



**DERİN ÖĞRENME TEKNİKLERİYLE AKCİĞER
RÖNTGENİNDEN COVID-19 HASTALIĞININ
TESPİT EDİLMESİ**

Ahmet KORKMAZ

**2021
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**Tez Danışmanı
Doç. Dr. Ümit ATİLA**

**DERİN ÖĞRENME TEKNİKLERİYLE AKCİĞER RÖNTGENİNDEN
COVID-19 HASTALIĞININ TESPİT EDİLMESİ**

Ahmet KORKMAZ

**T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır**

**Tez Danışmanı
Doç. Dr. Ümit ATİLA**

**KARABÜK
Kasım 2021**

Ahmet KORKMAZ tarafından hazırlanan “DERİN ÖĞRENME TEKNİKLERİYLE AKCİĞER RÖNTGENİNDEN COVID-19 HASTALIĞININ TESPİT EDİLMESİ” başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Ümit ATİLA

.....

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 13/11/2021

Unvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Dr. Öğr. Üyesi Yüksel ÇELİK (KBÜ)

.....

Üye : Doç. Dr. Ümit ATİLA (GÜ)

.....

Üye : Dr. Öğr. Üyesi İlyas ÖZER (BANÜ)

.....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Hasan SOLMAZ

.....

Lisansüstü Eğitim Enstitüsü Müdürü

“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Ahmet KORKMAZ

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME TEKNİKLERİYLE AKCİĞER RÖNTGENİNDEN COVID-19 HASTALIĞININ TESPİT EDİLMESİ

Ahmet KORKMAZ

Karabük Üniversitesi

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:

Doç Dr. Ümit ATİLA

Kasım 2021, 52 sayfa

2019 yılı aralık ayında Çin'in Wuhan kentinde ortaya çıktığı tespit edilen Covid-19 hastalığı çok kısa bir süre içinde tüm dünyayı etkisi altına almıştır. Hastalığın ölümcül sonuçları olduğu ve halen bilinen bir tedavisi olmadığı için yayılmasının önlenmesi amacıyla hasta olan kişilerin erkenden belirlenmesi ve karantinaya alınması hayati önem taşımaktadır. Günümüzde virüsün tespiti için asıl tanı kiti olarak Polymerase Chain Reaction (PCR) testleri kullanılmaktadır.

Ancak farklı araştırma sonuçları, hastalığın erken ve geç dönemlerinde PCR testinin negatif çıkabileceğini ve PCR testine alternatif olarak göğüs röntgeninin de hastalığın teşhisinde kullanılabileceğini göstermiştir. Araştırmacılar, Covid-19 belirtileri olan hastaların akciğerlerinde buzlu cam opaklığına benzer bazı görsel işaretler olduğunu saptamışlardır.

Derin öğrenme teknikleri ve bilgisayar donanımlarının her geçen gün daha da gelişmesiyle tıbbın birçok alanında kullanılan derin öğrenme, Covid-19 hastalığının teşhisinde de kullanılabilir.

Bu çalışmanın amacı göğüs röntgeninden Covid-19 kaynaklı akciğer enfeksiyonu tespitini yapabilen evrişimli sinir ağı tasarlamaktır. Çalışmada Covid-19 ve zatürre hastaları ile sağlıklı bireylerden alınan 1000'er adet göğüs röntgeni görüntülerinden oluşan veri seti kullanılmıştır. Çalışmada önerilen evrişimli sinir ağı ortalama %97 doğrulukla görüntüleri sınıflandırabilmektedir.

Anahtar Sözcükler : Covid-19, akciğer röntgeni, zatürre, görüntü işleme, derin öğrenme

Bilim Kodu : 92431

ABSTRACT

M. Sc. Thesis

DETECTION OF COVID-19 INFECTION FROM THE LUNG X-RAY BY DEEP LEARNING TECHNIQUES

Ahmet KORKMAZ

**Karabük University
Institute of Graduate Programs
Department of Computer Engineering**

Thesis Advisor:

Assoc. Prof. Dr. Ümit ATILA

November 2021, 52 pages

Covid-19 disease, which has been detected in Wuhan, China in December 2019, has affected the whole world in a very short time. Since the disease has fatal consequences and there is currently no known cure, it is vital that people who are sick are identified and quarantined early to prevent its spread. Today, Polymerase Chain Reaction (PCR) tests are used as the main diagnostic kit for the detection of the virus.

However, according to different research results, it shows that the PCR test may be negative in the early and late stages of the disease and that chest X-ray can be used in the diagnosis of the disease as an alternative to the PCR test. Researchers have found that patients with symptoms of Covid-19 have some visual signs in their lungs similar to ground glass opacities.

With the development of deep learning techniques and computer equipment, deep learning, which is used in many areas of medicine, can also be used in the diagnosis of Covid-19.

The aim of this study is to design a convolutional neural network that can detect Covid-19 disease from chest X-ray images. In the study, a data set consisting of 1000 chest X-ray images taken from individuals of three different classes as Covid-19, pneumonia and healthy was used. The convolutional neural network proposed in the study can classify images with an average accuracy of 97%.

Key Words : Covid-19, chest x-ray, pneumonia, image processing, deep learning

Science Code : 92431

TEŐEKKÜR

Bu tez alıőmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteęini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle alıőmamı bilimsel temeller ışığında őekillendiren sayın hocam Do. Dr. Ümit ATILA'ya teşekkürlerimi sunarım.

Hayatım boyunca hep yanımda olup bana desteęini hiç esirgemeyen Sevgili eşime tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ	xiv
SİMGELER VE KISALTMALAR DİZİNİ	xv
BÖLÜM 1	1
GİRİŞ	1
1.1. LİTERATÜR TARAMASI.....	2
BÖLÜM 2	4
COVID-19 HASTALIĞI	4
2.1. TANIM.....	4
2.2. BULAŞMA YOLLARI.....	4
2.3. TEŞHİS YÖNTEMLERİ	4
BÖLÜM 3	6
BİLGİSAYARLI GÖRÜ	6
3.1. BOYUT VE TİP DÖNÜŞÜMÜ, EŞİKLEME.....	7
3.1.1. Yeniden Boyutlandırma	7
3.1.2. Renk Uzayı Dönüşümü	8
3.1.3. Eşikleme	9
3.2. RESİM ZENGİNLEŞTİRME	10
3.2.1. Kontrast Yayma.....	10
3.2.2. Gürültü Giderme	11

	<u>Sayfa</u>
BÖLÜM 4	12
MAKİNE ÖĞRENMESİ	12
4.1. ÖĞRENME TÜRLERİNE GÖRE MAKİNE ÖĞRENMESİ.....	12
4.1.1. Denetimli Öğrenme	13
4.1.2. Denetimsiz Öğrenme.....	14
4.2. YAPAY SİNİR HÜCRESİ.....	15
4.2.1. Girdiler	15
4.2.2. Ağırlıklar	16
4.2.3. Aktivasyon Fonksiyonu.....	16
4.2.4. Çıktı.....	17
4.3. YAPAY SİNİR AĞLARI.....	17
4.3.1. Doğrusal Regresyon	18
4.3.2. Kayıp Fonksiyonu	18
4.3.3. Eğitim Düşümü Metodu	19
4.4. YAPAY SİNİR AĞLARI EĞİTİMİNDE KARŞILAŞILAN SORUNLAR .	20
4.4.1. Eksik/Aşırı Öğrenme.....	20
4.4.2. Kaybolan ya da Patlayan Eğitim	21
BÖLÜM 5	23
DERİN ÖĞRENME.....	23
5.1. DÜZENLİLEŞTİRME	23
5.1.1. L2-L1 Düzenleştirme	23
5.1.2. İletim Sönümü	24
5.1.3. Erken Durdurma.....	25
5.2. OPTİMİZASYON.....	25
5.2.1. Stokastik Eğitim Düşümü.....	26
5.2.2. AdaGrad	26
5.2.3. RMSProp.....	27
5.2.4. Adam	27
5.3. EVRİŞİMLİ SİNİR AĞLARI	27
5.4. YİNELEMELİ SİNİR AĞLARI	29
5.5. LİTERATÜRDEKİ POPÜLER DERİN ÖĞRENME ÇALIŞMALARI	30

	<u>Sayfa</u>
5.5.1. Alexnet	30
5.5.2. VGG16	30
5.5.3. Resnet50	31
5.5.4. Inception V3	32
5.5.5. Densenet	34
BÖLÜM 6	35
ÖNERİLEN YÖNTEM.....	35
BÖLÜM 7	38
DENEYSEL SONUÇLAR VE DEĞERLENDİRME	38
BÖLÜM 8	46
SONUÇ VE TARTIŞMA	46
KAYNAKLAR	47
ÖZGEÇMİŞ	52

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Görüntünün dijital olarak kaydedilmesi.....	6
Şekil 3.2. Görüntü üzerinde yapılan işlemler	7
Şekil 3.3. RGB renk uzayı.....	8
Şekil 3.4. HSV ve YUV renk uzayları	8
Şekil 3.5. Gri tonlamalı ve siyah beyaz resimler.....	10
Şekil 3.6. Kontrast yayma işlemi öncesi ve sonrası.	11
Şekil 3.7. Tuz biber gürültüsünün medyan filtre ile düzeltilmesi.	11
Şekil 4.1. Makine Öğrenmesi Çeşitleri.....	13
Şekil 4.2. Sınıflandırma ve regresyon	14
Şekil 4.3. K-Means kümeleme ve PCA temelli boyut azaltma	14
Şekil 4.4. Yapay sinir hücresi yapısı	15
Şekil 4.5. Bazı aktivasyon fonksiyonları.....	16
Şekil 4.6. Temel YSA yapısı	17
Şekil 4.7. Doğrusal model grafik gösterimi.....	19
Şekil 4.8. Eğim düşümü metodu.....	20
Şekil 4.9. Eksik ve aşırı öğrenme.	21
Şekil 4.10. Kaybolan ya da patlayan eğim	22
Şekil 5.1. L1 ve L2 düzenlileştirmeleri	24
Şekil 5.2. İletim sönümü öncesi ve sonrası DSA	25
Şekil 5.3. Erken durdurma.....	25
Şekil 5.4. EDM ve SED teknikleri yakınsamaları.....	26
Şekil 5.5. ESA temel yapısı.....	28
Şekil 5.6. Evrişim katmanı	28
Şekil 5.7. Biriktirme katmanı	29
Şekil 5.8. Yinelemeli sinir ağı yapısı.....	29
Şekil 5.9. Alexnet modeli	30
Şekil 5.10. VGG16 modeli	31
Şekil 5.11. Resnet50 modeli.....	31
Şekil 5.12. Inception V1 evrişim katmanı.....	32

	<u>Sayfa</u>
Şekil 5.13. Inception V3 evrişim katmanı.....	33
Şekil 5.14. Inception V3 modeli.....	33
Şekil 5.15. Densenet modeli.....	34
Şekil 6.1. Önerilen yöntem ağ modeli.....	36
Şekil 7.1. Katlama-1 doğruluk ve kayıp grafiği.....	38
Şekil 7.2. Katlama-2 doğruluk ve kayıp grafiği.....	39
Şekil 7.3. Katlama-3 doğruluk ve kayıp grafiği.....	39
Şekil 7.4. Katlama-4 doğruluk ve kayıp grafiği.....	39
Şekil 7.5. Katlama-5 doğruluk ve kayıp grafiği.....	40
Şekil 7.6. Uygulama karmaşıklık matrisleri.....	42
Şekil 7.7. Hatalı sınıflandırılan resimler.....	43
Şekil 7.8. Doğru sınıflandırılan bazı resimler.....	44
Şekil 7.9. ROC eğrileri ve AUC değerleri.....	45

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 1.1. Literatürdeki Covid-19 ile ilgili bazı çalışmalar	3
Çizelge 6.1. Model özet bilgisi.....	37
Çizelge 7.1. 5-katlamalı çapraz doğrulama için eğitim ve test verisi dağılımları	38
Çizelge 7.2. Uygulama sonuçları	41

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

- Σ : toplama fonksiyonu
 ∂ : kısmi türev operatörü
 η, λ : öğrenme katsayısı
 β : regresyon katsayısı

KISALTMALAR

- AUC : Area Under the ROC Curve
DSA : Derin Sinir Ağı (Deep Neural Network)
DÖ : Derin Öğrenme (Deep Learning)
EDM : Eğim Düşümü Metodu (Gradient Descent Method)
ESA : Evrişimli Sinir Ağı (Convolutional Neural Network)
MÖ : Makine Öğrenmesi (Machine Learning)
PCR : Polymerase Chain Reaction
ROC : Receiving Operating Characteristic Curves
SED : Stokastik Eğim Düşümü (Stochastic Gradient Descent)
YİSA : Yinelemeli Sinir Ağı (Recurrent Neural Network)
YSA : Yapay Sinir Ağı (Artificial Neural Network)
YSH : Yapay Sinir Hücresi (Artificial Neuron)

BÖLÜM 1

GİRİŞ

2019 yılı sonu, kısa zamanda tüm dünyayı etkisi altına alıp yıllar sürecektir bir pandeminin başlangıcı olarak tarih sayfalarına geçmiştir. Çin’de ortaya çıkan yeni tip bir koronavirüs, insanların evden işe, okuldan restoranlara kadar birçok alanda hayatlarının seyrini belki uzun süre eski haline gelmeyecek şekilde değiştirmiştir. Bir yanda maskesiz birini gördüğünde insanlar o kimseden uzak durmaya çalışırken diğer yandan normalde hastalık derecesinde sayılan temizlik alışkanlıklarının rutin hale geldiği görülmektedir. Günümüzde farklı ülkelerde geliştirilen aşı türleri mevcut olsa da halen etkili bir tedavi yöntemi bulunmadığından ve dahi virüsün de sürekli yeni varyantları ortaya çıktığından hastalık halen insanlık için fazlasıyla tehlikeli olmaya devam etmektedir. Bu süreçte hastalığın yayılmasını engellemenin en önemli yolu hasta olan bireylerin çok hızlı bir şekilde sağlıklı bireylerle temasının kesilmesinden geçmektedir.

Hasta olan bireylerin tespiti için farklı yöntemler mevcutsa da günümüzde Sağlık Bakanlığımızın hastalığın teşhisi için kabul ettiği asıl tanı kiti PCR testleridir. Bu testler boğaz ve burundan alınan sürüntüde virüsün izinin aranması şeklinde uygulanmaktadır. Ancak Covid-19 hastalığının bulaşma ve ilerleme süresinde hastalık ilk olarak ağız ve burun yoluyla bulaşırken daha sonraları solunum yolunu kullanarak akciğerlere inmektedir. Akciğerlere indiğinde araştırmalara göre PCR testleri negatif çıkabilmektedir [1-3]. Bu nedenle hastalığın belirtilerini taşıdığı halde PCR testi negatif çıkan hastalardan doktorlar akciğer röntgeni istemekte ve teşhisini bu röntgene bakarak koyabilmektedir. Araştırmalara göre Covid-19 hastalığı akciğerlerde buzlu cam opaklığına benzer görüntülerin oluşmasına sebep olmaktadır [2].

Çalışmanın amacı Covid-19 hastalığının teşhis edilmesine yardımcı olabilecek yüksek doğruluk oranına sahip bir yöntem geliştirmektir. Çalışmalar neticesinde elde edilen

yöntem, üç farklı sınıfta ortalama %97 doğrulukla tespit yapabilmektedir. Hastalığın teşhisinde hali hazırda kullanılan PCR testleri maliyetli olduğundan, her hastanede PCR testi verme imkânı bulunmadığından ve sonucu nispeten uzun sürede çıktığından, bu yöntem karar vericilere yardımcı olabilir.

Çalışma genel olarak giriş ve literatür taraması, kullanılan materyallerin ve literatürde kabul görmüş popüler yöntemlerin tanıtımı, önerilen yöntemin tanıtılması, değerlendirme ve sonuç olmak üzere dört ana başlık altında ele alınabilir.

Giriş bölümünde çalışmanın amacı, kısa bir özeti ve kullanılabileceği alanlar ile bu alanda yapılmış bazı çalışmalarla ilgili literatür taramasına yer verilmiştir. İki, üç, dört ve beşinci bölümlerde sırasıyla Covid-19 hastalığı ile ilgili bilgiler, bilgisayarlı görü, makine öğrenmesi (MÖ) ve derin öğrenme (DÖ) teknikleri ile ilgili detaylı bilgiler verilmiş ve literatürde popüler olan bazı DÖ teknikleri detaylı olarak anlatılmıştır. Altıncı bölümde önerilen yöntemin ve kullanılan veri setinin yapısı ile ilgili detaylı bilgiler verilmiştir. Yedinci bölümde ise önerilen yöntem ile literatürdeki diğer popüler DÖ tekniklerinin uygulama sonuçlarına yer verilmiş ve bu sonuçların performans analizleri yapılmıştır. Son bölümde ise çalışmanın kullanılabileceği alanlarla ilgili önerilere yer verilmiştir.

1.1. LİTERATÜR TARAMASI

Covid-19 hastalığının güncel bir konu olması ve tüm dünyayı etkilemesi sebebiyle bu alanda birçok çalışma yapılmış ve her geçen gün bu çalışmalara bir yenisini daha eklenmektedir. DÖ teknikleriyle akciğer tomografisi ve/veya akciğer röntgeninden hastalığı tespit etmeye çalışan Khan vd. (2021), tasarladığı Deep Boosted Hybrid Learning (DBHL) yönteminde iki adet alan ve kenar bazlı birbirinden bağımsız evrişimli sinir ağını (ESA) eğiterek sondan bir önceki katmanda transfer yöntemiyle birleştirmiştir [3]. Albahli vd. (2021), Densenet ve Google tarafından yayınlanan ESA tabanlı BiT model kullanılmıştır [4]. Gilanie vd. (2021), tasarladığı ESA yapısıyla hastalığı tahmin etmeye çalışmıştır [5]. Okolo vd. (2021), Imagenet ağırlıkları kullanılarak önceden eğitilmiş EfficientNetB4 adını verdiği bir ESA modeli geliştirmiştir [6]. Ohata vd. (2021), Mobilenet ve Densenet mimarileri ile tasarlanmış

iki yapı ayrı ayrı eğitilmiş ve transfer learning metoduyla birleştirilmiştir [7]. Ozturk vd. (2021), DarkCovidNet ismini verdikleri Darknet'in daha az katman ve filtre içeren bir model geliştirmişlerdir [8]. Khan vd. (2021), Coronet adı verdikleri modellerini derinlemesine ayrıştırılabilir artık bağlantılar içeren convolüsyon katmanlarından oluşan Xception üzerine kurulu bir model olarak tasarlamışlardır [2]. Li vd. (2021), Resnet50 üzerine kurulmuş yerel iki boyutlu ve global üç boyutlu özellik çıkartabilen modellerine Covnet adını vermişlerdir [9]. Brunese vd. (2021), öncelikle akciğer röntgeninden kişide akciğer hastalığı olup olmadığını daha sonra eğer varsa bu hastalığın Covid-19 mu başka bir hastalık mı olduğunu, eğer Covid-19 ise hastalığın etkilediği bölgeleri görselleştirmeyi amaçlayan bir model geliştirmişlerdir [10]. İlk iki işlem VGG16 temelli bir yapıyla sağlanırken üçüncü işlem Gradcam yapısıyla sağlanmaktadır [10]. Wang vd. (2021), Covid-Net adını verdikleri, Covid-19 için özel tasarlanmış, mimari çeşitliliği yüksek, uzun menzilli seçici bağlantıları olan bir model tasarlamışlardır [11].

