



T.C.

ALTINBAŞ ÜNİVERSİTESİ

Lisansüstü Eğitim Enstitüsü

Bilişim Teknolojileri

**DERİN ÖĞRENME MODELİ İLE ZATÜRRE
TESPİTİ**

Uraz Kağan GÜNEŞ

Yüksek Lisans Tezi

Danışman

Doç. Dr. Oğuz ATA

İstanbul, 2025

DERİN ÖĞRENME MODELİ İLE ZATÜRRE TESPİTİ

Uraz Kağan GÜNEŞ

Bilişim Teknolojileri

Yüksek Lisans Tezi

ALTINBAŞ ÜNİVERSİTESİ

2025

Uraz Kağan GÜNEŞ tarafından hazırlanmış ve 24/06/2025 tarihinde sunulmuş DERİN ÖĞRENME İLE ZATÜRRE TESPİTİ başlıklı tez Bilişim Teknolojileri Ana Bilim Dalında Yüksek Lisans Tezi olarak **oy birliği** ile kabul edilmiştir.

Doç. Dr. Oğuz ATA

Danışman

Tez Savunma Sınavı Jüri Üyeleri:

Doç. Dr. Oğuz ATA

Yazılım Mühendisliği

Bölümü,

Altınbaş Üniversitesi

Doç. Dr. Aytuğ BOYACI

Bilgisayar Mühendisliği

Bölümü,

Milli Savunma Üniversitesi

Doç. Dr. Can EYÜPOĞLU

Bilgisayar Mühendisliği

Bölümü,

Milli Savunma Üniversitesi

Bu tezin Yüksek Lisans tezi olarak bütün şartları sağladığımı beyan ederim.

Bu tezdeki tüm bilgilerin akademik kurallara ve etik davranışlara uygun olarak edinildiğini ve sunulduğunu beyan ederim. Ayrıca, bu kuralların ve davranışların gerektirdiği şekilde, bu çalışmada, orijinal olmayan tüm materyalleri ve sonuçları tamamen alıntı yaptığımı ve referans gösterdiğimi de beyan ederim.

Uraz Kağan GÜNEŞ

İmzası

XXXXXX

İTHAF

Bu çalışmayı, hayatım boyunca beni koşulsuz destekleyen, sevgisi ve fedakârlığıyla her zaman yanımda olan kıymetli aileme ithaf ediyorum. Her adımda sabır ve anlayışla yanımda olan, desteğiyle bana güç veren sevgili anne ve babama; hayat yolculuğunda bana ilham kaynağı olan kardeşime en derin teşekkürlerimi sunuyorum. Akademik sürecimde bilgi ve tecrübesiyle yolumu aydınlatan, rehberliğiyle çalışmamın şekillenmesinde büyük rol oynayan değerli danışmanım Doç. Dr. Oğuz Ata'ya minnetle teşekkür ederim. Ayrıca, bu süreçte manevi desteğini esirgemeyen tüm yakınlarıma ve yanımda olduklarını hissettiren dostlarıma içtenlikle teşekkür ederim.



ÖZET

DERİN ÖĞRENME MODELİ İLE ZATÜRRE TESPİTİ

GÜNEŞ, Uraz Kağan

Yüksek Lisans, Bilişim Teknolojileri, Altınbaş Üniversitesi

Danışman: Doç. Dr. Oğuz ATA

Tarih: 06/2025

Sayfa: 154

Zatürre, her bireyin yaşamının herhangi bir döneminde maruz kalabileceği bir akciğer enfeksiyonudur. Zatürre, akciğerleri hedef alan ve solunum yollarında ciddi enfeksiyonlara neden olan bir hastalıktır. Hastalık, hava keseciklerinin irin ve sıvılarla dolmasına yol açarak hayati tehlike oluşturabilir. Tedavi edilmediği takdirde ciddi sağlık sorunlarına ve hatta ölüme neden olabilir. Bu nedenle, hastalığın erken teşhisi büyük bir öneme sahiptir. Zatürrenin tespitinde çeşitli yöntemler kullanılmakla birlikte, en yaygın yöntemlerden biri akciğer röntgen görüntülerinin incelenmesidir. Bu görüntüler uzmanlar tarafından titizlikle değerlendirilir. Ancak bu sürecin doğruluğu, süresi ve harcanan emek önemli faktörler arasında yer almaktadır. Görüntü analizi alanında, yapay zekanın hızla gelişmesiyle birlikte önemli çalışmalar yapılmıştır. Bu tezde, akciğer röntgen görüntüleri kullanılarak makine öğrenmesi ile zatürre teşhisini destekleyen bir yöntem geliştirilecektir. Böylece, teşhis sürecinde zaman yönetimi, doğruluk oranı ve harcanan çabanın optimize edilmesi sağlanacaktır. Literatürdeki çalışmalar ve geliştirilen modeller dikkate alınarak, evrimsel sinir ağı tabanlı yöntemler incelenecektir. Çalışmada kullanılan akciğer görüntüleri üç gruba ayrılmıştır: eğitim, doğrulama ve test. Sonuç olarak, evrimsel sinir ağı modellerinin kullanılmasıyla elde edilen sonuçların, aynı konu üzerinde araştırma yapan araştırmacıların sonuçları kıyaslanarak literatüre katkıda bulunması ve doktorlara karar desteği sağlaması amaçlanmaktadır.

Anahtar Kelimeler: Makine Öğrenmesi, Evrişimsel Sinir Ağları, Zatürre, Akciğer Röntgeni, Derin Öğrenme.



ABSTRACT

PNEUMONIA DETECTION WITH DEEP LEARNING MODEL

GÜNEŞ, Uraz Kağan

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Assoc. Prof. Dr. Oğuz Ata

Date: 06/2025

Pages: 154

Pneumonia remains a critical global health concern due to its high morbidity and mortality rates, particularly when diagnosis and treatment are delayed. As a pulmonary infection, it can affect individuals across all age groups and, in severe cases, may result in life-threatening complications. Early and accurate diagnosis is therefore essential for effective treatment and improved patient outcomes. Traditionally, chest X-ray imaging has been one of the most used tools in the clinical diagnosis of pneumonia. However, manual interpretation of these images by radiologists is a time-consuming process and is often subject to intra- and inter-observer variability. These limitations highlight the need for intelligent, automated systems capable of assisting healthcare professionals in the diagnostic process. With the rapid evolution of artificial intelligence (AI), particularly in the domain of deep learning, significant strides have been made in medical image analysis. Convolutional neural networks (CNNs), a class of deep learning algorithms, have demonstrated exceptional performance in various classification and detection tasks, including medical imaging. Leveraging these advances, this thesis aims to develop a CNN-based method to support the diagnosis of pneumonia using chest X-ray images. The methodology involves the use of a labeled dataset that is divided into training, validation, and testing subsets. The proposed model's performance will be rigorously evaluated and compared with existing approaches in literature. Ultimately, this study seeks to contribute to the academic discourse by presenting

a reliable and efficient diagnostic tool that can aid clinicians in making more accurate and timely decisions in pneumonia detection.

Keywords: Machine Learning, Convolutional Neural Networks, Pneumonia, Chest X-Ray, Deep Learning.



İÇİNDEKİLER

Sayfa

ÖZET	vi
ABSTRACT	viii
TABLO LİSTESİ.....	xiii
ŞEKİL LİSTESİ	xiv
KISALTMALAR.....	xviii
1. GİRİŞ	1
2. GENEL BİLGİLER.....	6
2.1 LİTERATÜR TARAMASI	6
2.2 ZATÜRRE TESPİTİNDE KULLANILAN GÖRÜNTÜLEME TEKNİKLERİ	9
2.3 ZATÜRRE TANISINDA GÖĞÜS RÖNTGENİ VE YAPAY ZEKÂ DESTEKLİ SİSTEMLER.....	10
2.4 YAPAY ZEKÂ TEMELLERİ.....	11
2.4.1 Yapay Zeka Nedir?	11
2.4.2 Yapay Zekanın Tarihçesi	13
2.4.3 Yapay Zekanın ile Gelecek	16
2.5 MAKİNE ÖĞRENMESİ TEMELLERİ.....	18
2.5.1 Makine Öğrenmesi Nedir?	18
2.5.2 Makine Öğrenimi Teknikleri	21
2.5.2.1 Denetimli öğrenme (Supervised learning)	22
2.5.2.2 Denetimsiz öğrenme (Unsupervised learning).....	27
2.5.2.3 Yarı denetimli öğrenme (Semi supervised learning)	30
2.5.2.4 Güçlendirici öğrenme (Reinforcement learning)	31
2.5.2.5 Çoklu görev öğrenme (Multitasking learning).....	32
2.5.2.6 Topluluk öğrenmesi (Ensemble learning)	32
2.5.2.7 Sinir ağları öğrenmesi (Neural network learning)	34
2.5.2.8 Örnek tabanlı öğrenme (Instance based learning).....	34

2.5.3 Makine Öğrenmesinde Veri Ön İşleme	35
2.6 DERİN ÖĞRENME	36
2.6.1 Derin Öğrenmeye Giriş.....	36
2.6.2 Derin Öğrenmenin Makine Öğrenmesinden Farkı	37
3. GEREÇ VE YÖNTEM.....	39
3.1 GEREÇ	39
3.2 DONANIM	39
3.3 GELİŞTİRME ORTAMI.....	39
3.3.1 Python	40
3.1.2 Kullanılan Kütüphaneler	41
3.3 KULLANILAN VERİ SETİ	42
3.4 İKİNCİ VERİ SETİ	44
3.5 DERİN ÖĞRENME YÖNTEMLERİ VE KULLANILAN YÖNTEM.....	46
3.5.1 Derin Öğrenme Yöntemleri	46
3.5.1.1 Yapay sinir ağları	46
3.5.1.2 Tekrarlayan sinir ağları	62
3.5.1.3 Uzun kısa süreli bellek	64
3.5.1.4 U-Net.....	65
3.5.1.5 Otomatik kodlayıcılar	67
3.5.1.6 Üretken çatışmalı ağlar	68
3.5.1.7 Evrişimli sinir ağları.....	69
3.5.2 Performans Değerlendirme Metrikleri	89
3.5.2.1 Doğruluk	90
3.5.2.2 Hassasiyet.....	91
3.5.2.3 Duyarlılık	91
3.5.2.4 F1-ölçütü	91
3.5.2.5 ROC eğrisi altındaki alan	92
3.5.2.6 K-turlu çapraz doğrulama	92

3.5.3 Kullanılan Yöntemler.....	93
3.5.3.1 Veri ön işleme	93
3.5.3.2 Özellik çıkarımı ve seçimi.....	94
3.5.3.3 Modelin eğitimi ve hiperparametreler.....	94
3.5.3.4 Derin öğrenme modeli.....	95
3.5.3.5 Performans ve değerlendirme kriterleri	96
3.6 YAŞANAN VE YAŞANABİLECEK ZORLUKLAR.....	96
3.6.1 Veri Zorlukları	97
3.6.2 Model Zorlukları	98
3.6.3 Klinik Zorlukları	100
3.6.4 Etik ve Gizlilik Zorlukları.....	101
3.6.5 Analiz ve Yorumlama Zorlukları.....	102
4. ARAŞTIRMA BULGULARI.....	103
4.1 MODEL YAPISI	103
4.2 MODELİN EĞİTİLMESİ.....	104
4.3 EĞİTİM GEÇMİŞİNİ GÖRSELLEŞTİRME.....	105
4.4 MODEL DEĞERLENDİRME	107
4.5 KARIŞIKLIK MATRİSİ	107
4.6 SINIFLANDIRMA RAPORU.....	109
4.7 ROC EĞRİSİ DEĞERLENDİRME	110
4.8 DOĞRU VE YANLIŞ SINIFLANDIRILMIŞ GÖRÜNTÜLER.....	111
5. TARTIŞMA VE SONUÇ.....	113
5.1 KARŞILAŞTIRMALI ANALİZ VE TARTIŞMA.....	113
5.2 SONUÇ	117
5.3 ARAŞTIRMA BOŞLUKLARI VE GELECEK KAPSAMLARI.....	118
REFERANSLAR	120

TABLO LİSTESİ

	<u>Sayfa</u>
Tablo 2.1: Veri Ön İşleme Teknikleri.	35
Tablo 3.1: Donanım İçeriği.	39
Tablo 3.2: Modellerin Sınırlamaları ve Potansiyel Çözümleri [136].	76
Tablo 3.3: Veri Zorlukları [151-156].	97
Tablo 3.4: Model Zorlukları [151-156].	98
Tablo 3.5: Klinik Zorluklar [151-156].	100
Tablo 3.6: Etik ve Gizlilik Zorlukları [151-156].	101
Tablo 3.7: Analiz ve Yorumlama Zorlukları [151-156].	102
Tablo 5.1: Diğer Araştırmacıların Çalışma Sonuçları ve Kendi Sonuçlarım.	114

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1: 2017 Yılında Beş Yaş Altı Çocuklarda Hastalıklara Bağlı Ölüm Sayıları.	1
Şekil 1.2: Üç Taraftan Göğüs Radyografı Örneği.	2
Şekil 1.3: Örnek Zatürre Tespit Çerçevesinin Gösterimi [24].	4
Şekil 2.1: Derin Öğrenme Modelleri Sistemi Kullanılarak Göğüs Röntgeni Görüntülerinde Zatürre Tespitleri [34].	8
Şekil 2.2: Kedi Türleri.	19
Şekil 2.3: Çocuklarda Nesne Tanımlama Örneği.	20
Şekil 2.4: Klasik Programlama Örneği.	21
Şekil 2.5: Makine Öğrenmesi Teknikleri.	22
Şekil 2.6: Karar Ağaçları Yapısı.	24
Şekil 2.7: Naive Bayes Örneği.	25
Şekil 2.8: Destek Vektör Makineleri (Support Vector Machines- SVM).	26
Şekil 2.9: Rasgele Orman (Random Forest).	27
Şekil 2.10: K-Ortalamlar Kümeleme (K-Means Clustering).	28
Şekil 2.11: Hiyerarşik Kümeleme (Hierarchical Clustering).	29
Şekil 2.12: Bir Robotun Labirentten Çıkışı Öğrenmesi.	32
Şekil 2.13: Derin Öğrenmeye Gelene Kadar Genelden Özele Gösterim.	37
Şekil 2.14: Makine Öğrenmesi ve Derin Öğrenme Arasındaki Fark.	38
Şekil 3.1: Birinci Veri Seti İçerisinden Göğüs Röntgeni Örnekleri.	42
Şekil 3.2: Birinci Veri Seti Veri Dağılımı Oranı.	43

Şekil 3.3: Birinci Veri Seti Veri Dağılımı.	43
Şekil 3.4: İkinci Veri Setinden Göğüs Röntgeni Örnekleri.	44
Şekil 3.5: İkinci Veri Seti Veri Dağılımı Oranı.	45
Şekil 3.6: İkinci Veri Seti Veri Dağılımı.	45
Şekil 3.7: Biyolojik ve Yapay Sinir Ağı.	46
Şekil 3.8: Nöronlar Arası İletişim.	48
Şekil 3.9: Yapay Sinir Ağı Modeli.	48
Şekil 3.10: Yapay Sinir Ağının Aritmetik Modeli.	50
Şekil 3.11: Her Matematik Fonksiyonun Çalıştığı Nöron Bu Görselde Gösterildiği Gibi Çalışır.	50
Şekil 3.12: Yapay Sinir Hücresi.	52
Şekil 3.13: Basamak (Step) Fonksiyonu.	53
Şekil 3.14: Doğrusal (Linear) Fonksiyon.	54
Şekil 3.15: Sigmoid Fonksiyonu.	54
Şekil 3.16: tanh (Hiperbolik Tanjant) Fonksiyonu.	55
Şekil 3.17: ReLu Fonksiyonu.	56
Şekil 3.18: Sızıntı (Leaky ReLu) Fonksiyonu.	56
Şekil 3.19: Aktivasyon Fonksiyonları [110].	58
Şekil 3.20: Backpropagation (Geri Yayılım).	60
Şekil 3.21: RNN Algoritmasının İşleyiş Biçimi.	63
Şekil 3.22: RNN Algoritmasının Beş Farklı Çalışma Yapısı Gösterilmiştir.	64
Şekil 3.23: Herhangi t Anında Genel LSTM Hücresinin Diyagramı.	65

Şekil 3.24: U-Net Mimarisindeki Ana Ağ Bileşenleri.	67
Şekil 3.25: Otomatik Kodlayıcılar (Autoencoders- AE).	68
Şekil 3.26: Üretken Çatışmalı Ağlar (GANs).....	69
Şekil 3.27: Konvolüsyonel Sinir Ağı Katmanları [31].	70
Şekil 3.28: Konvolüsyonel Katmanı İşlem Basamakları.	71
Şekil 3.29: Relu Fonksiyonu Kullanarak Düzleştirme Katmanı ve Maksimum Havuzlama Yöntemi.	72
Şekil 3.30: Tam Bağlantı Katmanında Elde Edilen Vektördeki En Yüksek Değerlerin Ortalaması Doğruluk Oranı.	73
Şekil 3.31: Model Uyum Örnekleri.	76
Şekil 3.32: İnce Ayar Tabanlı Yöntemler (Fine Tuning).	79
Şekil 3.33: Boyut İndirgeme Teknikleri.	80
Şekil 3.34: Transfer Öğrenme Yönteminin Operasyonel Diyagramı: InceptionNetV3, XceptionNet, GoogleNet, MobileNet, ResNet50, DenseNet121, VGG-16, AlexNet ve LeNet [144].	86
Şekil 3.35: Hata Matriksi.....	90
Şekil 3.36: ROC Eğrisi.	92
Şekil 4.1: Model Yapısı ve Çıktısı İki Veri Kümesi İçin Aynıdır.	103
Şekil 4.2: İkinci Veri Seti İçin Modelin Eğitilmesi Sonucu Ortaya Çıkan Çıktısı.	104
Şekil 4.3: Birinci Veri Seti İçin Modelin Eğitilmesi Sonucu Ortaya Çıkan Çıktısı.	104
Şekil 4.4: Birinci Veri Seti İçin Eğitim ve Doğrulama Kaybı.	105
Şekil 4.5: Birinci Veri Seti İçin Eğitim ve Doğrulama Doğruluğu.	105
Şekil 4.6: İkinci Veri Seti İçin Eğitim ve Doğrulama Kaybı.	106

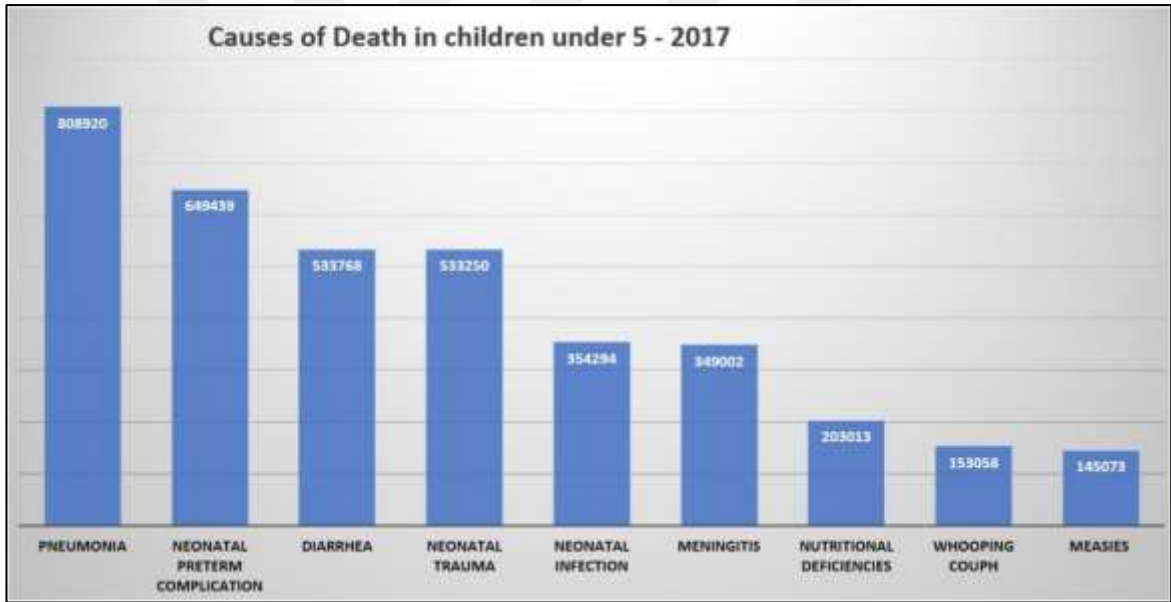
Şekil 4.7: İkinci Veri Seti İçin Eğitim ve Doğrulama Doğruluğu.	106
Şekil 4.8: İkinci Veri Seti İçin Alınan Sonuçlar.	107
Şekil 4.9: Birinci Veri Seti İçin Alınan Sonuçlar.	107
Şekil 4.10: Birinci Veri Seti İçin Karışıklık Matrisi Sonuçları.	108
Şekil 4.11: İkinci Veri Seti İçin Karışıklık Matrisi Sonucu.	108
Şekil 4.12: Birinci Veri Seti Sonucu.	109
Şekil 4.13: İkinci Veri Seti Sonucu.	109
Şekil 4.14: İkinci Veri Seti İçin ROC Eğrisi Grafiği.	110
Şekil 4.15: Birinci Veri Seti İçin ROC Eğrisi Grafiği.	110
Şekil 4.16: Birinci Veri Seti İçin Yapılan Deneydeki Doğru Tahminlerden Birkaçı.	111
Şekil 4.17: Birinci Veri Seti İçin Yapılan Deneydeki Yanlış Tahminlerden Birkaçı.	111
Şekil 4.18: İkinci Veri Seti İçin Yapılan Deneydeki Doğru Tahminlerden Birkaçı.	112
Şekil 4.19: İkinci Veri Seti İçin Yapılan Deneydeki Yanlış Tahminler.	112

KISALTMALAR

AE	:	Otomatik Kodlayıcılar
BT	:	Bilgisayarlı Tomografi
CNN	:	Evrişimsel Sinir Ağları
CXR	:	Göğüs Radyografisi
DL	:	Derin Öğrenme
ESA	:	Evrişimsel Sinir Ağları
FN	:	Yanlış Negatifler
FP	:	Yanlış Pozitifler
FPR	:	Yanlış Pozitif Oranı
GAN	:	Üretken Düşmanca Ağlar
LSTM	:	Uzun Kısa Süreli Bellek
ML	:	Makine Öğrenmesi
PCA	:	Temel Bileşen Analizi
ROC	:	Alıcı İşletim Karakteristiği
RNN	:	Tekrarlayan Sinir Ağı
SVM	:	Destek Vektör Makineleri
TN	:	Gerçek Negatifler
TP	:	Gerçek Pozitifler
TPR	:	Gerçek Pozitif Oranı
XAI	:	Açıklanabilir Yapay Zeka
YSA	:	Yapay Sinir Ağı
YZ	:	Yapay Zeka

1. GİRİŞ

Zatürre, akciğerleri hedef alan ve solunum yollarında ciddi enfeksiyonlara neden olan bir hastalıktır. Hastalık, hava keseciklerinin irin ve sıvılarla dolmasına yol açarak hayati tehlike oluşturabilir [1]. Zatürre genel olarak iki ana gruba ayrılmaktadır: bakteriyel ve viral. Çoğu durumda, bakteriyel zatürre daha ağır belirtiler gösterir. Bu iki tür arasındaki en önemli fark, tedavi süreçlerinde görülmektedir. Bakteriyel zatürre antibiyotiklerle tedavi edilirken, viral zatürre vakaları genellikle zamanla kendiliğinden iyileşir [2]. Dünya çapında yaygın olarak görülen bu hastalığın en büyük nedenlerinden biri hava kirliliğidir. Zatürre, Amerika Birleşik Devletleri'nde en sık görülen 10 ölüm sebebinden biri olup, sekizinci sırada yer almaktadır [3]. Hindistan'da ise her yıl 3,7 milyon çocuğun zatürre nedeniyle hayatını kaybettiği, bu sayının ülkedeki toplam zatürre ölümlerinin yüzde 50'sini oluşturduğu belirtilmektedir [4].



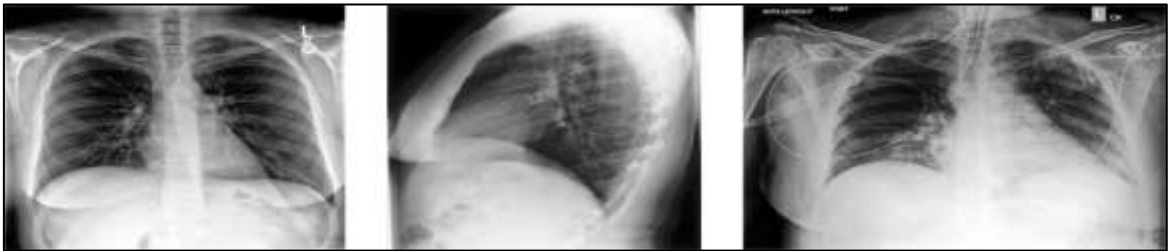
Şekil 1.1: 2017 Yılında Beş Yaş Altı Çocuklarda Hastalıklara Bağlı Ölüm Sayıları.

