmiştir.



Şekil 4.11. Veri artırma sonrası DenseNet201 derin öğrenme mimarisinin eğitimi sonucunda elde edilen kayıp ve doğruluk grafikleri

Şekil 4.11 incelendiğinde DenseNet201 derin öğrenme modelinin eğitimi esnasında en düşük kayıp değeri olarak 0.5972 (0.8807 doğruluk değeri ile), en yüksek doğruluk değeri olarak ise 0.8857 (0.6245 kayıp değeri ile) değeri tespit edilmiştir. DenseNet201 derin öğrenme modeli 50 devirlik eğitimin sonunda 0.6308 kayıp değeri ile 0.8837 doğruluk değeri elde edilmiştir. DenseNet201 derin öğrenme modelinin eğitiminden elde edilen sonuçlar test veri seti üzerinde yapılan değerlendirme sonucunda en iyi sonuç ve eğitim sırasında en yüksek doğruluk değerini veren model tez çalışmasında kullanılmıştır.

DenseNet201 derin öğrenme mimarisi ile test veri seti üzerinden elde edilen karmaşıklık matrisi Şekil 4.12'de verilmiştir.



Şekil 4.12. DenseNet201 derin öğrenme mimarisine ait karmaşıklık matrisleri

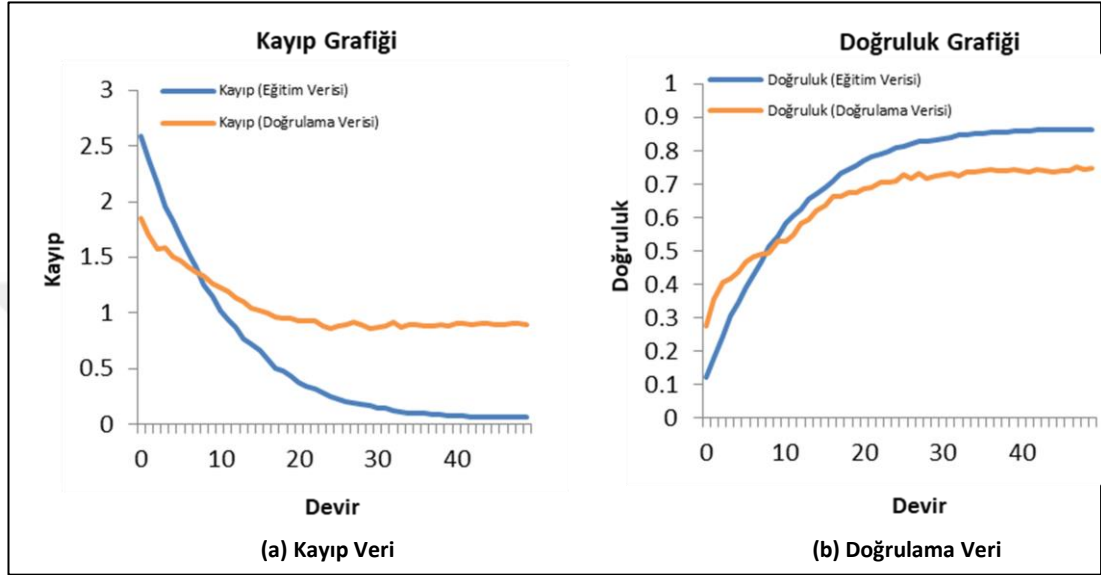
Şekil 4.12’de verilen karmaşıklık matrisi incelendiğinde DenseNet201 derin öğrenme mimarisinin en başarılı yüz modeli olarak Oblong (Köşeli yüz tipi) sınıfını, en başarısız olarak ise Oval (Oval Yüz Tipi) yüz modelini sınıflandırdığı görülmektedir. Modelin test veri seti üzerinde gerçekleştirmiş olduğu tahminler kullanılarak mimariyi değerlendirmek için doğruluk, kesinlik, duyarlılık ve F1 skor performans değerlendirme metriklerine göre değerlendirilerek elde edilen sonuçlar Çizelge 4.4’te verilmiştir.

Çizelge 4.4. DenseNet201 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor değeri sonuçları

DenseNet201	Doğruluk (Test)	Kesinlik	Duyarlılık	F1 skor
	0.884	0.896	0.913	0.905

Çizelge 4.4 incelendiğinde DenseNet201 derin öğrenme mimarisi ile doğruluk veri seti üzerinde en yüksek 0.884’lük bir doğruluk değeri olduğu tespit edilmiştir. DenseNet201 derin öğrenme mimarisinin test veri seti üzerinde ise 0.884 doğruluk değeri, 0.896 kesinlik, 0.913 duyarlılık ve 0.905 F1 skor değerleri elde edilmiştir.

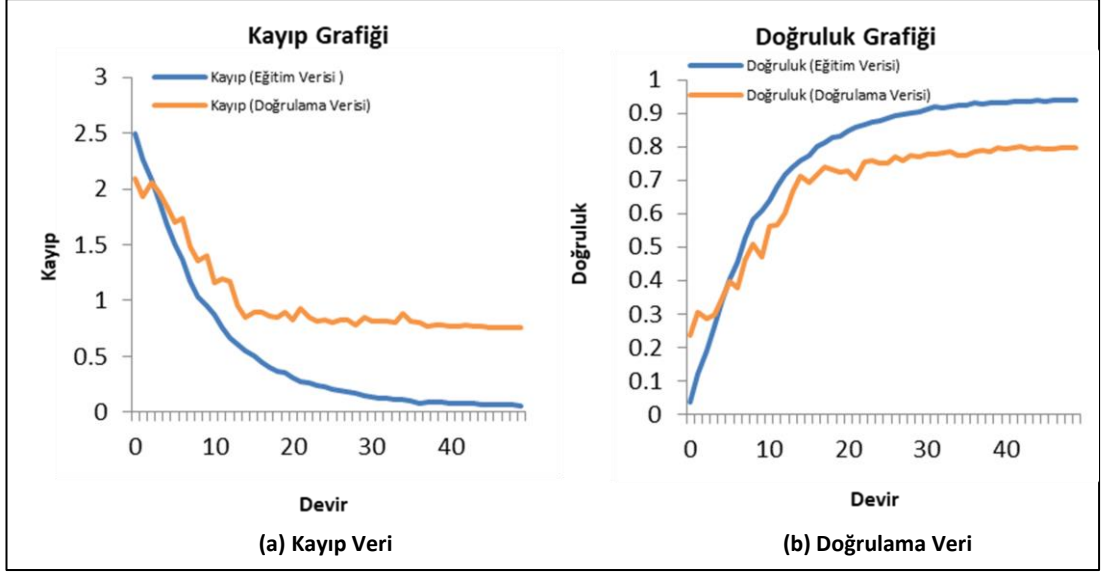
Tez çalışmasında son olarak MobileNet_V3 derin öğrenme mimarisi kullanılmıştır. MobileNet_V3 derin öğrenme mimarisi, veri artırma işlemi öncesi eğitim veri seti ile 50 devir boyunca eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.13’de verilmiştir.