Çizelge 1.1. Literatürdeki Covid-19 ile ilgili bazı çalışmalar.

Model Adı	Yazar	Sınıf Sayısı	Veri Seti	Başarı
1 DBHL	Khan vd. (2021)	2	6448 akciğer röntgeni	98,53%
2 ---	Albahli vd. (2021)	3	15498 akciğer röntgeni	92%
3 ---	Gilanie vd. (2021)	3	akciğer tomografisi ve akciğer röntgeni	96,68%
4 EfficientNetB4	Okolo vd. (2021)	3	2905 akciğer röntgeni	98%
5 ---	Ohata vd. (2021)	2	390 akciğer röntgeni	96%
6 DarkCovidNet	Ozturk vd. (2021)	2,3	1125 akciğer röntgeni	2 sınıf için 98% 3 sınıf için 87%
7 Coronet	Khan vd. (2021)	2,3,4	1300 akciğer röntgeni	2 sınıf için 99% 3 sınıf için 89,6% 4 sınıf için 83,5%
8 COVNet	Li vd. (2021)	3	4356 akciğer tomografisi	96%
9 ---	Brunese vd. (2021)	3	6523 akciğer röntgeni	97%
10 COVID-Net	Wang vd. (2021)	3	13,975 akciğer röntgeni	93.3%

BÖLÜM 2

COVID-19 HASTALIĞI

2.1. TANIM

Son yıllarda tüm dünyayı etkisi altına alan Yeni Koronavirüs Hastalığının (Covid-19) ilk olarak 2019 yılı Aralık ayında, Çin'in Vuhan Eyaleti'nde ortaya çıktığı bilinmektedir [12]. Ateş, öksürük, nefes darlığı şeklinde solunum yolu belirtileri gelişen bir grup hasta üzerinde yapılan araştırmalar sonucunda 13 Ocak 2020'de tanımlanmıştır [12].

2.2. BULAŞMA YOLLARI

Virüs, hasta bireylerin öksürmeleri aksırmaları ile ortama saçılan damlacıkların solunması ile bulaşır. Hastaların solunum parçacıkları ile kirlenmiş yüzeylere dokunulduktan sonra ellerin yıkanmadan yüz, göz, burun veya ağıza götürülmesi ile de virüs alınabilir [12].

2.3. TEŞHİS YÖNTEMLERİ

Hastalığın ölümcül olması ve kolay bulaşarak hızlı yayılması sebebiyle erken teşhisi ve pozitif vakaların izolasyonu hayati önem taşımaktadır. Özellikle pozitif olan bir hastaya yapılan Covid-19 testinin sonucunun negatif çıkması durumunda tedaviye başlanmayacak ve hasta karantinaya alınmayacak, dolayısıyla hem hasta için hem de çevresindeki insanlar için büyük bir risk oluşacaktır.

Günümüzde virüsün tespiti için asıl tanı kiti olarak kullanılan PCR testlerinin doğruluk yüzdesi çeşitli kaynaklarda farklı oranlar belirtilmekle birlikte Sağlık Bakanlığımızın yetkili birimlerince %90'ın üzerinde olduğu açıklanmıştır [13].

Ancak farklı arařtırma sonuçlarına gre hastalığın erken ve ge dönemlerinde PCR testinin negatif ıkabileceğini ve PCR testine alternatif olarak gğs rntgeninin de hastalığın teřhisinde kullanılabileceğini gstermektedir [14-16]. Arařtırmacılar, Covid-19 belirtileri olan hastaların akcięerlerinde buzlu cam opaklığına benzer bazı grsel iřaretler olduęunu saptamıřlardır [17].

BÖLÜM 3

BİLGİSAYARLI GÖRÜ

Bilgisayarlı görü, yapay zekânın bir alt disiplini olarak insan gözleri ve beyninin algılama yeteneğine benzer bir sistem oluşturmak amacıyla 1970’lerde ortaya çıkmıştır. Tıbbi görüntüleme, güvenlik, uzaktan algılama, parmak izi tanıma gibi birçok alanda kendine kullanım alanı bulmuştur [18].

Bilgisayarlar görüntü dosyalarını insanlar gibi göremezler. Şekil-1’deki gibi her bir pikseli sayısal değerler olarak kaydedip işleyebilirler. Çok farklı görüntü formatları bulunmakla birlikte genel olarak görüntü gri tonlarından oluşan bir resim ise her bir piksel 0-255 arası bir değer olarak, RGB formatında bir resim ise her bir piksel 0-255 arası üç değer olarak kaydedilir.

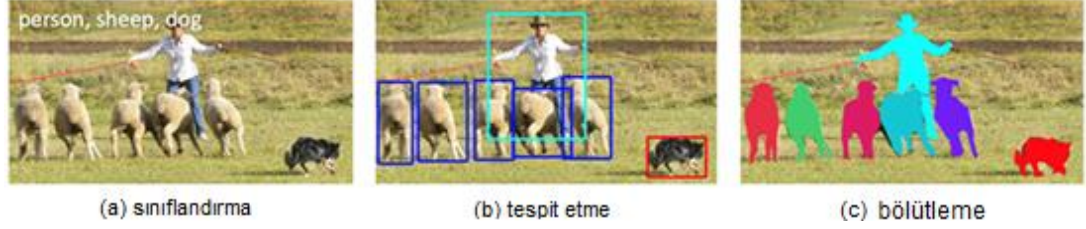


62	62	63	64	65	66	67	68	69	70	71	72	73	73	73	73	72	72	71	70	69	67	66	66	66	65	63	62	61	60	6	
61	62	63	64	66	66	68	68	69	70	71	71	72	72	73	73	72	72	71	71	69	68	67	66	66	65	65	64	63	62	61	6
61	63	64	64	66	67	68	68	69	70	71	71	73	73	74	73	73	71	70	69	68	66	66	65	64	63	62	61	61	6		
61	63	64	65	67	68	69	69	70	70	71	71	72	65	63	69	72	72	71	71	69	68	67	66	65	64	63	62	60	60	6	
63	64	65	66	67	68	69	69	70	71	71	72	62	4	5	11	68	72	71	71	69	68	67	66	65	64	62	62	60	59	5	
63	65	66	66	68	68	69	70	71	71	71	72	64	4	27	24	54	33	29	62	64	68	68	67	66	65	64	63	62	61	59	5
64	65	66	66	68	69	70	71	71	71	72	64	12	17	24	68	60	37	33	62	66	68	67	66	65	64	63	61	60	59	58	5
65	66	67	67	68	69	71	69	6	6	6	5	34	36	12	47	34	17	29	54	43	63	67	66	65	64	63	62	60	59	58	5
64	65	66	66	68	69	68	6	6	5	5	7	16	19	4	47	44	27	24	40	67	66	65	65	64	63	61	60	59	58	5	
63	64	65	65	67	69	6	6	5	5	6	8	9	20	27	51	78	41	44	66	65	65	65	65	64	63	62	60	59	58	5	
63	64	65	65	64	5	5	5	5	5	5	4	19	6	7	54	64	20	59	65	65	64	64	64	63	62	61	60	59	57	5	
63	64	64	65	14	5	6	5	4	5	4	18	7	5	4	19	10	11	05	64	64	64	63	61	66	62	61	60	59	58	5	
63	64	64	65	53	7	4	5	6	6	7	10	6	5	4	21	24	18	64	64	64	63	62	64	65	62	62	60	59	58	5	
64	64	64	64	65	50	4	4	4	5	11	16	6	6	4	6	35	16	26	66	64	64	63	61	72	67	63	62	61	59	58	5
64	64	64	64	65	46	4	4	4	5	6	9	8	5	29	10	53	56	29	57	64	64	63	61	70	67	62	64	65	59	58	5
64	64	64	65	66	27	5	4	4	5	6	6	6	18	66	20	57	60	46	35	75	70	62	61	70	67	62	61	60	58	58	5
49	50	62	65	57	5	5	6	5	6	6	6	6	31	59	26	60	58	44	22	63	71	72	60	69	68	61	60	58	59	5	
42	52	57	52	26	5	5	5	5	5	5	5	5	70	60	43	61	62	64	39	42	64	60	62	56	63	65	65	67	61	53	5
32	32	32	33	6	5	5	5	5	6	6	11	19	21	33	51	60	45	46	18	32	36	33	23	44	70	71	51	42	27	4	
50	50	51	29	5	5	5	6	5	6	6	42	69	68	34	42	39	43	37	26	29	40	26	29	26	35	42	35	33	18	1	
52	63	51	22	5	5	5	6	5	6	5	44	56	17	51	54	63	64	66	51	22	54	54	55	55	54	53	53	53	52	5	
54	54	53	8	5	5	5	6	5	6	13	32	40	21	51	54	51	49	49	50	22	41	45	42	42	41	40	41	44	43	4	
52	52	54	36	8	5	5	6	6	5	6	26	56	32	54	53	51	51	51	51	44	25	51	51	49	49	50	49	48	46	4	
54	54	52	53	30	7	5	6	6	5	6	40	54	29	52	51	53	56	55	52	52	51	38	52	52	50	49	46	46	45	46	4
51	52	51	53	27	14	5	4	5	4	7	47	51	23	59	49	47	49	52	52	52	49	35	31	48	46	47	47	46	46	4	
48	50	51	53	25	14	17	8	4	4	17	46	40	18	43	47	46	49	52	54	53	54	10	50	49	46	47	47	47	47	4	
49	49	49	49	22	12	20	24	6	14	35	51	30	48	48	50	51	51	49	51	51	52	50	41	58	48	47	47	47	45	45	4
51	49	50	50	22	13	19	38	13	12	42	50	40	73	50	50	50	49	48	49	49	48	49	45	51	46	44	44	44	42	45	4
47	49	49	47	20	16	26	30	21	15	36	48	42	61	47	48	51	47	50	51	51	51	49	47	47	47	47	44	43	45	4	

Şekil 3.1. Görüntünün dijital olarak kaydedilmesi [19].

Görüntü dosyaları üzerinde temelde üç farklı operasyon yapılmaktadır. Bunlardan birincisi bir görüntünün tamamının bir bütün olarak değerlendirilerek belli bir sınıfa

atanması anlamına gelen sınıflandırma, ikincisi görüntünün içindeki farklı nesnelere tespit etme anlamına gelen tespit etme ve son olarak da her bir nesnenin tam olarak görüntü içindeki diğer nesnelere ayrılması anlamına gelen bölütlemedir [20].



Şekil 3.2. Görüntü üzerinde yapılan işlemler [20].

Ancak bu tip işlemlerin yapılmasından önce görüntü dosyalarının bazı ön işlemlerden geçirilmesi gerekebilir. Bu işlemler de temel olarak iki ana başlık altında ele alınabilir.

3.1. BOYUT VE TİP DÖNÜŞÜMÜ, EŞİKLEME

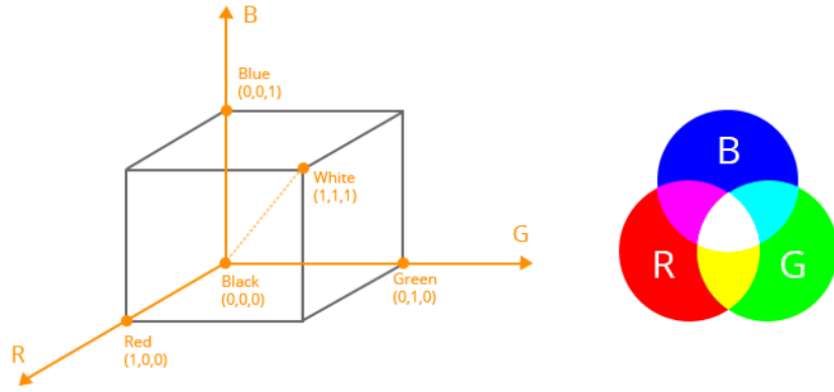
Veri kaynağından alınan veri çoğu zaman istenilen boyutta ve/veya formatta olmadığından bir dönüşüm yapılması gerekir. Bu dönüşüm renk uzayının değiştirilmesi, resmin formatının değiştirilmesi veya resmin yeniden boyutlandırılması şeklinde olabilir.

3.1.1. Yeniden Boyutlandırma

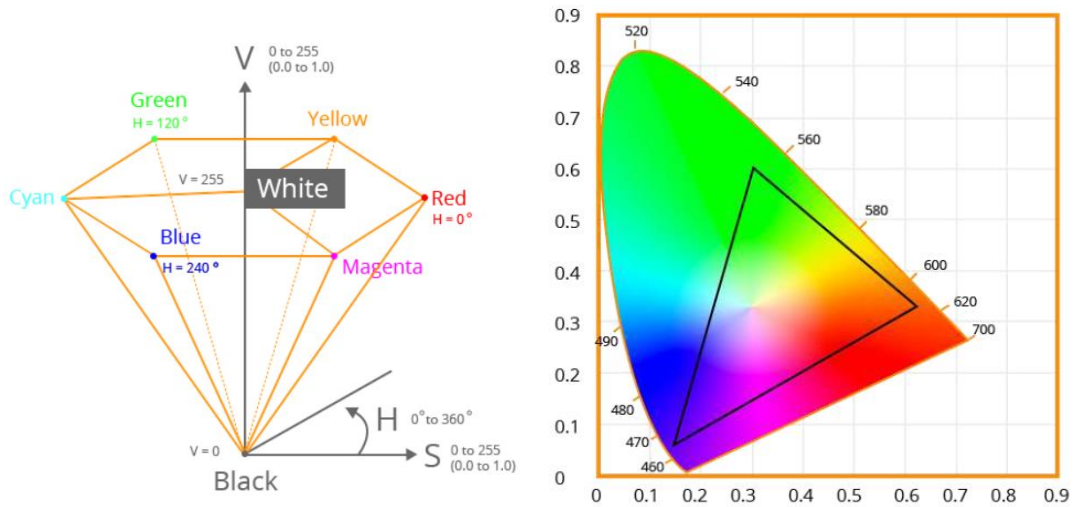
Bir resmi yeniden boyutlandırma, resmin içindeki piksel sayılarını değiştirmekle olur. Bu işlem bazen resmi bir çerçeveye sığdırmak için bazen de resimdeki bazı detayları görmek için yapılabilir. Ancak DÖ tekniklerinde farklı kaynaklardan gelen resimlerin boyutlarının eşitlenmesi veya resmin özelliklerini kaybetmeden işlenmesini daha kolay hale getirmek için yapılmaktadır. Ayrıca derin sinir ağlarında (DSA) işlenen veri miktarı fazla olduğundan yüksek çözünürlüklü resim dosyalarının çözünürlüğünün düşürülmesi için de yeniden boyutlandırma gerekebilir.

3.1.2. Renk Uzayı Dönüşümü

Renklerin bir sayı dizisi olarak nasıl temsil edilebileceğini açıklayan soyut matematiksel modele renk modeli denir (Örneğin RGB, CMYK vb.) [21]. Renk modelleri genellikle bir koordinat sistemi kullanılarak tanımlanabilir ve sistemdeki her renk, koordinat uzayında tek bir nokta ile temsil edilir [21]. Belirli bir renk modeli için, bir sayı demetini bir renk olarak yorumlamak için, renkleri doğru bir şekilde kalibre etmek ve oluşturmak için kullanılan kurallar, başka bir deyişle bir haritalama fonksiyonu tanımlanmalıdır [21]. Renk uzayı, renk modeli ve haritalama fonksiyonunun özel bir kombinasyonu olarak tanımlanabilir [21].



Şekil 3.3. RGB renk uzayı [21].



Şekil 3.4. HSV ve YUV renk uzayları [21].

Kullanılan yere ve amaca göre belirli renk uzayında kaydedilmiş resim dosyalarının başka bir renk uzayına dönüştürülmesi gerekebilir. Örneğin RGB formatındaki bir resim gri tonlamalı bir formata Eşitlik 3.1'deki formüllerden biriyle dönüştürülebilir.

$$\text{Gri Tonlamalı} = R / 3 + G / 3 + B / 3 \text{ (Ortalama yöntem)} \quad (3.1a)$$

$$\text{Gri Tonlamalı} = 0.299R + 0.587G + 0.114B \text{ (Ağırlıklı yöntem)} \quad (3.1b)$$

RGB renk uzayından YUV renk uzayına dönüştürmek için de Eşitlik 3.2'deki formüller kullanılır.