Zatürre, özellikle yaşlı bireylerde ölümcül bir aşamaya gelene kadar genellikle göz ardı edilmekte ve tedavi edilmemektedir. Küresel ölçekte, çocuk ölümlerinin en büyük sebeplerinden biri olup özellikle beş yaş altı çocuklarda yaygın olarak görülmektedir [5]. Dünya Sağlık Örgütü'ne göre, "Her yıl tahmini olarak 1,4 milyon beş yaş altı çocuğun ölümüne yol açarak, dünya genelindeki beş yaş altı ölümlerin %18'ini oluşturmaktadır. Zatürre, dünya çapında çocukları ve aileleri etkilemektedir, ancak en çok Güney Asya ve Sahra Altı Afrika bölgelerinde görülmektedir. Ancak çocuklar zatürreden korunabilir; basit

önlemler ile hastalık önlenabilir ve düşük maliyetli, düşük teknolojiye sahip tedavi yöntemleriyle iyileştirilebilir" [2]. Bu nedenle, özellikle çocuklarda zatürreye bağlı ölümleri azaltabilmek adına bilgisayar destekli tanı sistemleri üzerine daha fazla araştırma ve geliştirme yapılması gerekmektedir.

Zatürre tanısı koymak amacıyla çeşitli testler uygulanabilir. Bunlar arasında göğüs röntgeni, akciğer bilgisayarlı tomografisi (BT), göğüs ultrasonografisi, akciğer iğne biyopsisi ve göğüs MR görüntülemesi yer almaktadır [6]. Göğüs radyografisi (CXR), uzun süredir radyolojik görüntülemenin temel unsurlarından biri olup, halen dünya genelinde en yaygın kullanılan yöntemlerden biridir. Sanayileşmiş ülkelerde her 1000 kişiden 238'ine yılda en az bir kez ayaktan göğüs röntgeni çekildiği bildirilmektedir [7]. 2006 yılı verilerine göre sadece Amerika Birleşik Devletleri'nde 129 milyon CXR görüntüsü elde edilmiştir [8]. CXR'nin yaygın kullanımı; düşük maliyetli olması, radyasyon dozunun azlığı ve birçok hastalığı tespit edebilme yeteneğiyle açıklanabilir. Göğüs röntgeni, genellikle ilk görüntüleme aracı olarak tercih edilmekte ve pek çok hastalığın tespiti, teşhisi ve yönetiminde kritik bir rol oynamaktadır [9].

Göğüs röntgenleri, X-ışını kaynağının ve dedektör panelinin hastaya göre konumuna ve hastanın duruşuna göre üç ana başlıkta sınıflandırılır: posteroanterior (PA), anteroposterior (AP) ve lateral. PA ve AP, X-ışını kaynağının hastanın arkasına ya da önüne yerleştirilmesiyle alınan, frontal düzlemdeki görüntülerdir. AP görüntüler genellikle sırtüstü yatan hastalardan alınırken, PA görüntüler genelde ayakta duran hastalardan elde edilir. Lateral görüntüleme çoğunlukla PA görüntüleme ile gerçekleştirilir ve X-ışını, genellikle hastanın sağ tarafından sol tarafına doğru geçirilerek görüntüleme yapılır. Bu görüntü tiplerine ait örnekler Şekil 1.2'de sunulmuştur.



Şekil 1.2: Üç Taraftan Göğüs Radyografisi Örneği.

Göğüs radyografilerinin değerlendirilmesi, görüntüleme sırasında anatomik yapıların aynı projeksiyon düzlemine üst üste yansması nedeniyle oldukça karmaşık olabilir. Bu durum,

özellikle bazı anormalliklerin (örneğin kalbin arkasındaki nodüller gibi) görünmesini engelleyebilir, ayrıca küçük ya da belirsiz bulguların fark edilmesini ve farklı patolojik durumların doğru şekilde ayırt edilmesini zorlaştırabilir. Tüm bu etkenler, radyologlar arasında göğüs röntgeni değerlendirmelerinde ciddi ölçüde farklılıkların ortaya çıkmasına neden olabilmektedir [11-13].

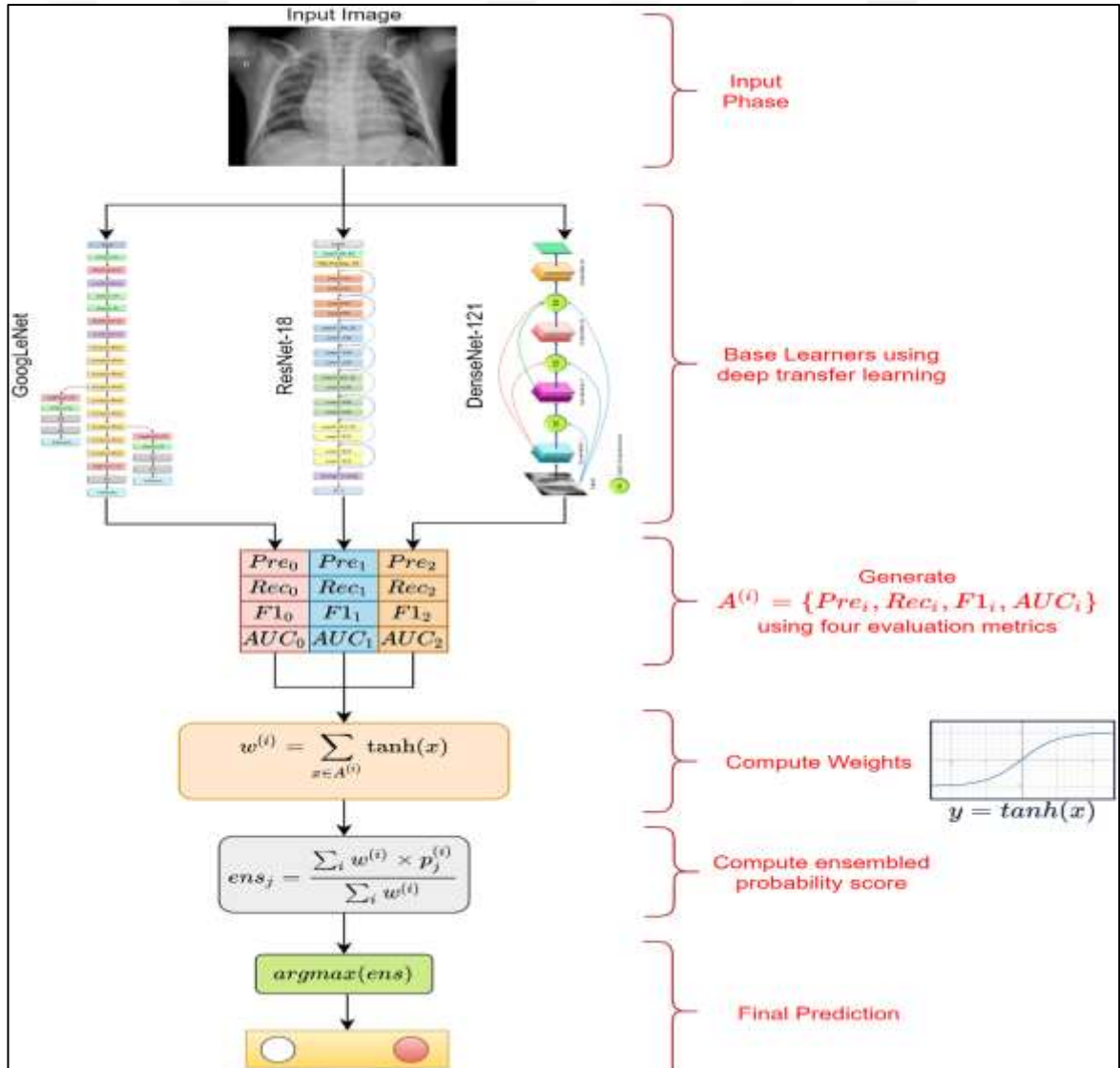
CXR görüntülerinin yüksek sayıda olması, yorumlama sürecinin zorlukları ve bu görüntülerin klinikteki önemi, araştırmacıların uzun süredir CXR analizine yönelik otomatik algoritmalar geliştirmesine ilham vermiştir. Bu doğrultuda, otomatik anormallik tespiti yapan sistemlerle ilgili ilk akademik çalışmalar 1960'lı yıllarda yayımlanmaya başlamış ve bu konu o zamandan beri ilgi çekici bir araştırma alanı olmuştur [14-18]. Otomatik CXR analizinden sağlanabilecek başlıca faydalar arasında; daha hassas tespitler, acil vakaların daha hızlı tanınması, tekrarlayan işlemlerin otomatikleştirilmesi ve radyolog eksikliği yaşanan bölgelerde analiz desteği sağlanması yer alır.

Derin öğrenme, birçok karmaşık bilgisayar görüşü sorununu çözmede önemli bir rol oynayan önemli bir yapay zekâ aracıdır [19, 20]. Derin öğrenme modelleri, özellikle evrimsel sinir ağları (CNN'ler), çeşitli görüntü sınıflandırma sorunları için yaygın olarak kullanılır. Ancak, bu tür modeller yalnızca büyük miktarda veri sağlandığında optimum performans gösterir. Biyomedikal görüntü sınıflandırma sorunları için, bu kadar büyük miktarda etiketli veriyi elde etmek zordur çünkü uzman doktorların her bir görüntüyü sınıflandırmasını gerektirir ve bu da pahalı ve zaman alıcı bir görevdir. Transfer öğrenme, bu engeli aşmak için bir geçici çözümdür. Bu teknikte, küçük bir veri kümesini içeren bir sorunu çözmek için, büyük bir veri kümesi üzerinde eğitilmiş bir model yeniden kullanılır ve bu modelde belirlenen ağ ağırlıkları uygulanır. 14 milyondan fazla görüntüden oluşan ImageNet [21] gibi büyük bir veri kümesi üzerinde eğitilmiş CNN modelleri, biyomedikal görüntü sınıflandırma görevleri için sıklıkla kullanılır.

Topluluk öğrenmesi, birden fazla sınıflandırıcının kararlarının bir test örneği için nihai tahmini elde etmek üzere birleştirildiği popüler bir stratejidir. Tüm temel sınıflandırıcılardan ayırt edici bilgileri yakalamak için gerçekleştirilir ve böylece daha doğru tahminlerle sonuçlanır. Literatürdeki çalışmalarda en sık kullanılan topluluk tekniklerinden bazıları ortalama olasılık, ağırlıklı ortalama olasılık ve çoğunluk oylamasıdır. Ortalama olasılık tabanlı topluluk, her bir bileşen temel öğrenicisine eşit öncelik atar. Ancak, belirli bir sorun

için, belirli bir temel sınıflandırıcı bilgileri diğerlerinden daha iyi yakalayabilir. Bu nedenle, daha etkili bir strateji tüm temel sınıflandırıcılara ağırlık atamaktır [22]. Ancak, topluluğun gelişmiş performansını sağlamak için, her bir sınıflandırıcıya atanan ağırlıkların değeri en önemli faktördür. Çoğu yaklaşım bu değeri deneysel sonuçlara göre belirler. Literatürdeki çalışmalarda, genel olarak, temel öğrenenlere ağırlık atamak için yalnızca sınıflandırma doğruluğu dikkate alınmıştır [23]; bu, özellikle veri kümeleri sınıf-dengesiz olduğunda yetersiz bir ölçüt olabilir. Diğer metrikler, temel öğrenenlere öncelik verme konusunda daha iyi bilgi sağlayabilir. Önerilen topluluk çerçevesinin genel iş akışı, Şekil 1.3'te sunulmuştur.

Bu çalışmada, CXR görüntülerinden pnömoni tanısına ilişkin güncel literatür sunulmuş ve katkılar aşağıdaki şekilde özetlenmiştir:



Şekil 1.3: Örnek Zatürre Tespit Çerçevesinin Gösterimi [24].

- a. Zatürre tanısında kullanılan algoritmaların kullanılabilirliği, iyilik faktörleri ve hesaplama karmaşıklığının incelenmesi ve analizi.
- b. Mevcut CXR görüntü veri kümelerinin kalitesinin, kullanılabilirliğinin, boyutunun ve sınıf dengesi kapsamının incelenmesi ve analizi.
- c. Zatürre tanısında kullanılan algoritmaların tartışılması ve karşılaştırmalı analizinin yapılması.
- d. Tıbbi uygulayıcılar ve araştırma topluluğu için mevcut literatüre dayalı gelecekteki yönleri önermek, onların gerçek zamanlı bir perspektiften en iyi yöntemleri seçmelerine, mevcut veri kümelerini algılamalarına ve bu alanda elde edilen sonuçları anlamalarına yardımcı olabilir.

2. GENEL BİLGİLER

2.1 LİTERATÜR TARAMASI

Derin Öğrenme ile zatürre teşhisine ilişkin literatürde çeşitli araştırmalar yer almakta olup; bu çalışmaların büyük çoğunluğunda evrimsel sinir ağları (Convolutional Neural Networks - CNN) temel yöntem olarak kullanılmaktadır [30, 35]. Ancak bu ağların oluşturulması hem zaman gerektirmekte hem de büyük boyutlu veri kümelerine ihtiyaç duymaktadır. Bununla birlikte, çalışmalarda sıklıkla öğrenme aktarımı (transfer learning) yöntemiyle desteklenen önceden eğitilmiş modeller tercih edilmiştir [30, 35].

Chandra ve meslektaşları [25], göğüs röntgeni görüntülerinde akciğer alanlarını bölgelere ayırmış ve bu bölgelere ait sekiz farklı istatistiksel özelliği elde ederek sınıflandırma sürecinde kullanmışlardır. Çalışmada çok katmanlı algılayıcı (MLP), rastgele orman, sıralı minimal optimizasyon (SMO), regresyon temelli sınıflandırma ve lojistik regresyon olmak üzere beş farklı yöntem uygulanmış; analizler 412 görüntü üzerinden gerçekleştirilmiştir.

Benzer şekilde Kuo ve arkadaşları [26], 185 şizofreni hastasına ait verilerden zatürre tanısı koymak için 11 özellik belirlemiş ve bu özellikleri farklı sınıflandırma teknikleri (karar ağaçları, destek vektör makineleri, lojistik regresyon) aracılığıyla değerlendirmiştir.

Yue ve çalışma arkadaşları [27], zatürreyi tespit edebilmek amacıyla 52 hastaya ait göğüs BT taramalarından altı farklı özellik çıkarmışlardır. Ancak bu yaklaşımlar, sınırlı veri kümesi kullanımı nedeniyle genellenebilirlik açısından kısıtlıdır.

Sharma vd. [28] ile Stephen vd. [29], pnömoniye ait göğüs röntgeni görüntülerinin sınıflandırılmasında basit CNN mimarileri geliştirmişlerdir. Veri yetersizliği sorununu azaltmak amacıyla, çalışmalarında veri artırma (data augmentation) tekniklerinden faydalanmışlardır.

Rajpukar ve arkadaşları, zatürre sınıflandırması için gelişmiş bir evrimsel sinir ağı modeli olan DenseNet-121 mimarisini kullanmışlardır. Yoğun Evrişim Ağı (Dense Convolutional Network- DenseNet) tabanlı bu modeli "CheXNet" olarak adlandırmışlardır. Modelde sigmoid aktivasyon fonksiyonu kullanılmış ve ağırlıklar, daha önce ImageNet üzerinde eğitilmiş bir modelin ağırlıklarıyla başlatılmıştır [30].

Tuncer ve çalışma arkadaşları [31], görüntülere bulanık ağaç dönüşümü uygulamış ve ardından örnek bölme işlemi gerçekleştirdikleri bir makine öğrenimi yaklaşımı önermişlerdir. Bu işlem sonrasında, çok çekirdekli yerel ikili desen yöntemiyle özellik çıkarımı yapılmış ve elde edilen örnekler geleneksel sınıflandırma algoritmaları ile sınıflandırılmıştır. Yöntem, COVID-19 ve zatürre içeren sınırlı bir veri kümesi üzerinde test edilmiştir.

Wang [32], veri kümesini %70 eğitim, %10 doğrulama ve %20 test olarak ayırarak, konvolüsyonel sinir ağları (CNN) temelli bir yöntem uygulamıştır. Başarı oranını artırmak amacıyla özellikle ResNet-50 mimarisi üzerine yoğunlaşmış, eğitimi ise ImageNet üzerinde gerçekleştirmiştir. Ayrıca AlexNet, GoogLeNet, VGGNet-16 ve ResNet-50 gibi önceden eğitilmiş modelleri değerlendirmiştir.

Wang'ın çalışmasını takiben, Li Yao [33] aynı veri kümesini kullanarak araştırmasını yürütmüş ve bu süreçte tekrarlayan sinir ağlarını (RNN) kullanmıştır. Wang gibi, veri setini %70 eğitim, %10 doğrulama ve %20 test olacak şekilde ayırmış ve modelini OpenI veri seti üzerinde eğitmiştir. Yao, modelinde sigmoid aktivasyon fonksiyonunu tercih etmiş ve Wang'ın sonuçlarıyla kendi bulgularını karşılaştırarak değerlendirme yapmıştır [32, 33].

Method	Approach	Merits	Demerits
Albahli et al. [9]	• Transfer Learning using InceptionResNet-V2	Reuse of models pretrained on a large dataset	Oversimplified for a complex pattern recognition task; Performance obtained is poor and not fit for practical use
Rahman et al. [10]	• Transfer Learning using DenseNet-201		
Liang et al. [11]	• Transfer learning using ResNet-50 pre-trained on ChestX-ray14 dataset		
Ibrahim et al. [12]	• Transfer learning using AlexNet		
Zubair et al. [13]	• Transfer learning using VGG-16		
Rajpukar et al. [14]	• Transfer learning using DenseNet-121		
Albahli et al. [15]	• Used generative adversarial networks to generate synthetic data. • Classification using ResNet-152	Generation of synthetic data to balance the classes of the data because medical data are scarce	Classification results (41% accuracy rate) are not fit for practical use
Chandra et al. [16]	• Segmentation of lung X-rays using image processing • Extraction and classification of eight statistical features	Segmentation of lungs before classification allows localization of the disease	The use of handcrafted features limits its ability to perform in complex pattern recognition tasks; Evaluation on a small dataset (412 images) cannot be generalized
Kuo et al. [17]	• Used 11 features from patient data to fit traditional classifiers	Use of 10-fold cross validation with 3 repeats avoids overfitting	Patient data are often private and not publicly available to fit to classification models
Yue et al. [18]	• Segmented lung lobes using U-Net • Extracted and classified radiomic features from CT-scan images	Segmentation before classification helps extract important features for radiologists and allows localization of the disease	Method evaluated on a small dataset (72 lesion segments) and thus difficult to generalize
Sharma et al. [19]	• Devised a CNN model for classification of X-ray images	Automatic feature learning for complex tasks	Simple linearly progressing CNN model increases computation cost without providing strong boost to performance
Stephen et al. [20]	• Developed a simple seven-layer CNN model for classification of X-ray images		
Janizek et al. [21]	• Developed a deep learning framework based on adversarial optimization	Adversarial optimization removed dependency on the source of the dataset and view of the X-rays for classification	Results (AUC 74.7%) are not fit for deployment in the field
Zhang et al. [22]	• Developed a confidence-aware module for anomaly detection in lung X-ray images	Posing the detection task as a one-class problem helped improve the model performance	The sensitivity obtained on the dataset was too low (71.70%) for practical use
Tuncer et al. [23]	• Applied fuzzy tree transformation to X-ray images • Extracted local features for classification using an ensemble of traditional classifiers	Generation of three different feature images improves the model performance	Handcrafted feature extraction limits performance in complex pattern recognition tasks; Evaluation on a small dataset cannot be generalized
Jaiswal et al. [24]	• Developed a mask region-based CNN for segmentation • Used an ensemble model for image thresholding	Use of threshold value in background boosts the performance	An irregular trend was observed, where results of the training set were lower than those of the testing set
Gabruseva et al. [25]	• Localized pulmonary opacity based on a single-shot detector • Used a snapshot ensemble model for segmentation	One-shot detector alleviates the problem of scarcity of data	Irregular trend of validation loss over epochs during model training
Pan et al. [26]	• Used an ensemble of Inception-ResNet v2, XceptionNet, and DenseNet-169 for bounding box prediction	Ensemble learning allows the fusion of salient properties of all its base learners	Pan et al. [26] suspect that their model evaluated on only one dataset may not generalize over data acquired from a different source

Şekil 2.1: Derin Öğrenme Modelleri Sistemi Kullanılarak Göğüs Röntgeni Görüntülerinde Zatürre Tespitleri [34].

2.2 ZATÜRRE TESPİTİNDE KULLANILAN GÖRÜNTÜLEME TEKNİKLERİ

Pnömoni (zatürre), enfeksiyöz etkenlerin neden olduğu akciğer dokusunun inflamasyonu ile karakterize, ciddi seyirli bir solunum sistemi hastalığıdır [36, 37]. Hastalığın erken evrede ve doğru biçimde teşhis edilmesi, klinik sürecin yönetimi açısından büyük önem taşımakta olup bu bağlamda tıbbi görüntüleme teknikleri tanısal değerlendirme sürecinin temel bileşenlerinden biri haline gelmiştir [38, 39].

Günümüzde zatürre teşhisinde en sık başvuru yöntem göğüs radyografisi (X-ray) olmaktadır. X-ray görüntüleme; düşük maliyetli oluşu, yaygın erişilebilirliği ve hızlı sonuç elde etme avantajları nedeniyle birinci basamak tarama yöntemi olarak öne çıkmaktadır. Ancak, bu yöntemle elde edilen görüntülerde zatürreye ilişkin bulguların diğer pulmoner patolojilerle benzerlik göstermesi, tanısal doğruluğun sağlanabilmesi için deneyimli radyologların yorumuna ihtiyaç duyulmasına yol açmaktadır [40, 41].

Bilgisayarlı tomografi (BT), X-ray'e oranla daha yüksek çözünürlükte ve detaylı kesitsel görüntüler sağlayarak özellikle kompleks veya tipik dışı olguların analizinde kullanılmaktadır. BT; enfeksiyonun yayılımı, lokalizasyonu ve eşlik eden diğer patolojilerin saptanmasında etkin bir yöntem olmakla birlikte, yüksek radyasyon dozu ve maliyet gibi bazı dezavantajlar barındırmaktadır [42, 43].

Son dönemlerde öne çıkan bir diğer görüntüleme tekniği ise akciğer ultrasonografisi (LUS)'dur [47]. Radyasyon içermemesi, taşınabilirliği ve yatak başı uygulanabilirliği sayesinde özellikle pediatrik ve gebelik gibi hassas hasta gruplarında alternatif bir tanı aracı olarak tercih edilmektedir. Ayrıca yoğun bakım ortamlarında hızlı ve tekrarlanabilir değerlendirme imkânı sunmaktadır [44, 45].

Bu görüntüleme teknolojileri, son yıllarda yapay zekâ ve derin öğrenme temelli sistemlerle bütünleştirilerek, otomatik tanı modellerinin geliştirilmesine olanak sağlamaktadır. Bu sayede, zatürre tanısında daha hızlı, objektif, güvenilir ve yeniden üretilebilir sonuçlara ulaşılması mümkün hâle gelmiştir [46].

2.3 ZATÜRRE TANISINDA GÖĞÜS RÖNTGENİ VE YAPAY ZEKÂ DESTEKLI SİSTEMLER

Zatürre [48], akciğer dokularında iltihaplanmaya neden olan ve solunum işlevini engelleyerek oksijen alımını zorlaştıran viral, bakteriyel veya fungal enfeksiyonlar sonucu gelişen ciddi bir sağlık problemidir. Bu durum, nefes darlığına yol açabilir ve tedavi edilmediği takdirde ölümcül sonuçlara neden olabilir.

X-ışınları, 1895 yılında Wilhelm Conrad Röntgen'in Lenard ve Crookes tüpleri ile yaptığı deneyler sonucunda keşfedilmiş olup, o zamandan bu yana tıbbi görüntüleme alanında vazgeçilmez bir araç haline gelmiştir [49]. X-ışını teknolojisi, enfeksiyonların tanı ve takibinde etkin biçimde kullanılmakta ve birçok hastalığın teşhisinde önemli bir rol oynamaktadır [50]. Günümüzde, özellikle göğüs röntgeni (CXR), zatürre teşhisinde sıklıkla başvurulmuş temel görüntüleme yöntemlerinden biridir.