Şekil 4.13. Veri artırma öncesi MobileNet_V3 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri

Şekil 4.13 incelendiğinde MobileNet_V3 derin öğrenme modelinin eğitimi esnasında en düşük kayıp değeri olarak 0.8612 (0.7234 doğruluk değeri ile) değeri, en yüksek doğruluk değeri olarak ise 0.7494 (0.8997 kayıp değeri ile) değeri tespit edilmiştir. MobileNet_V3 derin öğrenme modeli 50 devirlik eğitimin sonunda 0.8642 kayıp değeri ile 0.7418 doğruluk değeri elde edilmiştir.

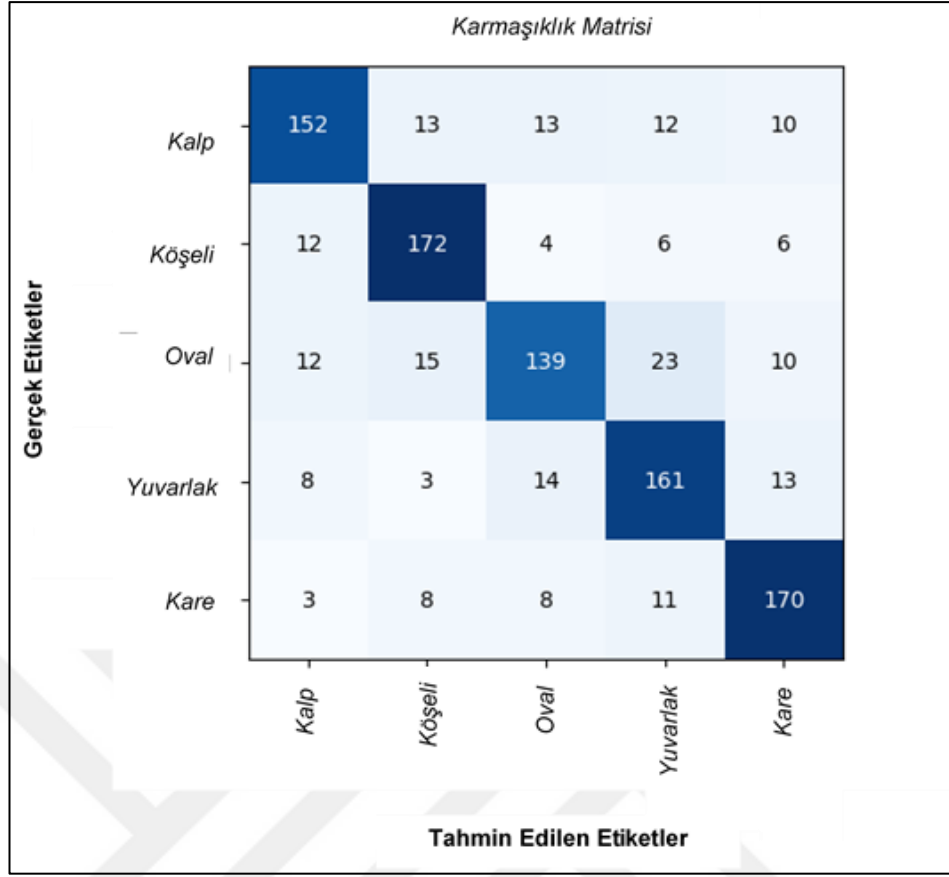
Veri ön işleme ile veri artırma işlemi yapılarak elde edilen eğitim veri seti ile 50 devir boyunca MobileNet_V3 derin öğrenme mimarisinin eğitimi sonucunda eğitim veri seti üzerinden elde edilen kayıp ve doğruluk grafikleri Şekil 4.14’te gösterilmiştir.



Şekil 4.14. Veri artırma sonrası MobileNet_V3 mimarisinin eğitimi sonucunda elde edilen (a) kayıp ve (b) doğruluk grafikleri

Şekil 4.14 incelendiğinde MobileNet_V3 derin öğrenme modelinin eğitimi esnasında en düşük kayıp değeri olarak 0.7558 (0.7955 doğruluk değeri ile) değeri, en yüksek doğruluk değeri olarak ise 0.7985 (0.7759 kayıp değeri ile) değeri tespit edilmiştir. MobileNet_V3 derin öğrenme modeli 50 devirlik eğitimin sonunda 0.7601 kayıp değeri ile 0.7955 doğruluk değeri elde edilmiştir. MobileNet_V3 derin öğrenme modelinden elde edilen sonuçlar test veri seti üzerinde yapılan değerlendirme sonucunda en iyi sonuç ve eğitim sırasında en yüksek doğruluk değerini veren model tez çalışmasında kullanılmıştır.

MobileNet_V3 derin öğrenme mimarisi ile test veri seti üzerinden elde edilen karmaşıklık matrisi Şekil 4.15'te verilmiştir.



Şekil 4.15. MobileNet_V3 derin öğrenme mimarisine ait karmaşıklık matrisi

Şekil 4.15'te verilen karmaşıklık matrisi incelendiğinde MobileNet_V3 derin öğrenme mimarisinin en başarılı yüz modeli olarak Oblong (Köşeli yüz tipi) sınıfını, en başarısız sınıflandırma olarak ise Oval (Oval Yüz Tipi) yüz modelini sınıflandırdığı görülmektedir. Modelin test veri seti üzerinde gerçekleştirmiş olduğu tahminler kullanılarak derin öğrenme mimarisi doğruluk, kesinlik, duyarlılık ve F1 skor performans değerlendirme metriklerine göre değerlendirilerek elde edilen sonuçlar Çizelge 4.5'te verilmiştir.