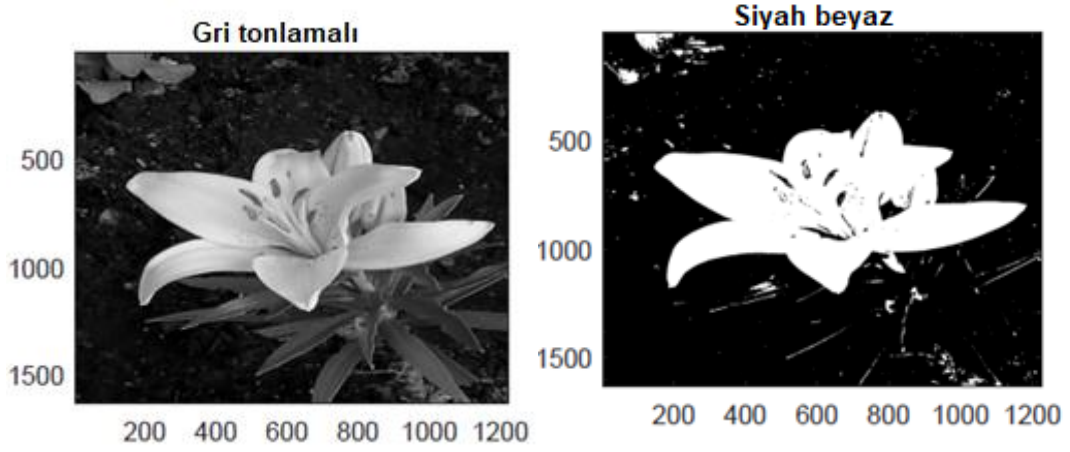
$$Y = 0.299R + 0.587G + 0.114B \quad (3.2a)$$

$$U' = (B-Y) * 0.565 \quad (3.2b)$$

$$V' = (R-Y) * 0.713 \quad (3.2a)$$

3.1.3. Eşikleme

Bir resimde sadece belli değerlere sahip yeğinlikteki piksellerin bilgilerinin tutulması istendiğinde eşikleme işlemi yapılır. Örneğin gri tonlamalı bir resim sadece siyah ve beyaz piksellerden oluşan bir resme dönüştürmek istendiğinde her bir pikselin üzerinde bir işlenen gezdirilerek o pikselin yeğinlik değeri eşik değerinin üzerindeyse beyaz altındaysa siyah olacak şekilde işlem yapılabilir [21].



Şekil 3.5. Gri tonlamalı ve siyah beyaz resimler [21].

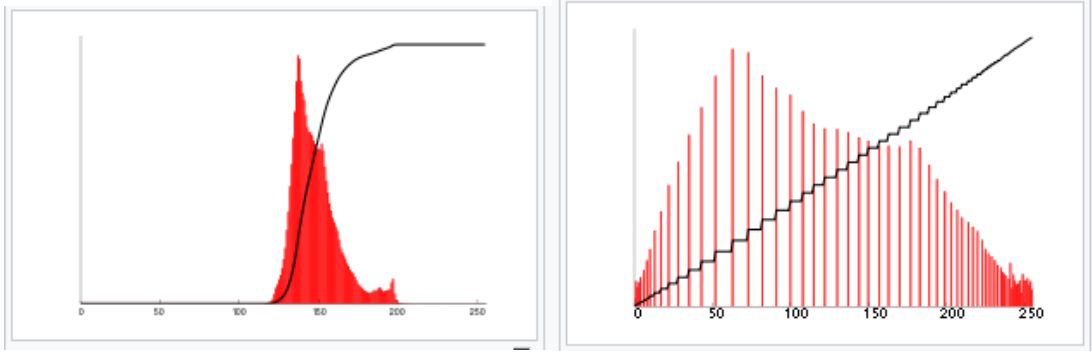
3.2. RESİM ZENGİNLEŞTİRME

Resim dosyaları bazen istenilen formatta ve boyutta olsalar bile içerdiği verinin daha sağlıklı olarak elde edilebilmesi için resim üzerinde ek işlemler yapılması gerekebilmektedir. Örneğin karanlık bir resmin içindeki ayrıntıları daha iyi ayırt edilebilir hale getirmek ya da gürültülü bir resmin içindeki gürültüyü temizlemek için birtakım operasyonlar gerekir.

3.2.1. Kontrast Yayma

Bu yöntemle belli bir aralıkta yoğunluğu fazla olan resmin renk değerlerinin tüm renk uzayına yayarak resmin görünürlüğü artırılmaktadır.

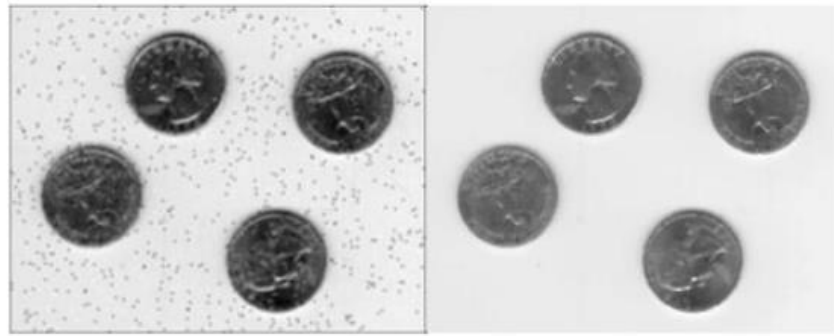




Şekil 3.6. Kontrast yayma işlemi öncesi ve sonrası [21].

3.2.2. Gürültü Giderme

Görüntü dosyasında gürültü, herhangi bir pikselin yeğinlik değerinin doğru değerinden farklı olmasıdır [22]. Görüntüyü algılayan algılayıcının çevresel faktörlerden etkilenmesiyle, veri iletimi esnasında ya da çevresel faktörler sebebiyle (toz, ışık azlığı vb.) gürültü oluşabilir. Bazen de görüntü dosyasına yapay gürültü eklenebilir. Her iki durumda da gürültü giderme algoritmaları, temel olarak gürültünün görünürlüğü azaltmak veya tamamen ortadan kaldırmak için gürültü pikselin değerini tüm resmin karşıtlık sınırları içinde bir değer ile değiştirir [22]. Gürültünün çeşidine göre uygulanacak filtreler de değişiklik göstermektedir.



Şekil 3.7. Tuz biber gürültüsünün medyan filtre ile düzeltilmesi [22].

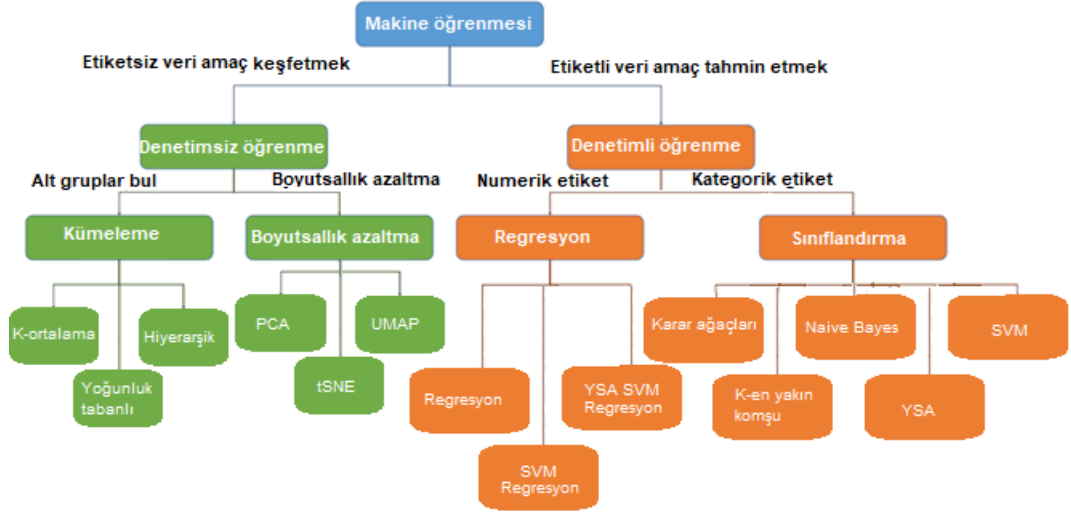
BÖLÜM 4

MAKİNE ÖĞRENMESİ

En temel düzeyde MÖ, bir insan tarafından açıkça programlanmasına gerek kalmadan kendi kendine "öğrenebilen" her tür bilgisayar programını ifade eder [23]. MÖ, "Belli sınıflara bağlı bir şekilde belli görevlerin sonucunda elde ettiği deneyimlerin performansını ölçerek bu ölçümleri kullanıp deneyimlerini iyileştiren bilgisayar bir bilgisayar programıdır." denilebilir [24]. MÖ, literatürde yapay zekânın bir alt dalı olarak değerlendirilmektedir.

4.1. ÖĞRENME TÜRLERİNE GÖRE MAKİNE ÖĞRENMESİ

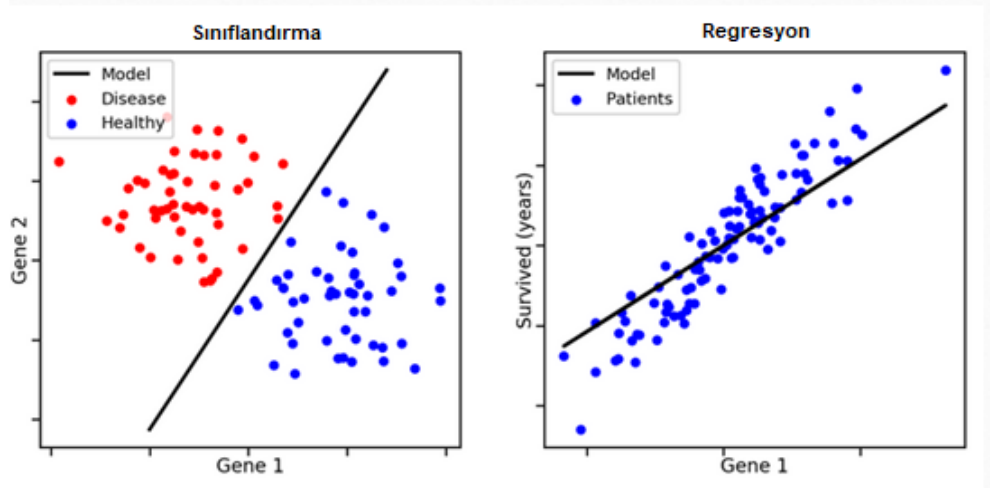
Veri setlerinde var olan her bir verinin belli bir etiketi olup olmaması durumuna göre öğrenme yöntemleri değişmektedir. Eğer veriler etiketli ise bu etiketler kullanılarak bir öğrenme modeli oluşturulabilir. Daha sonra bu model kullanılarak öncesinde hiç görülmeyen bir verinin etiketi tahmin edilebilmektedir. Ancak bazen de veri setinde herhangi bir etiket değeri bulunmaz. Bu durumda da belli özelliklere göre verilerin keşfedilmesi ve gruplandırılması için bir model oluşturulabilir.



Şekil 4.1. Makine Öğrenmesi Çeşitleri [25].

4.1.1. Denetimli Öğrenme

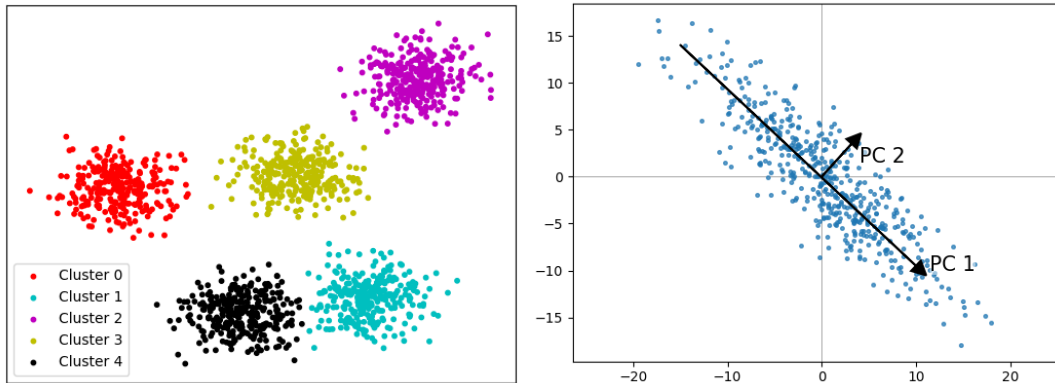
Denetimli öğrenmede bilgisayar, belli girdiler karşılığında belli çıktılar verdiği bilinen eğitim verileriyle beslenir [25]. Amaç, girdi değerlerini bilinen ancak bunlarla ilişkili çıktıları bilinmeyen, yeni görünen veriler için çıktıyı tahmin etmeyi mümkün kılacak şekilde girdileri çıktılarıyla eşleştiren genel kuralları (genellikle "model" olarak da adlandırılır) öğrenmektir [25]. Temelde iki ana başlık altında ele alınabilir. İlk olarak her bir girdiyi belli sayıdaki sınıflardan birine atamak anlamına gelen sınıflandırma, ikincisi ise girdilere karşılık sayısal bir çıktı değeri hesaplama anlamına gelen regresyondur [25].



Şekil 4.2. Sınıflandırma ve regresyon [26].

4.1.2. Denetimsiz Öğrenme

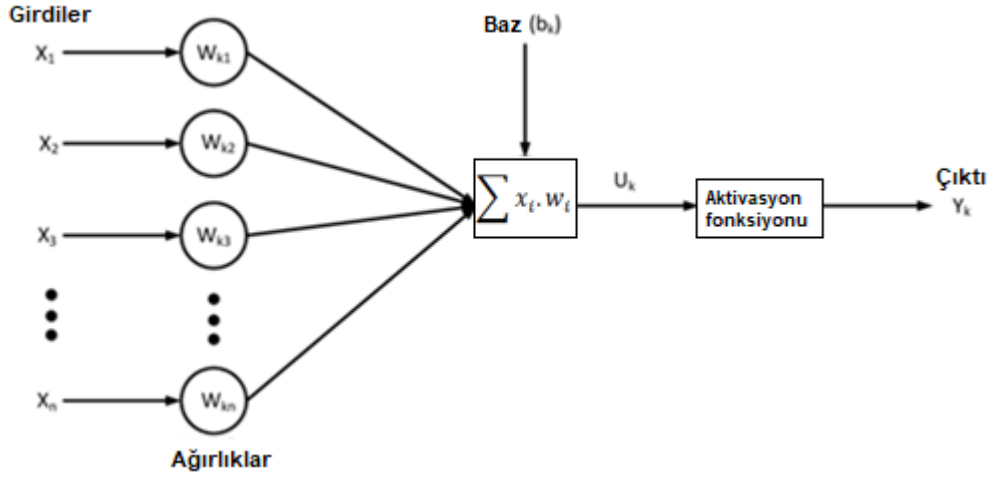
Denetimsiz öğrenmede modeller veri setinin bazı özelliklerini tanımlayarak öğrenir [27]. Farklı denetimsiz öğrenme teknikleri farklı özellikleri keşfetmeyi sağlamakla birlikte, verileri benzer bölütler halinde gruplandırır ve veri kümesindeki kalıpları ortaya çıkaran kümeleme (clustering) veya veri noktaları arasında en fazla farklılaşmayı sağlayan veri değerlerinin özelliklerini tanımlayan boyut azaltma (dimensionality reduction) şeklinde iki ana başlık altında ele alınabilir [27]. Çünkü denetimsiz öğrenme doğası gereği keşfedici olduğundan, doğru veya yanlış cevap kavramı yoktur.



Şekil 4.3. K-Means kümeleme ve PCA temelli boyut azaltma [28].

4.2. YAPAY SİNİR HÜCRESİ

İnsan beyninin öğrenme ve karar verme yeteneğini bilgisayarlara da kazandırmayı amaçlayan bilim adamları biyolojik sinir hücrelerinin bir benzetimi diyebileceğimiz yapay sinir hücrelerini geliştirmişlerdir. Yapay sinir hücreleri (YSH) de aynı biyolojik sinir hücreleri gibi kendisine verilen verileri belli işlemlerden geçirerek belli çıktılar üretirler. Bir yapay sinir hücresinin temelde beş bileşeni vardır [29]. Yapay sinir hücresine tekil hesaplama katmanı (single computational layer) veya algılayıcı (perceptron) da denir [30].



Şekil 4.4. Yapay sinir hücresi yapısı [29].

4.2.1. Girdiler

Dış dünyadan ya da diğer YSH'den gelen bilgilerdir [29]. Günümüzde sayısallaştırılabilen her türlü veri, MÖ'nün konusu olabilmektedir. Girdiler farklı sayısal değer aralıklarında olabileceği için girdinin 0-1 aralığına çekilmesi yani normalize edilmesi elde edilecek sonuçların doğruluğu açısından önemlidir. Örneğin havanın sıcaklık ve nem değerlerini girdi olarak alıp yağış tahmini yapan bir YSH'ye sıcaklık değerinin 20 santigrat derece ve nem değerinin de 0,40 olarak verildiğini düşünelim. Bu iki değerden biri diğerinin 50 katı olduğundan bu sayılar normalize edilmeden alınırsa sıcaklığın sonuca etkisi neme göre çok daha fazla olacaktır. Bir YSH'nin girdileri genellikle (x_1, x_2, \dots, x_n) şeklinde gösterilir.

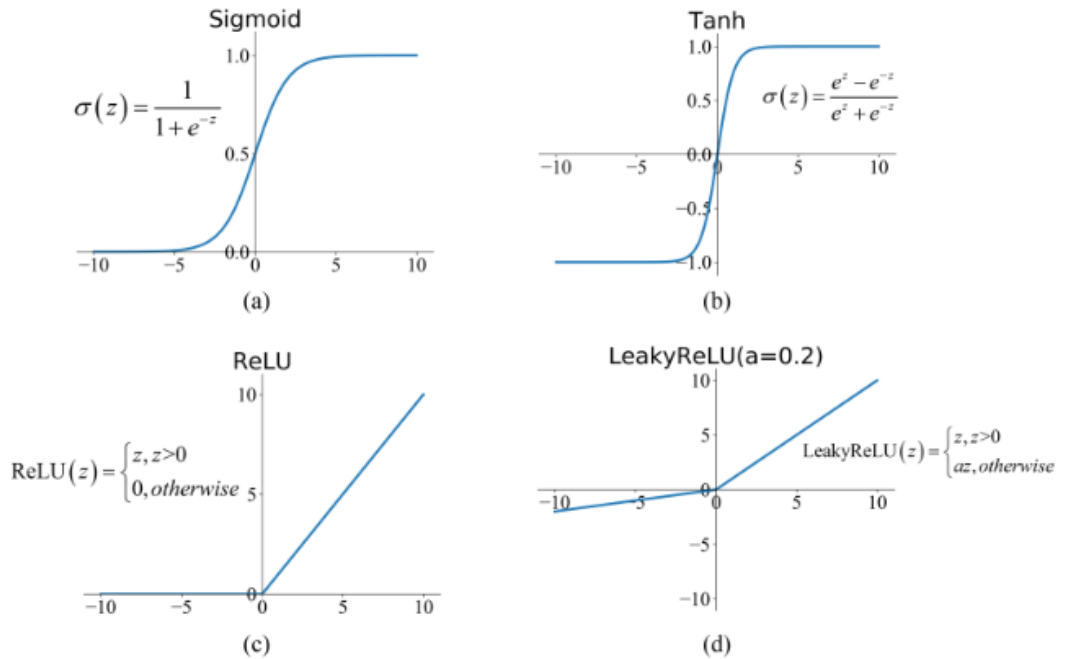
4.2.2. Ağırlıklar

Bir hücreye gelen her bir verinin etkisini gösteren değerlerdir [29]. Negatif, pozitif ya da sıfır olabilmektedir. Değerin negatif olması o girdinin sonuca negatif yönlü etkisi olduğunu gösterir. Değerin sıfır olması o girdinin sonuca etkisi olmadığını gösterir. Ağırlıklar genellikle (w_1, w_2, \dots, w_n) şeklinde gösterilir. Daha sonra her bir girdi kendisine tanımlanmış ağırlıkla çarpılır ve çıkan sayılar toplanır. Buna toplama fonksiyonu denir. Bu fonksiyonun sonucunda net girdi elde edilir.

$$NET = \sum x_i \cdot w_i \quad (4.1)$$

4.2.3. Aktivasyon Fonksiyonu

Bir YSH'ye gelen net girdiyi işleyerek o girdilere karşılık bir sonuç üretir [29]. Aktivasyon fonksiyonunda çıktıyı hesaplamak için farklı formüller kullanılabilir. Ağın tüm işlem elemanlarında aynı aktivasyon fonksiyonunun kullanılması gerekmez [29].