Genel olarak görüntüleme temelli bir tanı süreci üç aşamadan oluşur: tarama hazırlığı, görüntü elde etme ve tanı aşaması. İlk aşamada, hasta teknisyen rehberliğinde taramaya hazırlanır. Görüntü elde etme aşamasında, uygun görüntüleme modaliteleri (örneğin X-ışını veya BT) kullanılarak gerekli görseller toplanır. Son aşamada ise bu görüntüler analiz edilerek klinik değerlendirme yapılır [51].

Zatürre tanısında kullanılan bu görüntüler, genellikle radyologlar tarafından yorumlanmaktadır. Ancak zatürre vakalarının artmasıyla birlikte, radyologlara olan ihtiyaç ve iş yükü de paralel biçimde artmaktadır. Özellikle viral kaynaklı zatürrelerde bulaş riski yüksek olduğundan, bu durum sağlık çalışanları açısından ciddi bir tehdit oluşturmaktadır. Bu bağlamda, [52] referanslarında da vurgulandığı üzere, yapay zekâ tabanlı temassız tanı sistemlerinin kullanımı, sağlık profesyonellerinin üzerindeki tanı baskısını azaltmak, enfeksiyon riskini en aza indirmek ve teşhis sürecini hızlandırmak açısından büyük önem taşımaktadır [53].

Bu bölümde, göğüs röntgeni görüntüleri kullanılarak zatürre teşhisinin nasıl yapıldığını daha iyi kavrayabilmek adına, yapay zekâ temelli derin öğrenme yaklaşımlarını ayrıntılı şekilde ele alacağız.

2.4 YAPAY ZEKÂ TEMELLERİ

2.4.1 Yapay Zeka Nedir?

İnsanlık, tarih boyunca beynin işleyişine duyduğu ilgiyi sürdürmüş ve bu karmaşık yapıyı taklit eden makineler geliştirme yönündeki çabalarını kesintisiz biçimde devam ettirmiştir. Bu ilgi günümüzde de bilim camiasında canlılığını korumaktadır. Ateşin ve tarımın keşfi, üretim teknolojilerinin geliştirilmesi ve makinelerin icadı gibi dönüm noktaları, hep bu zihinsel arayışın uzantısı olarak değerlendirilebilir.

Bugünün teknolojik seviyesinde yapay zekâ, zeki makineler ve özellikle akıllı yazılımlar geliştiren bir mühendislik ve bilim alanı olarak tanımlanmaktadır. Alanın temel amacı, insan zekâsına ait temel işlevleri -örneğin öğrenme, problem çözme ve muhakeme- bilgisayarlara kazandıran algoritmalar geliştirmektir. Bazı çalışmalar, bu hedefin ötesine geçerek insanın bilişsel yetilerini aşan sistemler tasarlamayı da hedeflemektedir. Ancak bilimsel yaklaşımla değerlendirildiğinde yapay zekâ; deneyimlere dayalı öğrenme, mantıksal analiz, görsel örüntü ve biçim tanıma, doğal dil işleme gibi becerileri içeren, bilişim alanına yenilikçi katkılar sunan bir disiplindir [54].

İnsan zekâsının taklit edilmesi temelinde geliştirilen yapay zekâ çalışmaları, makinelerin insan öğrenme süreçlerini modellemesi şeklinde tanımlanabilir. İnsan öğrenmesinin beyinde gerçekleştiği dikkate alındığında, bu süreci simüle etmek amacıyla beynin yapısının analiz edilerek bilgisayarlarda benzetim modelleri geliştirilmesi mümkün görünmektedir. İnsan beyinde öğrenmenin, nöron adı verilen hücreler arasındaki etkileşim aracılığıyla gerçekleştiği bilgisi doğrultusunda, bilgisayarlarda yapay sinir ağları inşa edilmiş ve bu sayede öğrenme süreci benzetim yoluyla modellenmiştir [55]. Yapay zekâ araştırmalarının öncülerinden biri olan McCarthy'ye göre, insan zekâsı ve öğrenme mekanizmaları en ince ayrıntısına kadar çözümlenip tanımlanabildiği takdirde, makineler bu süreci taklit ederek zeka ve öğrenme kapasitesine sahip hale gelebilecektir. Makinelerin belirli kurallarla programlanarak zekice davranışlar sergilemesi "zayıf yapay zekâ" olarak adlandırılmakta ve genel kabul görmektedir. Buna karşın, makinelerin sadece zekice değil, aynı zamanda bilinçli davranışlar sergileyebilmesi anlamına gelen "güçlü yapay zekâ" kavramı ise hâlâ tartışma konusudur [56].

Güçlü yapay zekâya yönelik eleştiriler, makinelerin biyolojik olarak insanlardan farklı olmaları nedeniyle duygusal tepkiler veremeyecekleri, etik değerleri içselleştiremeyecekleri ve insan benzeri bilişsel süreçleri gerçekleştiremeyecekleri yönündedir. Buna karşın savunucular, insan beyninin düşünce ve bilinç üretmeyen bileşenlerden oluşmasına rağmen, bu bileşenlerin etkileşimi ile zihin ve bilinç ortaya çıktığını belirterek, aynı prensibin makinelerde de geçerli olabileceğini ileri sürmektedirler [57].

Yapay zekâ sistemlerinin değerlendirilmesi, bu alandaki ilk çalışmalarla birlikte gündeme gelmiş ve bu bağlamda ilk ciddi öneri Alan Turing tarafından sunulmuştur. Turing, “Computing Machinery and Intelligence” başlıklı makalesinde, makinelerin zekâya sahip olabileceği tezini ortaya koymuş ve bu durumu ölçümlemek için “Imitation Game” adını verdiği, günümüzde ise “Turing Testi” olarak bilinen yöntemi önermiştir. Bu testte, yapay zekâya sahip bir sistem ile bir insan arasında yazılı bir iletişim gerçekleştirilir ve farklı bir odada bulunan sorgulayıcı kişi, hangisinin makine olduğunu ayırt edemezse, test edilen yapay zekâ sisteminin zekâ sahibi olduğu kabul edilir [58]. Turing Testi, günümüzde de yapay zekâ sistemleri için önemli bir değerlendirme ölçütü olarak kabul edilmeye devam etmektedir [59].

Yapay zekâ düşüncesini ilk kez gündeme getiren kişi, İngiliz matematikçi ve mantıkçı Alan Turing olmuştur. 1950 yılında, Dartmouth Konferansı’ndan altı yıl önce, Turing “Mind” dergisinin Ağustos sayısında yayımladığı “Computing Machinery and Intelligence” başlıklı makalesinde “Makineler düşünebilir mi?” sorusunu ortaya atmış ve bu soruyu felsefi bir şekilde irdelemiştir. Aynı makalede, makinelerin düşünemeyeceğini öne süren görüşlere karşı çeşitli yanıtlar da vermiştir. Turing’in 1936 yılında yazdığı başka bir makalede ise bilgisayarların kuramsal temellerine değinilmiştir. Bu çalışmada, matematiksel mantıkla ilgili bir soyut problemi çözerken, günümüzde “Turing makinesi” olarak bilinen ve genel amaçlı, programlanabilir hesaplama sistemlerinin temelini oluşturan modeli geliştirmiştir. Turing makinesi, işlemlerini, üzerinde sadece tek bir sembol bulunan bölmelere ayrılmış bir bant aracılığıyla gerçekleştirir. Bu model sınırlı sayıda içsel duruma sahiptir ve bir sembolü okuduğunda hem mevcut durumuna hem de sembole göre durum değiştirerek işlem yapar [59].

Alan Turing, ayrıca bir bilgisayar ya da benzeri bir sistemin insanlar ile aynı bilişsel yetilere sahip olup olmadığını değerlendirmeye yönelik bir test geliştirmiştir. Bu test günümüzde

"Turing Testi" olarak bilinmektedir. Genel çerçevede, bu testin amacı bir uzmanın, bir makinenin verdiği cevapları bir insanınkilerden ayırt edip edemeyeceğini belirlemektir. Eğer uzman bu ayrımı yapamıyorsa, makinenin insan seviyesinde zihinsel yeterliliğe sahip olduğu varsayılır [60].

Test sırasında bir insan ve bir bilgisayar, değerlendirmeyi yapan kişiden gizlenerek konumlandırılır. Sorgulayıcı, hangisiyle iletişim kurduğunu bilmeden her iki tarafla yazılı olarak iletişim kurar. Soru-cevap şeklinde ilerleyen bu süreçte amaç, sorgulayıcının hangi katılımcının insan, hangisinin makine olduğunu tahmin etmesidir. Eğer bu ayrım güvenilir biçimde yapılamıyorsa, makine Turing testini geçmiş sayılır ve insan düzeyinde anlama becerisine sahip olduğu kabul edilir [60].

Anlama ve farkındalık, insan aklının doğal işlevleridir. Makinelere gerçek anlamda bir "akıl" kazandırmanın mümkün olmadığı yaygın bir görüştür. Ancak asıl araştırma konusu, zekânın ve onun işlevlerinin makineler aracılığıyla taklit edilip edilemeyeceğidir. Günümüzde bu alandaki çalışmalar ağırlıklı olarak bu noktaya odaklanmaktadır [61].

2.4.2 Yapay Zekanın Tarihçesi

‘Yapay zekâ’ kavramının tarihsel gelişimini dönemsel olarak inceleyelim:

- a. Tarih Öncesi Dönem: Binlerce yıl önce, Yunan mitolojisinde rüzgâr tanrısı olarak bilinen Daedalus’un “yapay insan” üretme girişimi, bu alana dair ilk izlerden biri olarak değerlendirilebilir.
- b. Karanlık Dönem (1965-1970): Bu yıllar arasında ilerleme oldukça sınırlı kalmıştır. Bilgisayar uzmanları, yalnızca veri yükleyerek akıl yürütebilen sistemler üretme beklentisi içerisindeydiler. Ancak bu çabalar, sürecin duraksadığı bir döneme neden olmuştur.
- c. Rönesans Dönemi (1970-1975): Bu süreç, yapay zekâ alanındaki gelişmelerin ivme kazandığı ve yeni açılımların zeminini oluşturduğu bir dönemdir. Bu yıllarda özellikle hastalık teşhisi gibi alanlara yönelik sistemler geliştirilmiştir.
- d. Ortaklık Dönemi (1975-1980): Yapay zekâ araştırmacıları bu dönemde psikoloji ve dil bilimi gibi farklı disiplinlerle iş birliği yaparak çok yönlü yaklaşımlar geliştirmeye başlamışlardır.

- e. Giriřimcilik Donemi (1980-?): Bu surele birlikte yapay zeka, laboratuvar ortamından ıkarılarak gerek dunyadaki karmařık problemlere uygulanabilir hale gelmiřtir. Bu donem gunumuzde de surmektedir.

Gunumuzde hızla geliřen ve hemen her disiplinde uygulama alanı bulan yapay zeka kavramının teorik zemini ve ilk uygulama ornekleri 17. yuzyıla dayanmaktadır. Bu donemde, zellikle yonetici elitler ve aristokrasi arasında insan ya da hayvan davranıřlarını taklit eden otomatik sistemler uretme iřteėi yaygınlařmıřtır. Bu yaklařım donemin felsefi gorřlerine de yansımıřtır. Descartes (1596-1650), insanı mekanik bir duzenek gibi ele almıř ve saat mekanizmasına benzeterek aıklamıřtır. Taklit giriřimlerinin arttıėı bu donemin ardından, İngiliz matematiki Charles Babbage (1792-1871), bu defa fiziksel niteliklerden ok zihinsel faaliyetleri ornek almayı hedeflemiř ve “Fark Motoru” adını verdiėi ilk hesaplama makinesini ortaya koymuřtur. Bu makine yalnızca temel matematik iřlemlerini yapmakla kalmayıp, iřlem arası verileri depolayabilen belleėi ile dikkat ekmiřtir. Ayrıca satran ve dama gibi oyunları oynayabilme yeteneėi de mevcuttur. Babbage’in bu giriřimi, kendi aėında yapay zeka geliřimi adına ileriye donuk onemli bir basamak olarak deėerlendirilmektedir [62].

Yapay zeka ile ilgili ilk giriřimler Cezeri’nin (1136-1206) mekanik robot izimlerine kadar uzanmakla birlikte, modern yapay zeka arařtırmalarının asıl ivme kazandıėı donem İkinci Dunya Savařı ve sonrasıdır. Alan Mathison Turing, savař sırasında “Bombe” adını verdiėi ve tamamen otomatik alıřan bir řifre kırıcı cihaz geliřtirerek savařın seyrini onemli olude etkilemiřtir [63]. Savař sonrası surete Turing bařta olmak uzere pek ok bilim insanı, yapay zeka alanında bireysel arařtırmalar yapmaya yonelmiřtir. Turing, 1947’de yapay zekaya iliřkin ilk kez bir konferans vermiř ve bilgisayar programları aracılıėıyla zeki makinelerin uretilebileceėini ifade etmiřtir [64]. 1950 yılında yayımlanan “Computing Machinery and Intelligence” adlı alıřmasında “Makineler dünebilir mi?” sorusu uzerinden teorik bir ereve izmiř; “makine” ve “dünmek” kavramlarını sorgulayarak yapay zekanın temel kuramsal altyapısını oluřturmuřtur [60].

Yapay zekanın oncsu olarak Alan Turing anılsa da “yapay zeka” terimi ilk kez 1956 yılında Dartmouth Koleji’nde yapılan bir alıřtayda John McCarthy tarafından dile getirilmiřtir. Bu onemli etkinlikte McCarthy’nin yanı sıra Marvin L. Minsky (MIT), Nathaniel Rochester (IBM) ve Claude Shannon (Bell Laboratories) da yer almıřlardır [65]. Bu toplantının

ardından yapay zekâ çalışmalarında büyük bir ilerleme kaydedilmiş; Aziz (1961), Benzeşim (1963), Eliza (1965), Bilgin (1970) ve Stajyer (1979) gibi yazılımlar geliştirilmiştir [66]. 1972 yılında Japon mühendisler tarafından geliştirilen WABOT-I, insana benzeyen ilk akıllı robot olarak tarihe geçmiştir [60]. Ancak 1974-1980 döneminde yapay zekâyâ karşı yöneltilen eleştiriler sonucu birçok hükümet bu alana sağlanan finansmanı kesmiş, bu durum da “Yapay Zekâ Kışı” olarak tanımlanmıştır. 1980’li yıllarda İngiltere’nin yeniden kaynak ayırması ile bu alan tekrar canlanmıştır [67]. 1997 yılına gelindiğinde IBM’in geliştirdiği “Deep Blue”, dünya satranç şampiyonu Garry Kasparov’u yenerek büyük bir çığır açmıştır. Bu sistem, saniyede 200 milyon hamleyi değerlendirebilme yeteneğine sahipti ve bilgisayarların belirli alanlarda insanı geçebileceğinin güçlü bir göstergesi olmuştur [62].

Zatürre tanısı için geliştirilen derin öğrenme modelini değerlendirmeden önce, sağlık sektöründe yapay zekânın tarihçesine kısa bir bakış atmak yararlı olacaktır. Sağlık alanındaki ilk yapay zekâ uygulaması, 1970’lerin başında bakteriyel enfeksiyonu olan hastalarda en uygun tedaviyi belirlemeye yardımcı olması amacıyla geliştirilen MYCIN sistemidir [68]. 1980’li ve 1990’lı yıllarda, bu teknolojinin sağlık hizmetlerinde ve tıbbi görüntü analizlerinde yaygınlaşmasıyla birlikte, bilgisayar destekli tanı sistemleri (CAD) öne çıkmaya başlamıştır. Bu sistemler, kanser tanısı [69], kalp ve damar hastalıkları, beyin hastalıkları, Alzheimer ve diyabetik retinopati gibi alanlarda başarıyla kullanılmıştır. Ayrıca veri madenciliği yöntemleri, risk tahmini, hastalıkların ilerleyişinin modellenmesi, genetik veri analizi ve tedavi sürecinin planlanması gibi birçok farklı uygulama alanında kullanılmıştır. Bu teknolojiler, büyük boyutlu ve farklı yapıya sahip veri setleriyle etkili bir şekilde çalışarak, veri içindeki gizli bilgileri gün yüzüne çıkarmak için güçlü araçlar sunmuştur [65]. 1993’te Lo ve ekibi, tıbbi görüntülemelerde Konvolüsyonel Sinir Ağları’nı ilk kez akciğer nodülü tespiti için kullandı. Takip eden yıllarda bu yöntem, mamografi görüntülerinde mikrokalsifikasyon ve kitle saptama işlemlerine de uyarlandı. Ancak bu erken CNN modelleri, modern derin öğrenme ağlarında yaygın olan çok katmanlı yapı ve havuzlama katmanlarını içermemekteydi [70].

2010’lu yıllarla birlikte, modern derin öğrenme (DL) yöntemleri sağlık alanında, özellikle de tıbbi görüntüleme uygulamalarında yaygın olarak benimsenmiştir. Bu teknolojiler; görüntü sınıflandırma, nesne algılama, segmentasyon, görüntü kaydı ve görüntü iyileştirme gibi çeşitli görevlerde etkili bir şekilde değerlendirilmiştir. DL tabanlı sistemler, kanser

taraması, diyabetik retinopati ve glokom teşhisi, meme nodüllerinin ultrason ile saptanması ve pulmoner nodül tespiti gibi klinik alanlarda da başarıyla kullanılmıştır [68].

Segmentasyon özelinde; kalp, beyin dokuları ve akciğer ile hava yolları üzerine yapılan çalışmalar öne çıkmaktadır [63].

DL yaklaşımları; göğüs röntgeni, bilgisayarlı tomografi, manyetik rezonans görüntüleme, pozitron emisyon tomografisi, ultrasonografi, optik koherens tomografi ve histopatolojik görüntüler gibi çeşitli görüntüleme modaliteleri üzerinde yaygın olarak uygulanmaktadır [71].

Son dönemde, yapay zekâ sistemlerinin doğrudan klinik ortamlara entegrasyonu artmakta olup, bu süreçte sistemlerin karar mekanizmalarının anlaşılabilir olması gerektiğine vurgu yapılmaktadır. Bu bağlamda, yalnızca doğruluk performansı değil, aynı zamanda algoritmaların karar süreçlerinin açıklanabilirliği de önem kazanmaktadır. Avrupa Birliği gibi düzenleyici kurumlar, klinik uygulamaya geçmeden önce yapay zekâ sistemlerinin karar alma mekanizmalarının şeffaf biçimde açıklanmasını zorunlu kılmaktadır. Bu gelişmeler, "açıklanabilir yapay zekâ" (XAI) adı verilen yeni bir araştırma alanının doğmasına neden olmuştur [71].

2.4.3 Yapay Zekanın ile Gelecek

Teknolojik gelişmelere paralel olarak toplumsal dönüşümün yaşanması, doğal olarak yapay zekânın sosyal yaşam üzerinde de önemli etkilerinin görülmesine yol açmaktadır.

Yapay zekâ sistemleri, geçmiş dönemde insanlar için güç olan birçok durumu günümüz insanı için basitleştirerek hayal dahi edilemeyecek üstünlükler sağlamıştır. Herhangi bir adresi bulmak için kullanılacak en uygun rotayı belirleyen yapay zekâ sistemleri, alışverişlerde kişisel öneri listesi oluşturarak en uygun ürünleri sunan uygulamalar, otoyollarda kaza riskini en aza indiren yapay zekaya dayalı sürücüsüz araçlar gibi pek çok yapay zekâ ürünü; insanlara zaman, emek, maliyet, iş gücü açısından büyük üstünlükler sağlamaktadır [72].

Massachusetts Teknoloji Enstitüsü (Massachusetts Institute of Technology, MIT) Yapay Zekâ Laboratuvarı yöneticisi ve Carnegie Mellon Üniversitesi öğretim üyelerinden Edward Fredkin'e göre tarih boyunca meydana gelmiş üç önemli durum söz konusudur. Bunların ilki

evrenin oluşumu, ikincisi canlılığın başlangıcıdır. Üçüncü ve yaşamın başlangıcına denk derecede önemli gördüğü diğer bir olay ise yapay zekanın ortaya çıkışıdır. Yapay zekanın geliştirilmesiyle birlikte insana gerek duyulan alan ve rollerde değişiklikler yaşanmıştır. Canlılığın başlangıcında yaşamda kalmak için avcılık ve toplayıcılığa gereksinim duyan insanın günümüzde hayatta kalabilmesi için yapay zekaya sahip makinelerle uyum içinde çalışabilme, yapay zekâ sistemlerini kullanabilme ve geliştirebilme becerisine sahip olması beklenmektedir. Yapay zekaya sahip makinelerin iş kollarına dahil edilmesiyle beraber insandan beklenen fiziki güce ve basit bilişsel becerilere duyulan gereksinim azalmıştır [72]. Yapay zekaya dayalı sistemlerin yaygınlaşması onun insanların yerine geçeceği ve insanların işsiz kalacağı korkularını da beraberinde getirmiştir. Ancak bir kısım meslek dallarına duyulan gereksinimin azalması insanların işsiz kalacağı anlamına gelmemelidir. Yapay zekanın yaygınlaşmasıyla beraber bazı meslek dalları önemini yitirirken bazı alanlarda yeni meslekler ortaya çıkmaktadır. Kısa bir süre öncesine kadar adı duyulmayan sosyal medya uzmanlığı, bilişim avukatı, bilişim savcısı gibi meslek alanları popüler duruma gelmektedir. Yapay zekâ sistemleriyle pek çok meslek dalı yok olmaktan çok değişim geçirmektedir [73].

Yapay zekâ sistemleri, insana dayalı ucuz iş gücünün önemini azaltırken yapay zekâ sistemleriyle beraber çalışabilecek nitelikli insan gücüne olan gereksinimi arttırmaktadır. Yapay zekâ sistemlerinin geliştirilmesi ve yaygınlaşması eğitim sisteminden iş dünyasına, yöneticilerden çalışanlara değin toplumun her kesimini etkilemektedir. Bugün için gerekli olan iş gücünün niteliği yakın bir tarihte gereksiz duruma gelebilmektedir [74].

Yapay zekanın insan beyninin düşünme, tepki verme ve etkileşim gibi özelliklerine benzetilip benzetilemeyeceği halen tartışılıyor da günümüz yazılım ve donanımlarının giderek insan anlayışına daha da yakınlaştığı söylenebilir.

Sonuç olarak, günümüzde yapay zekâ sistemleri çok hızlı bir şekilde gelişim ve yayılım gösterdiğinden, geline nokta yapay zekanın karşısında durmanın gereksiz bir uğraş olacağı açıktır. İnsanlar yapay zekâ sistemleriyle yarışmak yerine onu kullanma ve yönetebilme becerilerini geliştirmeye odaklanmalıdır. Devletlerin en küçük birimlerinden en üst birimlerinde çalışanlara, toplumun en yaşlı bireylerinden en genç bireylerine değin yapay zekâ sistemlerini anlayacak ve uyum sağlayacak düzeye gelebilmeleri için eğitimlere tabi tutulmaları gerekmektedir. Ayrıca yapay zekâ sistemlerinin kötüye kullanımını engellemek

için kontrol ve denetim mekanizmalarının kurulması, çalışmaların daha sağlıklı yürütülmesi açısından yarar sağlayacaktır [75].

2.5 MAKİNE ÖĞRENMESİ TEMELLERİ

Bu bölümde makine öğrenmesi hakkında genel bilgiler verilmiştir.

2.5.1 Makine Öğrenmesi Nedir?

Makine öğrenimi, yapay zekânın alt disiplinlerinden biri olarak özellikle büyük veri kümelerinin yaygın hale gelmesiyle birlikte giderek daha fazla önem kazanmıştır. Bu terim, 1959 yılında Amerikalı bilgisayar bilimcisi Arthur Samuel tarafından, "bilgisayarların açıkça programlanmaksızın öğrenme yetisi" olarak tanımlanmıştır. Makine öğrenimi algoritmaları, genellikle belirli özelliklere ve örüntülere dayanarak çalışır. Bu algoritmalar, karmaşık yapıya sahip büyük veri setlerini analiz etmek ve bu verilerden anlamlı sonuçlar çıkarmak amacıyla geliştirilmiştir [76].

Bir makinenin davranışlarında ya da algoritmasında meydana gelen değişiklikler, sistemin dış çevreden aldığı veri ya da tepkiler doğrultusunda performansını geliştirmesini sağlar. Bu bağlamda, makine öğrenmesi; makinelerin doğru verilerle beslendiğinde, belirli görevleri öğrenebilme ve çözüm üretebilme yeteneğini ifade eden bir yapay zekâ alanı olarak değerlendirilmektedir. Matematiksel modeller ve istatistiksel yöntemlerden faydalanan makine öğrenimi, geleneksel olarak insan zekâsına atfedilen bilişsel işlevlerin otomatikleştirilmesini amaçlar. Aynı zamanda, yapay zekâ kapsamındaki örüntü tanıma ve hesaplamalı öğrenme teorilerinden türemiştir. Bu sistemlerde algoritmalar, veriler üzerinde analizler yaparak tahminlerde bulunur [76].