Çizelge 4.5. MobileNet_V3 derin öğrenme mimarisine ait doğruluk, kesinlik, duyarlılık ve F1 skor sonuçları

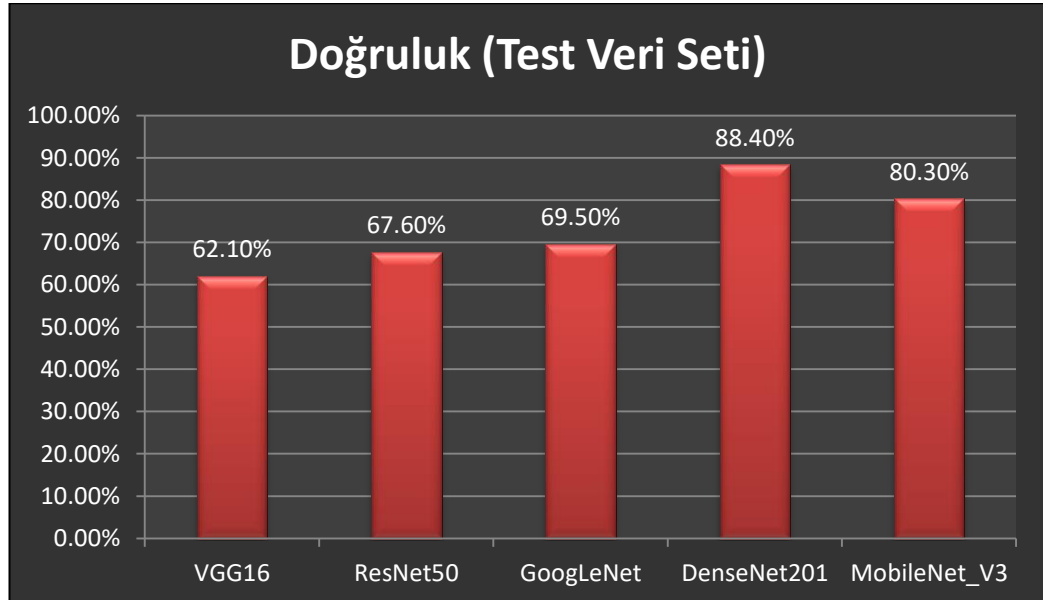
MobileNet_V3	Doğruluk (Test)	Kesinlik	Duyarlılık	F1 skor
	0.803	0.842	0.820	0.831

Çizelge 4.5 incelendiğinde MobileNet_V3 derin öğrenme mimarisi ile doğruluk veri seti üzerinde en yüksek 0.803'lük bir doğruluk değeri elde edilmiştir. MobileNet_V3 derin öğrenme mimarisinin test veri seti üzerinde ise 0.803 doğruluk değeri, 0.842 kesinlik, 0.820 duyarlılık ve 0.831 F1 skor değerleri elde edilmiştir.

4.2. Derin Öğrenme Mimarilerin Karşılaştırılması

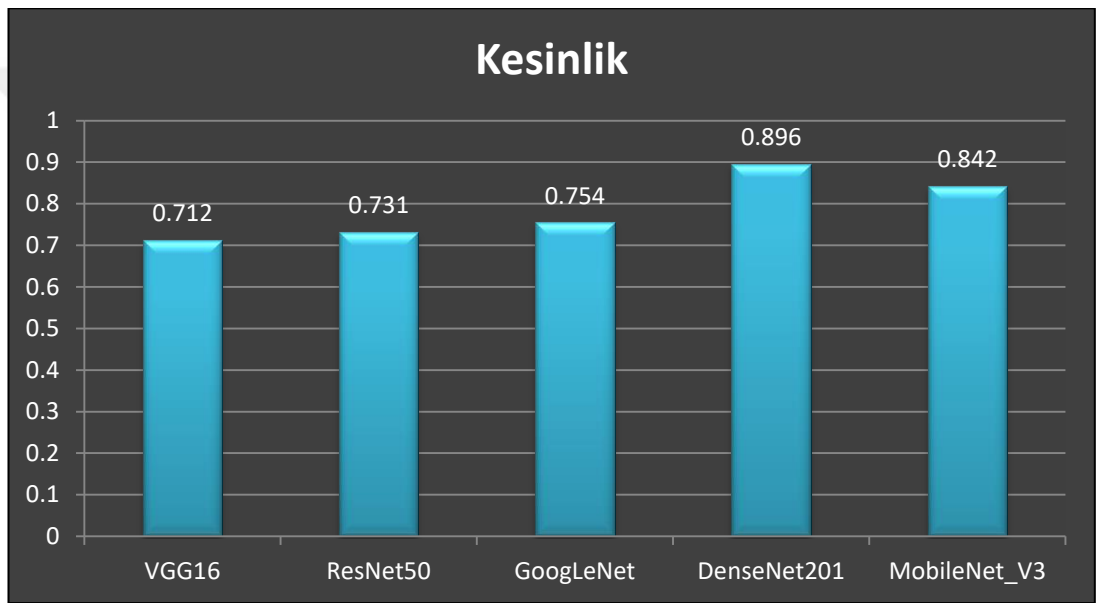
Çalışmada kullanılan VGG16, ResNet50, GoogLeNet, DenseNet201 ve MobileNet_v3 derin öğrenme mimarilerin test veri seti üzerinde değerlendirilmesi sonucunda VGG16 %58.91, ResNet50 %63.32, GoogLeNet %64.37, DenseNet201 %77.45 ve MobileNet_v3 mimarisi ile %74.94 doğruluk değerleri elde edilmiştir. Tüm derin öğrenme mimarilerinin doğruluk oranları %80'in altında olduğu için veri ön işleme yöntemi kullanılarak veri artırma işlemi uygulanmıştır.

Veri artırma işlemi ile VGG16, ResNet50, GoogLeNet, DenseNet201 ve MobileNet_v3 derin öğrenme mimarilerin, test veri seti üzerinde tekrar değerlendirilmesi sonucunda VGG16 %62.10, ResNet50 %67.6, GoogLeNet %69.5, DenseNet201 %88.40 ve MobileNet_v3 mimarisi ile %80.30 doğruluk değerleri elde edilmiştir. Çalışmada kullanılan tüm derin öğrenme mimarileri karşılaştırıldığında en yüksek doğruluk oranı elde eden modelin %88.40 ile DenseNet201 olduğu, en düşük doğruluk oranının ise %62.10 ile VGG16 ile elde edildiği görülmektedir. Tüm mimarilere ait doğruluk grafikleri Şekil 4.16'da görülmektedir.