Şekil 4.5. Bazı aktivasyon fonksiyonları [31].

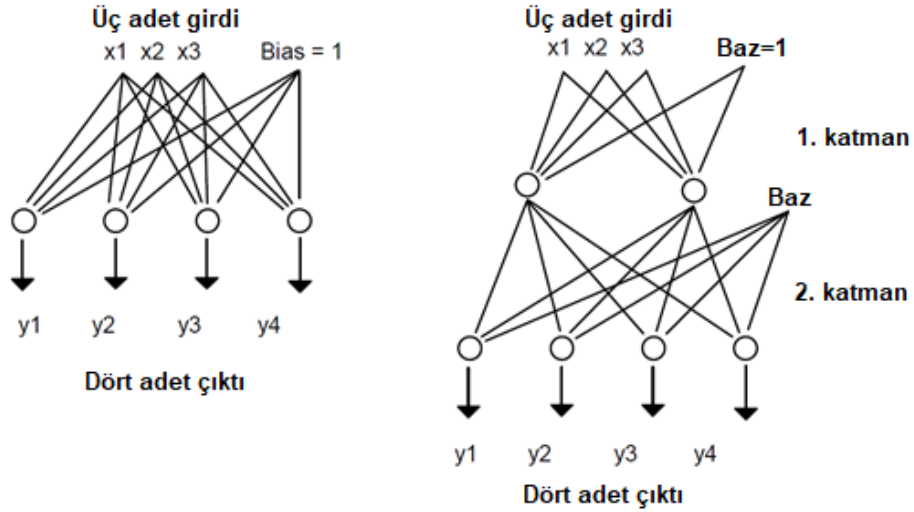
Şekil-4.5'te bazı aktivasyon fonksiyonları yer almaktadır. Problemin türüne göre seçilmeleri gerekir.

4.2.4. Çıktı

Aktivasyon fonksiyonunda üretilen çıktı değeridir. Bu değer nihai çıktı olabileceği gibi çok katmanlı yapılarda başka YSH'lere veya kendine girdi olarak da gönderilebilir [29]. Çıktılar genellikle y harfiyle gösterilir.

4.3. YAPAY SİNİR AĞLARI

Birden çok YSH'nin seri ve/veya paralel olarak birbirine bağlanmasıyla ortaya çıkan yapıya yapay sinir ağı (YSA) denir. Çoğunlukla katmanlı bir yapıda tasarlanırlar. İleri yönlü hesaplamada yukarıda anlatılan yöntemlerle belli girdilerden belli sonuçlar elde edilir.



Şekil 4.6. Temel YSA yapısı [32].

Daha sonra geri yönlü hesaplama yöntemleri kullanılarak hesaplanan sonuçla olması gereken sonuç arasındaki fark (hata) üzerinden eğitim düşümü algoritmalarıyla ağırlıklar güncellenir ve hatanın azaltılması sağlanmaya çalışılır.

4.3.1. Doğrusal Regresyon

Regresyon, bir veya daha fazla bağımsız değişken ile bağımlı bir değişken arasındaki ilişkiyi modellemeye yönelik bir dizi yöntemi ifade eder [33]. Doğa bilimlerinde ve sosyal bilimlerde, regresyonun, çoğunlukla girdiler ve çıktılar arasındaki ilişkiyi karakterize etmek için kullanılırken, MÖ'de çoğunlukla tahmin için kullanılır [33].

Sayısal bir değeri tahmin etmek istediğimizde regresyon sorunları ortaya çıkar. Yaygın örnekler arasında fiyatları tahmin etmek (evlerin, stokların vb.), kalış süresinin tahmin edilmesi (hastanedeki hastalar için), talep tahmini (perakende satışlar için) sayılabilir [33].

Doğrusal Regresyonda girdi ile çıktı arasındaki ilişki doğrusal kabul edilir [33]. Örneğin bir evin fiyatını belirleyen iki özellik yüzölçümü ve yaşı ise bu evin fiyat fonksiyonu Eşitlik 4.2a'daki gibi yazılabilir.

$$Fiyat = w * yüzölçümü + w * yaş + b \quad (4.2a)$$

Burada w değerleri ağırlıkları b değeri ise sabit başlangıç değerini ifade eder. Bu işlemde değişken sayısı arttırılabilir.

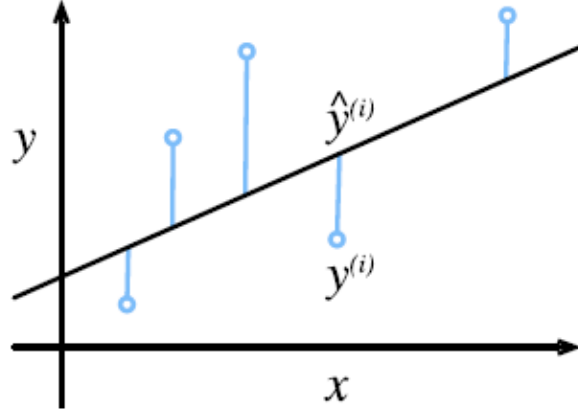
$$\hat{y} = w_1x_1 + \dots + w_dx_d + b \quad (4.2b)$$

4.3.2. Kayıp Fonksiyonu

Kayıp fonksiyonu, tahmin edilen değerle gerçek değer arasındaki farkı azaltmak için ağırlık değerlerinin güncellenmesinde kullanılacak eğitim düşümü algoritmalarına veri olarak gelen hesaplamalardır. Problemin türüne göre farklı kayıp fonksiyonlarından yararlanır. Regresyon problemlerinde ortalama kare hatası (mean squared error) veya ortalama mutlak hata (mean absolute error) kullanılırken sınıflandırma problemlerinde eğer tek sınıf varsa ikili çapraz entropi daha fazla sınıf varsa kategorik çapraz entropi hatası kullanılabilir.

$$l^{(i)}(w, b) = \frac{1}{2} (\hat{y}^{(i)} - y^{(i)})^2 \quad (4.3)$$

Örneğin yapılan tahminler ile gerçek çıktı değerleri arasındaki farktan yararlanılarak en küçük kareler yöntemi ile kayıp miktarı hesaplanır.



Şekil 4.7. Doğrusal model grafik gösterimi [33].

Tahmin edilen ve gerçek değer arasındaki fark arttıkça kayıp miktarı karesel olarak daha fazla artacaktır. Modelin n adet örnek üzerindeki kalitesini ölçmek için kayıp fonksiyonu Eşitlik 4.4'teki gibi ifade edilebilir [33].

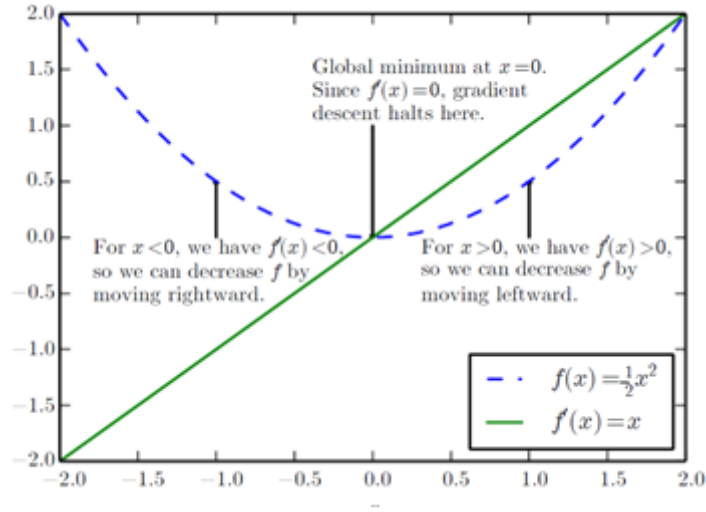
$$L(w, b) = \frac{1}{n} \sum_{i=1}^n l^{(i)}(w, b) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (w^T x^{(i)} + b - y^{(i)})^2 \quad (4.4)$$

Burada amaç kayıp değerini minimize eden w ve b değerlerini bulmaktır.

$$w^*, x^* = \underset{w, b}{\operatorname{argmin}} L(w, b) \quad (4.5)$$

4.3.3. Eğim Düşümü Metodu

Neredeyse tüm MÖ modellerini optimize etmek için kullanılan ve parametrelerin aşamalı olarak değiştirilerek hatanın azaltılmasını amaçlayan metoda eğim düşümü metodu (EDM) denir [33].



Şekil 4.8. Eğim düşümü metodu [19].

Bir fonksiyonun türevi, o fonksiyonun girdisinde meydana gelecek küçük bir değişimin çıktısı üzerinde ne kadar etkisi olduğunu gösterir. Yukarıdaki şekilde verilen $f(x) = \frac{1}{2} x^2$ fonksiyonunun türevi $f'(x)=x$ olduğundan fonksiyonun türevini sıfıra eşitleyen $x=0$ noktası fonksiyonun global minimum noktasıdır. Bu nokta haricinde seçilen bir x değeri için hangi yönde hareket edileceği fonksiyonun türevi ile bulunur. Bu işlem, EDM'nin temelini oluşturmaktadır [19].

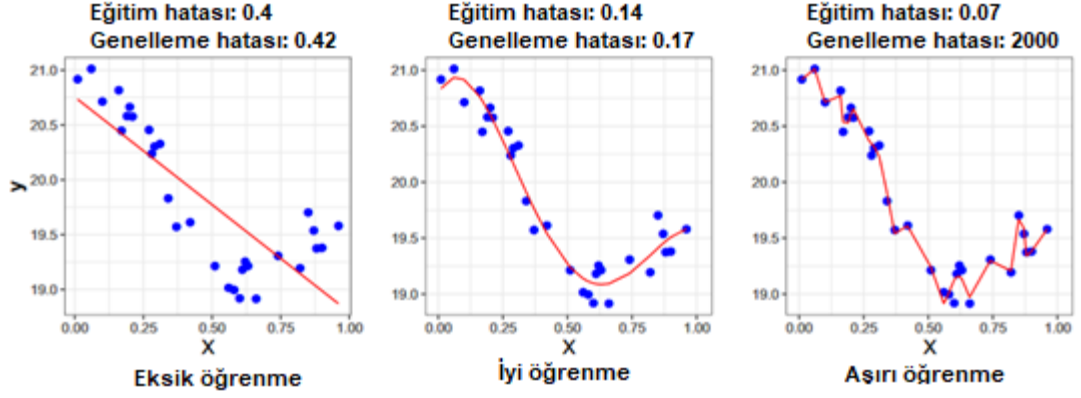
4.4. YAPAY SİNİR AĞLARI EĞİTİMİNDE KARŞILAŞILAN SORUNLAR

YSA'ların eğitimi, belli standartlarda yapılan ve her yapıldığında aynı sonucu veren işlemler değildir. Verinin miktarına, örneklerin dağılımının düzgün olup olmamasına, ağırlık katman sayısına ve katmanların yapısına göre ortaya çıkan sonuçlar farklılık gösterebilir. Ancak eğitim esnasında yaşanabilen temel sorunlardan iki ana başlık halinde bahsetmek mümkündür.

4.4.1. Eksik/Aşırı Öğrenme

YSA eğitimi süresince elde edilen modelin doğruluk değeri istenilen değerlere ulaşamadığı zaman eksik öğrenmeden söz edilebilir. Diğer taraftan girdi olarak verilen veriye uygulanan algoritmalar neticesinde elde edilen model, veriye o kadar adapte olur ki adeta eğitim verisini ezberler. Model, eğitim verisi üzerinde iyi bir performans

gösterse bile daha önce görmediği bir veri ile karşılaştığında aynı performansı gösteremez. Bu duruma aşırı öğrenme denir.



Şekil 4.9. Eksik ve aşırı öğrenme [33].

Aşırı öğrenmeyi engellemek için bazı algoritmalar geliştirilmiştir. Bir sonraki bölümde bu algoritmalarından bahsedilecektir.

4.4.2. Kaybolan ya da Patlayan Eğim

YSA'ların eğitimi esnasında başta ağırlık değerleri belirlenir ve bu ağırlıklar kullanılarak ileri yönlü hesaplamalar yapılır. Elde edilen sonuçlar kayıp fonksiyonuna girer ve hata miktarı hesaplanır. Daha sonra geri yönlü eğim düşümü algoritmaları kullanılarak ağırlık değerleri güncellenir. Ancak YSA derinleştikçe geri yönlü hesaplamada türev işlemi kullanıldığından ve sigmoid ve tanh fonksiyonlarının türevleri 1'den küçük sayılar olduğundan giriş katmanına yaklaştıkça hesaplanan ağırlıklar da daha az güncellenecektir. YSA veya ESA'larda en baştaki katmanlar girdinin en temel özelliklerini belirlemek için kullanıldığından bu katmanlardaki ağırlık değerleri yeterince güncellenemediğinde ve ağırlık eğitimi belli bir seviyenin ilerisine geçemeyecektir. Aynı şekilde eğer başlangıç ağırlıkları çok yüksek sayılar ise bu sefer de zincir kuralı gereği hesaplanan ağırlık güncellemeleri çok yüksek sayılara ulaşacak ve patlayan eğim sorunu ortaya çıkacaktır.

$$\left. \frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \frac{\partial c_T}{\partial c_{T-1}} \cdot \frac{\partial c_{T-1}}{\partial c_{T-2}} \cdot \dots \cdot \frac{\partial c_{t+1}}{\partial c_{c_t}} \right\} \frac{\partial \mathcal{L}}{\partial w} \ll 1 \Rightarrow \text{Kaybolan E\u011fim}$$

$$\left. \frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \frac{\partial c_T}{\partial c_{T-1}} \cdot \frac{\partial c_{T-1}}{\partial c_{T-2}} \cdot \dots \cdot \frac{\partial c_1}{\partial c_{c_t}} \right\} \frac{\partial \mathcal{L}}{\partial w} \gg 1 \Rightarrow \text{Patlayan E\u011fim}$$

Şekil 4.10. Kaybolan ya da patlayan eğim [32].

BÖLÜM 5

DERİN ÖĞRENME

DÖ her ne kadar MÖ'nün bir alt alanı olsa da diğer alt alanları bu tezin kapsamına girmediğinden DÖ ayrı bir başlık altında ele alınacaktır. DÖ tekniklerinin YSA'dan en önemli farkı, veri miktarının fazla olmasından dolayı sınıflandırılacak olan girdinin içinden belli özelliklerin tespit edilerek daha hızlı bir şekilde doğruluğu yüksek bir model üretilmesidir. Zaman içinde kullanıldıkları alana göre farklı modeller ortaya çıkmıştır. Örneğin ESA'lar genellikle resim dosyalarının sınıflandırılmasında kullanılırken yinelemeli sinir ağları (YİSA) sıralı gelen bilgilerin (metin tanıma ses/video dosyası tanıma vb.) sınıflandırılmasında/anlamlandırılmasında kullanılmaktadır. Derin sinir ağı (DSA) modellerinden önce ağların eğitimi esnasında kullanılan bazı algoritmalarından bahsetmekte fayda olabilir.

5.1. DÜZENLİLEŞTİRME

Bir DSA'da öğrenme algoritmasında genelleme hatasını azaltarak aşırı öğrenmenin önüne geçmeyi amaçlayan herhangi bir değişiklik düzenleme olarak tanımlanabilir [19]. Literatürde sıkça rastlanan bazı düzenleme yöntemleri aşağıda açıklanmıştır.