Makine öğrenmesi, sabit ve önceden tanımlanmış programlama kurallarını izlemek yerine, veri odaklı karar verme veya tahmin süreçlerini mümkün kılan algoritmalara dayanır. Bu algoritmalar, genellikle “eğitim verisi” olarak adlandırılan örnek veri kümelerini kullanarak, belirli görevleri yerine getirmeye yönelik matematiksel modeller geliştirir. Daha teknik bir tanımla makine öğrenmesi, belirli bir probleme ilişkin çevresel koşullardan elde edilen veriler doğrultusunda, bu problemi modelleyebilen bilgisayar algoritmalarının genel başlığı altında değerlendirilir [77].

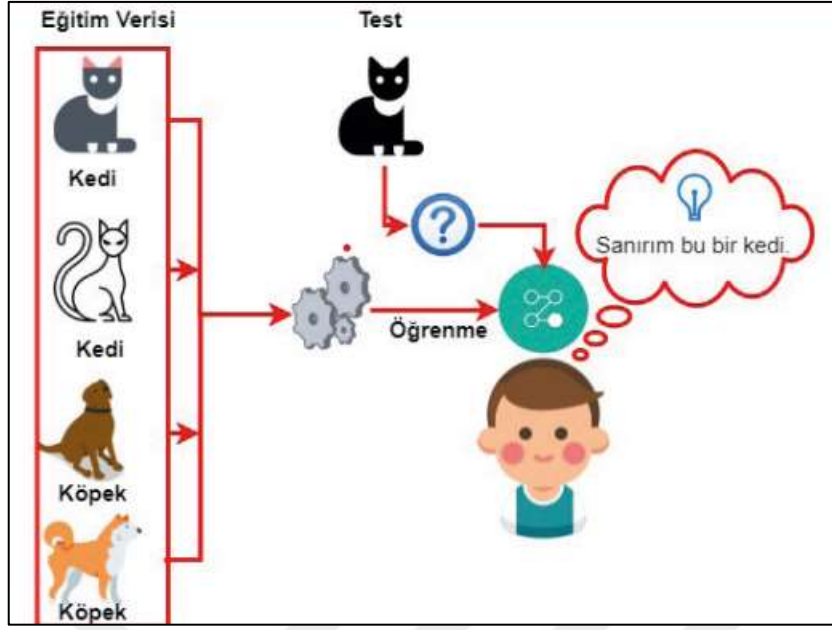
Bu yöntem, günümüzde saldırı tespit sistemlerinden kötü amaçlı yazılım analizine, kanserli hücrelerin tespiti ve görüntü tabanlı kişi veya nesne tanıma gibi birçok alanda başarıyla kullanılmaktadır [77].

Makine öğrenmesini günlük yaşamdan bir örnekle açıklamak gerekirse: Küçük bir çocuk ilk defa bir kedi gördüğünde, nesneyi tanımlamakta zorlanabilir. “Bu bir kedi” şeklinde bir yanıt aldığı anda, zihninde bu canlıya dair genel bir kavramsal model oluşur. Zamanla farklı renk, boyut ve türde kedilerle karşılaştıkça, bu kavramsal şema gelişir ve çeşitlenir. Bu süreç, makine öğrenmesinde bir modelin eğitim verileriyle öğrenerek genelleme yapmasına benzetilebilir [77].



Şekil 2.2: Kedi Türleri.

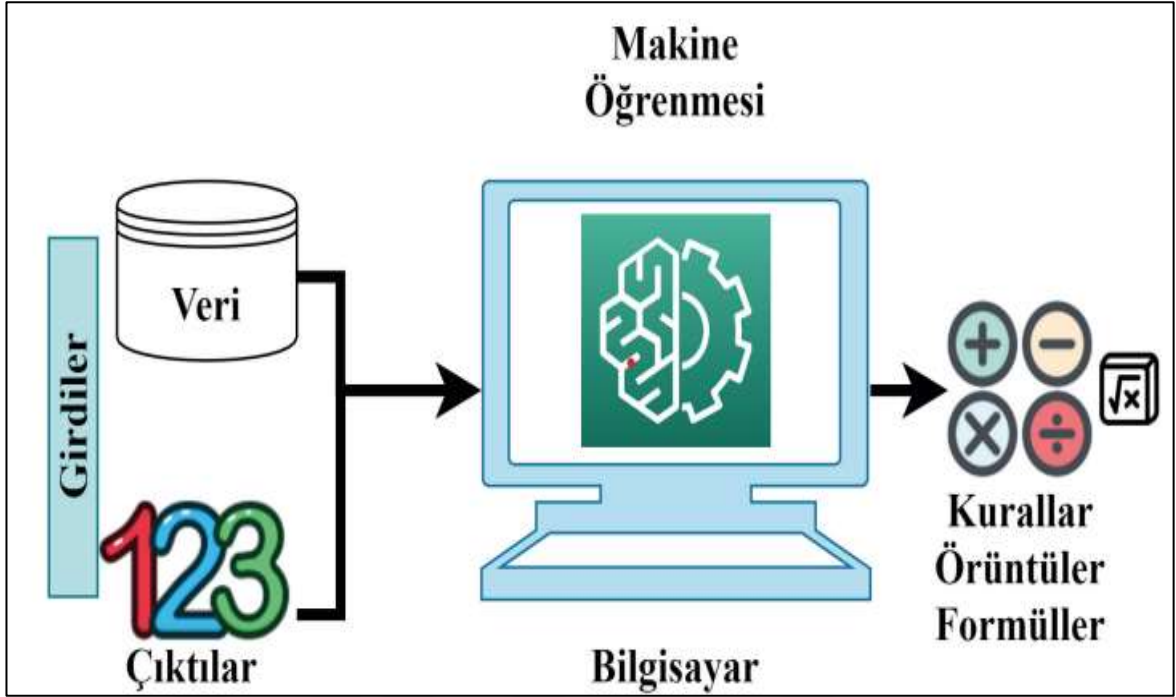
Bir çocuk, kedilere ait örneklerle karşılaştıkça, zihninde bu hayvanlara özgü ortak karakteristikler —örneğin ayak sayısı, kuyruk, kulak yapısı ve bıyıklar— giderek belirginleşen bir şablon hâlini almaya başlar. Bu özellikler, kedileri diğer hayvan türlerinden ayırmada temel referans noktalarıdır. Çocuk, yeni bir kediyle karşılaştığında, bu zihinsel şablonla karşılaştırma yaparak gördüğü canlının kedi olup olmadığına karar verir. Bu öğrenme süreci doğal bir şekilde ilerler; çocuklar bu sınıflandırmaları yapmak için bilinçli bir şekilde zaman ayırmazlar [77].



Şekil 2.3: Çocuklarda Nesne Tanımlama Örneği.

Benzer bir süreç, yapay öğrenme sistemlerinde de gözlemlenebilir. Örneğin, kedi ve köpek görsellerinden oluşan etiketli bir veri seti kullanılarak bir makine öğrenmesi modeli eğitilir. Bu model, sınıfları temsil eden belirgin özellikleri verilerden çıkartır. Sonrasında kendisine sunulan yeni bir görseli analiz ederek, sahip olduğu sınıf temsilleriyle ne ölçüde örtüştüğüne göre bir sınıflandırma yapar [77].

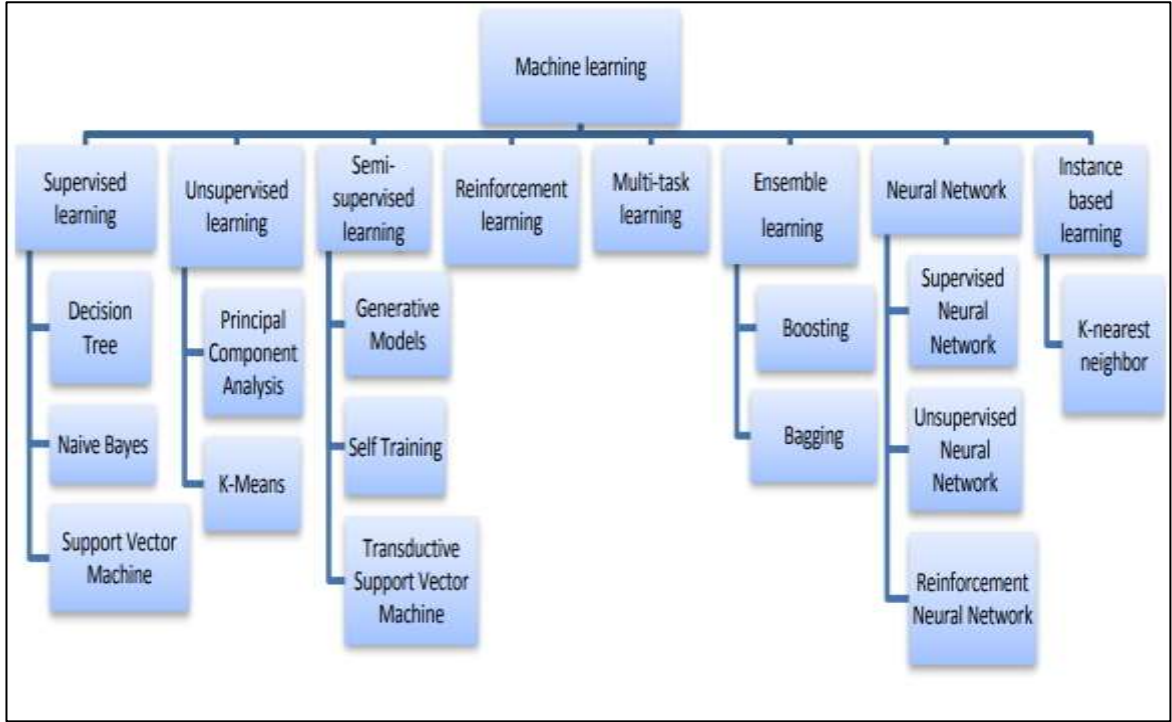
Bu yaklaşımı klasik programlama ile karşılaştırmak, makine öğrenmesini daha net kavramak açısından faydalı olabilir. Geleneksel programlamada geliştiriciler, tüm olasılıkları göz önüne alarak koşullu yapılar ve sabit kurallar tanımlar. Program yalnızca bu kurallar çerçevesinde çalışır ve dışına çıkamaz. Oysa makine öğrenmesinde, veri ile çıktı arasındaki ilişkiler, algoritmalar tarafından öğrenilir. Böylece manuel kod yazımı yerine, sistemin kendisi örüntüleri çıkarır. Bu durum, özellikle büyük veri setlerinde klasik programlamaya kıyasla çok daha etkin ve esnek çözümler sunar [78].



Şekil 2.4: Klasik Programlama Örneği.

2.5.2 Makine Öğrenimi Teknikleri

Makine öğrenimi, temel olarak bilgisayarların belirli bir konuda öğrenmesini sağlayan algoritmaların geliştirilmesiyle ilgilenir. Farklı makine öğrenimi algoritmaları; kullandıkları veriler, elde ettikleri çıktılar ve çözmeye çalıştıkları problemler açısından birbirlerinden ayrılmaktadır. Bu tekniklerin bazıları, insanların bir problemi çözme biçimlerini model olarak tasarlanmıştır. Birçok matematikçi ve yazılımcı, aşağıda sunulan şemada yer alan şekilde sınıflandırılmış çeşitli makine öğrenimi yöntemleri üzerine çözümler geliştirmiştir. Bu bağlamda, farklı makine öğrenimi teknikleri ve bu tekniklerin uygulanabileceği örnek alanlar aşağıda ele alınmaktadır [79].



Şekil 2.5: Makine Öğrenmesi Teknikleri.

2.5.2.1 Denetimli öğrenme (Supervised learning)

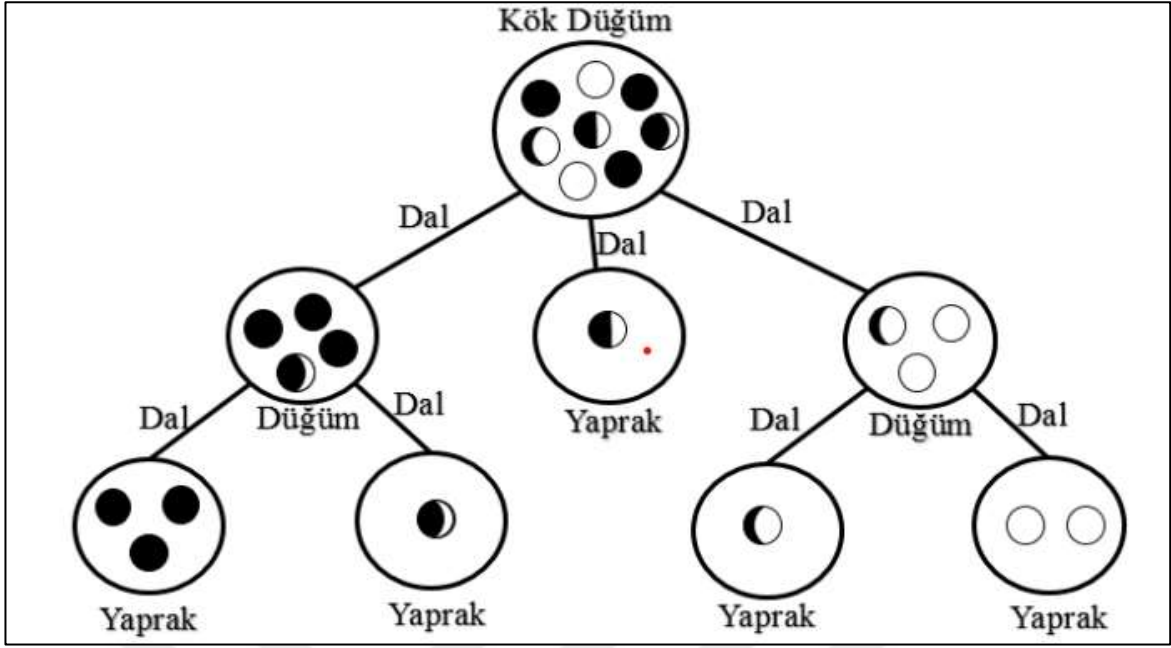
Hem girdileri hem de bu girdilere karşılık gelen hedef çıktıları içeren veri kümeleri üzerinden matematiksel modeller kurar. Bu veriler, “eğitim verisi” olarak adlandırılır ve her biri bir veya daha fazla girdiyle birlikte “hedef çıktı” ya da “denetleyici sinyal” içeren örneklerden oluşur. Her bir örnek, genellikle “özellik vektörü” olarak bilinen bir dizi ya da vektörle temsil edilirken, tüm veri kümesi bir matris biçiminde modellenir. Bu süreçte algoritmalar, yinelemeli şekilde optimize edilen bir amaç (loss) fonksiyonu aracılığıyla, yeni veriler için doğru çıktıları tahmin edebilecek bir fonksiyon öğrenir. Öğrenilen fonksiyonun ideal olması, algoritmanın daha önce görmediği girdilere yönelik doğru tahminler yapabilmesini sağlar [80]. Tahmin başarımı zamanla artan bir algoritmanın, ilgili görevi öğrenmiş olduğu kabul edilir. Denetimli öğrenme kapsamına giren başlıca iki alt alan; sınıflandırma ve regresyondur. Sınıflandırma, çıktının belirli sınırlı bir değerler kümesinden seçildiği durumlar için kullanılırken, regresyon ise çıktının sürekli bir sayısal aralıkta olabildiği durumlara uygulanır. Ayrıca, benzerlik öğrenmesi olarak bilinen alan da denetimli öğrenmeyle yakından ilişkilidir ve iki öge arasındaki benzerliği ölçen bir benzerlik fonksiyonu aracılığıyla öğrenme gerçekleştirilir. Bu teknik; sıralama, öneri sistemleri, yüz tanıma ve konuşmacı doğrulama gibi pek çok uygulamada kullanılmaktadır. Yarı denetimli

öğrenmede ise veri setinde yer alan bazı örneklerin etiketleri eksiktir; buna rağmen modelin başarımını artırmada bu veriler de faydalı olabilir. Zayıf denetimli öğrenme ise sınırlı, hatalı ya da net olmayan etiketlerin kullanıldığı bir yaklaşımdır; bu tür etiketlerin elde edilmesi daha ekonomik olduğundan, daha geniş kapsamlı eğitim verileri ile çalışma imkânı sağlar [81].

Denetimli öğrenme için çeşitli algoritmalar mevcuttur. Yaygın olarak kullanılan denetimli öğrenme algoritmalarından bazıları şunlardır:

Karar ağaçları (Decision trees)

Sınıflandırma problemlerinde yaygın olarak kullanılan yöntemlerden biridir. Bunun temel nedeni, bu algoritmaların yapılandırılmasının, anlaşılmasının ve yorumlanmasının oldukça kolay olmasıdır. Denetimli bir öğrenme yöntemi olan karar ağaçları, etiketlenmiş veriler kullanılarak oluşturulmaktadır. Karar ağaçları, verilerin niteliklerini dikkate alarak bu nitelikleri farklı değerlere göre gruplar ve sıralar. Bu ağaç yapıları, düğüm ve dallardan meydana gelir ve çoğunlukla sınıflandırma işlemleri için tercih edilir. Düğümler, bir veri kümesindeki belirli nitelikleri temsil ederken; dallar, bu niteliklerin alabileceği olası değerleri gösterir. Karar ağaçlarının oluşturulma sürecinde dallanmanın hangi kritere göre yapılacağı konusunda çeşitli yöntemler mevcuttur. En sık başvuru alan yaklaşımlar arasında bilgi kazancı, Gini İndeksi, Twoing kuralı ve Ki-Kare olasılık tablo istatistiği yer almaktadır. Karar ağaçlarının dallanma süreci, eğitim verilerini benzer özellikler gösteren daha küçük alt kümelere ayırmayı hedefler. Bu işlem, belirlenen kesme kuralları sağlanana kadar devam eder. [81]



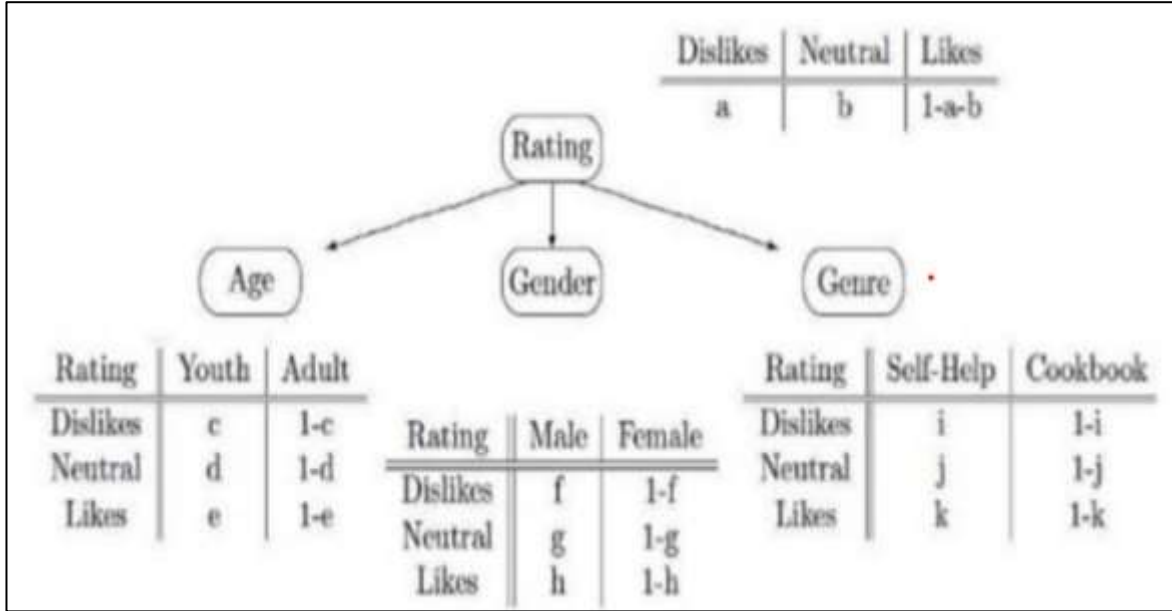
Şekil 2.6: Karar Ağaçları Yapısı.

Naive bayes

Naive Bayes algoritması, İngiliz matematikçi Thomas Bayes'in geliştirdiği koşullu olasılık kuramına dayanan bir sınıflandırma yöntemidir. Bu algoritma, daha önceden sınıfları bilinen verilerle eğitim sürecini tamamladıktan sonra, sınıfı bilinmeyen yeni verileri sınıflandırmak amacıyla kullanılır. Ayrıca, sadece sınıflandırma değil, kümeleme gibi amaçlarla da uygulanabilmektedir. Naive Bayes algoritmasının temelini oluşturan koşullu olasılık yaklaşımı, gerçekleşme olasılıklarına göre yapılandırılmış olasılık ağaçları meydana getirir ve bu yapılar Bayes Ağı olarak adlandırılabilir. Naive Bayes sınıflandırıcısı, Bayes Teoremi'ne dayalıdır [82].

- $P(A)$: A olayının meydana gelme olasılığı,
- $P(B)$: B olayının meydana gelme olasılığı,
- $P(A|B)$: B gerçekleştiğinde A'nın gerçekleşme olasılığı,
- $P(B|A)$: A gerçekleştiğinde B'nin gerçekleşme olasılığıdır.

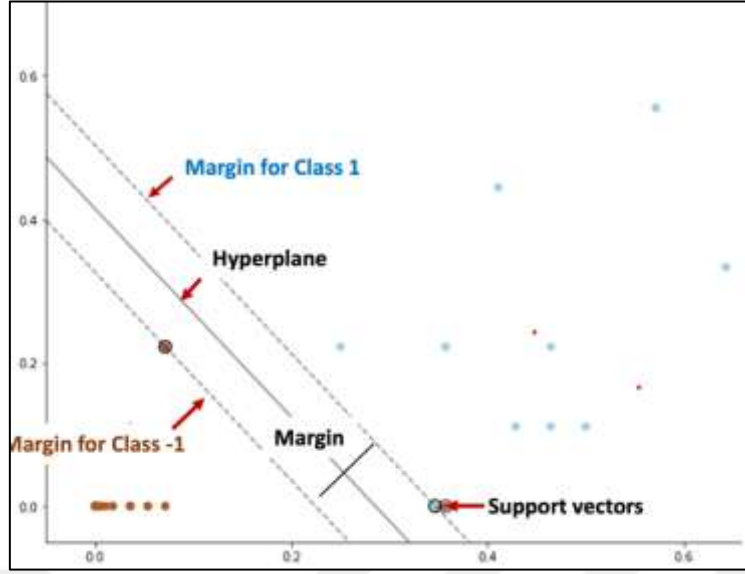
Naive Bayes algoritmasının uygulama alanına bağlı olarak çeşitli türleri bulunmaktadır. Özellikler sürekli değerler alıyorsa Gaussian Naive Bayes, çok sınıflı verilerle çalışılması gerekiyorsa örneğin e-ticaret sitelerindeki ürünlerin spor, giyim, kozmetik gibi kategorilere ayrılması durumunda Multinomial Naive Bayes tercih edilmektedir. Eğer sınıflandırma yalnızca ikili (örneğin: Evet/Hayır, Spam/Spam Değil, 1/0 gibi) değerler alıyorsa, bu durumda Bernoulli Naive Bayes algoritması kullanılmaktadır [82].



Şekil 2.7: Naive Bayes Örneği.

Destek vektör makineleri (Support vector machines- SVM)

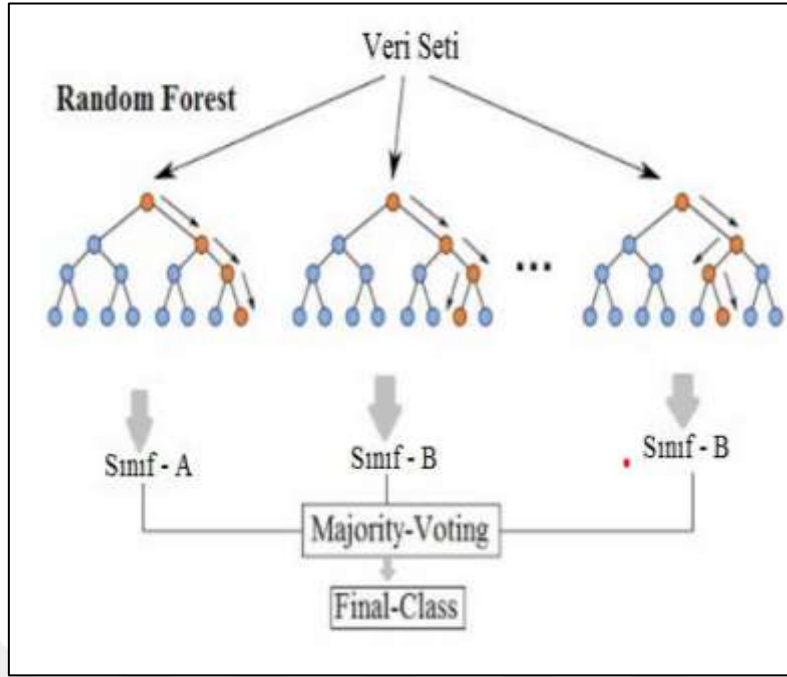
Lojistik regresyon, verileri bir doğru yardımıyla iki sınıfa ayırmakla sınırlı kalırken, SVM bu ayrımı maksimum marjini (aralığı) sağlayacak şekilde gerçekleştirir. Bu sınırın belirlenmesinde destek vektörleri rol oynar. Sınıflar arasındaki örnekler, ayırıcı sınır çizgisine en yakın mesafedeki veriler olarak yer alır. Amaç, modelin genelleme yeteneğini artırmak ve aşırı öğrenmeyi önlemek için, farklı sınıfları birbirinden en uzak mesafeyle ayırabilmektir [83].