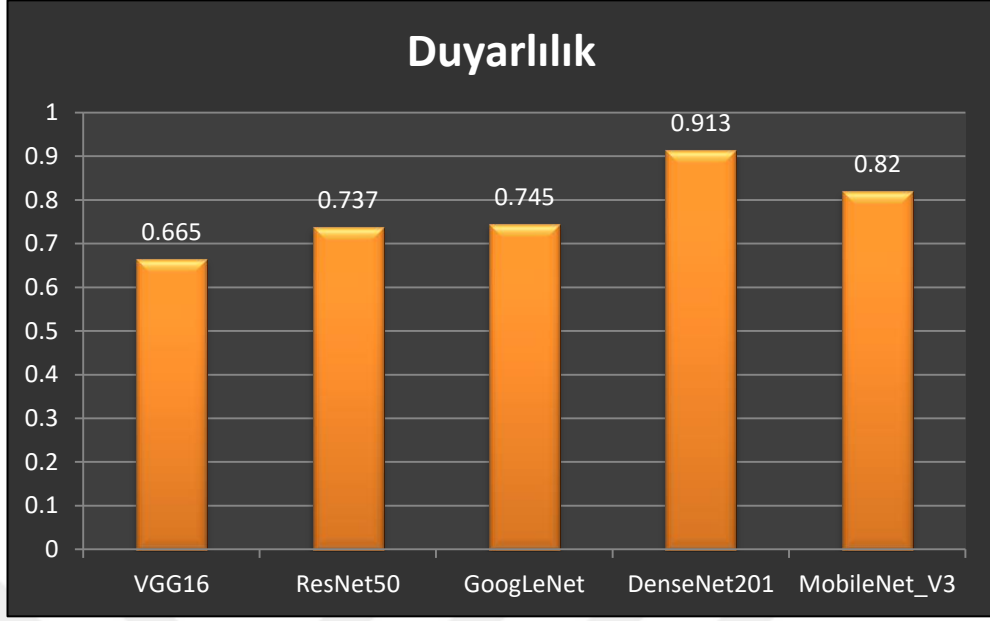
Şekil 4.16. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait doğruluk değerlerinin karşılaştırılması

Derin öğrenme mimarilerin test veri seti üzerinde kesinlik performans değerlendirme metriğine göre VGG16 derin öğrenme mimarisinde 0.712, ResNet50 derin öğrenme mimarisinde 0.731, GoogLeNet derin öğrenme mimarisinde 0.754, DenseNet201 derin öğrenme mimarisinde 0.896, MobileNet_v3 derin öğrenme mimarisinde ise 0.842 değerleri elde edilmiştir. Kesinlik performans değerlendirme metriğine göre ise en başarılı mimari 0.896 ile DenseNet201 mimarisi iken en başarısız derin öğrenme modeli ise 0.712 değeri ile VGG16 derin öğrenme mimarisinin olduğu tespit edilmiştir. Tez çalışmasında kullanılan tüm derin öğrenme mimarileri için kesinlik değerlerine ait grafik Şekil 4.17’de gösterilmiştir.



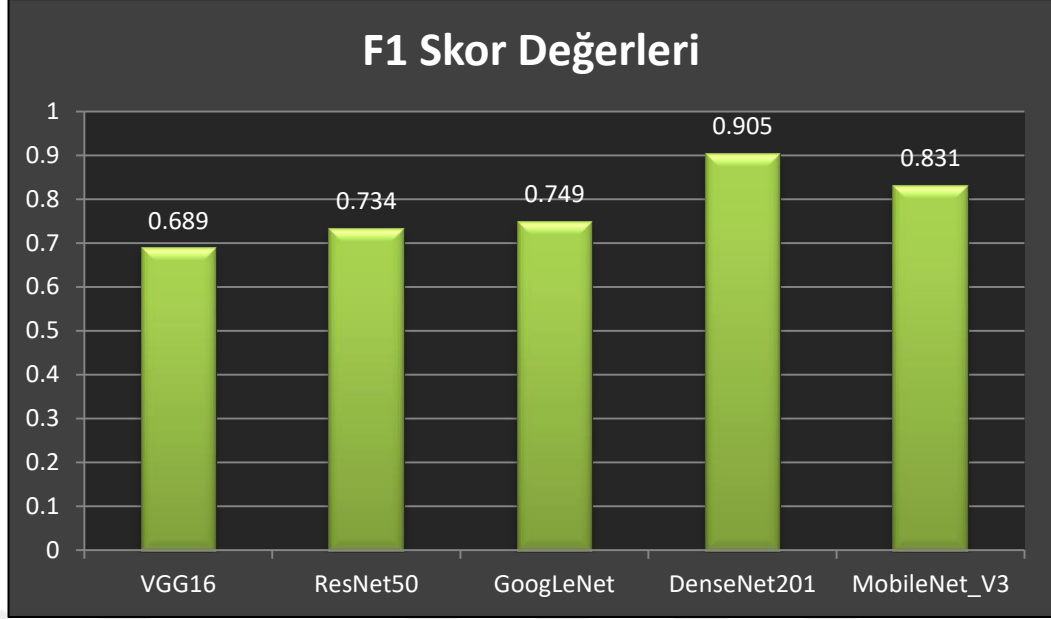
Şekil 4.17. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait kesinlik değerlerinin karşılaştırılması

Tez çalışmasında kullanılan derin öğrenme mimarileri test veri seti üzerinde duyarlılık performans değerlendirme metriğine göre VGG16 derin öğrenme mimarisinde 0.665, ResNet50 derin öğrenme mimarisinde 0.737, GoogLeNet derin öğrenme mimarisinde 0.745, DenseNet201 derin öğrenme mimarisinde 0.913, MobileNet_v3 derin öğrenme mimarisinde ise 0.820 değerleri elde edilmiştir. Duyarlılık performans değerlendirme metriğine göre en başarılı mimari 0.913 değeri ile DenseNet201 derin öğrenme mimarisi iken en başarısız derin öğrenme modeli ise 0.665 ile VGG16 derin öğrenme mimarisinin olduğu tespit edilmiştir. Tez çalışmasında kullanılan tüm derin öğrenme mimarileri için duyarlılık değerlerine ait grafik Şekil 4.18’te gösterilmiştir.



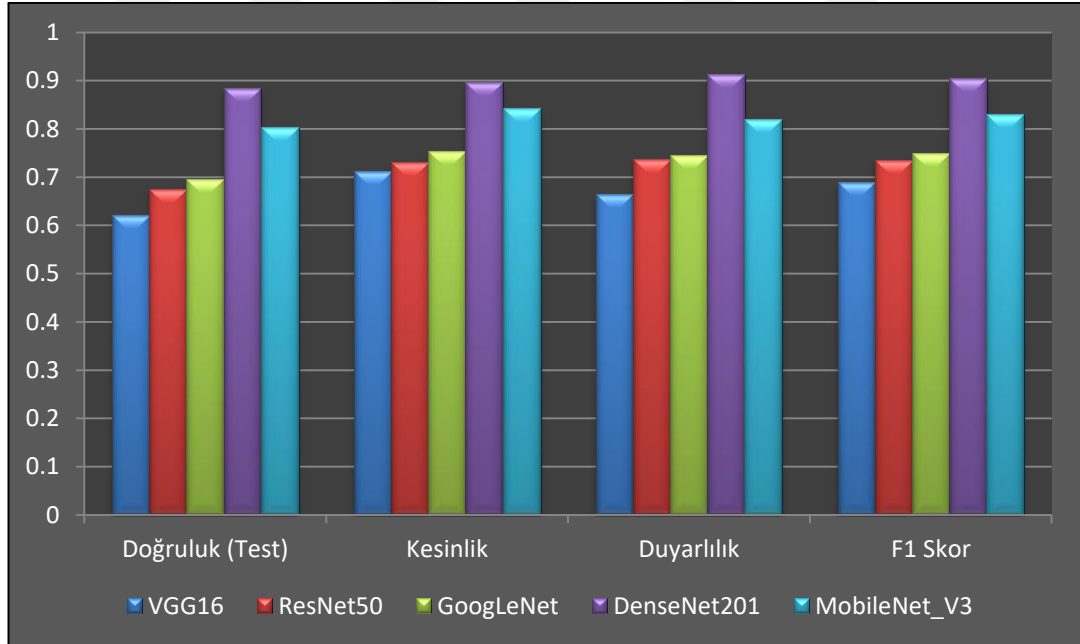
Şekil 4.18. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait duyarlılık değerlerinin karşılaştırılması