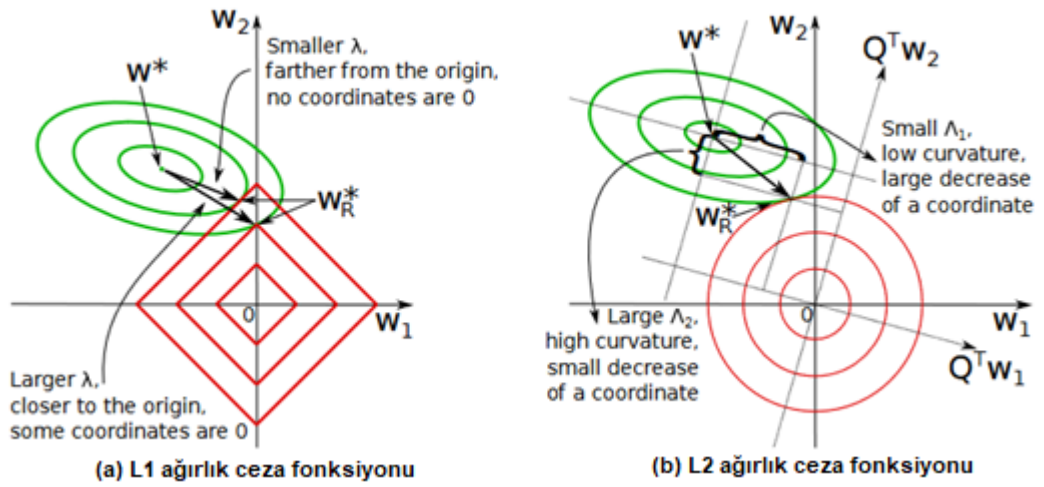
5.1.1. L2-L1 Düzenleme

Genel anlamda düzenleme ağırlıkların cezalandırılması olarak tanımlanabilir. L1 (Lasso Regresyonu) ve L2 (Ridge Regresyonu), YSA'larda ağırlık değerlerinin güncellenmesi esnasında bazı ağırlıkların aşırı önem kazanmasının ya da kaybolmasının önüne geçmek için geliştirilmiştir. L1 düzenlemesi bazı ağırlıkların katsayılarını sıfıra çekerek katsayıların seyrekleştirilmesinde kullanılır. Genellikle önemsiz ağırlık değerleri çok olan durumlarda tercih edilir. L2

düzenlileştirilmesi ise katsayıları sıfıra indirmez, makul tahminler üretmeye yardımcı olur ve L1'e göre daha yaygındır [34].

$$\hat{\beta}_{L1} = \underset{\beta}{\operatorname{argmin}} (\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i,1} + \dots + \beta_p x_{i,p}))^2 + \lambda \sum_{j=0}^p |\beta_j|) \quad (5.1a)$$

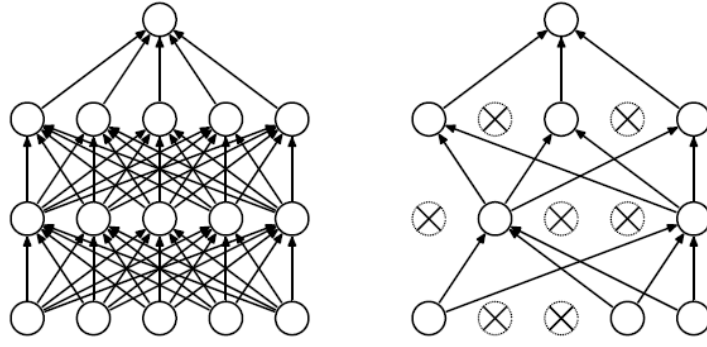
$$\hat{\beta}_{L2} = \underset{\beta}{\operatorname{argmin}} (\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i,1} + \dots + \beta_p x_{i,p}))^2 + \lambda \sum_{j=0}^p |\beta_j|^2) \quad (5.1b)$$



Şekil 5.1. L1 ve L2 düzenliştirmeleri [34].

5.1.2. İletim Sönümü

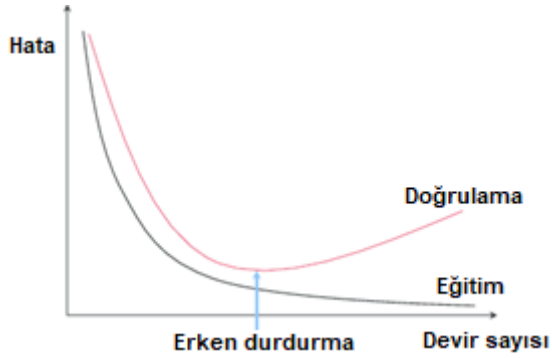
DSA'lar birden çok doğrusal olmayan gizli katman içerir ve bu sayede girdiler ve çıktılar arasındaki çok karmaşık ilişkileri temsil eden etkileyici modeller üretebilirler [35]. Bununla birlikte, sınırlı eğitim verilerinde, bu karmaşık ilişkilerin çoğu, örnekleme gürültüsünün sonucu olacaktır. Bu nedenle, aynı dağılımdan alınsa bile, eğitim kümesinde var olan bazı özellikler test verilerinde olmayabilir [35]. Bu durumda model eğitim verisine gereğinden fazla uyum sağlayacağından aşırı öğrenme sorunuyla karşı karşıya kalınmaktadır. İletim sönümü işlemi ile bazı bağlantılar rastgele iptal edilerek farklı mimaride ağlar oluşturulur ve bu ağların ortalama değerleri ile bir model ortaya çıkar.



Şekil 5.2. İletim sönümü öncesi ve sonrası DSA [35].

5.1.3. Erken Durdurma

DSA'nın eğitimi esnasında güncellenen modelin performansı eğitim veri seti üzerinde iyileşmeye devam etse de doğrulama veri seti üzerinde iyileşmeyebilir. Belli bir süre doğrulama veri seti üzerinde performans iyileşmesi görülmediği zaman eğitimin durdurulması için erken durdurma kullanılır.



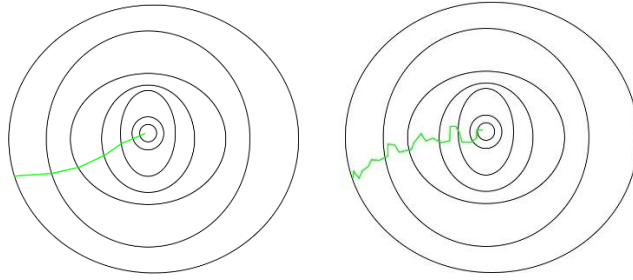
Şekil 5.3. Erken durdurma [36].

5.2. OPTİMİZASYON

Derin öğrenme tekniklerinin en başında genellikle bir kayıp fonksiyonu tanımlanır. İşte bu kayıp fonksiyonunu minimize etmeye yarayan algoritmalara en iyileyici denir [33]. Literatürde sıkça rastlanan bazı en iyileyici algoritmalar aşağıda açıklanmıştır.

5.2.1. Stokastik Eğim Düşümü

Tipik bir EDM kullanıldığında, veri setinde çok fazla örnek olduğu durumlarda bir yinelemeyi tamamlamak için tüm örnekleri her birinin kullanılması gerektiğinden minimum noktasına ulaşmak hesaplama açısından çok pahalı hale gelmektedir [37]. SED'de, her bir yinelemeyi gerçekleştirmek için veri setinden rastgele bir veri seçilir ve yineleme bu veri üzerinde yapılır. Ancak bu durumda minimum noktasına ulaşmak için daha fazla yineleme yapılması gerekir. Yine de geleneksel EDM'ye göre çok daha az maliyetlidir.



Şekil 5.4. EDM ve SED teknikleri yakınsamaları [37].

5.2.2. AdaGrad

AdaGrad, (Adaptive Gradient Optimizer) metodunun EDM ve SED'den farkı her bir ağırlık değerinin farklı bir öğrenme katsayısına sahip olmasıdır [38].

$$\hat{\eta}_t = \frac{\eta}{\text{sqrt}(\alpha_t + \epsilon)} \quad (5.2a)$$

$$(w)_{\text{yeni}} = (w)_{\text{eski}} - \eta \frac{\partial L}{\partial w(\text{eski})} \quad (5.2b)$$

$$w_t = w_{t-1} - \hat{\eta}_t \frac{\partial L}{\partial w(t-1)} \quad (5.2c)$$

Bu durum, öğrenme katsayısının manuel ayarlanmasını gerektirmemesi, daha hızlı yakınsaması ve daha güvenilir olması açısından avantajlıdır [38]. Ancak yineleme

sayısı arttıkça alfa değeri artabileceğinden öğrenme katsayısının değeri ve bununla birlikte öğrenme hızı düşebilir [38].

5.2.3. RMSProp

Eğimlerin büyüklüğü, farklı ağırlıklar için farklı değerler alabilir ve öğrenme sırasında değişebilir, bu da tek bir küresel öğrenme katsayısı seçmeyi zorlaştırır [39]. RMSProp (Root Mean Square Propagation), eğimlerin karesinin hareketli bir ortalamasını tutarak ve ağırlık güncellemelerini bu büyüklüğe göre ayarlayarak bu sorunu çözer [39].

$$E[g^2](t) = \beta E[g^2](t-1) + (1-\beta)\left(\frac{\partial c}{\partial w}\right)^2 \quad (5.3a)$$

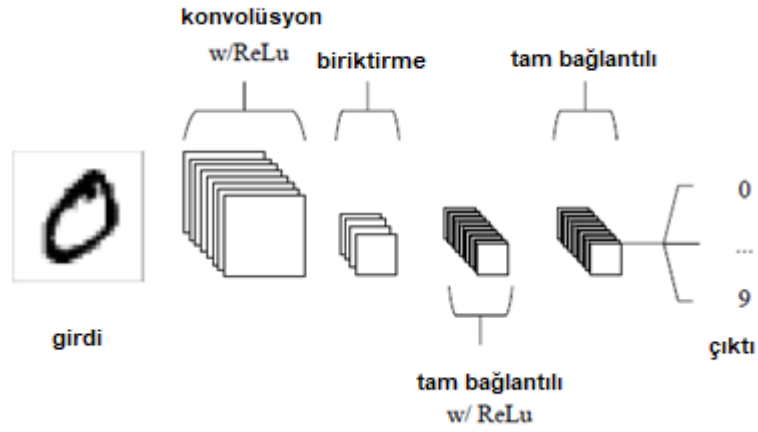
$$w_{ij}(t) = w_{ij}(t-1) - \frac{\eta}{\sqrt{E[g^2]}} \frac{\partial c}{\partial w_{ij}} \quad (5.3b)$$

5.2.4. Adam

Adam (Adaptive Moment) metodu, AdaGrad ve RMSProp yöntemlerinin avantajlarını birleştirmek için geliştirilmiştir [40]. Yöntem, eğimlerin birinci ve ikinci momentlerinin tahminlerinden farklı parametreler için bireysel uyarlanabilir öğrenme oranlarını hesaplar [40].

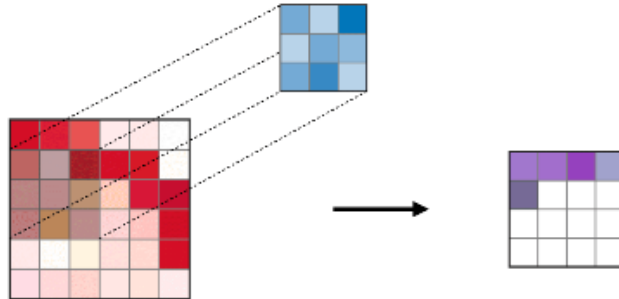
5.3. EVRİŞİMLİ SİNİR AĞLARI

ESA'lar çok katmanlı algılayıcıların bir türü olarak düşünülebilir. Çok katmanlı sinir ağlarına ilave olarak özellik çıkartma (feature extraction) yeteneği olan yapay sinir ağlarıdır. Bunu evrişim ve biriktirme (pooling) işlemleriyle gerçekleştirmektedirler [7]. Sistemin genel yapısı Şekil 5.5'te gösterildiği gibidir.



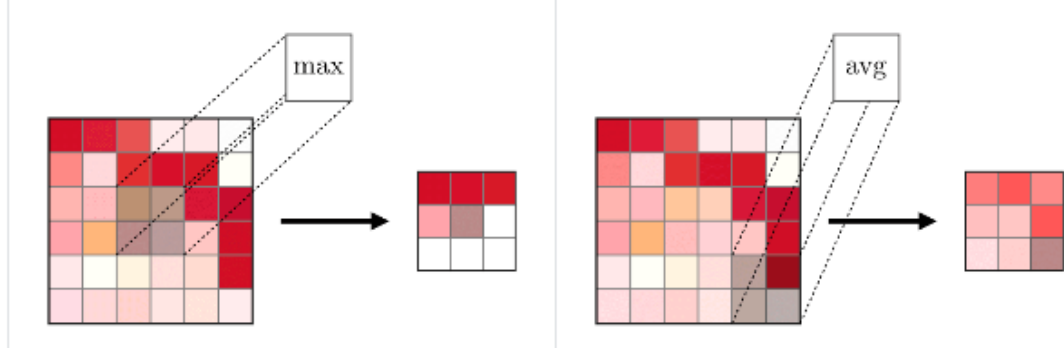
Şekil 5.5. ESA temel yapısı [41].

Burada girdi olarak diğer yapay sinir ağlarında olduğu gibi resmin piksel değerleri yer almaktadır. Daha sonra gelen evrişim katmanında resim üzerine bazı filtreler uygulanarak bazı özellikler tespit edilmeye çalışılır. Burada belirlenen çekirdekler resmin üzerinde kaydırılarak skaler çarpım yapıp özellik haritaları çıkartılır.



Şekil 5.6. Evrişim katmanı [42].

Birçok ESA temelli yapıda evrişim katmanından sonra biriktirme adı verilen bir katman gelmektedir. Bu katmanın amacı işlenecek veri miktarını verinin özelliklerini kaybetmeden azaltmaktır.

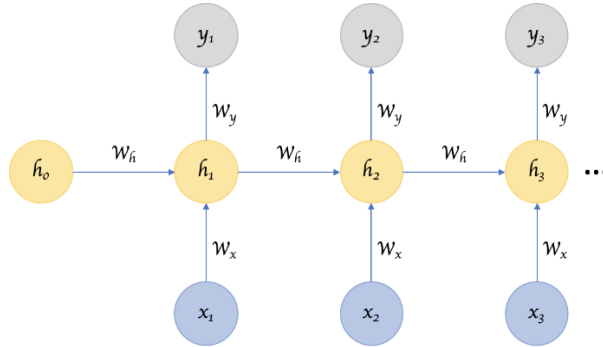


Şekil 5.7. Biriktirme katmanı [42].

ESA yapılarının sonuna doğru tam bağlantılı katmanlar kullanılır. Bu katmanın amacı daha önceki bölümlerde tespit edilen farklı özelliklerin toplanarak resmi daha güçlü bir şekilde ifade eden bir çıktı alınmasıdır [43].

5.4. YİNELEMELİ SİNİR AĞLARI

Birçok sinyal, doğası gereği doğrusal olmayan bir fiziksel mekanizmadan üretilir ve örneğin konuşma gibi istatistiksel olarak durağan olmayan özelliklere sahiptir [44]. Orijinal adı Recurrent Neural Networks olan bu yapı genellikle sıralı gelen örneğin doğal dil işleme, çeviri programları, video analizi, konuşma tanıma vb. gibi sıralı olarak gelen verilerin sınıflandırılmasında kullanılmaktadır. Bu yapıda her bir girdinin bir önceki girdi ile bir bağlantısı vardır. Bu sebeple sisteme giren verilerin içerdiği özellikler kadar sisteme geliş sırası da önemlidir.



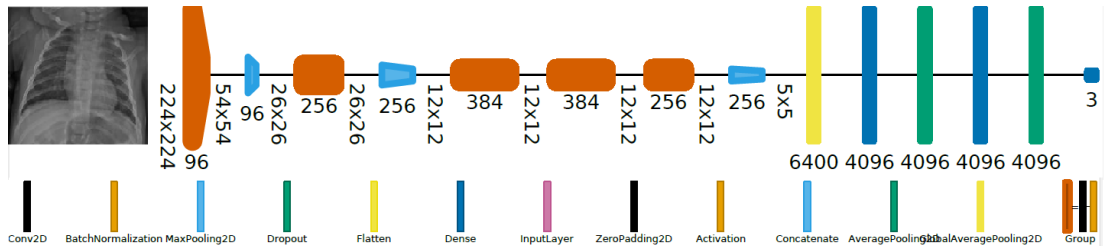
Şekil 5.8. Yinelemeli sinir ağı yapısı [45].

5.5. LİTERATÜRDEKİ POPÜLER DERİN ÖĞRENME ÇALIŞMALARI

Tez çalışması kapsamında geliştirilen ESA modeli, literatürde başarısını kanıtlamış beş popüler ESA modeli ile karşılaştırılmıştır. Bu modellerin mimarileri ilerleyen alt bölümlerde verilmektedir.

5.5.1. Alexnet

AlexNet, Krizhevsky vd. tarafından geliştirilen ve ilk olarak ImageNet LSVRC-2010 yarışmasında 1.2 milyon yüksek çözünürlükteki resmi 1000 farklı sınıfa top-1 ve top-5 hata oranı kıstaslarında sırasıyla 37.5% ve 17.0% değerlerini yakalayan ve daha sonra ILSVRC-2012 yarışmasında top-5 test hata oranında 15.3% değerini elde eden bir modeldir [46]. Bu yarışmada en yakın rakibinin hata oranı 26.2%'dir. Alexnet, 5 evrişim ve 3 tamamen bağlı katmandan oluşmaktadır. 1000 sınıfta 60 milyon üzerinde parametreye sahiptir.

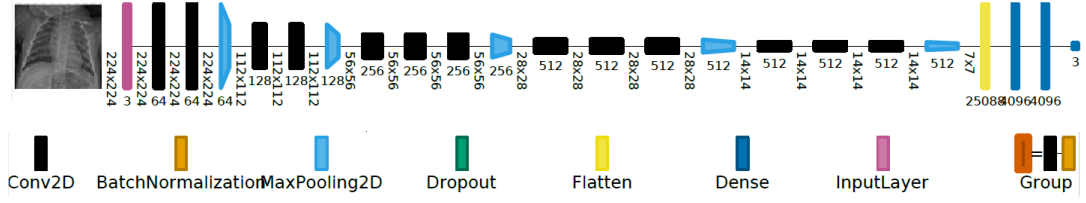


Şekil 5.9. Alexnet modeli.

Tez çalışması kapsamında son katmanı yeniden düzenlenen Alexnet modeli üç farklı sınıfta yaklaşık 45 milyon parametre içermektedir.

5.5.2. VGG16

VGG16 modelinin yapısı, Alexnet'in yapısına benzemektedir. Ancak bu modelde evrişim katmanlarındaki çekirdek boyutunu küçülterek ve katman sayısı artırılarak daha derin ve istikrarlı bir yapı oluşturulması hedeflenmiştir [47]. Adından da anlaşılacağı üzere toplamda 16 katmandan oluşmaktadır [48].



Şekil 5.10. VGG16 modeli.

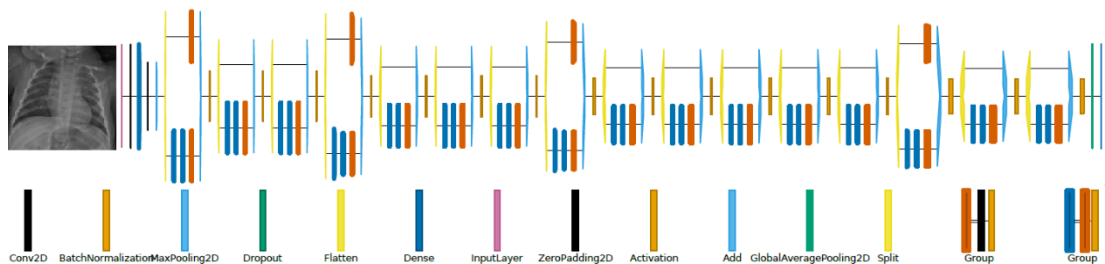
Tez çalışması kapsamında son katmanı yeniden düzenlenen VGG16 modeli üç farklı sınıfta yaklaşık 134 milyon parametre içermektedir.