Şekil 2.8: Destek Vektör Makineleri (Support Vector Machines- SVM).

Rasgele Orman (Random Forest)

Random Forest yöntemi, Bagging algoritmasının temelleri üzerine kurulmuştur. Bagging algoritması ilk olarak 1996 yılında Breiman tarafından geliştirilmiş olup, eğitim verisinden farklı alt örnekleme kümeleri elde ederek bu alt kümelerle birçok karar ağacı oluşturur. Kullanıcı tarafından belirlenen sayıda karar ağacı üretilir ve ağaç sayısı arttıkça sınıflandırma doğruluğu artış gösterir. Bu ağaçlar, genellikle budama yapılmaksızın maksimum derinliğe ulaşacak şekilde geliştirilir. Ağaçların oluşturulmasında CART algoritması tercih edilir. Modelin tahmin aşamasında, gelen yeni bir veri tüm ağaçlara ayrı ayrı sunulur, her ağacın sınıflandırma kararı alınır ve en fazla oyu alan sınıf (Majority Voting) son karar olarak belirlenir. Random Forest algoritmasının bu oylama süreci Şekil 2.9’da gösterilmiştir [84].



Şekil 2.9: Rasgele Orman (Random Forest).

2.5.2.2 Denetimsiz öğrenme (Unsupervised learning)

Denetimsiz öğrenme algoritmaları, yalnızca giriş verilerini temel alarak, etiketlenmemiş veri kümelerinde örüntüleri ve yapıları ortaya çıkarma amacı taşır. Bu algoritmalar, sınıflandırılmamış ya da kategorize edilmemiş verilerle çalışır ve verilerdeki benzerlikleri analiz ederek bu benzerliklere göre gruplandırmalar yapar. Denetimsiz öğrenme, kullanıcı yorumları gibi dış yönlendirmeler olmadan, veriler arasındaki ilişkilere dayanarak yeni bilgileri organize eder. Sağlık analizleri gibi istatistiksel uygulamalar, denetimsiz öğrenmenin öne çıktığı alanlardan biridir. Ayrıca veri özetleme ve yapılandırma gibi işlemler de bu öğrenme türünün kapsadığı diğer uygulama sahalarıdır [84].

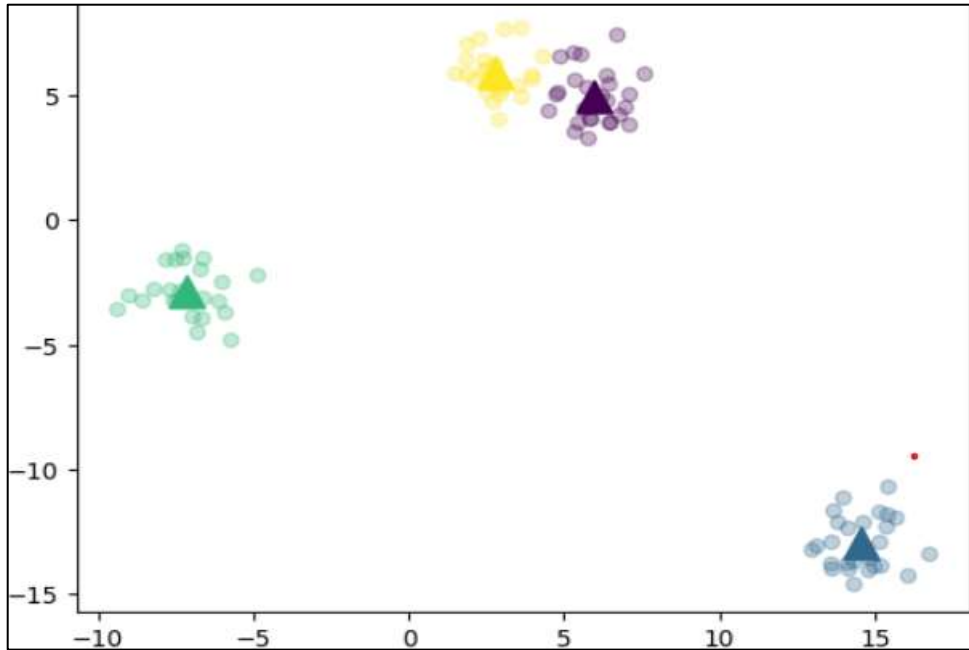
Kümeleme analizi, bir gözlem grubunu benzer özelliklere sahip alt gruplara – yani kümelere – ayırma işlemidir. Aynı kümede yer alan gözlemler, belirli ölçütler doğrultusunda benzerlik gösterirken, farklı kümelerden alınan gözlemler arasında belirgin farklar bulunur. Uygulanan kümeleme yöntemleri, örnekler arasındaki içsel benzerlik, kümeler arasındaki ayrışma gibi kriterlere göre farklı varsayımlar geliştirir. Bunun dışında bazı teknikler, veri yoğunluğunu ya da bağlantı yapısını esas alır. Aşağıda denetimsiz öğrenme algoritmalarından olan k-ortalama ve hiyerarşik kümeleme anlatılmıştır. Ancak denetimsiz öğrenme algoritmaları bu ikisiyle sınırlı değildir [85].

K-ortalamlar kümeleme (K-means clustering)

K-Means, gözetimsiz öğrenme kapsamında kullanılan bir kümeleme algoritmasıdır. Bu yöntemde K değeri, verinin kaç kümeye ayrılacağını belirler; dolayısıyla doğru bir kümeleme için K değerinin uygun şekilde seçilmesi kritik öneme sahiptir. K değerinin belirlenmesinde farklı yöntemler kullanılabilirken, bu değeri otomatik olarak belirleyebilen x-Means gibi geliştirilmiş versiyonlar da mevcuttur [85].

K değeri tanımlandıktan sonra algoritma, rastgele K adet merkez noktası belirleyerek çalışmaya başlar. Veri setinde yer alan her bir nokta, bu merkezlerle arasındaki mesafeye göre değerlendirilir ve en yakın merkeze atanır. Bu işlemin ardından, her küme için yeni bir merkez noktası (kümenin ağırlık merkezi) hesaplanır ve bu merkezler temel alınarak kümeleme işlemi tekrar edilir. Süreç, merkez noktaları sabitlenene kadar yani sistem kararlı hale gelene kadar devam eder [85].

Ancak, başlangıçta merkez noktalarının rastgele belirlenmesi bazen kümelerin dengesiz veya hatalı oluşmasına neden olabilir. Bu sorunun önüne geçmek amacıyla K-Means++ adı verilen geliştirilmiş bir algoritma sürümü ortaya konmuştur. K-Means++'ta ilk merkez rastgele seçilir, ardından diğer merkezler, mevcut merkezle aralarındaki uzaklıkların karelerine göre hesaplanan bir olasılık dağılımına göre belirlenir [85].



Şekil 2.10: K-Ortalamlar Kümeleme (K-Means Clustering).

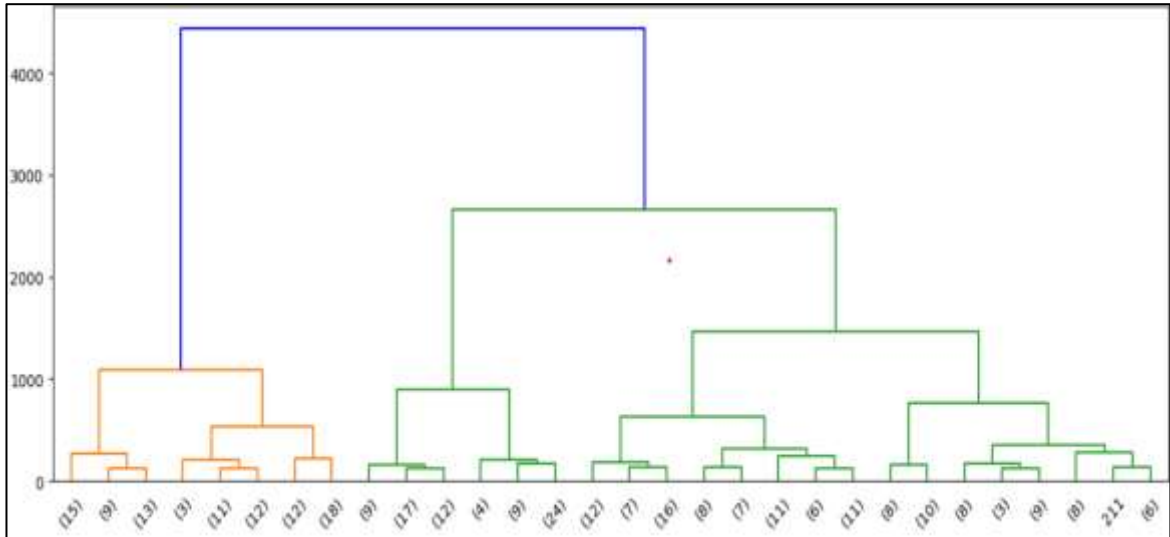
Hiyerarşik kümeleme (Hierarchical clustering)

Hiyerarşik kümeleme algoritmaları, temel olarak iki ana yaklaşım altında incelenir: Yığınsal (Agglomerative- parçadan bütüne) ve Bölümlemeli (Divisive - bütünden parçaya) yöntemler [86].

Agglomerative yönteminde her bir veri noktası başlangıçta bağımsız bir küme olarak kabul edilir. Daha sonra aralarındaki benzerlik esas alınarak en yakın kümeler birleştirilir. Bu birleşme işlemi, sistemde yalnızca tek bir küme kalana kadar devam eder. Öte yandan, Divisive yaklaşım bu sürecin tersini izler. Başlangıçta tüm veriler tek bir küme içinde yer alır ve bu küme, süreç boyunca alt parçalara ayrılarak her eleman tek başına kalana kadar devam eder [86].

Agglomerative hiyerarşik kümelemede, kümeler arasındaki mesafeyi belirlemek için kullanılan ölçüm yöntemi sonuçları doğrudan etkiler. Bu ölçüm teknikleri aynı zamanda dendrogram adı verilen ağaç yapılarının oluşturulmasında da kullanılmaktadır:

- a. Single Linkage (Tek bağlantı): İki küme arasındaki en kısa mesafeyi baz alır.
- b. Complete Linkage (Tam bağlantı): Kümeler arasındaki en uzun mesafeyi dikkate alır.
- c. Average Linkage (Ortalama bağlantı): Kümeler arasındaki tüm mesafelerin ortalaması temel alınır.



Şekil 2.11: Hiyerarşik Kümeleme (Hierarchical Clustering).

2.5.2.3 Yarı denetimli öğrenme (Semi supervised learning)

Yarı denetimli öğrenme, içinde az sayıda etiketlenmiş örnek ile bol miktarda etiketlenmemiş örneğin yer aldığı veri kümeleri üzerinde çalışan hibrit bir öğrenme yaklaşımıdır. Bu teknik, denetimli ve denetimsiz öğrenme prensiplerini birleştirerek, sınıflandırma, regresyon ve öngörüsöl modelleme gibi görevlerde kullanılabilir. Öncelikle etiketli verilerden elde edilen bilgi, denetimsiz yöntemlerle oluşturulan kümelerin anlamlandırılmasında kullanılır. Tanımlanan bu kümelere daha sonra sözde etiketler atanır ve hem bu sahte etiketli veriler hem de gerçek etiketli veriler birlikte kullanılarak modelin eğitimi devam eder. Bu yöntem; etiketli verinin az bulunduğu, ancak çok sayıda etiketsiz örneğe ulaşıldığı durumlarda – özellikle bilgisayarlı görü, doğal dil işleme gibi alanlarda – oldukça etkilidir. Aynı zamanda veri madenciliği çalışmalarında da yaygın bir şekilde uygulanmaktadır [87].

Üretken modeller (Generative models)

Yarı denetimli öğrenme alanında kullanılan en erken yöntemlerden biri olan generatif modeller, Gaussian karışım modelleri gibi yapıların, veri üzerinde karışık dağılım varsayımıyla çalıştığı tekniklerdir. Bu modeller, etiketlenmemiş verilerdeki örüntüleri tanımlayarak karışım bileşenlerini ortaya koyabilir [87].

Kendi kendine eğitim (self training)

Self-training yaklaşımında, öncelikle model etiketli verilerle eğitilir. Sonrasında modelin etiketsiz veriler üzerinde yaptığı yüksek güvenli tahminler, sanki etiketliymiş gibi tekrar modele geri beslenir. Bu işlem, iteratif şekilde devam eder ve model her döngüde kendini geliştirir [88].

Transdüktif destek vektör makinesi (Transductive support vector machine)

Transdüktif SVM, klasik SVM algoritmasının bir türevidir ve hem etiketli hem de etiketlenmemiş veriyi hesaba katar. Bu yaklaşım, karar sınırlarını belirlerken, etiketsiz verilerin bu sınırlara olan mesafesini de dikkate alarak daha geniş marjlı ve genelleyici sınıflamalar yapmayı amaçlar [86].

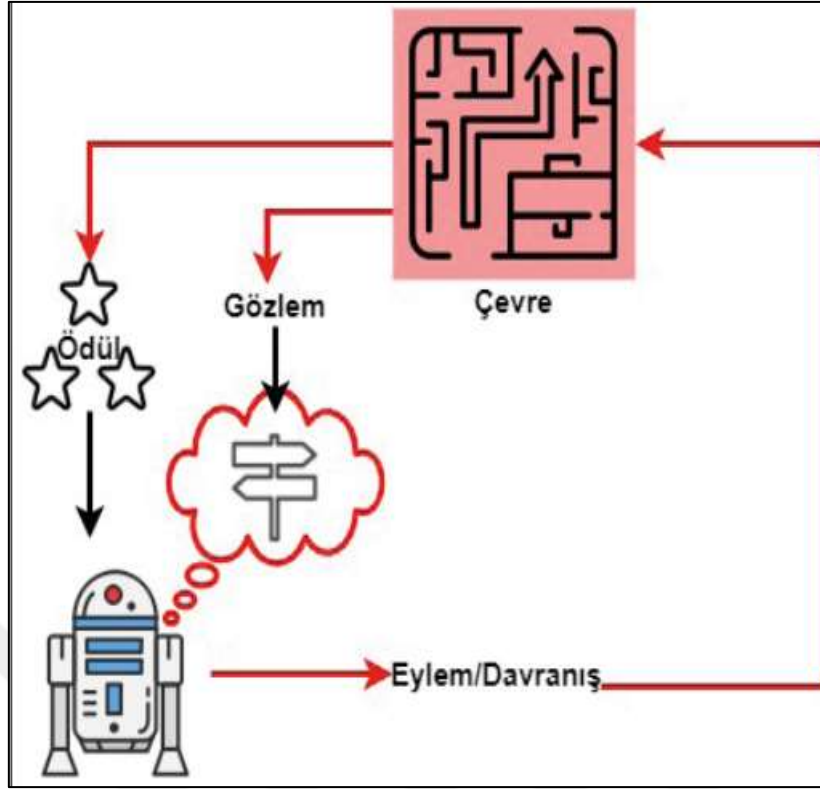
2.5.2.4 Güçlendirici öğrenme (Reinforcement learning)

Pekiştirmeli öğrenme, diğer makine öğrenme yöntemlerinden ayrılarak, öğrenme sürecini ödül ve ceza mekanizmasına dayandırır. Bu öğrenme biçimi, bir ajanın, içinde bulunduğu ortamda en yüksek toplam ödülü elde etmek adına hangi adımları atması gerektiğini deneme-yanılma yöntemiyle keşfetmesini esas alır [85].

Bu yöntem, yazılım ajanlarının çevreyle nasıl etkileşim kurarak kümülatif ödülü en üst düzeye çıkarabileceklerini belirlemeye odaklanır. Genelleştirilebilir doğası sayesinde; operasyon araştırması, kontrol kuramı, oyun kuramı, bilgi kuramı, simülasyon tabanlı optimizasyon, çoklu ajan sistemleri, istatistik, sürü zekâsı ve genetik algoritmalar gibi birçok disiplinde araştırma konusu olmuştur [85].

Makine öğrenmesinde pekiştirmeli öğrenme süreci genellikle Markov Karar Süreci (MDP) ile modellenir. Çoğu pekiştirmeli öğrenme algoritması, çalıştıkları sırada dinamik karar verme teknikleri kullanır. Bu algoritmalar, MDP'nin tam matematiksel modelinin bilinmesini gerektirmez, bu nedenle modelin belirli olmadığı ya da modellenmesinin mümkün olmadığı senaryolarda tercih edilir.

Örneğin, bir robotun bir labirentten çıkışı öğrenmesi, ya da bir yazılımın rakibe karşı stratejik oyun oynaması gibi durumlarda uygulanır. Bu bağlamda, ajanın ödül alması ya da skorunun artması gibi geri bildirimler, öğrenme sürecine yön verir (Görsel 2.13). Gözetimli öğrenmeden farkı, çevre hakkında önceden tanımlı bir etiket veya öğretici bilgi olmamasıdır. Pekiştirmeli öğrenme özellikle etiketli verinin sınırlı olduğu, ya da ortam hakkında bilgi sahibi olunmayan durumlarda tercih edilir. Robotik, oyun geliştirme ve kişiselleştirilmiş öneri sistemleri bu tekniğin yaygın kullanıldığı alanlardır. Pekiştirmeli öğrenme kapsamında en çok bilinen algoritmalar arasında Q-öğrenme, SARSA ve Zamansal Farklama (Temporal Difference – TD) yer almaktadır [87].



Şekil 2.12: Bir Robotun Labirentten Çıkışı Öğrenmesi.

2.5.2.5 Çoklu görev öğrenme (Multitasking learning)

Çoklu görev öğrenmesi, bir algoritmanın belirli bir problemi nasıl çözdüğünü ve sonuca nasıl ulaştığını hatırlamasına dayanır [54]. Bu çözümleme süreci, başka görevler için de bir referans olarak kullanılır.

Bu yöntem, aktarım mekanizması olarak da tanımlanabilir [73]. Öğrenciler arasındaki bilgi paylaşımı sayesinde, öğrenme süreci bireysel değil, eş zamanlı gerçekleşir ve bu da genel olarak daha hızlı öğrenme sağlar.

2.5.2.6 Topluluk öğrenmesi (Ensemble learning)

Topluluk öğrenmesi (Ensemble Learning), birden fazla makine öğrenmesi algoritmasının bir araya getirilerek tek bir problemin çözümünde birlikte kullanılması temeline dayanır. Bu yöntem, genellikle "zayıf öğrenici" (weak learner) olarak adlandırılan bireysel modellerin bir araya gelmesiyle daha güçlü ve daha doğru bir tahmin mekanizması oluşturulabileceği varsayımına dayanır [88].

Her bir zayıf öğrenici, eğitim verileri üzerinde ayrı ayrı eğitilerek tahminlerde bulunur ve bu tahminler daha sonra birleştirilerek nihai karar üretilir.

Topluluk öğrenme yöntemleri, özellikle model başarımını artırmak, doğruluğu yükseltmek veya hata oranını azaltmak gibi hedefler söz konusu olduğunda tercih edilir. Ayrıca, modelin aşırı öğrenme (overfitting) eğiliminde olduğu durumlarda da faydalı olabilir. Çünkü topluluk yapısı, birden fazla modelin öngörülerini dengeleyerek daha kararlı sonuçlar üretir [89].

Tahminlerin birleşimi farklı yollarla yapılabilir. En yaygın teknikler arasında ortalama alma (averaging), oylama (voting) ve istifleme (stacking) yöntemleri yer almaktadır.

Topluluk öğrenmesinin temel türleri

- a. En Çok Oy (Majority Voting): Sınıflandırma problemlerinde, her bir model bir oy verir ve en çok oy alan sınıf tahmin edilir.
- b. Ortalama Alma (Averaging): Özellikle regresyon problemleri için uygundur; tahminler aritmetik ortalamayla birleştirilir. Örneğin, random forest regresyon modelleri, bireysel ağaçların tahminlerinin ortalamasını alarak sonuca ulaşır.
- c. Ağırlıklı Ortalama (Weighted Average): Bu yöntemde, temel modellerin performansları dikkate alınarak her birine farklı ağırlıklar atanır. Tahmin gücü yüksek olan modellere daha fazla önem verilmesi esas alınır. Önceki örnek bağlamında, toplam ağırlıkların 1 olacak şekilde dağıtılması gereklidir [89].

İleri seviye topluluk öğrenme teknikleri

Yukarıda bahsedilen temel yöntemlerin ötesinde, daha karmaşık topluluk teknikleri de mevcuttur. Bunlar aşağıda açıklanmıştır [90]:

- a. İstifleme (Stacking): Farklı tahminleme algoritmaları, hatalarını azaltmak amacıyla birlikte kullanılır. Bu yöntem, birden çok modelin çıktılarını “meta model” adı verilen bir üst düzey öğreniciye girdi olarak verir. Meta model, genellikle çapraz doğrulama (cross-validation) yoluyla eğitilir. İstifleme yöntemi hem regresyon hem de sınıflandırma problemlerinde kullanılabilir. Regresyonda meta modele sayısal çıktılar girilirken, sınıflandırmada sınıf etiketleri veya olasılık değerleri kullanılır.

- b. Karıştırma (Blending): İstiflemeye benzer şekilde çalışır; ancak burada modeller, eğitim verisinin belirli bir bölümünde eğitilirken, kalan kısmı test verisi olarak değerlendirilir. Bu teknik, istiflemeye göre daha basit bir veri ayrımı kullanır. Eğitim setinin bir bölümü dışarıda bırakılarak modeller eğitilir, ardından bu bölüm üzerinden meta model oluşturulur.
- c. Torbalama (Bagging): Torbalama, veri setlerinden rasgele alınan örneklerle birden fazla model eğitilerek sonuçların ortalaması alınır. Bu yöntem, önyüklemeli toplama (bootstrap aggregating) olarak da bilinir. Torbalama yöntemi, model çeşitliliği yoluyla genelleme başarısını artırmayı hedefler.
- d. Güçlendirme (Boosting): Bu yöntem, zayıf öğrencileri art arda geliştirerek güçlü bir modele dönüştürür. Her yeni model, bir öncekinin yaptığı hatalara odaklanarak eğitilir. Böylece hem önyargı (bias) hem de varyans azaltılarak daha başarılı sonuçlar elde edilir.

2.5.2.7 Sinir ağları öğrenmesi (Neural network learning)

Sinir ağı öğrenme tekniği, dendrit, soma, çekirdek ve akson gibi dört ana bileşeni bulunan biyolojik nöron yapısından esinlenilerek geliştirilmiştir [58]. Bu nöronların birbirine bağlanmasıyla sinir ağı meydana gelir. Yapay sinir ağları da benzer bağlantılar kurarak çalışır. Sistemdeki giriş katmanı verileri alır, gizli katman bu verileri işler ve çıkış katmanı nihai sonucu üretir.

Yapay sinir ağları, denetimli, denetimsiz ve pekiştirmeli olmak üzere üç ana gruba ayrılır. Bu modeller, özellikle görüntü tanıma, ses işleme ve doğal dil gibi zorlu görevlerde oldukça etkilidir [95].

Yapay Sinir Ağları (YSA), kapsamlı ilerleyen bölümlerde anlatılacaktır.

2.5.2.8 Örnek tabanlı öğrenme (Instance based learning)

Örnek tabanlı öğrenme yöntemi, yalnızca belirli bir örüntü türüne aşina olan ve bu bilgiyi yeni veriler üzerinde uygulamaya çalışan algoritmalarla ilgilidir [91]. Bu yaklaşım, test verisi gelmeden herhangi bir genelleştirme yapmadığı için "tembel öğrenme" olarak adlandırılır. Ancak veri miktarı arttıkça yöntem daha karmaşık hale gelir. Bu tür bir öğrenme tekniğine k-en yakın komşu (k-NN) algoritması örnek verilebilir. Tarım uygulamalarında,

örnek tabanlı öğrenme, özellikle yeşil dokuya sahip bitkilerin diğerlerinden ayrılması amacıyla bitki görüntülerinin ayrıştırılmasında kullanılır [92].