Tez çalışmasında kullanılan derin öğrenme mimarileri test veri seti üzerinde F1-Skor performans değerlendirme metriğine göre VGG16 derin öğrenme mimarisinde 0.689, ResNet50 derin öğrenme mimarisinde 0.734, GoogLeNet derin öğrenme mimarisinde 0.749, DenseNet201 derin öğrenme mimarisinde 0.905, MobileNet_v3 derin öğrenme mimarisinde ise 0.831 değerleri elde edilmiştir. F1-Skor performans değerlendirme metriğine göre en başarılı mimari 0.905 değeri ile DenseNet201 derin öğrenme mimarisi iken en başarısız model 0.689 değeri ile VGG16 derin öğrenme modelinin olduğu tespit edilmiştir. Tez çalışmasında kullanılan tüm derin öğrenme mimarileri için F1-Skor değerlerine ait grafik Şekil 4.19’da gösterilmiştir.



Şekil 4.19. Tez çalışmasında kullanılan derin öğrenme mimarilerine ait F1 skor değerlerinin karşılaştırılması

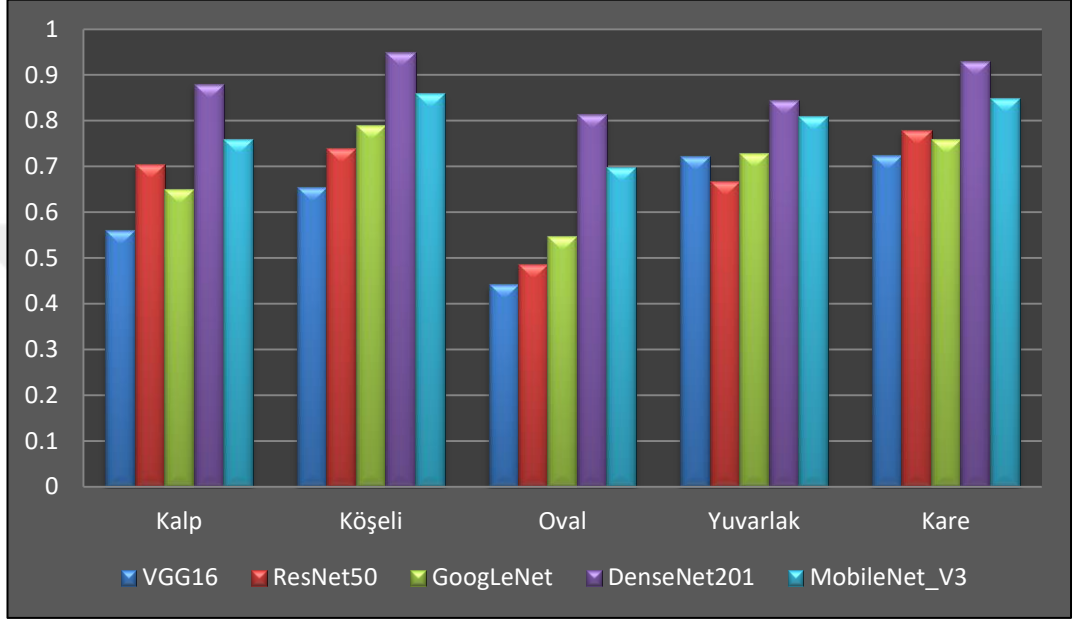
Tez çalışmasında kullanılan tüm derin öğrenme mimarilerin doğruluk, duyarlılık, kesinlik ve F1 skor karşılaştırmaları Şekil 4.20’de gösterilmiştir.



Şekil 4.20. Tez çalışmasında kullanılan derin öğrenme mimarilerinin karşılaştırılması

Tez çalışmasında kullanılan tüm derin öğrenme mimarilerinin performans değerlendirme metriklerine göre değerlendirme sonuçları incelendiğinde tez kapsamındaki veri seti ile en uyumlu ve en doğru sonucu veren derin öğrenme modelinin DenseNet201 derin öğrenme mimarisinin olduğu tespit edilmiştir.

Tez çalışmasında kullanılan tüm derin öğrenme mimarilerin test veri seti üzerinde “Kalp Yüz Tipi”, “Köşeli Yüz Tipi”, “Oval Yüz Tipi”, “Yuvarlak Yüz Tipi”, “Kare Yüz Tipi” sınıflarına ait başarılarını analiz etmek için sınıf bazında F1 skoru değerleri de incelemiştir. Tez çalışmasında kullanılan derin öğrenme mimarilerin sınıflara göre F1 skoru karşılaştırması Şekil 4.21’de gösterilmiştir.



Şekil 4.21. Tez çalışmasında kullanılan derin öğrenme mimarilerinin sınıflara göre F1 skorları

Şekil 4.21 incelendiğinde tüm derin öğrenme mimarilerin kare yüz tipini sınıflandırmada oldukça başarılı oldukları görülmektedir. Benzer bir şekilde tez çalışması kapsamında kullanılan tüm derin öğrenme mimarilerin oval yüz tipi sınıfını tanımlamada ise zorlandığı belirlenmiştir. Oval yüz tipi sınıflamasında, tez çalışması kapsamında kullanılan tüm derin öğrenme mimarileri arasında yine DenseNet201 mimarisinin diğer mimarilerden çok daha iyi bir performans sergilediği açıkça görülmektedir.

Karmaşıklık matrisleri incelendiğinde ise yuvarlak yüz tipi sınıfının çoğunlukla oval yüz tipi sınıfıyla, köşeli yüz tipi sınıfında kare yüz tipi sınıfı ile birbirlerine benzerlik göstermesinden dolayı derin öğrenme mimarilerinin eğitiminde birbirleriyle karıştırıldığı gözlemlenmiştir. Bunun nedeni ise, oval yüz tipi ve yuvarlak yüz tipi sınıflarında bulunan görüntüler ile köşeli yüz tipi ve kare yüz tipi

sınıflarındaki görüntülerin birbirlerine büyük ölçüde benzerlik göstermesinin neden olabileceği düşünülmektedir.