5.5.3. Resnet50

Derin sinir ağlarının derinliği artırıldığında ağın öğrenme becerisi sığ katmanları bilginin kaybolmasından dolayı azalmaktadır. Buna bozulma sorunu (degradation problem) denir [49].

Microsoft Araştırma Merkezi mühendislerince geliştirilen bu modelde bazı bilgiler evrişim katmanına girmeden ilerdeki katmanlara aktarılarak bu bozulmanın önüne geçilmeye çalışılmıştır [50]. Bu model ILSVRC 2015 yarışmasında tasarımcılarını birinciliğe taşımıştır.

Resnet'in 34, 50, 101 ve 152 katmandan oluşan farklı biçimleri bulunmaktadır [51]. Performans ve işlem maliyeti açısından değerlendirildiğinde, tez çalışmasında 50 katmandan oluşan Resnet50 modeli kullanılmıştır.

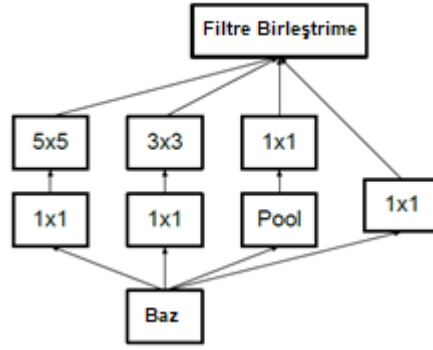


Şekil 5.11. Resnet50 modeli.

Tez çalışması kapsamında son katmanı yeniden düzenlenen Resnet50 modeli üç farklı sınıfta yaklaşık 23 milyon 500 bin parametre içermektedir.

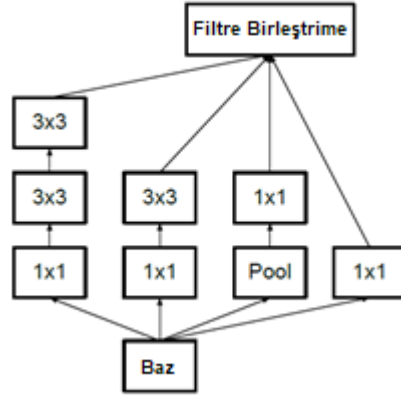
5.5.4. Inception V3

Google mühendisleri tarafından geliştirilen Inception V1 modelinin temelinde, aynı evrişim katmanında farklı boyutlarda filtrelerin uygulanması fikri bulunmaktadır [52]. Bunun amacı görüntünün içinde aranılan bilgi için uygulanacak filtre boyutunun kesin olarak bilinmemesidir. Ancak işlem maliyetinin yüksek olmasından dolayı filtreleri 1x1 boyutunda filtrelerle çarparak parametre sayısı azaltılarak işlem maliyeti düşürülmeye çalışılmıştır [52].

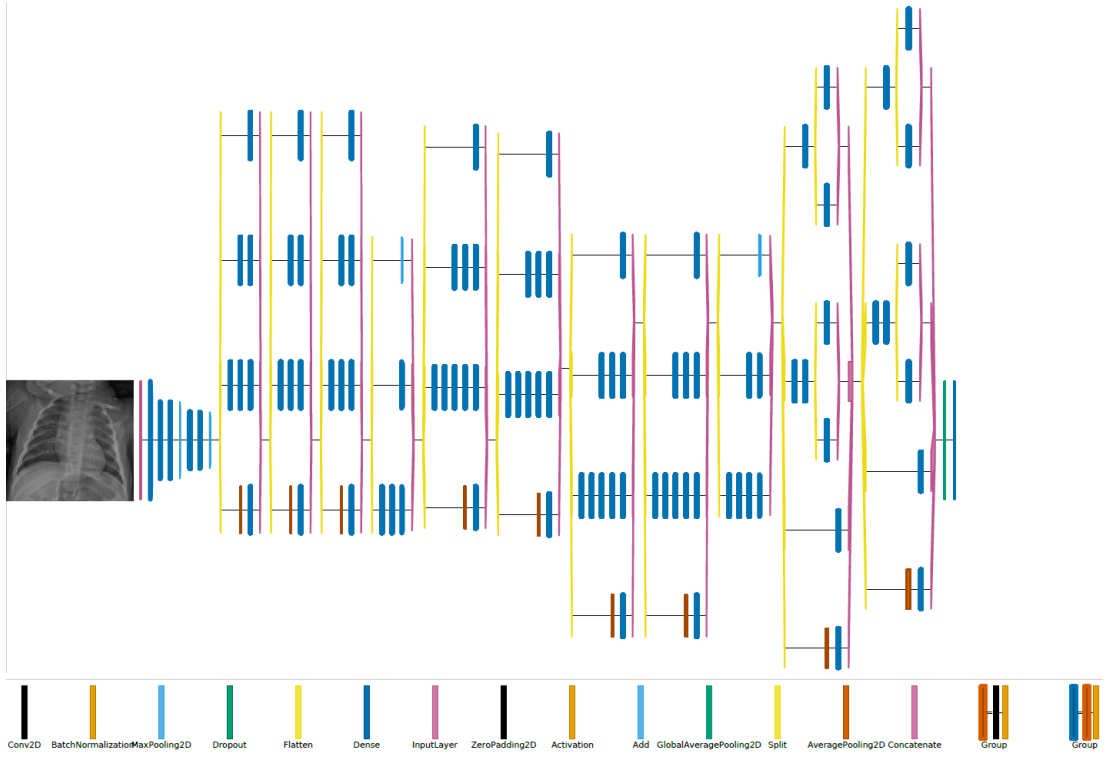


Şekil 5.12. Inception V1 evrişim katmanı [52].

Daha sonra modelin üçüncü versiyonunda 5x5 boyutunda filtre yerine 2 adet 3x3 boyutunda filtre kullanılarak parametre sayısı azaltılarak işlem yükü daha da hafifletilmiştir.



Şekil 5.13. Inception V3 evrişim katmanı [52].

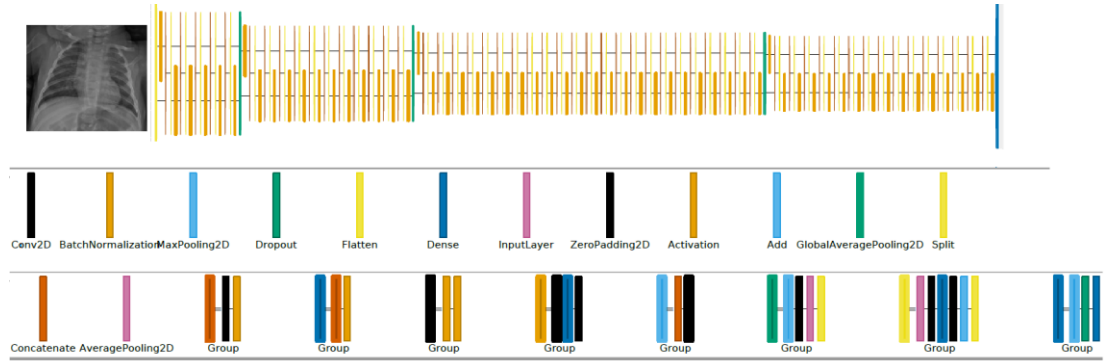


Şekil 5.14. Inception V3 modeli.

Tez çalışması kapsamında son katmanı yeniden düzenlenen InceptionV3 modeli üç farklı sınıfta yaklaşık 22 milyon parametre içermektedir.

5.5.5. Densenet

Derin ESA'ların başarısının katmanlar arası kısa bağlantılarla daha da arttığını gören araştırmacılar 2016 yılında bu modeli tasarlamışlardır. Densenet modelinde, sistemdeki her bir katmanın kendinden sonra gelen tüm katmanlarla bağlantısı bulunmaktadır.



Şekil 5.15. Densenet modeli [53].

Tez çalışması kapsamında Densenet adı altında bahsedilen model Densenet121 modelidir. Yine çalışma kapsamında son katmanı yeniden düzenlendiğinde, üç farklı sınıfta yaklaşık 7 milyon parametre içermektedir.

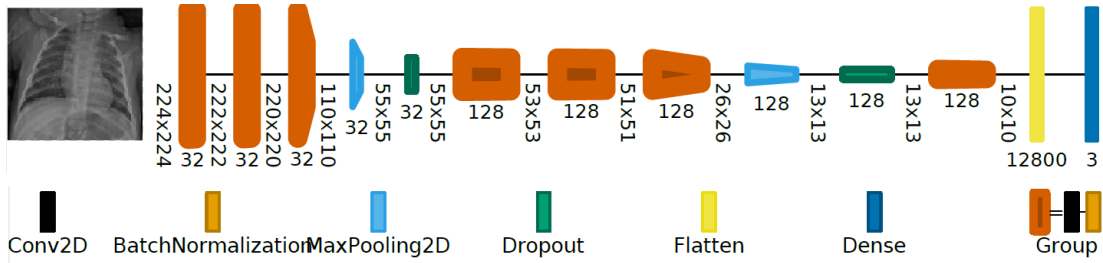
BÖLÜM 6

ÖNERİLEN YÖNTEM

Yapılan çalışmada kullanılan veri setinde yer alan akciğer röntgen görüntüleri temelde üç farklı kaynaktan elde edilmiştir. Kaynaklardan ilki 2021 yılında Chowdhury v.d'nin yapmış oldukları çalışmadır. Chowdhury v.d, çalışmalarında kullandıkları veri setini farklı kaynaklardan elde etmişlerdir. Veri seti, Covid-19, viral zatürre ve normal olmak üzere üç farklı sınıfta toplam 3487 görüntü dosyası içermektedir [54]. Bu veri setinden normal ve viral zatürre görüntüleri alınmıştır. İkinci veri kaynağı ise 2021 yılında Rahman v.d tarafından farklı kaynaklardan toplanan 3616 adet Covid-19 hastasının röntgen görüntüsünü içermektedir [55]. Bu çalışmada Covid-19 hastası olan ve olmayan diye iki sınıf üzerinde çalışıldığından sadece Covid-19 hastalarının görüntüleri alınmıştır. Bir diğer veri kaynağı ise 2018 yılında Kermay v.d'nin yaptıkları çalışmada kullanılmak için topladıkları 5232 hastanın röntgen görüntülerinden oluşmaktadır [56]. Bu görüntülerin 2538 tanesi bakterilere, 1345 tanesi virüse bağlı zatürre hastalarının, kalanlar da sağlıklı bireylerin akciğer görüntüleridir [56]. Bu veri setinden normal ve zatürre hastalarının akciğer görüntüleri alınmıştır. Tez çalışmasında kullanılan veri seti, yukarıda belirtilen kaynaklardan rastgele seçilen Covid-19, zatürre (viral ve bakteriyel) ve normal olmak üzere üç sınıfta, her bir sınıftan 1000 adet olmak üzere toplamda 3000 görüntü dosyasından oluşmaktadır. Görüntü dosyaları işlem süresini kısaltmak amacıyla $224*224*3$ boyutunda yeniden boyutlandırılmış ve piksel yeğinlik değerleri 0-1 arasına çekilerek normalize edilmiştir.

Tasarlanan ESA modelinde yedi adet evrişim katmanı ve bir adet tam bağlantılı katman bulunmaktadır. İlk üç evrişim katmanındaki filtre sayısı 32, çekirdek ölçüleri sırasıyla $3*3$, $3*3$ ve $5*5$ 'tir sonraki dört katmandaki filtre sayısı 128, çekirdek ölçüleri sırasıyla $3*3$, $3*3$, $5*5$, $4*4$ 'tür. Öğrenme katsayısı 0,001 olarak alınmıştır. Aşırı öğrenmeyi engellemek için her üç evrişim katmanından ve en son evrişim

katmanından sonra 0,4 oranında iletim sönümü yapılmıştır. Kayıp fonksiyonu olarak kategorik çapraz entropi, en iyileyici olarak Adam kullanılmıştır. Eğitim esnasında belli sayıda yineleme boyunca herhangi bir gelişme olmadığı durumda eğitimin durdurulması için erken durdurma uygulanmıştır. Ayrıca her bir yinelemede kayıp fonksiyonunun minimum değerini veren model kaydedilmiştir. Evrişim katmanlarında kaybolan eğitim problemi yaşamamak için aktivasyon fonksiyonu olarak Relu, tam bağlantılı son katmanda ise aktivasyon fonksiyonu olarak Softmax kullanılmıştır. Çalışma Kaggle ortamında GPU kullanılarak gerçekleştirilmiştir. Tasarlanan modelde toplamda 932 bin eğitilebilir parametre bulunmaktadır. Modelin şeması Şekil 6.1’de yer almaktadır.



Şekil 6.1. Önerilen yöntem ağ modeli.

Tasarlanan model Python dilinde kodlanmış olup Keras kütüphaneleri kullanılmıştır. Oluşturulan modelin özet bilgileri Çizelge 6.1’de yer almaktadır.

Çizelge 6.1. Model özet bilgisi.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
batch_normalization (Batch Normalization)	(None, 222, 222, 32)	128
conv2d_1 (Conv2D)	(None, 220, 220, 32)	9248
batch_normalization_1 (Batch Normalization)	(None, 220, 220, 32)	128
conv2d_2 (Conv2D)	(None, 110, 110, 32)	25632
batch_normalization_2 (Batch Normalization)	(None, 110, 110, 32)	128
max_pooling2d (MaxPooling2D)	(None, 55, 55, 32)	0
dropout (Dropout)	(None, 55, 55, 32)	0
conv2d_3 (Conv2D)	(None, 53, 53, 128)	36992
batch_normalization_3 (Batch Normalization)	(None, 53, 53, 128)	512
conv2d_4 (Conv2D)	(None, 51, 51, 128)	147584
batch_normalization_4 (Batch Normalization)	(None, 51, 51, 128)	512
conv2d_5 (Conv2D)	(None, 26, 26, 128)	409728
batch_normalization_5 (Batch Normalization)	(None, 26, 26, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 128)	0
dropout_1 (Dropout)	(None, 13, 13, 128)	0
conv2d_6 (Conv2D)	(None, 10, 10, 128)	262272
batch_normalization_6 (Batch Normalization)	(None, 10, 10, 128)	512
flatten (Flatten)	(None, 12800)	0
dropout_2 (Dropout)	(None, 12800)	0
dense (Dense)	(None, 3)	38403
Total params: 933,187		
Trainable params: 931,971		
Non-trainable params: 1,216		

BÖLÜM 7

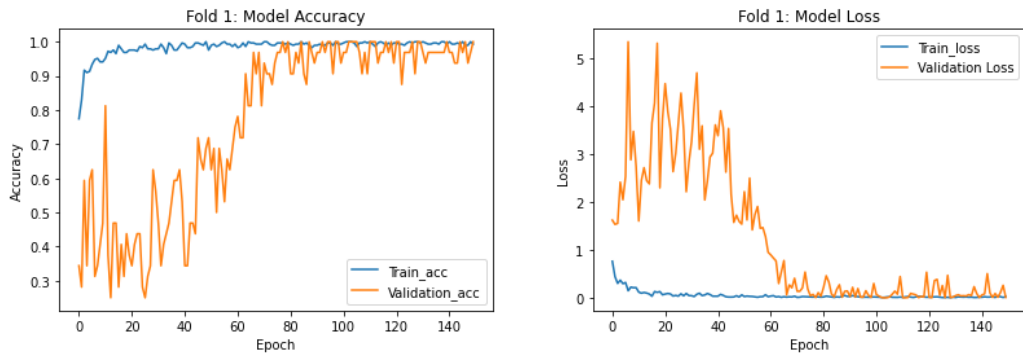
DENEYSEL SONUÇLAR VE DEĞERLENDİRME

Yapılan çalışmada 5-katlamalı çapraz doğrulama yöntemi uygulanmıştır. Veri setindeki 3000 görüntü her bir sınıf için 800 eğitim ve 200 test olacak şekilde Çizelge 7.1'deki gibi beş alt gruba ayrılmıştır. 5-katlamalı çapraz doğrulama tekniği ile elde edilen modellerin performansı ölçülmüştür.

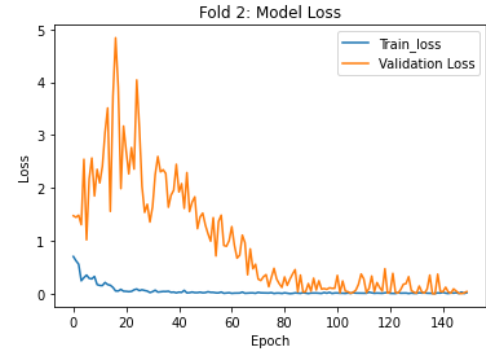
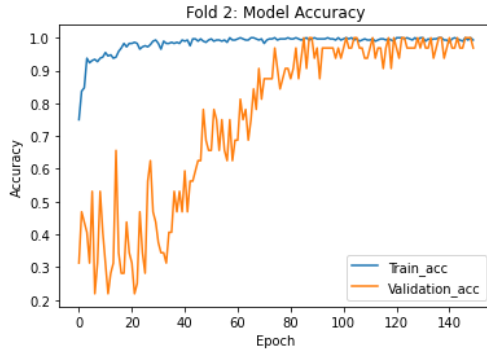
Çizelge 7.1. 5-katlamalı çapraz doğrulama için eğitim ve test verisi dağılımları.