2.5.3 Makine Öğrenmesinde Veri Ön İşleme

Veri ön işleme, herhangi bir uygulama adımına geçmeden önce mutlaka yapılması gereken kritik bir süreçtir. Bu aşamada amaç, ham verileri daha kullanılabilir ve analiz edilebilir bir forma dönüştürmektir. Analitik işlemlere başlamadan önce eksik, tutarsız ya da hatalı veriler belirlenerek düzeltilir. Bu süreç, veri madenciliğinde oldukça önemli bir yer tutar ve güvenilir sonuçlar elde edebilmek için çeşitli dönüşüm veya veri dışlama işlemlerini içerir [93].

Tablo 2.1: Veri Ön İşleme Teknikleri.

Renk Dönüşümü	Renk formatının dönüştürülmesi, piksel parlaklıklarının yeniden ayarlanması ve genel parlaklık düzeltmeleri [93].
Normalizasyon	Göğüs röntgeni görüntülerinin standart hale getirilmesi, ortalama normalizasyon ve geometrik dönüşümler sayesinde CNN doğruluğunun artırılması, BT görüntülerinin özellik çıkarımı amacıyla normalize edilmesi, histogram eşitleme yöntemi olan CLAHE, genel histogram eşitleme ile CNN performansının iyileştirilmesi. Ayrıca görüntülerin geri yüklenmesi ve standardizasyonu, öğrenme kalitesini artırmaktadır [93].
Filtreleme	Görüntü düzeltme, kenar tespiti ve görüntü keskinleştirme, filtreleme işlemleri hem BT hem de X-ışını görüntüleri üzerinde uygulanır. Görüntülerin yeniden şekillendirilmesi de bu süreçte yer alır [93].
Segmentasyon	Görüntü segmentasyonu [43], görüntüyü farklı parçalara bölerek içindeki objeleri ve kenarları tespit etmeyi sağlar. Özellikle akciğer görüntülerinde lezyon, lob veya enfekte alanları tespit etmekte kullanılır. Segmentasyon, modelin yalnızca ilgili bölgeyi öğrenmesini sağladığından zatürre teşhisinde önemli bir ön işleme adımıdır. Göğüs röntgeni görüntülerinde segmentasyon, Fourier dönüşümleri ve kenar tabanlı, eşik tabanlı, bölge tabanlı, küme tabanlı ve havza tabanlı yöntemlerle segmentasyon işlemleri gerçekleştirilmiştir [94].

Tablo 2.2: Veri Ön İşleme Teknikleri (Devamı).

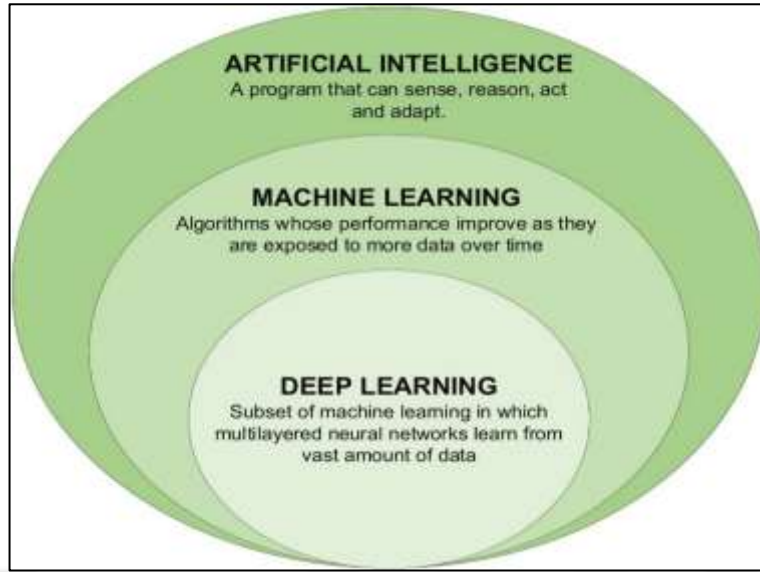
Veri Arttırma (Data Augmentation)	Makine öğrenimi modellerini eğitirken yeterli veri bulunmadığında, veri kümesinin boyutunu artırmak doğruluk oranını yükseltmek açısından kritik bir stratejidir. Bu süreç, özellikle azınlık sınıflarının temsilini artırarak çoğunluk sınıfına yönelik olası önyargıları azaltır ve bu sayede tüm sınıflarda daha isabetli tahminler yapılmasına olanak tanır [94]. Veri artırma için çeşitli yöntemler mevcuttur. Bunlardan ilki, çevirme, döndürme, kırpma, maskeleme ve yeniden ölçekleme gibi geometrik dönüşümleri içeren yaklaşımdır. İkinci yöntem, özellikle tıbbi görüntüleme gibi veri kısıtlılığı ve gizlilik sorunlarının ön planda olduğu alanlarda başvurulmuş sentetik veri üretimidir. Üçüncü yöntem ise Generative Adversarial Networks (GAN'lar) kullanılarak gerçekleştirilen veri artırmadır [93].
--	--

2.6 DERİN ÖĞRENME

Bu bölümde, derin öğrenmenin genel özellikleri üzerinde durulmuştur. Derin öğrenmeye dair temel kavramlar açıklandıktan sonra, bu alanla doğrudan bağlantılı olan yapay sinir ağları ve evrimsel sinir ağları konularına ilerleyen bölümlerde daha ayrıntılı olarak yer verilecektir [96].

2.6.1 Derin Öğrenmeye Giriş

Makine öğrenmesinin (ML) bir alt dalı olan derin öğrenme (DL), insan beyninin bilgi işleme yapısından ilham alınarak geliştirilmiştir. DL'nin çalışabilmesi için önceden belirlenmiş insan yapımı kurallara gerek yoktur; bunun yerine, büyük veri kümeleri kullanılarak giriş bilgileri belirli etiketlerle eşleştirilir. DL, katmanlı yapısıyla (çoğunlukla yapay sinir ağları aracılığıyla), her katmanda farklı bir temsil düzeyi oluşturacak şekilde çalışır. Son yıllarda derin öğrenme tabanlı yöntemler, makine öğrenmesinin en yaygın kullanılan tekniklerinden biri haline gelmiştir. Derin öğrenmenin bilim çevrelerinde büyük yankı uyandırması ilk kez 2012 yılında gerçekleşmiştir. Aynı yıl düzenlenen ve görsel tanıma alanında en prestijli yarışmalardan biri olan ImageNet Büyük Ölçekli Görsel Tanıma Yarışması'nda (Competition 2012), derin öğrenme mimarilerinden biri olan Evrimsel Sinir Ağı (ESA) kullanılarak birincilik elde edilmiştir. Bu başarıdan sonra derin öğrenme yöntemleri, birçok problemde üstün performans göstererek dikkat çekmiştir [97].

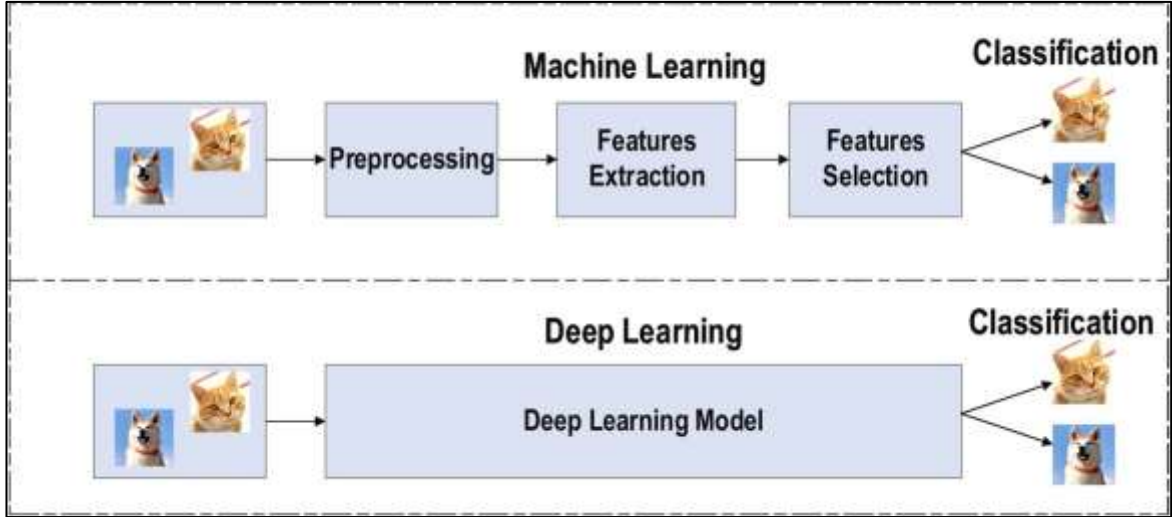


Şekil 2.13: Derin Öğrenmeye Gelene Kadar Genelden Özele Gösterim.

2.6.2 Derin Öğrenmenin Makine Öğrenmesinden Farkı

Derin öğrenmenin makine öğrenmesinden temel farkı, öğrenme sürecinde çok katmanlı yapay sinir ağlarının etkin biçimde kullanılmasıdır. Derin öğrenme algoritmalarının, girdilere ait özellikleri çok katmanlı yapılarla hiyerarşik düzeyde ve otomatik olarak öğrenebilmesi, bu yöntemleri geleneksel ve öz nitelik mühendisliğine dayanan yöntemlere kıyasla oldukça avantajlı kılar. Derin öğrenmenin başarısının temelinde, çok sayıda katmandan oluşan derin yapay sinir ağları yer almaktadır [96].

Bu yöntemler esasen yapay sinir ağı mimarileri üzerine kuruludur, ancak klasik YSA'lerden farklı olarak çok daha fazla sayıda nöron ve ara katman içermektedir. Klasik makine öğrenmesi sistemlerinde model kurmak için, öncelikle karmaşık yapıya sahip verilerin daha sade bir temsili olan özellik vektörleri elde edilmelidir. Bu özellik çıkarımı işlemi genellikle uzman bilgisine dayalı olup yüksek zaman ve emek maliyeti içerir. Bu bağlamda, ham veriler klasik algoritmalar tarafından doğrudan işlenemez. Derin öğrenme teknikleri bu problemi çözmüş ve veri işleme sürecinde büyük bir sıçrama sağlamıştır. Çünkü derin yapılar, ham veriyi doğrudan kullanarak gerekli temsilleri kendi katmanları içerisinde otomatik olarak oluşturur ve bu temsiller üzerinden öğrenme işlemini gerçekleştirir [97].



Şekil 2.14: Makine Öğrenmesi ve Derin Öğrenme Arasındaki Fark.

Bir makine öğrenimi algoritmasını evcil hayvan görsellerini doğru biçimde sınıflandıracak şekilde eğitmek için şu aşamalar gereklidir:

- Kartal, aslan, kanarya ve boğa gibi pek çok hayvanın binlerce görüntüsünü manuel olarak toplamak ve bu görüntüleri doğru sınıflarla etiketlemek.
- Modelin, hangi görsel özelliklere dikkat etmesi gerektiğini belirlemek. Örneğin, sistem bacak sayısını analiz ettikten sonra göz şekli, kulak yapısı, kuyruk biçimi ve tüy yapısını kontrol edebilir.
- Modelin doğruluğunu artırmak amacıyla etiketli veri seti üzerinde değerlendirme ve gerekirse düzeltmeler yapmak. Örneğin, eğitim verilerinde çok sayıda siyah kedi yer alıyorsa model siyah kedileri tanımada başarılı olurken, beyaz kedileri tanımakta zorlanabilir.
- Derin öğrenme yaklaşımında ise, bu tür özellik mühendisliğine gerek duyulmaz. Sinir ağı yapısı tüm görüntüleri kendisi analiz eder ve görüntüleri tespit ederek örneğin bacak sayısına, ardından yüz şekline ve son olarak kuyruk yapısına bakması gerektiğini kendi başına öğrenir [98].

3. GEREÇ VE YÖNTEM

3.1 GEREÇ

Bu bölümde, tez çalışmasında faydalanılan teknolojiler, kullanılan yardımcı kütüphaneler, geliştirme ortamları, veri setleri, oluşturulan model ve karşılaşılan zorluklara ilişkin detaylı bilgilere yer verilecektir. Kısaca özetlemek gerekirse, uygulama Google COLAB platformu üzerinde Python programlama dili ile geliştirilmiş olup, ilerleyen bölümde detayları verilecek olan kütüphaneler kullanılarak, derin öğrenme yöntemlerinden biri olan Konvolüsyonel Sinir Ağı (CNN) mimarisi ile akciğer röntgen görüntülerinden zatürre tespiti gerçekleştirilmiştir. Kendi sonuçlarımı dördüncü bölüm olan araştırma bulgularında paylaşmış olacağım. Daha önceden bu alanda yapılan çalışmaların sonuçları ve kendi sonucumun karşılaştırılmaları beşinci bölümde paylaşılacaktır.

3.2 DONANIM

Belirtilen yazılım geliştirmesi ve test edilmesinde kullanılan bilgisayarın donanım özellikleri Tablo 2’de verilmiştir.

Tablo 3.1: Donanım İçeriği.

İşlemci	İntel® Core™ i7-9750H
Ekran Kartı	NDIVIA GeForce GTX 1660Gti 6 GB
RAM	24 GB

3.3 GELİŞTİRME ORTAMI

Google genellikle Google Colab olarak bilinen Colaboratory, 2017 yılında Google Research tarafından geliştirilmiştir. Makine öğrenimi ve derin öğrenme modellerinizi CPU'lar, GPU'lar ve TPU'lar üzerinde eğitmenize olanak tanıyan ücretsiz bir çevrimiçi bulut tabanlı Jupyter Notebook ortamıdır. Kendi verilerinizi Google Drive hesabınızdan (e-tablolar dahil), GitHub'dan ve diğer pek çok kaynaktan Colab not defterlerine aktarabilirsiniz. Colab ile bir resim veri kümesini içe aktarabilir, üzerinde bir resim sınıflandırıcıyı eğitebilir ve modeli değerlendirebilirsiniz.

Hem de sadece birkaç satır kodla [99].

Colab, makine öğrenimi topluluğunda yaygın olarak şu uygulamalarla kullanılır:

- a. TensorFlow'u kullanmaya başlama
- b. Nöral ağ geliştirme ve eğitme
- c. TPU'lar ile deneme yapma
- d. Yapay zekâ araştırmalarını yayma
- e. Eğitici oluşturma

3.3.1 Python

Python; web uygulamaları, yazılım geliştirme, veri bilimi ve makine öğreniminde (ML) yaygın olarak kullanılan bir programlama dilidir. Geliştiriciler, etkili ve öğrenmesi kolay olduğu ve birçok farklı platformda çalıştırılabildiği için Python'ı kullanır. Python yazılımı ücretsiz olarak indirilebilir, her türlü sistemle iyi bir entegrasyon sağlar ve geliştirme hızını artırır [100].

Python'ın avantajlarından bazıları şunlardır:

- a. Geliştiriciler, basit ve İngilizceye benzer bir söz dizimine sahip olduğundan Python programını kolayca okuyabilir ve anlayabilir.
- b. Geliştiriciler, diğer birçok dile kıyasla çok daha az kod satırıyla bir Python programı yazabildiğinden Python, geliştiricilerin daha üretken olmasını sağlar.
- c. Python, neredeyse her görev için yeniden kullanılabilir kodları içeren geniş bir standart kitaplığa sahiptir. Dolayısıyla geliştiricilerin kodu sıfırdan yazmasına gerek yoktur.
- d. Geliştiriciler Python'ı Java, C ve C++ gibi diğer popüler programlama dilleriyle kolayca kullanabilir.
- e. Dünya genelinde aktif Python topluluğunda, destek sunmaya hazır milyonlarca geliştirici bulunmaktadır. Bir sorunla karşılaşırsanız bu topluluktan hızlıca destek alabilirsiniz.

- f. Python'ı öğrenmek isterseniz internette birçok yararlı kaynak bulunmaktadır. Örneğin, kolayca videolar, dersler, belgeler ve geliştirici kılavuzları bulabilirsiniz.
- g. Python; Windows, macOS, Linux ve Unix gibi farklı bilgisayar işletim sistemleri arasında taşınabilir.

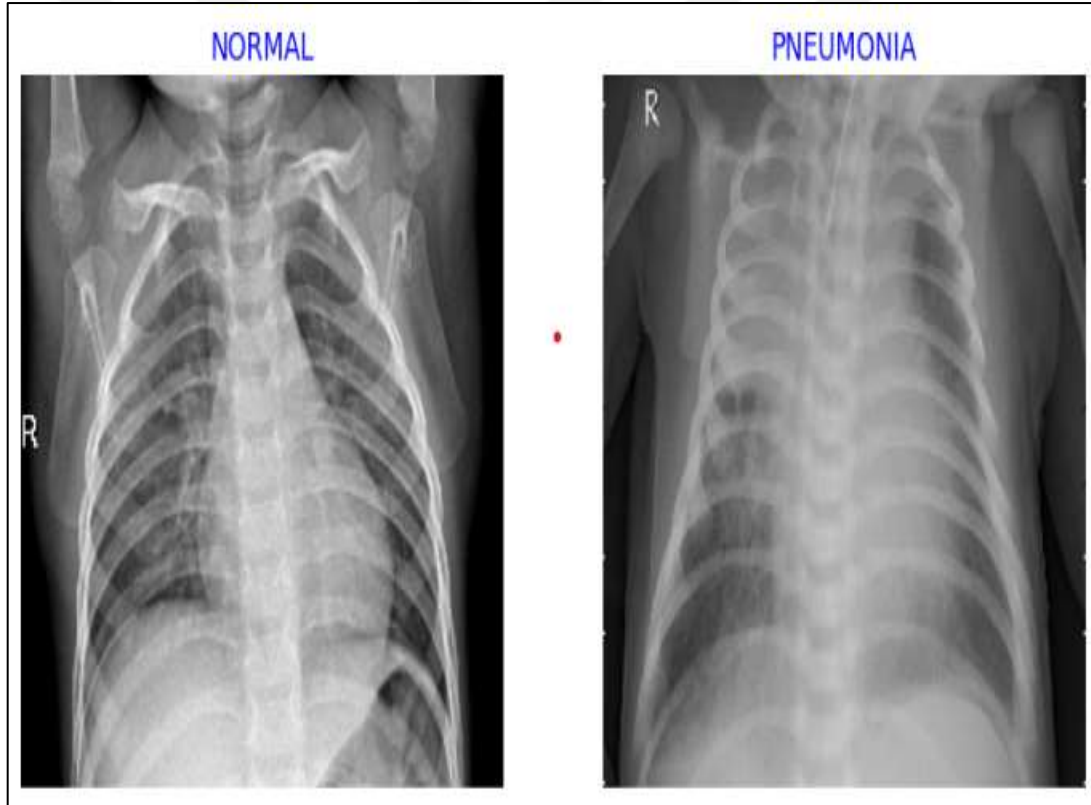
3.1.2 Kullanılan Kütüphaneler

- a. `opendatasets (od)`: Veri setlerini indirmek için kullanılır.
- b. `pathlib`: Dosya yollarıyla nesne tabanlı şekilde çalışmak için kullanılır.
- c. `os`: İşletim sistemiyle etkileşim kurmak için kullanılır (örneğin, bir klasördeki dosyaları listeleme).
- d. `PIL (Pillow)`: Görüntü (image) işleme için kullanılır.
- e. `itertools`: Verimli döngüler (loop) için iterator'lar oluşturmak amacıyla kullanılır.
- f. `cv2 (OpenCV)`: Görüntü işleme gibi bilgisayarla görme (computer vision) görevleri için kullanılır.
- g. `pandas (pd)`: `DataFrame` yapısıyla veri analizi ve düzenleme işlemleri için kullanılır.
- h. `numpy (np)`: Dizi (array) yapılarıyla sayısal (numerical) hesaplamalar yapmak için kullanılır.
- i. `matplotlib.pyplot (plt)`: Grafik ve çizelge (plot/chart) oluşturmak için görselleştirme kütüphanesi.
- j. `seaborn (sns)`: `Matplotlib` üzerine kurulmuş, istatistiksel veri görselleştirme için kullanılır.
- k. `sklearn`: Model seçimi, değerlendirme gibi makine öğrenmesi araçlarını içerir.
- l. `tensorflow (tf)`: Derin öğrenme (deep learning) uygulamaları için kullanılan bir framework'tür.
- m. `warnings`: Uyarı mesajlarını yönetmek için kullanılır.

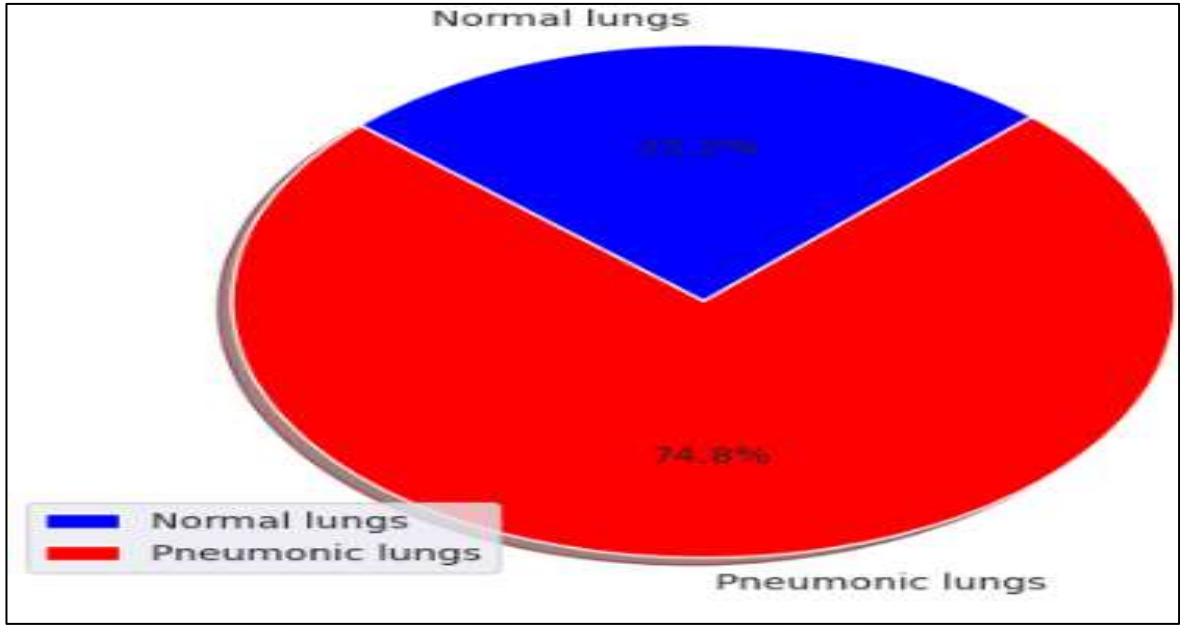
3.3 KULLANILAN VERİ SETİ

Bu çalışmada, iki farklı etiketlenmiş veri seti kullanılarak analiz gerçekleştirilmiştir.

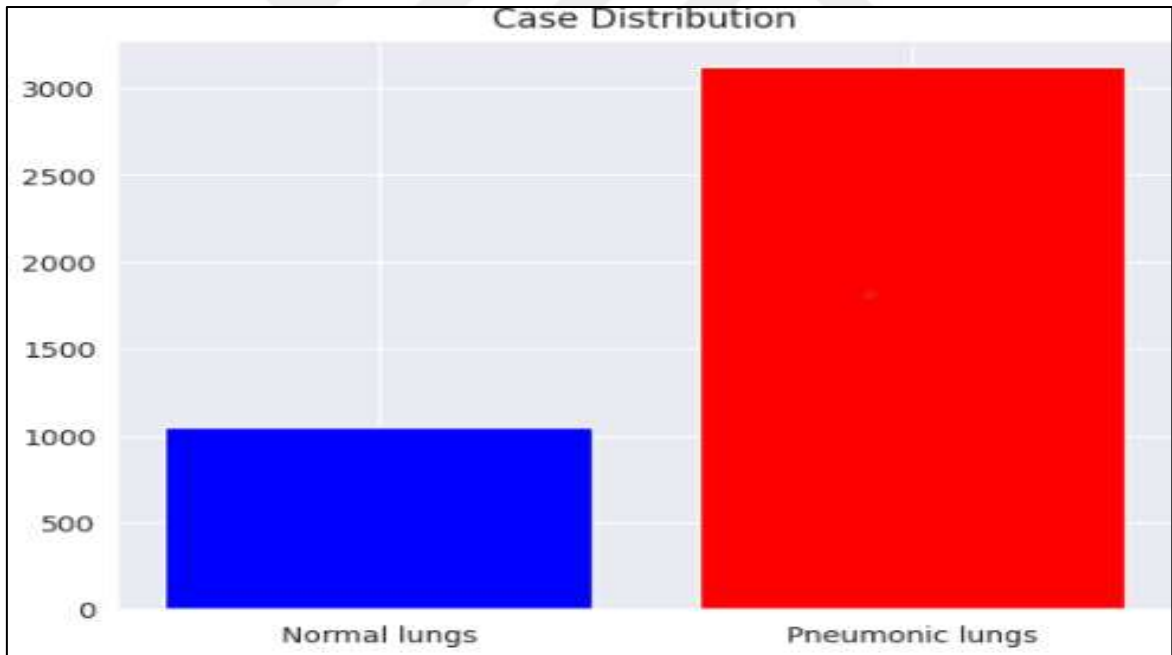
Birinci veri seti, Kermany, Zhang ve Goldbaum (2018) [101] tarafından oluşturulmuş olan ve Mooney (2017) [102] aracılığıyla paylaşılan göğüs röntgeni görüntülerinden oluşmaktadır. Bu veri seti; eğitim, doğrulama ve test olmak üzere üç ayrı klasöre ayrılmıştır. Görüntüler iki sınıfa ayrılmıştır: ‘normal’ ve ‘zatürre’. Sınıflar incelendiğinde, ‘normal’ sınıfında 1583, ‘zatürre’ sınıfında ise 4273 adet görüntü bulunduğu görülmektedir. Bu dağılım, sınıflar arasında ciddi bir dengesizlik olduğunu göstermektedir. Bu tür dengesizlikler, geliştirilecek yapay zekâ modelinin belirli bir sınıfa yönelik aşırı öğrenme (overfitting) eğilimi göstermesine yol açabilir.