Tez çalışmasında, VGG16, ResNet50, GoogLeNet, DenseNet201 ve MobileNet_V3 derin öğrenme mimarileri kullanılarak yüz tipi belirleme işlemi gerçekleştirilmiştir. Derin öğrenme mimarileri transfer öğrenme yöntem ile eğitim ve doğrulama veri setleri kullanılarak eğitilmiştir. Eğitilmiş olan tüm mimarilerin doğruluk değerleri incelenmiş ve elde edilen sonuçlar karşılaştırıldığında, insan yüz tipinin geometrik olarak tespitinde ve sınıflandırmadaki başarısı göz önüne alındığında, DenseNet201 derin öğrenme mimarisinin tez çalışması kapsamında kullanılan diğer derin öğrenme mimarilerinden daha başarılı olduğu tespit edilmiştir.



5. TARTIŞMA VE SONUÇ

Tez çalışmasında, mobil cihazlar ve nesnelerin interneti teknolojileri kullanılarak derin öğrenme algoritmaları ile insan yüz şeklinin belirlenmesi ve belirlenen yüz şekline uygun gözlük tavsiyesi sunan bir sistem tasarlanmıştır. Geliştirilen sistem, android cihazların fotoğraf çekme özelliği veya galerisinden seçilen görüntüleri kullanarak yüz tipini belirlemekte ve buna uygun gözlük tavsiyesinde bulunmaktadır.

Nesne tanıma, görüntü işleme, örüntü tanıma ve görüntü işlemlerine bağlı çoklu sınıflandırma problemlerinde başarılı sonuçlar veren, VGG16, ResNet50, GoogleNet, DenseNet201 ve MobileNet_V3 derin öğrenme mimarileri, eğitim veri seti üzerinden eğitilmiş ve test veri seti ile değerlendirilmiştir. İlk etapta, yapılan değerlendirme sonucunda tüm derin öğrenme mimarilerinin doğruluk oranları %80'in altında olduğu tespit edilmiştir. Veri ön işleme yöntemi ile görüntü verileri artırılarak veri setine eklenmiştir. Veri arttırımı gerçekleştirildikten sonra, görüntüler tekrar derin öğrenme mimarileri eğitime tabi tutulmuş ve test edilmiştir. Ayrıca tüm derin öğrenme mimarileri doğruluk, duyarlılık, kesinlik ve F1-skor performans değerlendirme metrikleri ile değerlendirilmiştir.

Elde edilen sonuçlar incelendiğinde, DenseNet201 derin öğrenme mimarisi, %88.40 doğruluk oranı ile yüz tipi sınıflandırmasında en yüksek doğruluk oranı ile en başarılı mimari olarak tespit edilmiştir. DenseNet201 derin öğrenme mimarisi, 0.896 kesinlik, 0.913 duyarlılık ve 0.905 F1 skor değerleri ile diğer mimarilerin performans metriklerine göre üstün sonuçlar elde etmiştir. DenseNet201 derin öğrenme mimarisi, özellikle kare ve köşeli yüz tiplerinin sınıflandırılmasında yüksek bir başarı göstermiştir. Bununla birlikte, tüm derin öğrenme modellerinin oval yüz tipini sınıflandırmasında zorlandığı gözlemlenmiş olup yine DenseNet201 derin öğrenme mimarisinin diğer mimarilere göre oval yüz tipini sınıflandırmasında daha başarılı olduğu tespit edilmiştir.

Çalışmanın yazılım kısmında ise, android cihazın fotoğraf çekme özelliği kullanılarak elde edilen görüntü ya da android cihazın galerisinden seçilen görüntü, görüntü ön işleme yöntemlerine tabi tutulmuştur. Görüntü ön işleme sonrasında iyileştirilen görüntü, derin öğrenme yöntemi ile bilimsel açıdan geometrik olarak beş

yüz tipinden hangisine uygun olduğuna karar vermesi ve kişiye belirlenen yüz tipine en uygun gözlük tavsiyesinde bulunması, çalışmanın özgün yönü olarak düşünülmüştür. Ayrıca tez kapsamında geliştirilen kompleks sistem ile kişinin hangi yüz tipine sahip olduğunu ve bu yüz tipine ait başlıca özellikleri, android cihaz üzerinden kullanıcıya aktarılması tez çalışmanın bir diğer yaygın etkisidir. Ayrıca, yapay zekâ, nesne tanıma, derin öğrenme algoritmalarının, çevrimiçi çalışan cihazlarda ve mobil platformlarda çalışabilmesini sağlayan disiplinler arası bir çalışma olması da önemli yaygın etkilerinden birisi olduğu düşünülmektedir.

Çalışma, mobil destekli derin öğrenme sistemlerinin yüz tipi belirleme ve buna uygun gözlük tavsiyelerinde etkili bir şekilde kullanılabilmesini göstermektedir. Sistemin mobil cihazlar üzerinden kolayca erişilebilir olması, kullanıcıların hem zamandan hem de mekândan tasarruf etmelerini sağlamaktadır. Ayrıca, geliştirilmiş olan yazılımların, derin öğrenme algoritmalarının, mobil cihazların ve çevrimiçi çalışan cihazların koordineli bir şekilde çalışma olması da literatüre ayrı bir katkı sağlayacağı düşünülmektedir.

Kişilerin geometrik olarak yüz tipinin belirlenmesi oldukça zor bir işlemdir. Özellikle “Kare Yüz Tipi” ve “Köşeli Yüz Tipi” gibi yüz tiplerinin belirlenmesi derin öğrenme mimarilerin başarı oranlarını azaltmaktadır. Mimarilerin başarı oranlarını arttırmak için daha çok görüntülere sahip veri setleri kullanılmalıdır. Ayrıca tüm mimarilerin “Kare Yüz Tipi” ile “Köşeli Yüz Tipi” ve “Oval Yüz Tipi” ile “Yuvarlak Yüz Tipi” görüntülerinin karıştırılarak sınıflandırmada zorluk çektiği tespit edilmiştir. Bu yüz tiplerine ait görüntülerin artırılarak, oluşturulacak yeni veri setleri ile eğitilecek olan derin öğrenme mimarilerinin optimizasyonu sağlanarak elde edilecek genel başarı oranını artırılabilmesi düşünülmektedir. Ayrıca uygulamanın gerçek hayatta kullanımının artması ile yeni yüz görüntülerinin veri tabanına eklenerek, görüntü verilerinin artması ile oluşturulacak yeni ve zengin bir veri seti ile derin öğrenme modellerinin tahmin başarısının arttırılabileceği de öngörülmektedir.