Katlama-1	Test (600)	Eğitim (600)	Eğitim (600)	Eğitim (600)	Eğitim (600)
Katlama -2	Eğitim (600)	Test (600)	Eğitim (600)	Eğitim (600)	Eğitim (600)
Katlama -3	Eğitim (600)	Eğitim (600)	Test (600)	Eğitim (600)	Eğitim (600)
Katlama -4	Eğitim (600)	Eğitim (600)	Eğitim (600)	Test (600)	Eğitim (600)
Katlama -5	Eğitim (600)	Eğitim (600)	Eğitim (600)	Eğitim (600)	Test (600)

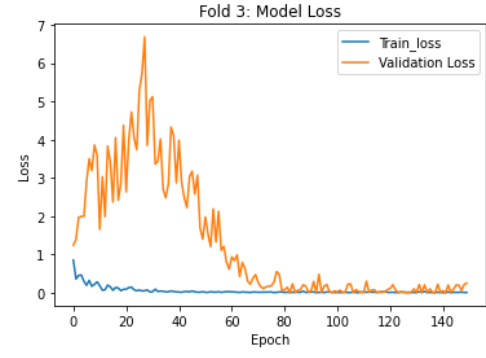
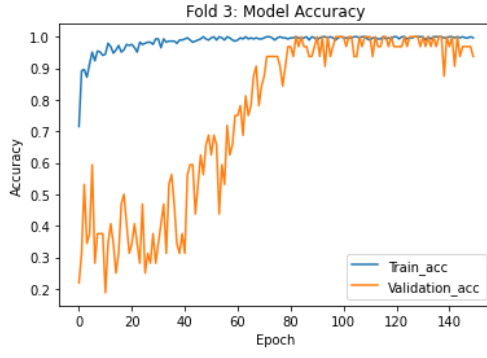
Her bir katlama için 150 olmak üzere toplamda 750 yinleme eğitim yapılmıştır. Her bir katlama sırasında hesaplanan doğruluk ve kayıp grafikleri Şekil 7.1-7.5'te yer almaktadır.



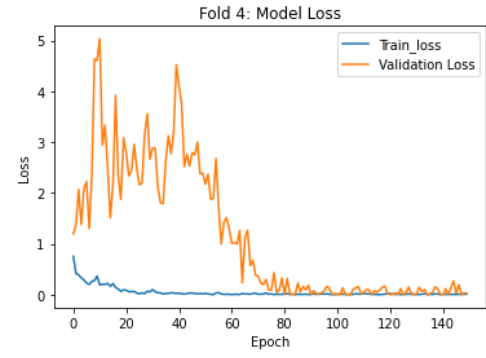
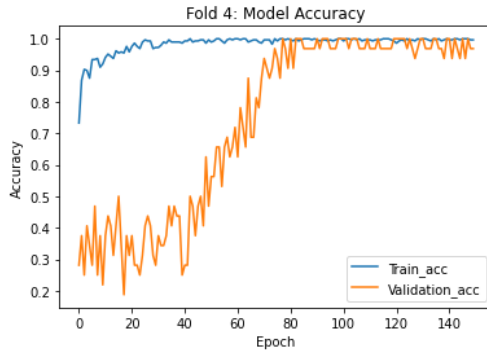
Şekil 7.1. Katlama-1 doğruluk ve kayıp grafiği.



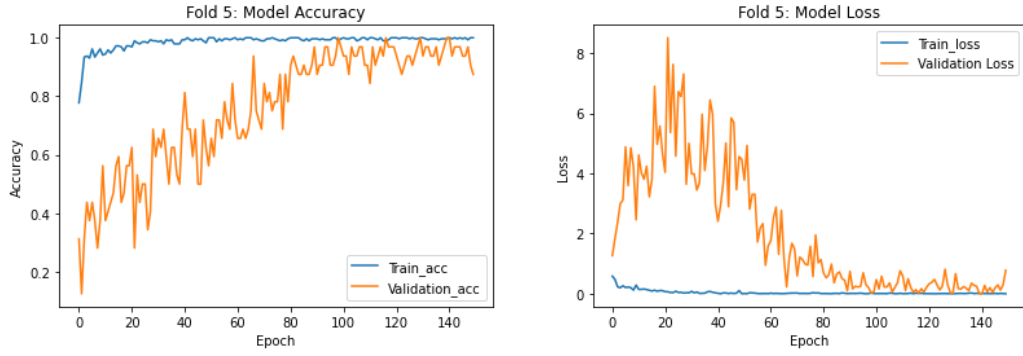
Şekil 7.2. Katlama-2 doğruluk ve kayıp grafiği.



Şekil 7.3. Katlama-3 doğruluk ve kayıp grafiği.



Şekil 7.4. Katlama-4 doğruluk ve kayıp grafiği.



Şekil 7.5. Katlama-5 doğruluk ve kayıp grafiği.

Beşinci bölümün sonunda yer alan literatürdeki diğer popüler modeller de aynı veri seti ile 0.001 öğrenme katsayısıyla eğitilmiş, kayıp fonksiyonu olarak kategorik çapraz entropi, en iyileyici olarak Adam kullanılmıştır. Tüm modellerin performansları (7.1) denklemlerindeki metriklere göre hesaplanmış olup sonuçlar Çizelge-3'te yer almaktadır.

Başarım kavramı modelin ne derece doğru tahmin yapabildiğini gösterir. Doğru tahmin edilen örnek sayısının tüm örnek sayısına oranıdır. Örneklerin düzgün dağılmadığı durumlarda gerçekçi sonuç vermeyebilir.

$$Başarım = \frac{Doğru Pozitif + Doğru Negatif}{Tüm Örnekler} \quad (7.1a)$$

Kesinlik kavramı, özellikle örneklerin düzgün dağılmadığı ve yanlış pozitif durumlarının tehlikeli olduğu durumlarda önem kazanmaktadır. Doğru tahmin edilen pozitif örnek sayısının pozitif tahmin edilen tüm örnek sayısına oranıdır.

$$Kesinlik = \frac{Doğru Pozitif}{Doğru Pozitif + Yanlış Pozitif} \quad (7.1b)$$

Hassasiyet kavramı da kesinlik kavramı gibi örnek dağılımının düzgün olmadığı durumlarda daha çok tercih edilir. Ancak kesinlikten farklı olarak yanlış negatif durumlarının tehlikeli olduğu durumlarda önem kazanmaktadır. Doğru tahmin edilen pozitif örnek sayısının tüm pozitif örnek sayısına oranıdır.

$$Hassasiyet = \frac{Doğru Pozitif}{Doğru Pozitif + Yanlış Negatif} \quad (7.1c)$$

F1-Skoru ise kesinlik ve hassasiyet değerlerinin harmonik ortalamasıdır. Hem yanlış negatif hem de yanlış pozitif durumlarının tehlikeli olduğu durumlarda kullanımı yaygındır.

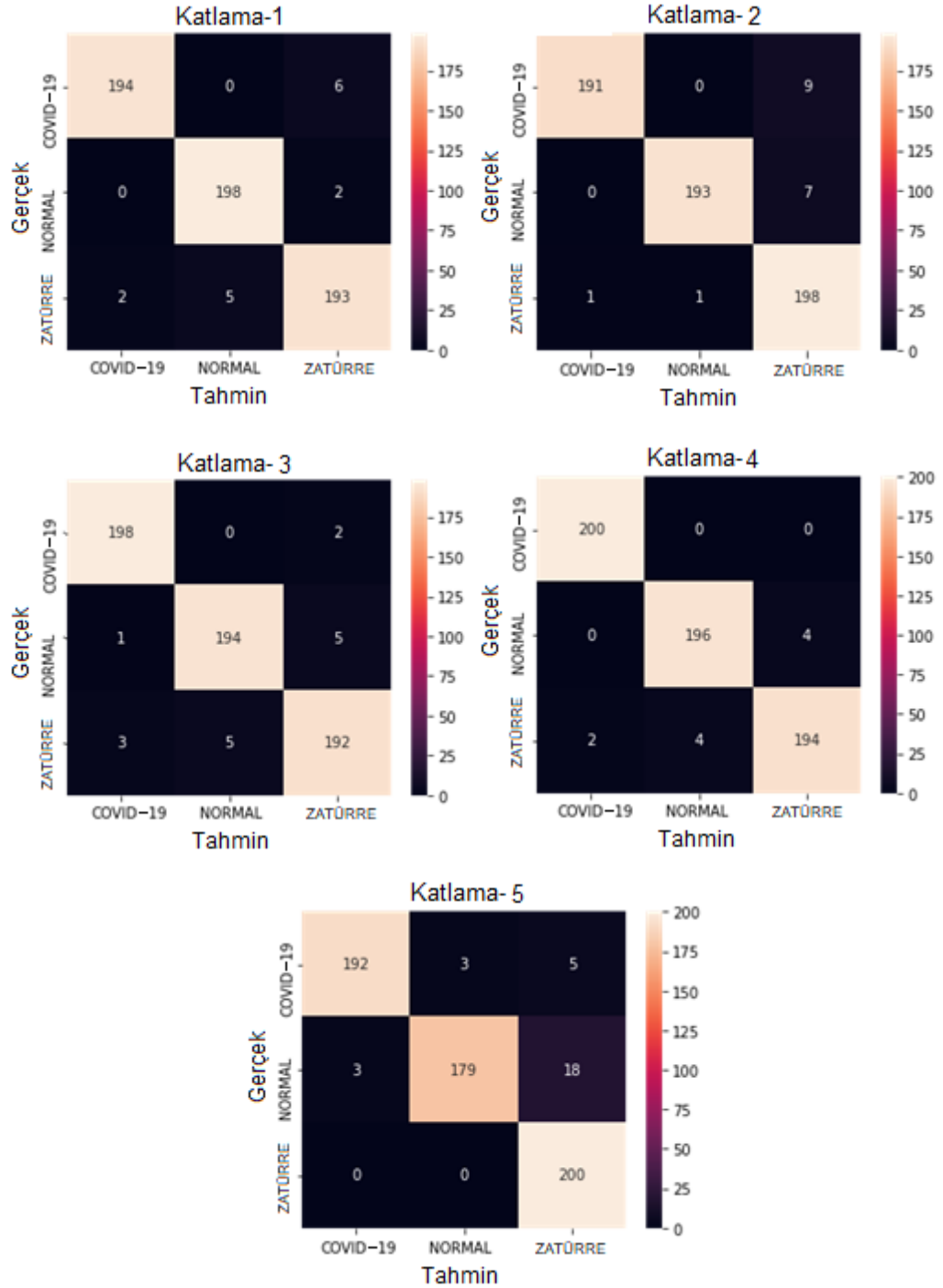
$$F1 - Skoru = \frac{Kesinlik * Hassasiyet}{Kesinlik + Hassasiyet} \quad (7.1d)$$

Verilen tüm metriklerde değerler 0-1 aralığında çıkacaktır. Sıfıra yakın sonuçlar, modelin o metriğe göre performansının düşük olduğunu, bire yakın sonuçlar ise modelin o metriğe göre performansının yüksek olduğunu gösterir.

Çizelge 7.2. Uygulama sonuçları.

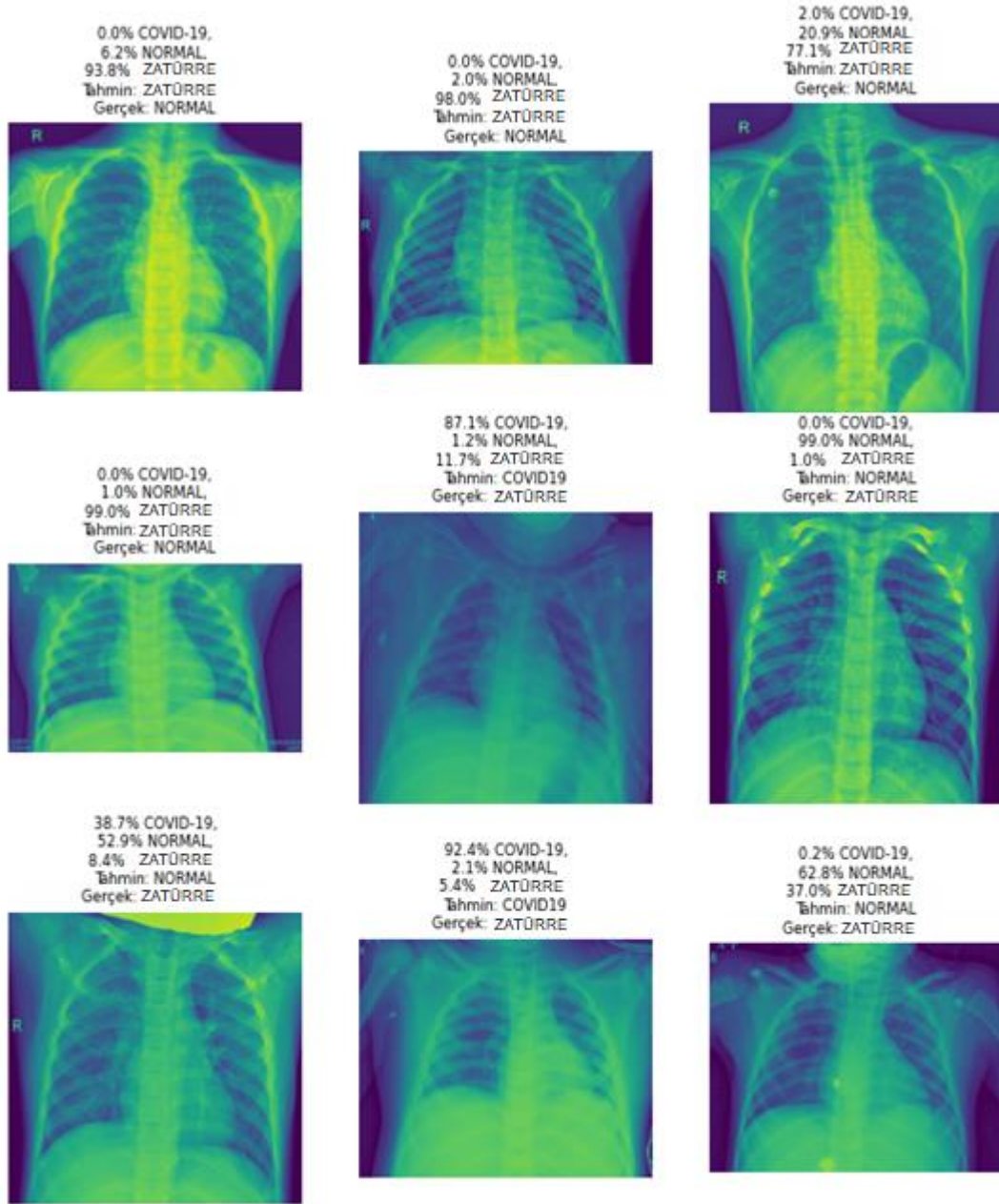
		Tasarlanan Yöntem %	Alexnet %	VGG16 %	Resnet50 %	Inception V3 %	Densenet %
Katlama-1	Başarım	97.500	96.667	33.330	98.000	98.167	97.333
	Kesinlik	97.512	96.681	16.670	98.005	98.207	97.344
	Hassasiyet	97.500	96.667	33.330	98.000	98.167	97.333
	F1-Skoru	97.506	96.674	16.670	98.002	98.187	97.339
Katlama-2	Başarım	97.000	96.333	97.167	98.500	98.167	98.667
	Kesinlik	97.162	96.383	97.182	98.508	98.175	98.679
	Hassasiyet	97.000	96.333	97.167	98.500	98.167	98.667
	F1-Skoru	97.081	96.358	97.174	98.504	98.171	98.673
Katlama-3	Başarım	97.333	96.667	96.333	95.833	97.000	97.333
	Kesinlik	97.330	96.686	96.346	95.910	97.007	97.384
	Hassasiyet	97.333	96.667	96.333	95.833	97.000	97.333
	F1-Skoru	97.332	96.676	96.340	95.872	97.003	97.359
Katlama-4	Başarım	98.333	97.000	97.500	99.167	98.333	98.500
	Kesinlik	98.330	97.022	97.500	99.175	98.354	98.546
	Hassasiyet	98.333	97.000	97.500	99.167	98.333	98.500
	F1-Skoru	98.332	97.011	97.500	99.171	98.344	98.523
Katlama-5	Başarım	95.167	94.500	93.000	95.333	95.167	96.667
	Kesinlik	95.500	94.703	93.266	95.564	95.584	96.873
	Hassasiyet	95.167	94.500	93.000	95.333	95.167	96.667
	F1-Skoru	95.333	94.601	93.133	95.449	95.375	96.770
Ortalama	Başarım	97.067	96.233	83.466	97.367	97.367	97.700

Uygulamada sonucunda her bir katlama işleminde elde edilen karmaşıklık matrisleri çıkartılmış olup Şekil 7.6’da yer almaktadır.

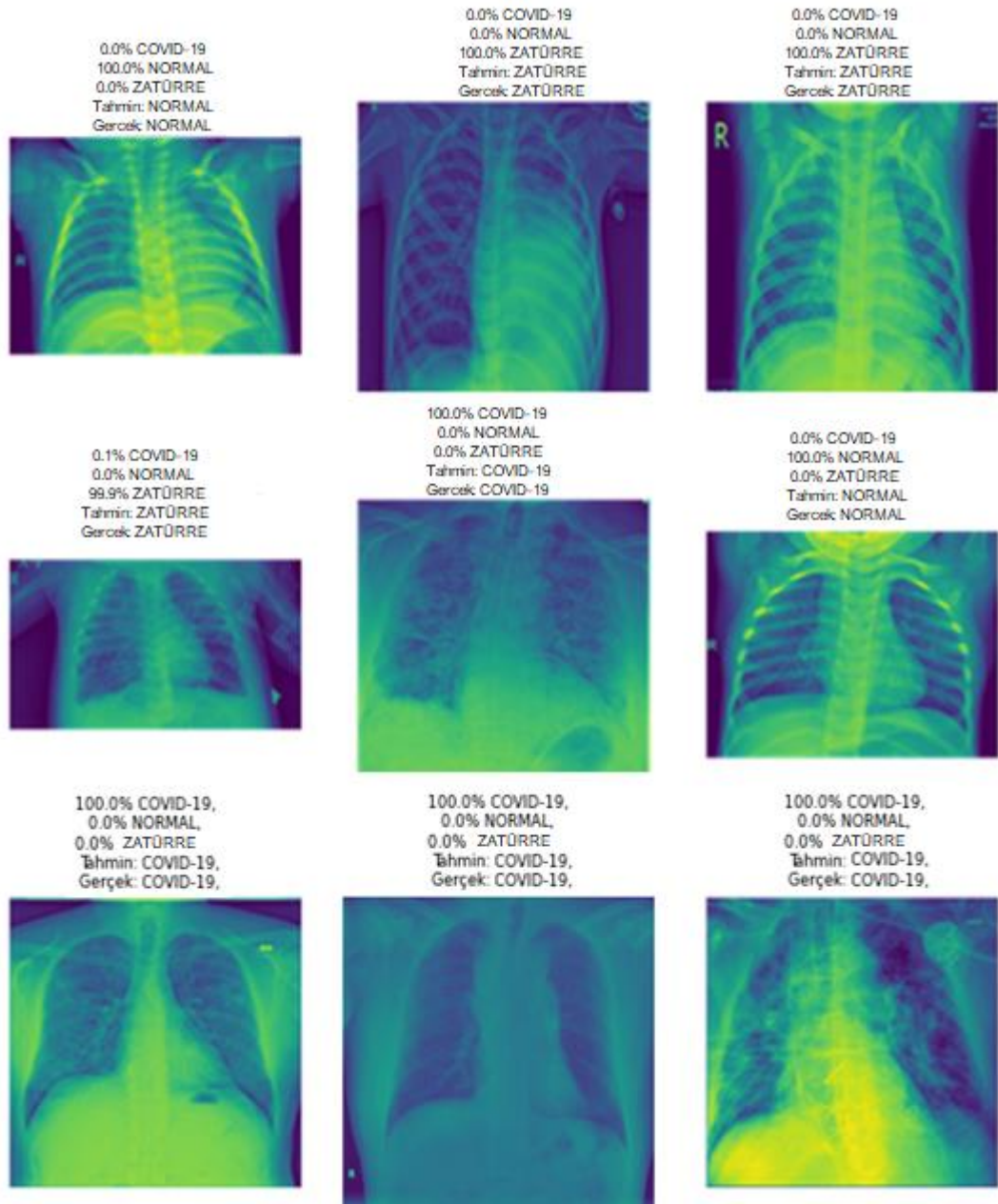


Şekil 7.6. Uygulama karmaşıklık matrisleri.