Şekil 3.1: Birinci Veri Seti İçerisinden Göğüs Röntgeni Örnekleri.



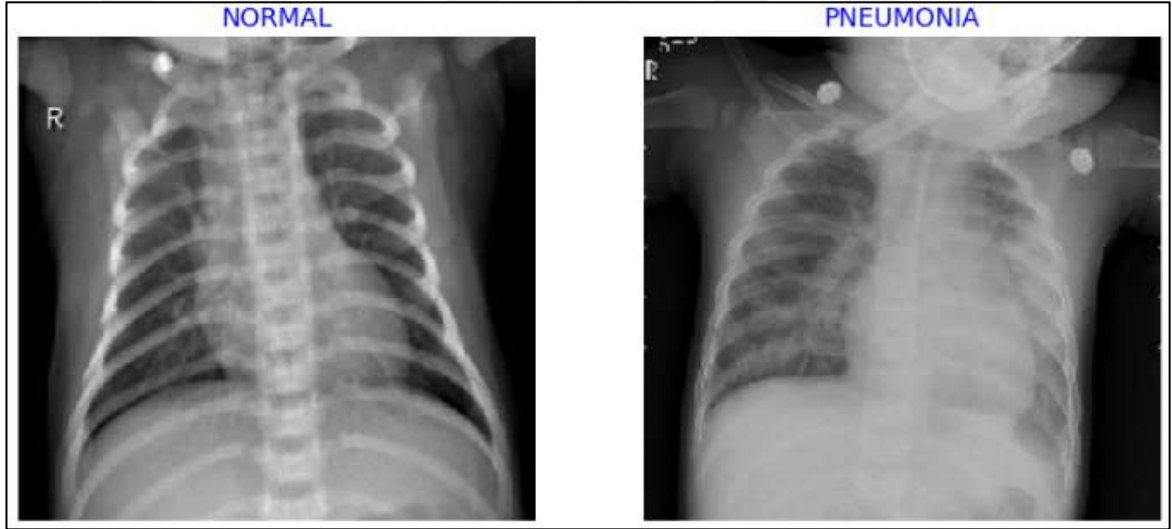
Şekil 3.2: Birinci Veri Seti Veri Dağılımı Oranı.



Şekil 3.3: Birinci Veri Seti Veri Dağılımı.

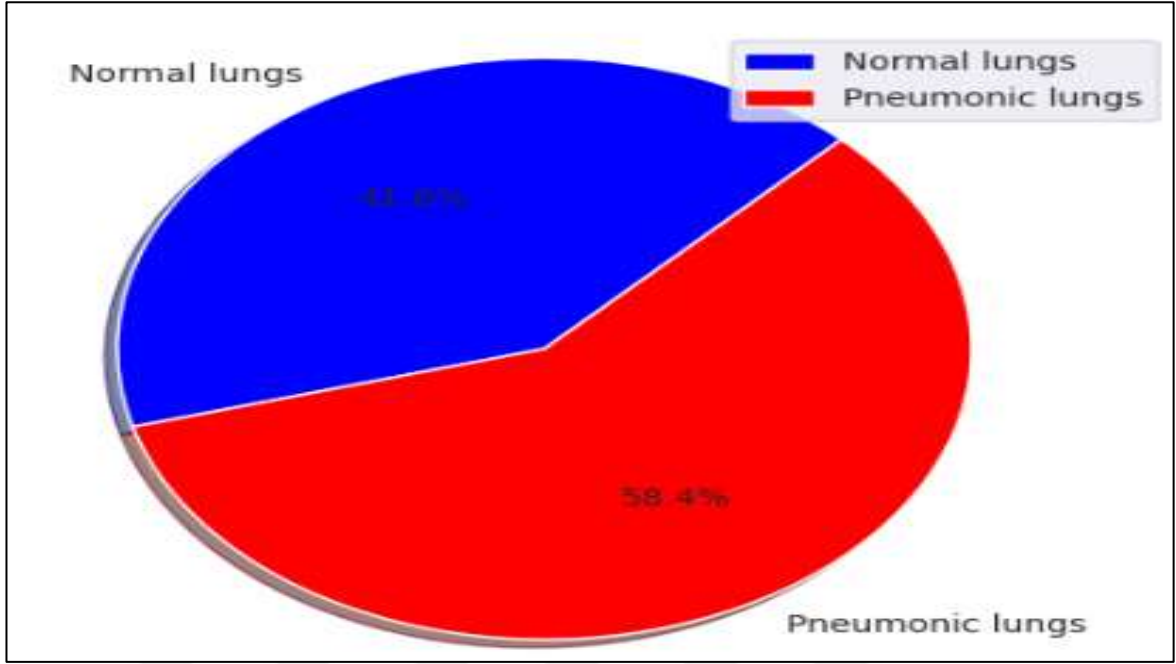
3.4 İKİNCİ VERİ SETİ

Sait, Unais; Lal KV, Gokul; Prakash Prajapati, Sunny; Bhaumik, Rahul; Kumar, Tarun; Shivakumar, Sanjana ve Bhalla, Kriti (2021) [103] tarafından sunulan veri kümesidir. Montalbo (2022) tarafından kaggle üzerinden düzenlenmiş hali paylaşılmıştır [104]. Bu veri setinin yapısı da ilk veri setine benzer şekilde eğitim, doğrulama ve test olmak üzere üç klasöre ayrılmıştır. Söz konusu veri seti, referanslar bölümünde belirtilen 15 farklı açık erişimli veri setinin bir araya getirilmesiyle oluşturulmuş, göğüs röntgeni görüntülerinden oluşan bütünleşmiş ve düzenlenmiş bir yapıya sahiptir. Bu birleşik veri seti, 1281 COVID-19, 3270 normal, 1656 viral pnömoni ve 3001 bakteriyel pnömoni röntgen görüntüsünü içermektedir. Bu görüntüler, ortak bir havuz oluşturacak şekilde birleştirilmiştir. Nihai birleşik havuzda toplamda 4558 COVID-19, 5403 normal, 4497 viral pnömoni ve 5768

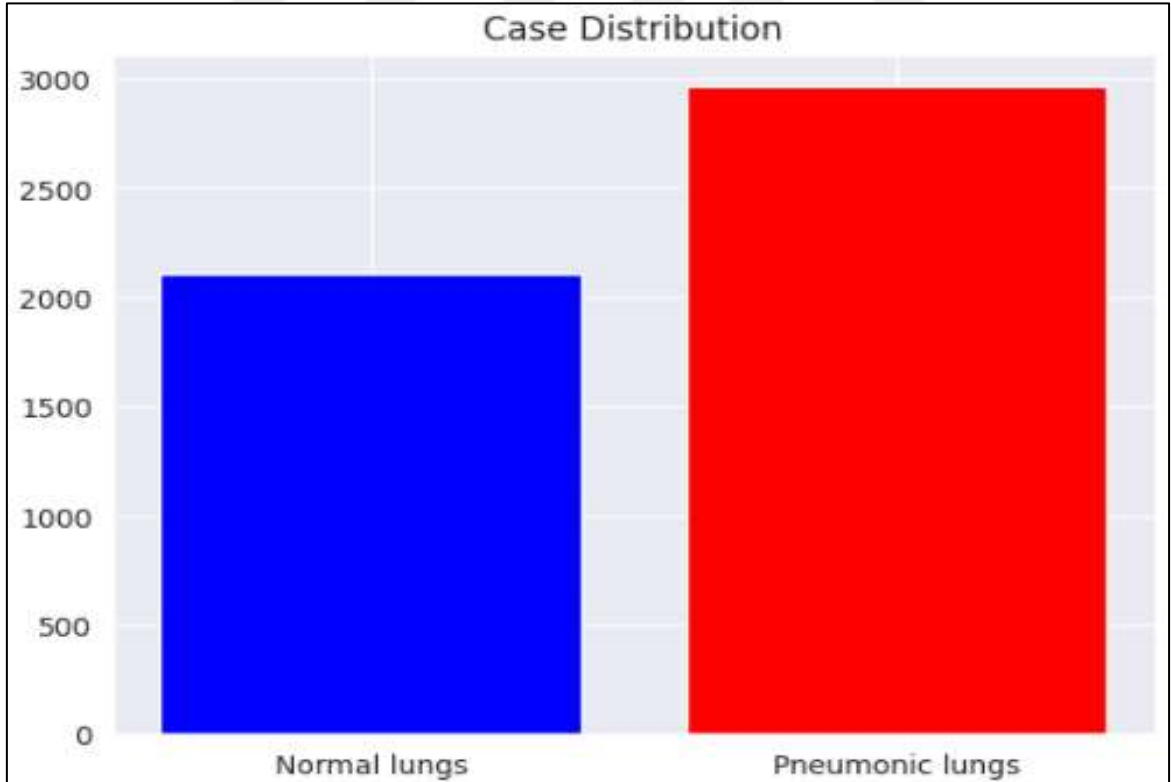


Şekil 3.4: İkinci Veri Setinden Göğüs Röntgeni Örnekleri.

bakteriyel pnömoni röntgen görüntüsü bulunmaktadır. Görüntü benzerlikleri temel alınarak yapılan analiz sonucunda, 1379 COVID-19, 1476 normal, 2690 viral pnömoni ve 2588 bakteriyel pnömoni görüntüsünün mükerrer olduğu belirlenmiş ve bu tekrar eden veriler veri setinden çıkarılmıştır. Son düzenlenmiş haliyle veri seti, Montalbo tarafından Kaggle platformu üzerinden paylaşılmıştır.



Şekil 3.5: İkinci Veri Seti Veri Dağılımı Oranı.



Şekil 3.6: İkinci Veri Seti Veri Dağılımı.

3.5 DERİN ÖĞRENME YÖNTEMLERİ VE KULLANILAN YÖNTEM

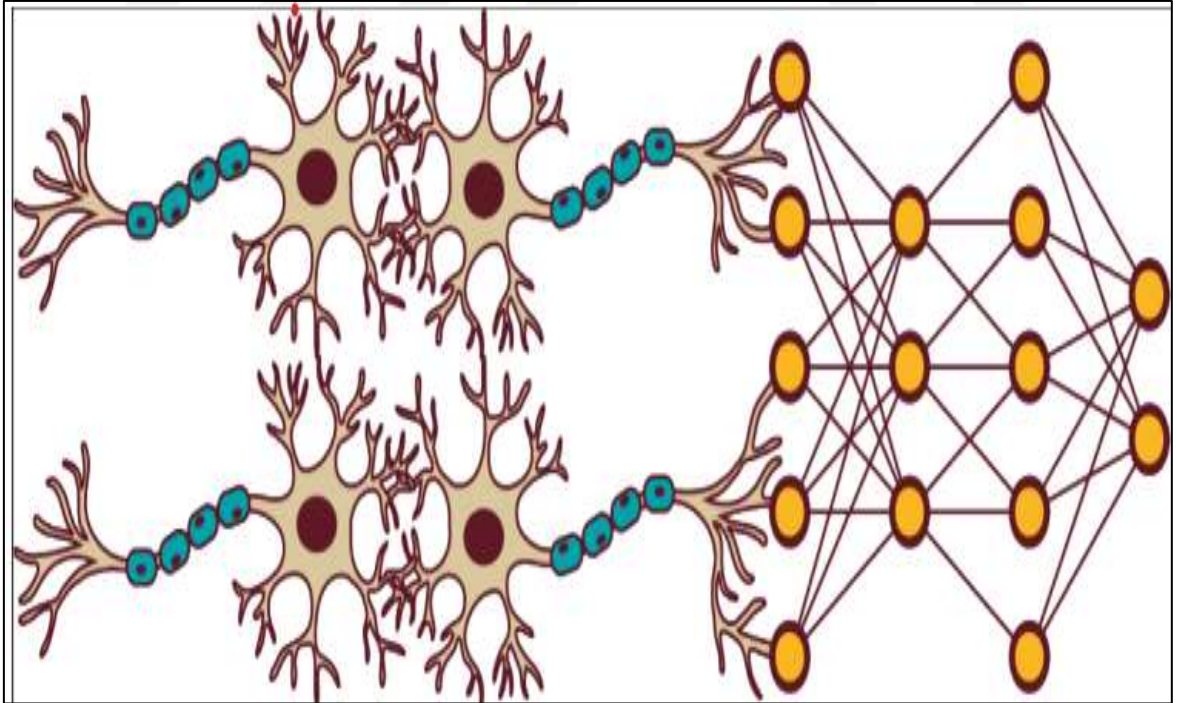
3.5.1 Derin Öğrenme Yöntemleri

Bu kısımda derin öğrenme modelleri detaylı olarak anlatılmıştır.

3.5.1.1 Yapay sinir ağları

Yapay sinir ağları, insan beyninde bulunan biyolojik sinir ağlarının işleyişini matematiksel biçimde modellemeyi amaçlayan, algoritmalar aracılığıyla geliştirilen yazılımsal sistemlerdir. Bu sistemler, daha önce öğrenilmiş ya da sınıflandırılmış veri kümelerinden faydalanarak yeni bilgiler üretebilir, kararlar alabilir ve verileri sınıflandırabilir. Yapay sinir ağları, beynin öğrenme, hafıza, genelleme ve karar verme yetilerini taklit etmeye çalışır. Bu ağlarda öğrenme süreci, örnekler ve veri setleri aracılığıyla gerçekleşir [105].

İlk yapay sinir ağı modeli, 1943 yılında nörolog Warren McCulloch ile matematikçi Walter Pitts tarafından kaleme alınan “Sinirsel Aktivite İçindeki Fikirlerin Mantıksal Hesabı” (A Logical Calculus of Ideas Immanent in Nervous Activity) adlı makale ile literatüre kazandırılmıştır [106].



Şekil 3.7: Biyolojik ve Yapay Sinir Ağı.

Biyolojik Sinir Ağı

İnsandaki biyolojik sinir sistemi, bilgi işleme yeteneğine sahip olan nöron isimli sinir hücrelerinin bir araya gelmesiyle oluşur. Bu nöronlar, birbirleriyle bağlantı kurarak çeşitli nörolojik işlevleri gerçekleştirir. Biyolojik sinir ağı, insanın davranışlarını yönlendirmesini ve çevresini anlamasını sağlayan temel yapıdır. Beş duyu organından gelen veriler sayesinde, bu ağ olaylar arasındaki ilişkileri algılayıp çözümleyebilir. Beynin farklı bölümleri farklı görevleri yerine getirir. Duyulardan gelen bilgiler sinir sistemi aracılığıyla beyne aktarılır ve beyinde oluşan kararlar tekrar sinir sistemi vasıtasıyla vücudun ilgili bölgelerine sinyallerle iletilir [107].

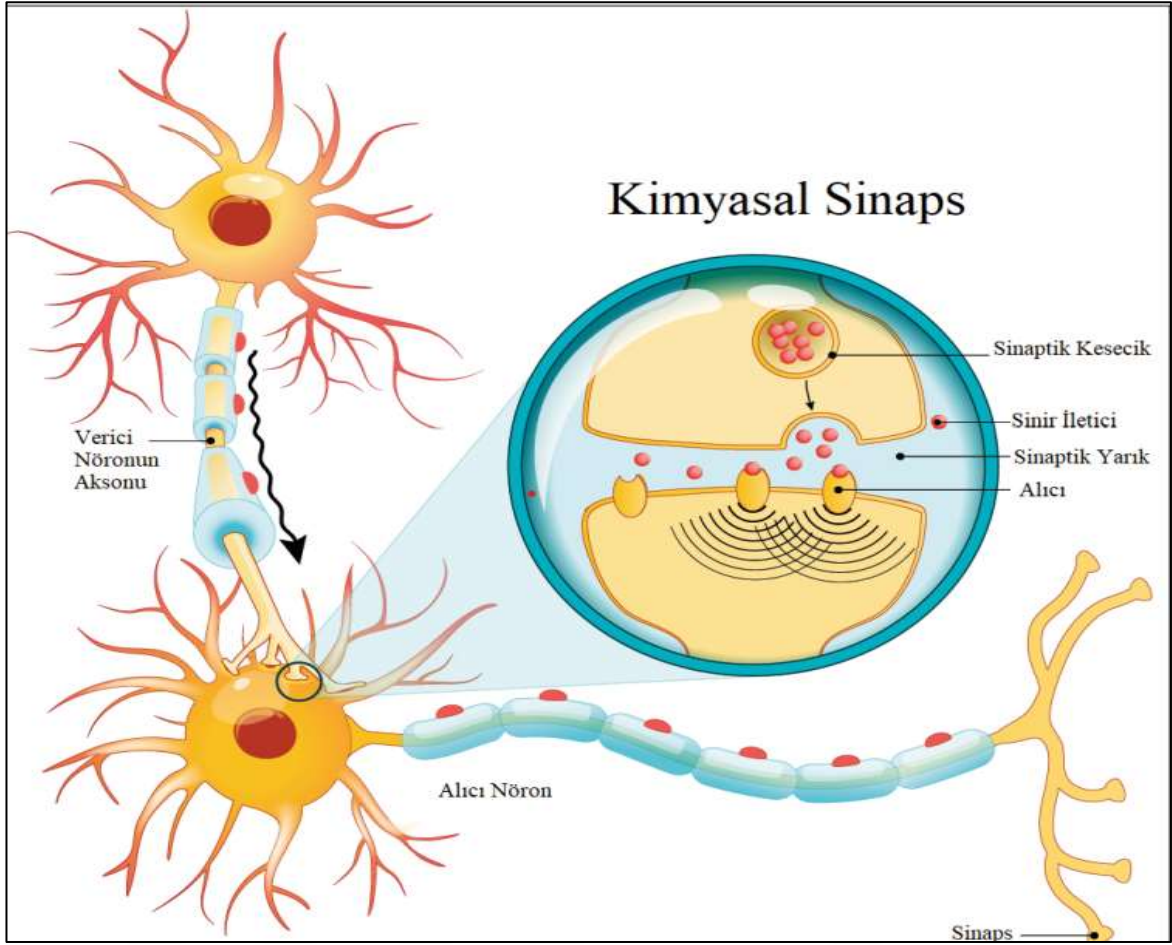
Nöronlar işlevlerine göre üç ana gruba ayrılır:

- a. Duyusal Nöronlar: Koku, tat, dokunma, görme ve işitme gibi dış uyarılardan gelen verileri beyne taşır.
- b. Motor Nöronlar: Kasların kasılmasını sağlayarak hareketi gerçekleştirir.
- c. Ara Nöronlar: Duyusal ve motor nöronlar ile merkezî sinir sistemi arasında bağlantı kurarak sinirsel devreleri oluşturur.

Nöronlar arasındaki sinyal iletimi, sinaps adı verilen yapılar üzerinden ve neurotransmitter denilen kimyasallar aracılığıyla gerçekleşir. Eğer alınan kimyasal uyarı, hücre içindeki elektriksel potansiyelin eşik seviyesini aşarsa, aksona bir sinyal iletilir. Bu sürece hücrenin uyarılması denir. Sinir hücresi, sinapslardan dentrite ulaşan sinyalleri alır ve bunlara karşılık aksiyon potansiyeli üretip üretmemeye karar verir. İletim, akson boyunca gerçekleşir. Girişteki duyu verileri çekirdekte ağırlıklandırılarak çıkış terminallerine ulaşır ve başka bir nörona iletilir. Bu biçimde nöronlar arasında iletişim sağlanır [108].

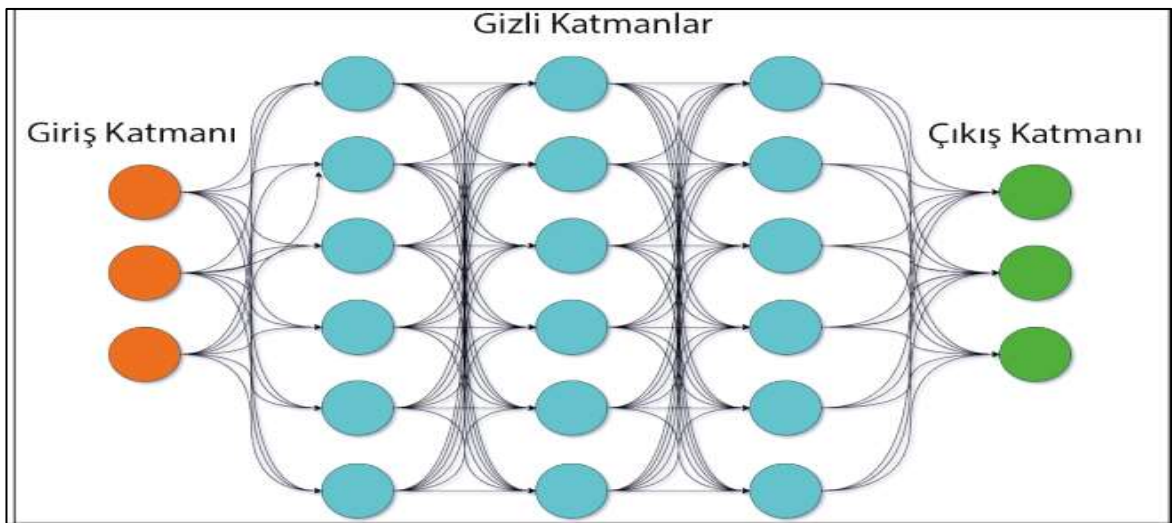
İnsan beyinde öğrenme üç şekilde gerçekleşir:

- a. Yeni akson oluşumu ile
- b. Aksonların daha sık yarılmasıyla
- c. Mevcut aksonların iletim gücünün değiştirilmesiyle



Şekil 3.8: Nöronlar Arası İletişim.

Yapay Sinir Ağının Yapısı



Şekil 3.9: Yapay Sinir Ağı Modeli.

Yapay sinir ağıları, yapay sinir hücrelerinin birbirine bağlanmasıyla oluşan sistemlerdir. Bu ağlar temel olarak üç ana katmandan meydana gelir:

- a. Giriş Katmanı
- b. Ara (Gizli) Katmanlar
- c. Çıkış Katmanı

Bilgiler ilk olarak giriş katmanından ağı iletilir. Bu bilgiler ara katmanlarda işlenerek, yani belirli ağırlık değerleri üzerinden değerlendirilerek, çıkış katmanına aktarılır. Ağı, verilen girdilere karşılık doğru çıktıları üretebilmesi için bu ağırlıkların uygun şekilde ayarlanmış olması gereklidir [107].