Sonuç olarak tasarlanan sistem, mobil destekli derin öğrenme ve nesnelerin interneti entegrasyonunun, günlük yaşamda pratik ve faydalı uygulamalar sunabileceğini göstermektedir. Yüz tipi belirleme ve buna uygun gözlük tavsiyesi sistemi, hem kullanıcılar hem de optik sektöründeki profesyonel gözlük firmaları için değerli bir

yazılım olarak hizmet verebileceđi öngörülmektedir. Bu alanda yapılacak ileri çalışmalar, teknolojinin sunduđu imkânlarla daha gelişmiş ve kullanıcı dostu sistemlerin geliştirilmesine katkı sağlayacaktır.



KAYNAKLAR

- Aksoy, B., & Salman, O. K. M. (2021). Detection of covid-19 disease in chest x-ray images with capsul networks: application with cloud computing. *Journal of Experimental & Theoretical Artificial Intelligence*, 33(3), 527-541.
- Al-Haija, Q. A. & Adebajo, A. (2020). Breast Cancer Diagnosis in Histopathological Images Using ResNet-50 Convolutional Neural Network. *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. April 21-24, Toronto, pp. 1-7.
- Al-Kababji, A., Shidqi, L., Boukhenoufa, I., Amira, A., Bensaali, F., Gastli, M. S., & Abdalla, A. (2019, August). IoT-Based Fall and ECG Monitoring System: Wireless Communication System Based Firebase Realtime Database. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)* (pp. 1480-1485). IEEE.
- Altuntaş, E. (1998). *İşletme Yönetiminde Yapay Zeka* (Yüksek Lisans Tezi, Anadolu Üniversitesi Sosyal Bilimler Enstitüsü).
- Anonim (2024a). <https://www.kaggle.com/datasets/> (Son erişim tarihi: 06 Mayıs 2024)
- Anonim (2024b). <https://developer.android.com/guide/platform?hl=tr> (Son erişim tarihi: 06 Mayıs 2024)
- Anonim (2024c). <https://developer.android.com/reference/android/app/Activity.html> (Son erişim tarihi: 06 Mayıs 2024)
- Anonim (2024d). <https://developer.android.com/develop/background-work/services?hl=tr> (Son erişim tarihi: 06 Mayıs 2024)
- Anonim (2024e). <https://firebase.google.com/> (Son erişim tarihi: 06 Mayıs 2024)
- Anonim (2024f). <https://firebase.google.com/products/auth> (Son erişim tarihi: 06 Mayıs 2024)
- Anonim (2024g). <https://firebase.google.com/docs/cloud-messaging?hl=tr> (Son erişim tarihi: 06 Mayıs 2024)
- Anonim (2024h). <https://www.kaggle.com/datasets/zeyadkhalid/faceshape-processed> (Son erişim tarihi: 06 Mayıs 2024)
- Arıç, E. & Turan, M. (2020). Video duygu analizi. *European Journal Of Science And Technology*, 59–67. <https://doi.org/10.31590/ejosat.779059>

- Ballester, P., & Araujo, R. (2016). On the performance of googlenet and alexnet applied to sketches. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).
- Bal, Y., & Bal, M. (2021). Proaktif insan kaynakları yönetiminin yeni gücü: İK analitiği ve yapay zekâ. *Business & Management Studies: An International Journal*, 9(3), 1198-1216.
- Bayrakdar, S., Akgün, D., & Yücedağ, İ. “Yüz ifadelerinin otomatik analizi üzerine bir literatür çalışması”. *Sakarya University Journal of Science*, SAÜ Fen Bil Der 20. Cilt, 2. Sayı, s. 383-398, 2016.
- Bhargavi, K. & Babu, B. S. (2021). Application of Convolved Neural Network and Its Architectures for Fungal Plant Disease Detection. *Artificial Intelligence and IoT-Based Technologies for Sustainable Farming and Smart Agriculture*, 314-324.
- Bush V., The Study Of The Geometric Face Shape, *How to Create the Perfect Eyebrow*, 1st Edition, Canada: Delmar Cengage Learning, 2003.
- Caliskan, A. ve Tasdemir, S., (2018), A practical approach to android mobile application security, *International Journal of Applied Mathematics Electronics and Computers*, 6 (4), 53-58.
- Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colab as a tool for accelerating deep learning applications. *Ieee Access*, 6, 61677-61685.
- Chen X. ve Cheng W. (2015), “Facial expression recognition based on edge detection”, *International Journal of Computer Science & Engineering Survey (IJCSES)*, 6(2), 1- 9.
- Çalışkan, A. (2019). *Android Akıllı Telefon Ve Tabletlerde Kullanıcı Aktivitelerini İzleme*. (Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü)
- Çetiner, M. (2020). *Makine Öğrenmesi Yöntemleri İle Yazılım Hata Tahmini*. (Yüksek Lisans Tezi, İstanbul Kültür Üniversitesi Lisansüstü Eğitim Enstitüsü)
- Ekman, P. & Friesen, W. V. (1986). A new pan-cultural facial expression of emotion. *Motivation and emotion*, 10(2), 159-168.
- Ekman P. ve Friesen W. V. (1971), “Constants across cultures in the face and emotion”, *Journal of Personality and Social Psychology*, 17(2), 124-129.
- Eldem, A., Eldem, H., & Palalı, A. (2017). Görüntü işleme teknikleriyle yüz algılama sistemi geliştirme. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 6(2), 44-48.

- Eusebi, P. (2013). Diagnostic accuracy measures. *Cerebrovascular Diseases*, 36(4), 267-272.
- Farahani F. S., Sheikhan M. ve Farrokhi A. (2013). "A fuzzy approach for facial emotion recognition", *13th Iranian Conference on Fuzzy Systems(IFS)*, 1-4. <https://doi.org/10.1109/IFSC.2013.6675597>.
- Farboudi, S. (2009). *Tıp Bilişiminde İstatistiksel Veri Madenciliği* (Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü).
- Feng X., Pietikäinen M. Ve Hadid A. (2005), "Facial expression recognition based on local binary patterns and linear programming", *Pattern Recognition and Image Analysis*, 15(2), 546-548.
- Güler, T. (2022). *Gerçek Zamanlı Duygu Durumu Analizi: Derin Öğrenme Tabanlı Akıllı Sistem Tasarımı* (Yüksek Lisans Tezi, Bartın Üniversitesi Fen Bilimleri Enstitüsü).
- He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 27-30, Las Vegas, pp. 770-778.
- Hu, H., Dey, D., Del Giorno, A., Hebert, M., & Bagnell, J. A. (2017). Log-densenet: how to sparsify a densenet. Arxiv. <https://arxiv.org/abs/1711.00002> (Son erişim tarihi: 6 Mayıs 2024).
- Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 21-26, Honolulu, pp. 4700-4708.
- Kazmi S. B., Qurat-ul-Ain ve Jaffar M. A. (2012), "Wavelets-based facial expression recognition using a bank of support vector machines", *Soft Computing*, 16(3), 369-379.
- Khasoggi, B., Ermatita, E., & Sahmin, S. (2019). Efficient mobilenet architecture as image recognition on mobile and embedded devices. *Indonesian Journal of Electrical Engineering and Computer Science*, 16(1), 389-394.
- Kosmatos, E.A., Tselikas, N.D. and Boucouvalas, A.C. (2011) Integrating RFIDs and smart ob-jects into a unified internet of things architecture. *Advances in Internet of Things: Scientific Research*, 1, 5-12. <http://dx.doi.org/10.4236/ait.2011.11002>
- Kurtboğan, H. (2023). *Yeni Dünyada Yapay Zeka Metaforu Ve Yapay Zekanın Çalışan Performansına Etkisi* (Yüksek Lisans Tezi, Karamanoğlu Mehmetbey Üniversitesi Sosyal Bilimler Enstitüsü).