Her bir katlama işleminde elde edilen modeller test için ayrılan 600 resimden 10 ila 29'unu hatalı tahmin etmiştir. Hatalı ve doğru sınıflandırılan bazı resimler Şekil 7.7 ve Şekil 7.8'de verilmiştir.



Şekil 7.7. Hatalı sınıflandırılan resimler.



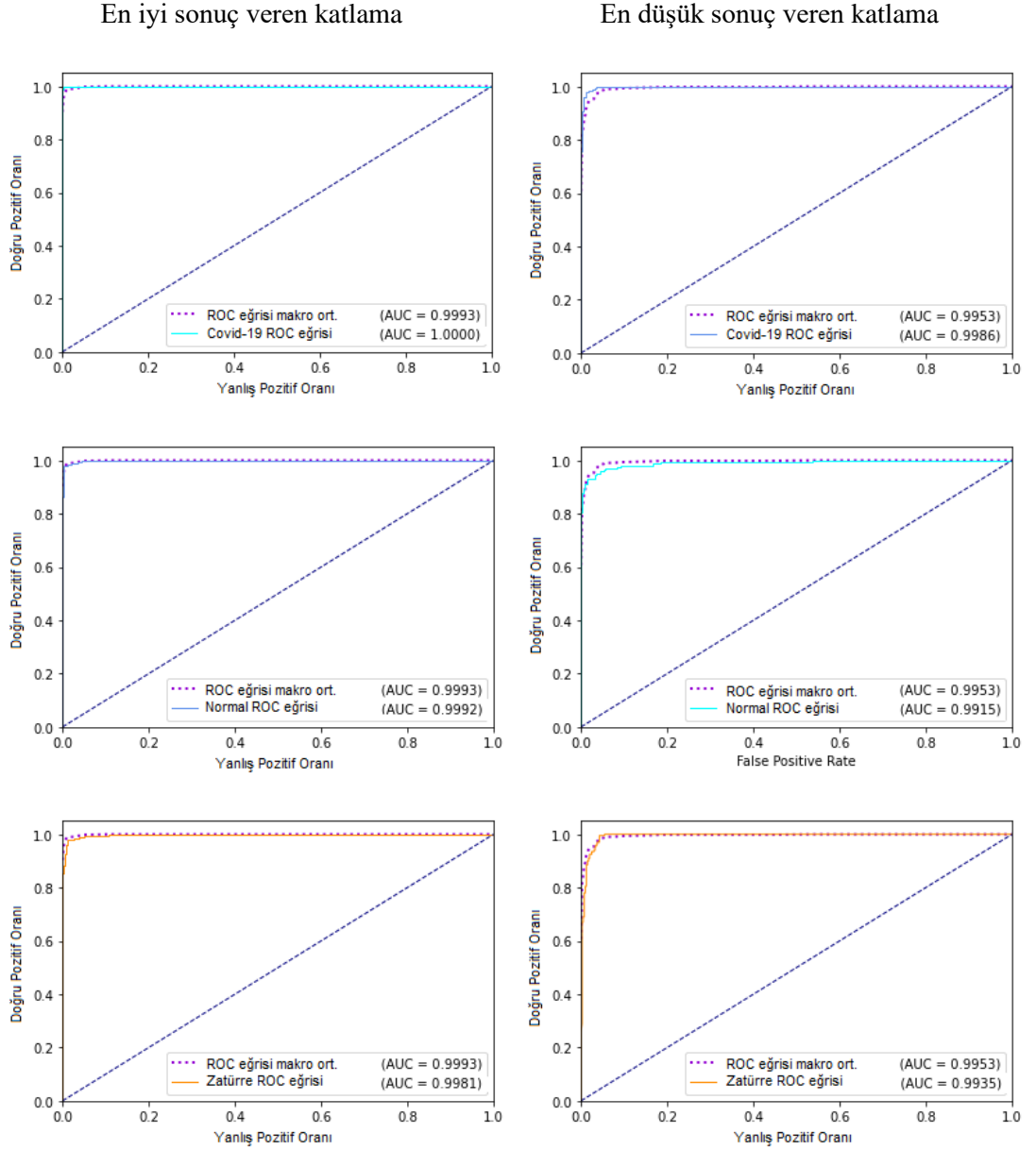
Şekil 7.8. Doğru sınıflandırılan bazı resimler.

Özellikle sağlık alanında yapılan çalışmalarda sıklıkla hesaplanan ROC eğrileri ve AUC değerleri için (7.2) denklemleri kullanılmıştır.

$$\text{Doğru Pozitif Oranı} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \quad (7.2a)$$

$$\text{Yanlış Pozitif Oranı} = \frac{\text{Yanlış Pozitif}}{\text{Yanlış Pozitif} + \text{Doğru Negatif}} \quad (7.2b)$$

ROC eğrisinin altında kalan alan AUC değerini vermektedir.



Şekil 7.9. ROC eğrileri ve AUC değerleri.

Şekil 7.9’da her bir sınıf için en düşük ve en yüksek doğruluk değeri veren katlamalardaki ROC eğrileri ve AUC değerleri verilmiştir.

BÖLÜM 8

SONUÇ VE TARTIŞMA

Çalışmada, günümüzde tüm dünyayı etkisi altına alan ve hızla yayılan Covid-19 hastalığının teşhis edilmesinde yardımcı olabilecek yüksek doğruluk oranına sahip bir yöntem geliştirmek amaçlanmıştır. Çalışmalar neticesinde elde edilen yöntem, üç farklı sınıfta ortalama %97 doğrulukla Covid-19 hastalığı ve zatürre tespiti yapabilmektedir. Hastalığın ilk safhalarında akciğerde Covid-19 belirtileri görülmediğinden ve çalışma tıp ana bilim dalında yapılmadığından kesin teşhis aracı olarak kullanılamasa da hali hazırda hastalığının teşhisinde kullanılan PCR testleri ile kıyaslandığında daha yüksek doğruluk oranı, düşük maliyet ve kısa işlem süresi dikkate alındığında yardımcı bir yöntem olarak kullanılabilir. Örneğin acile gelen hastaların ön muayene sonrasında istenecek bir röntgen görüntüsü üzerinden saniyeler içinde verdiği sonuçla, karar verecek olan doktora bir fikir verebilir. Ayrıca günümüzde her hastanede PCR testi verme imkânı bulunmamasına rağmen neredeyse tüm hastanelerde röntgen cihazı bulunmaktadır. Bu açıdan da bakıldığında yine hastalığın teşhisinde karar vericilere yardımcı olabilir.

Gelecekte ise bu model, röntgen cihazının bilgisayarına entegre edilerek röntgenin çekildiği anda hasta ile ilgili tahminde bulunarak doktorları desteklemesi sağlanabilir.

KAYNAKLAR

1. İnternet: Imaging Technology News "COVID-19 Genetic PCR Tests Give False Negative Results if Used Too Early", <https://www.itnonline.com/content/covid-19-genetic-pcr-tests-give-false-negative-results-if-used-too-early> (2021).
2. Khan, A. I., Shah, J. L., & Bhat, M. M., "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images", *Computer Methods and Programs in Biomedicine*, 196, (2020).
3. Khan, S. H., Sohail, A., Khan, A., Hassan, M., Lee, Y. S., Alam, J., ... & Zubair, S. , "COVID-19 Detection in Chest X-ray Images using Deep Boosted Hybrid Learning". *Computers in Biology and Medicine*, 137 (2021).
4. Albahli, S., Ayub, N., & Shiraz, M., "Coronavirus disease (COVID-19) detection using X-ray images and enhanced DenseNet", *Applied Soft Computing*, 110 (2021).
5. Gilanie, G., Bajwa, U. I., Waraich, M. M., Asghar, M., Kousar, R., Kashif, A., ... & Rafique, H. "Coronavirus (COVID-19) detection from chest radiology images using convolutional neural networks", *Biomedical Signal Processing and Control*, 66 (2021).
6. Okolo, G. I., Katsigiannis, S., Althobaiti, T., & Ramzan, N. "On the Use of Deep Learning for Imaging-Based COVID-19 Detection Using Chest X-rays", *Sensors*, 21(17): 5702 (2021).
7. E. Ohata, "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning", *IEEE/CAA Journal of. Automatica Sinica*, 8(1), 239-248, (2020).
8. Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., & Acharya, U. R., "Automated detection of COVID-19 cases using deep neural networks with X-ray images", *Computers in biology and medicine*, 121, (2020).
9. Li, L., Qin, L., Xu, Z., Yin, Y., Wang, X., Kong, B., ... & Xia, J., "Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT", *Radiology* (2020).
10. Brunese, L., Mercaldo, F., Reginelli, A., & Santone, A., "Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays", *Computer Methods and Programs in Biomedicine*, 196 (2020).

11. Wang, L., Lin, Z. Q., & Wong, A., "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images", *Scientific Reports*, 10(1): 1-12 (2020).
12. İnternet: T.C. Sağlık Bakanlığı, "COVID-19 Nedir ?", <https://covid19.saglik.gov.tr/TR-66300/covid-19-nedir-.html> (2021).
13. İnternet: Anadolu Ajansı, "Sağlık, Koronavirüs", <https://www.aa.com.tr/tr/koronavirus/saglik-bakani-yardimcisi-mese-PCR-testinin-yuzde-90in-uzerinde-duyarlilik-var/1917633> (2021).
14. Bernheim, A., Mei, X., Huang, M., Yang, Y., Fayad, Z. A., Zhang, N., ... & Chung, M. "Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection", *Radiology*, 200463 (2020).
15. Xie, X., Zhong, Z., Zhao, W., Zheng, C., Wang, F., & Liu, J., "Chest CT for typical coronavirus disease 2019 (COVID-19) pneumonia: relationship to negative RT-PCR testing", *Radiology*, 296(2): E41-E45 (2020).
16. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., ... & Ng, A. Y. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning", *arXiv preprint arXiv:1711.05225* (2017).
17. Frija-Masson, J., Debray, M. P., Boussouar, S., Khalil, A., Bancal, C., Motiejunaite, J., ... & Gonzalez-Bermejo, J. "Residual ground glass opacities three months after Covid-19 pneumonia correlate to alteration of respiratory function", *The post Covid M3 study. Respiratory medicine*, 184, 106435 (2021).
18. Rasche C. "Computer Vision An Overview For Enthusiasts", Bükreş, 9-10 (2021).
19. İnternet: Express Analytics, "10 Things about Computer Vision and Data Analytics You May Not Know", <https://expressanalytics.com/resources/topics/miscellaneous/10-things-about-computer-vision-and-data-analytics-you-may-not-know/> (2021).
20. İnternet: Facebook Research, "Learning to Segment", <https://research.fb.com/blog/2016/08/learning-to-segment/> (2021).
21. İnternet: Dynamsoft, "Image Processing", https://www.dynamsoft.com/Documents/image%20processing_compressed.pdf (2021).
22. Hambal, A. M., Pei, Z., & Ishabailu, F. L., "Image noise reduction and filtering techniques", *International Journal of Science and Research (IJSR)*, 6(3), 2033-2038 (2017).

23. Wehle, H. D. "Machine learning, deep learning, and ai: What's the difference?", *International Conference on Data scientist innovation day*, Brüksel (2017).
24. Goodfellow, I., Bengio, Y., & Courville, A. "Deep learning", *MIT press*, Cambridge, 98-99 (2016).
25. Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., ... & Zhang, J. D. "An introduction to machine learning", *Clinical Pharmacology & Therapeutics*, 107(4), 871-885 (2020).
26. İnternet: Microbiome Summer School, "Introduction to Machine Learning", <https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/> (2021).
27. İnternet: Alegion, " Supervised vs. unsupervised learning", https://orgs.mines.edu/daa/wp-content/uploads/sites/38/2019/07/Supervised_vs_Unsupervised_Learning.pdf (2021).
28. İnternet: Imperva, " Clustering and Dimensionality Reduction: Understanding the "Magic" Behind Machine Learning", <https://www.imperva.com/blog/clustering-and-dimensionality-reduction-understanding-the-magic-behind-machine-learning/> (2021).
29. Öztemel, E. Yapay sinir ağları. *PapatyaYayincilik*, Istanbul, 100-120 (2003).
30. Aggarwal, C. C., Neural networks and deep learning. *Springer*, 10, New York, 25-50 (2018).
31. Feng, J., He, X., Teng, Q., Ren, C., Chen, H., & Li, Y., "Reconstruction of porous media from extremely limited information using conditional generative adversarial networks", *Physical Review* (2019).
32. Zupan, J. "Introduction to artificial neural network (ANN) methods: what they are and how to use them", *Acta Chimica Slovenica*, 41, 327-327, Tarragona (1994).
33. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. "Dive into deep learning", *arXiv preprint arXiv:2106.11342* (2021).
34. Demyanov, S., "Regularization methods for neural networks and related models", Doctoral Dissertation, *Department of Computing and Information Systems, The University of Melbourne*, Melbourne, 48 (2015).

35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. "Dropout: a simple way to prevent neural networks from overfitting", *The journal of machine learning research*, 1929-1958, Kanada (2014).
36. Shin, S., Hoang, T. T., Le, T. H., & Lee, M. Y., "A new robust design method using neural network", *Journal of Nanoelectronics and Optoelectronics* (2016).
37. İnternet: GeeksforGeeks, "ML | Stochastic Gradient Descent (SGD)", <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/> (2021).
38. İnternet: GeeksforGeeks, "Intuition behind Adagrad Optimizer", <https://www.geeksforgeeks.org/intuition-behind-adagrad-optimizer/> (2021).
39. İnternet: Facebook AI, "RMSProp", <https://paperswithcode.com/method/rmsprop> (2021).
40. Kingma, D. P., & Ba, J. "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*. (2014).
41. O'Shea, K., & Nash, R. "An introduction to convolutional neural networks", *arXiv preprint* (2015).
42. İnternet: Stanford University "Convolutional Neural Networks", <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks#overview> (2021).
43. Wu, J., "Introduction to convolutional neural networks", *National Key Lab for Novel Software Technology*. Nanjing, 2-3 (2017).
44. Mandic, D., & Chambers, J., "Recurrent neural networks for prediction", *Wiley*, Chichester, 5-10 (2001).
45. İnternet: Gotensor, "Recurrent Neural Networks – Remembering what’s important", <https://gotensor.com/2019/02/28/recurrent-neural-networks-remembering-whats-important/> (2021).
46. Krizhevsky, A., Sutskever, I., & Hinton, G. E., "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, 25, 1097-1105 (2012).
47. Simonyan, K., & Zisserman, A., "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556* (2014).
48. İnternet: Neurohive, "VGG16 – Convolutional Network for Classification and Detection", <https://neurohive.io/en/popular-networks/vgg16/> (2021).

49. Internet: Towards Data Science, "Intuition behind Residual Neural Networks", <https://towardsdatascience.com/intuition-behind-residual-neural-networks-fa5d2996b2c7> (2021).
50. He, K., Zhang, X., Ren, S., & Sun, J., "Deep residual learning for image recognition", *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778) (2016).
51. Yeung S., "Artificial Intelligence in Healthcare Lecture Notes", *Stanford University* (2020).
52. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. "Rethinking the inception architecture for computer vision", *IEEE conference on computer vision and pattern recognition* (pp. 2818-2826) (2016).
53. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q., "Densely connected convolutional networks. In Proceedings of the *IEEE conference on computer vision and pattern recognition*, Washington, 4700-4708 (2017).
54. Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., ... & Islam, M. T. , "Can AI help in screening viral and COVID-19 pneumonia?", *IEEE Access*, 8 (2020).
55. Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S. B. A., ... & Chowdhury, M. E., "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images", *Computers in biology and medicine*, 132 (2021).
56. Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., ... & Zhang, K., "Identifying medical diagnoses and treatable diseases by image-based deep learning", *Cell*, 172(5) (2018).

ÖZGEÇMİŞ

Ahmet Korkmaz ilk ve orta öğrenimini Karabük’te tamamladı. Safranbolu İmam Hatip Lisesi’nden 1999 yılında mezun oldu. 2000 yılında Erciyes Üniversitesi Nevşehir İİBF İşletme Bölümüne başlayıp 2005 yılında fakülte birincisi olarak mezun oldu. 2006 yılında askerlik hizmetini tamamladı. Aynı yıl Türkiye Finans Katılım Bankası Kızılay Şubesi’nde işe başladı. 2008 yılında Türk Telekomünikasyon A.Ş. Karabük İl Müdürlüğünde çalışmaya başladı. 2013 yılında Karabük Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümüne başlayıp 2017 yılında iyi bir derece ile mezun oldu. 2019 yılında Karabük Üniversitesi Lisansüstü Eğitim Enstitüsünde Bilgisayar Mühendisliği Anabilim Dalı’da yüksek lisans eğitimine başladı. Halen Türk Telekomünikasyon A.Ş.’de çalışmaya devam etmekte olan Ahmet Korkmaz evli ve üç çocuk babasıdır.