- Lamba, P. S. & Virmani, D. (2021). Dcnn-Based Facial Expression Recognition Using Transfer Learning. In *Proceedings of International Conference on Artificial Intelligence and Applications* (pp. 509-520). https://doi.org/10.1007/978-981-15-4992-2_48
- Liao, Q., & Poggio, T. (2016). Bridging the gaps between residual learning, recurrent neural networks and visual cortex. Arxiv. <https://arxiv.org/abs/1604.03640> (Son erişim tarihi: 6 Mayıs 2024).
- Liu, M., & Zhu, M. (2018). Mobile video object detection with temporally-aware feature maps. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5686-5695).
- Meier, R., 2013, *Profesyoneller için Android ile Uygulama Geliştirme*, 3, İstanbul, Pusula Yayıncılık, p. 4-5, 246.
- Mufid, M. R., Basofi, A., Al Rasyid, M. U. H., & Rochimansyah, I. F. (2019, September). Design An Mvc Model Using Python for Flask Framework Development. In *2019 International Electronics Symposium (IES)* (pp. 214-219). IEEE.
- Narman, A. E., 2013, *Android Programlama*, 3, İstanbul, Kodlab Yayın Dağıtım Yazılım ve Eğitim Hizmetleri San. ve Tic. Ltd. Şti., p. 7.
- Nunberg, G. (2012). The Advent of the Internet. Berkeley. <https://courses.ischool.berkeley.edu/i103/f07/slides/HofI12-10Internet.pdf> (Son erişim tarihi: 06 Mayıs 2024)
- Oliveira L., Mansano M., Koerich A. ve Alceu de Souza Britto J. (2011), “2D principal component analysis for face and facial-expression recognition”, *Computing in Science & Engineering*, 13(3), 9-13.
- Ovatman, T., *Gerçek Zamanlı Bir Optik Karakter Tanıma Sistemi* (Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü).
- Peng, D., Cao, L., & Xu, W. (2011). Using JSON for data exchanging in web service applications. *Journal of Computational Information Systems*, 7(16), 5883-5890.
- Qian, S., Ning, C., & Hu, Y. (2021, March). MobileNetV3 For Image Classification. In *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)* (pp. 490-497). IEEE.
- Rao, A., Ajri, S., Guragol, A., Suresh, R. & Tripathi, S. (2020). Emotion Recognition from Facial Expressions in Children and Adults Using Deep Neural Network. In *Intelligent Systems, Technologies and Applications*. (pp. 43-51). https://doi.org/10.1007/978-981-15-3914-5_4

- Said, Y. & Barr, M. (2021). Human emotion recognition based on facial expressions via deep learning on high-resolution images. *Multimedia Tools and Applications*, 1-13. <https://doi.org/10.1007/s11042-021-10918-9>
- Selamet, F. (2023). *Derin öğrenme yöntemleri ile metalik yüzeylerde kusur tespiti ve sınıflandırılması*. (Doktora Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü)
- Silva L. C. D. ve Hui S. C., “Real-Time Facial Feature Extraction and Emotion Recognition”, *Fourth Pacific Rim Conference on Multimedia, Information, Communications and Signal Processing*, Singapore, 2003.
- Singh, M., Verma, A., Parasher, A., Chauhan, N. ve Budhiraja, G. (2019). Python flask çerçevesi kullanılarak veritabanının uygulanması. *Uluslararası Mühendislik ve Bilgisayar Bilimleri Dergisi*, 8 (12), 24890-24893.
- Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv. <https://arxiv.org/abs/1409.1556> (Son erişim tarihi: 6 Mayıs 2024).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going Deeper with Convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 7-12, Boston, pp. 1-9.
- Šimundić, A. M. (2009). Measures of diagnostic accuracy: basic definitions. *ejifcc*, 19(4), 203.
- Tsapatsoulis, N., Leonidou, M., & Kollias, S. (1998, December). Facial Expression Recognition Using HMM with Observation Dependent Transition Matrix. In *1998 IEEE Second Workshop on Multimedia Signal Processing (Cat. No. 98EX175)* (pp. 89-95). IEEE.
- Wu, C. D. & Chen, L. H. (2019). Facial emotion recognition using deep learning. Arxiv. <https://arxiv.org/abs/1910.11113> (Son erişim tarihi: 06 Mayıs 2024)
- Wu, X., Yang, T. & Xia, Z. (2020). Gait recognition based on densenet transfer learning. *International Journal of Science, Environment and Technology*, 9(1), 1-14.
- Wu, Z., Shen, C., & Van Den Hengel, A. (2019). Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern recognition*, 90, 119-133.
- Vujović, V., & Maksimović, M. (2015). Raspberry pi as a sensor web node for home automation. *Computers & Electrical Engineering*, 44, 153-171.
- Yacoob Y. ve Davis L. S. (1996), “Recognizing human facial expressions from long image sequences using optical flow”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6), 636-642.

- Yavuz, M. A. (2009). *Kaçak Su Kullanımının Tespitinde Veri Madenciliği Yaklaşımı* (Yüksek Lisans TEzi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü).
- Yeniad, M. (2006). *Uzaktan Eğitimde Kullanılmak Üzere Web Tabanlı Bir Portal Yazılımı Geliştirme* (Yüksek Lisans Tezi, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü).
- Zhang Z. (1999), “Feature-based facial expression recognition: sensitivity analysis and experiments with a multi-layer perceptron”, *International Journal of Pattern Recognition and Artificial Interlligence*, 13(6), 893-911.



ÖZGEÇMİŞ