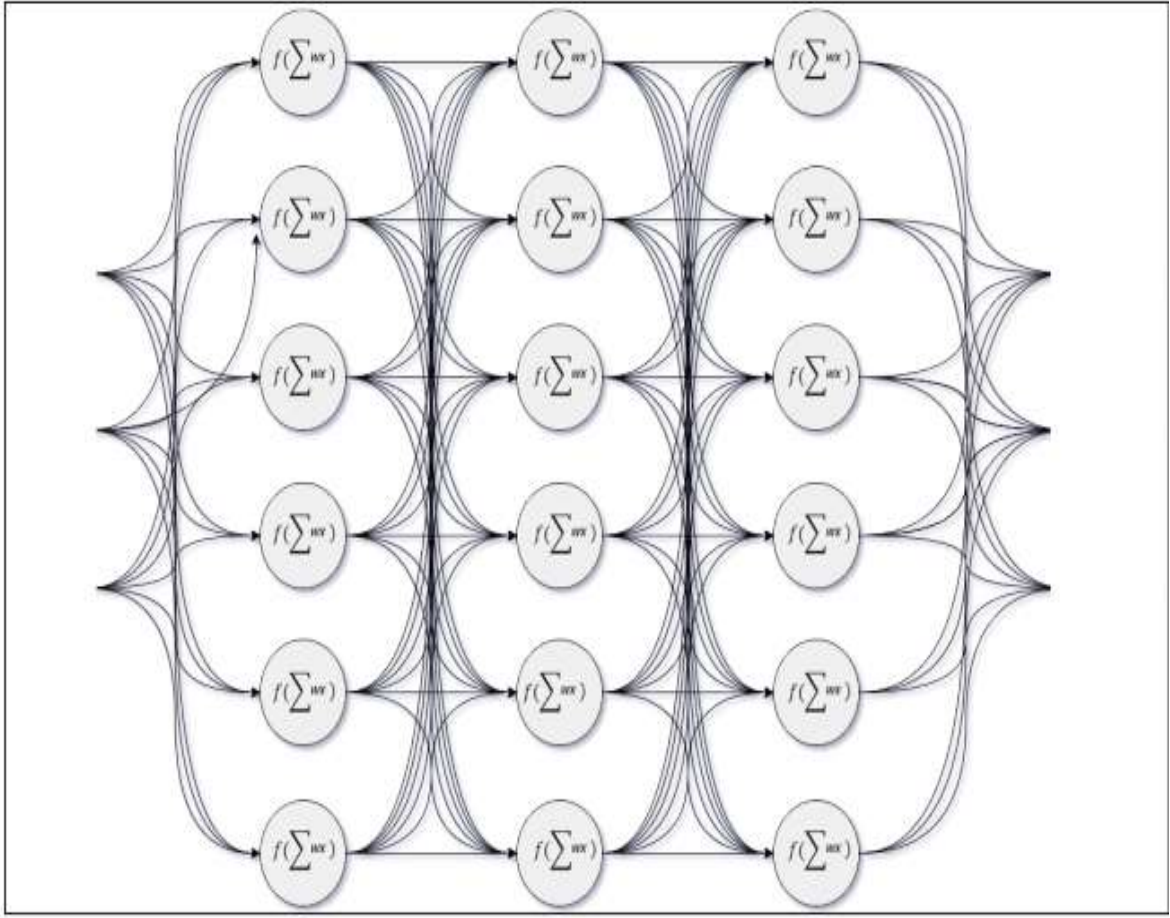
Katmanlar arasında gerçekleşen bilgi akışı şu şekilde özetlenebilir:

- a. Girdi olarak, belirli bir veri seti sisteme sunulur.
- b. Yapay sinir ağı bu verileri giriş katmanından alır ve başlangıçta belirlenen ağırlıklar doğrultusunda bir çıktı üretir. Çıktının doğruluğu analiz edilerek ağırlıklar tekrar düzenlenir. Bu işlem, doğru çıktılar elde edilinceye kadar devam eder. Ara (gizli) katmandaki nöronların çıktıları, ağırlıklarla belirlenmiş olur.
- c. Kabul edilebilir doğruluk seviyesine ulaşıldığında eğitim tamamlanır ve son çıktılar alınır.

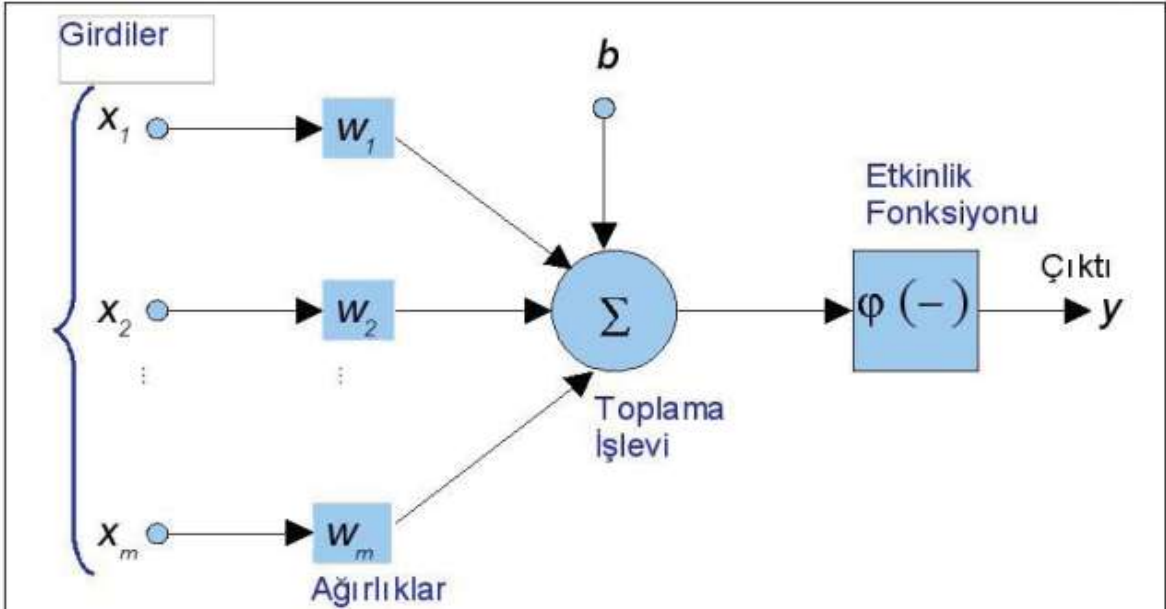
Yapay sinir ağlarında bu ağırlıkların ayarlanması ve nihai çıktının oluşturulması matematiksel bir model aracılığıyla gerçekleştirilir [107].

Yapay Sinir Ağıının Aritmetik Modeli

İnsandaki bir sinir hücresinin matematiksel karşılığı aşağıda gösterildiği şekilde modellenebilir. Bu bağlamda, yapay sinir ağının temel işlevi, giriş olarak verilen veri setine karşılık belirtilen biçimde bir çıktı üretebilmektir. Bu işlemin gerçekleştirilebilmesi için, ağ öncelikle belirli eğitim veri setleri kullanılarak eğitilmelidir. Eğitim süreci tamamlandıktan sonra, yapay sinir ağı genelleme yapabilir hale gelir ve karar verebilir seviyeye ulaşır. Son aşamada ise istenilen çıktılar elde edilir [108].



Şekil 3.10: Yapay Sinir Ağının Aritmetik Modeli.



Şekil 3.11: Her Matematik Fonksiyonun Çalıştığı Nöron Bu Görselde Gösterildiği Gibi Çalışır.

Ara katman yollarına karşılık gelen dentritler boyunca belirli ağırlıklar bulunur ve bu dentritlere, başka bir nörondan gelen giriş değerleri ($x_1, x_2, x_3, \dots, x_m$) iletilir. Hesaplama süreci aşağıdaki adımlarla gerçekleşir [108]:

- a. Giriş değeri ile dentritteki ağırlık (w_1) çarpılır ve sonuç ($w_1 \times x_1$) sinir hücresine gönderilir.
- b. Tüm dentritlerden gelen giriş ve ağırlık çarpımları toplanarak ağırlıklı toplam elde edilir.
- c. Bu toplam değere bir bias (b) eklenir ve ardından aktivasyon fonksiyonundan geçirilerek çıkışa aktarılır.
- d. Elde edilen bu çıkış, nihai bir çıktı olabileceği gibi başka bir sinir hücresinin girdisi de olabilir.
- e. Her bir nöron bu yöntemle hesaplanır ve hücreler birbirine seri ya da paralel biçimde bağlanabilir.
- f. Matematiksel olarak, girişler ağırlıklarla çarpılır, bias değeri eklenir ve böylece temel bir model oluşturulur. Yapay sinir ağlarında en önemli işlem, modelin en uygun sonuçları vereceği ağırlık (w) ve bias (b) değerlerinin hesaplanmasıdır.

Yapay Sinir Hücresi Bölümleri

Bir yapay sinir hücresi dört ana bileşenden meydana gelir [108]:

- a. Girdiler: Nöronlara ulaşan verileri temsil eder. Bu veriler, biyolojik sinir hücrelerinde olduğu gibi çekirdekte işlemek üzere nörona iletilir.
- b. Ağırlıklar: Girdi verileri, sinir hücresinin çekirdeğine aktarılmadan önce, geldikleri bağlantıların belirli ağırlıklarıyla çarpılır. Böylece her bir girdinin oluşturulacak çıktıya katkısı ayarlanabilir hale gelir.
- c. Toplama Fonksiyonu: Yapay nörona ağırlıklarla birlikte ulaşan tüm girdileri toplar ve bu toplam, hücrenin net girdisini belirler.
- d. Aktivasyon Fonksiyonu: Bir önceki katmandan gelen ağırlıklı toplam veriyi alarak, doğrusal olmayan bir çıktı üretir ve bu sonucu bir sonraki katmana aktarır.

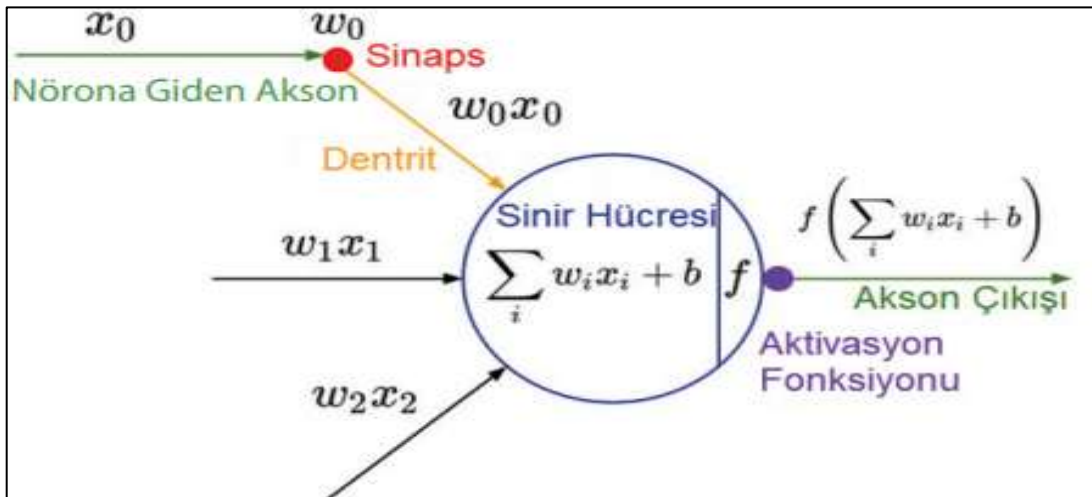
- e. Aktivasyon fonksiyonları, karmaşık verilerin öğrenilebilmesini sağlamak için önemlidir. Bu fonksiyonun temel amacı, ağırlık (w) ve bias (b) değerlerini ayarlamaktır. $L = x \cdot w + b$ formülünden sonra, sonuç bir aktivasyon fonksiyonundan geçirilir. Bu işlevler tensörleri etkiler ve bir nevi nöronun tetiklenmesini sağlar.

Aktivasyon fonksiyonunun gerekliliği nedir?

Yapay sinir ağlarına gerçek hayattaki doğrusal olmayan örüntülerin kazandırılabilmesi için aktivasyon fonksiyonlarına ihtiyaç duyulur. En basit haliyle, sinir ağına girilen değerler (x) ve bu değerlere ait ağırlıklar (w) ile işlem yapılır ve $f(x)$ şeklinde bir aktivasyon fonksiyonu uygulanır. Bu sonuç ya ağın çıktısı olur ya da bir sonraki katmanın girdisi olarak kullanılır [108].

Eğer aktivasyon fonksiyonu kullanılmazsa, çıkış yalnızca doğrusal bir fonksiyon olur. Bu durumda sinir ağı karmaşık örüntüleri öğrenemez. Ancak temel hedef, ağın doğrusal olmayan ilişkileri de kavrayabilmesidir. Bu sebeple, sinir ağına görüntü, ses, metin veya video gibi kompleks bilgiler verilir. Çok katmanlı derin yapay sinir ağları, bu sayede verilerden anlamlı örüntüler çıkarabilir.

Modelin öğrenmesi sırasında ağırlıklarla ilgili hataları belirlemek için geriye yayılım algoritması kullanılır. Giriş verisi ile ağırlıklar çarpılır, bias eklenir ve ardından aktivasyon fonksiyonu uygulanarak çıktı elde edilir [107].

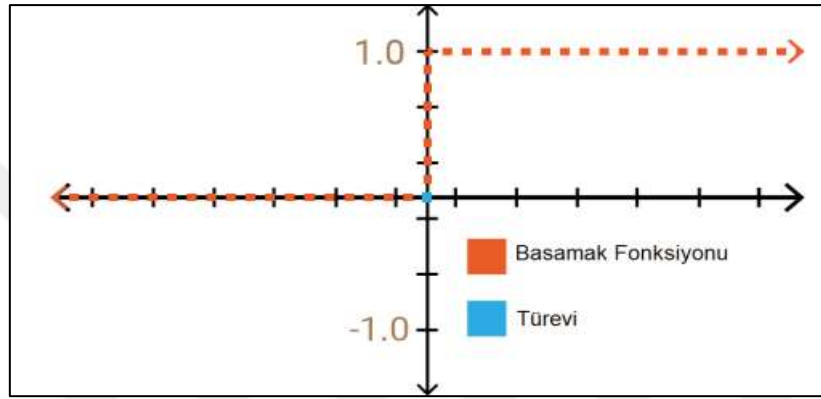


Şekil 3.10: Yapay Sinir Hücresi.

Bu kısımdan sonrasında yapay sinir ağlarında kullanılan aktivasyon fonksiyonlarına yer verilmiştir.

a. Basamak (Step) Fonksiyonu:

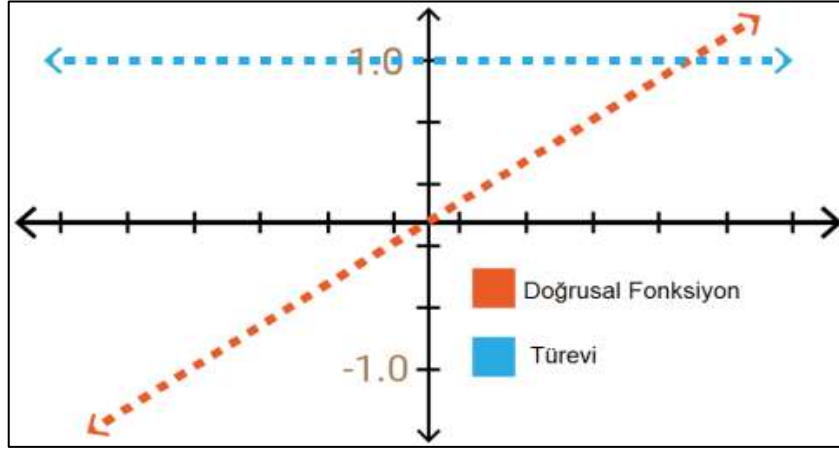
Çıkışı yalnızca iki farklı değer alabilen bir fonksiyondur ve bu özelliği sayesinde ikili sınıflama problemlerinde, özellikle de çıkış katmanlarında sıkça tercih edilir. Ancak, türevi sabit olmadığı ve öğrenme sürecine katkı sunmadığı için gizli katmanlarda kullanımı önerilmez [109].



Şekil 3.11: Basamak (Step) Fonksiyonu.

b. Doğrusal (Linear) Fonksiyon:

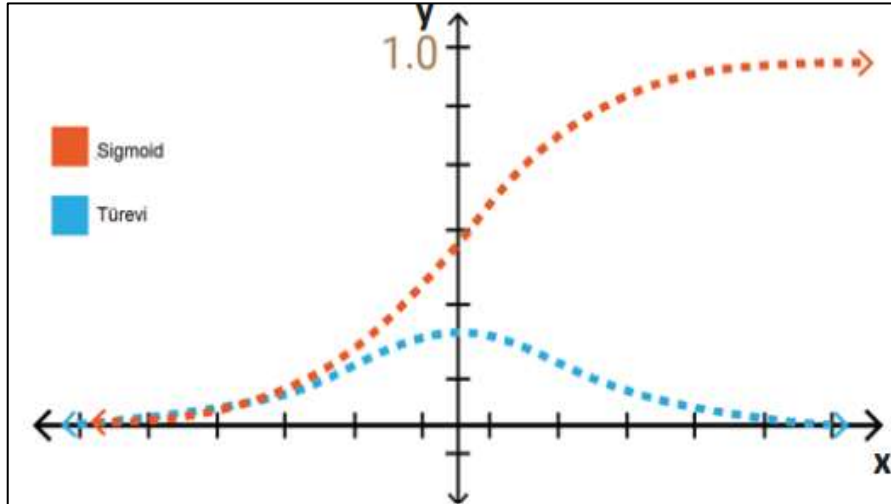
Çıkış olarak ikili olmayan sürekli değerler üreten bir aktivasyon fonksiyonudur. Bu yapı, sinir hücrelerinin birbirine bağlanmasına imkân tanır. Ancak türevinin sabit bir değere sahip olması önemli bir dezavantajdır; çünkü bu durumda model, öğrenme yapamaz. Ayrıca, tüm katmanlarda doğrusal fonksiyon kullanılırsa, sistem sadece giriş ve çıkış arasında sabit bir dönüşüm gerçekleştirir ve ara katmanlar devre dışı kalır. Bu durum derin öğrenme sistemlerinde istenmeyen bir sonuçtur [109].



Şekil 3.12: Doğrusal (Linear) Fonksiyon.

c. Sigmoid Fonksiyonu:

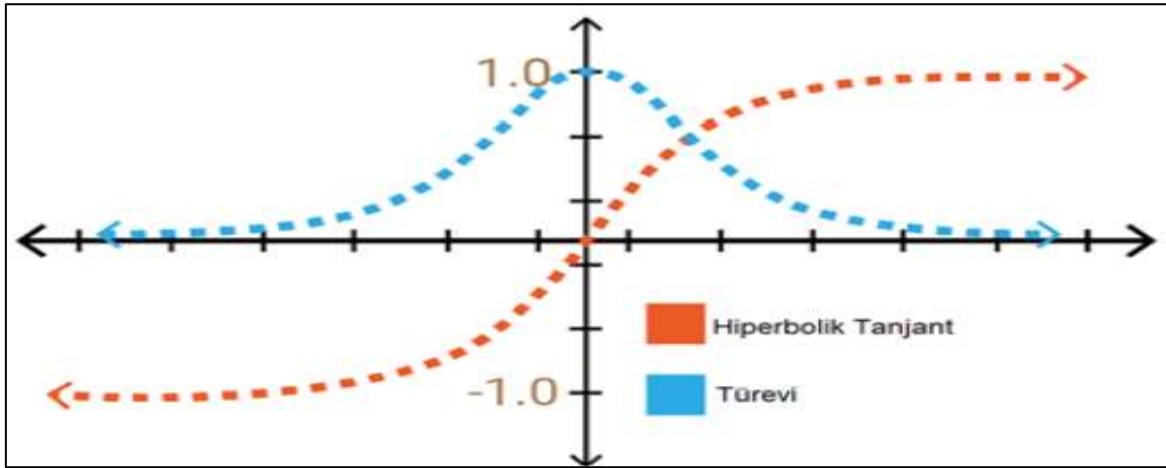
Geçmişte sıkça kullanılan bu fonksiyon, giriş değerlerini 0 ile 1 arasında sınırlar ve doğrusal değildir. Büyük girişlerde çıkışı 1'e, küçük girişlerde ise 0'a yaklaştırır. Böylece uç değerler aşılamaz. Türevi alınabilir bir yapıda olduğundan öğrenme işlemi desteklenebilir. Özellikle giriş değeri x , -2 ile +2 aralığındaysa, çıkış y değerleri çok daha hızlı değişir. Bu özellik, fonksiyonu etkili bir sınıflayıcı yapar. Ayrıca giriş değerleri ne kadar uç olursa olsun, çıktı her zaman 0 ile 1 arasında kalır; bu da modelin kararlılığını artırır [109].



Şekil 3.13: Sigmoid Fonksiyonu.

d. tanh (Hiperbolik Tanjant) Fonksiyonu:

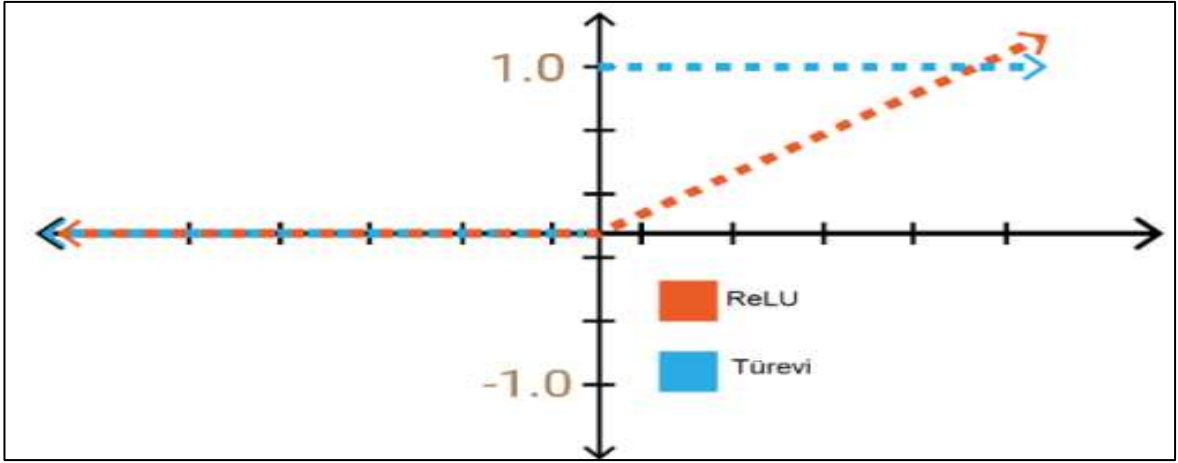
Tanh fonksiyonu da sigmoid'e benzer şekilde sıkıştırma işlemi yapar; ancak çıktıları $[-1, 1]$ aralığında kalır. Bu yönüyle 0 merkezli bir yapıya sahiptir. Sigmoid'e göre daha yüksek eğimli türev değerleri taşıdığından, daha hızlı öğrenme ve daha hassas sınıflandırma sağlayabilir. Bu geniş aralık, modelin daha fazla bilgi taşımasına olanak verir. Ancak yine de uç noktalarda gradyanların sıfıra yaklaşması (gradyan sönmesi) problemi mevcuttur [109].



Şekil 3.14: tanh (Hiperbolik Tanjant) Fonksiyonu.

e. ReLu Fonksiyonu:

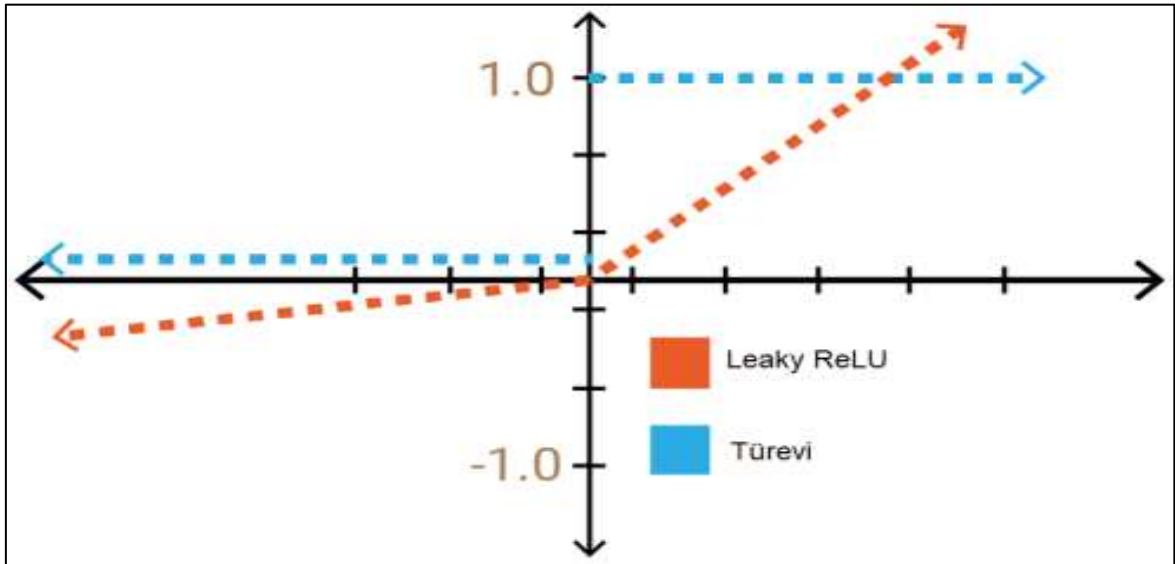
ReLU, günümüzde en sık tercih edilen aktivasyon fonksiyonlarından biridir. Bu fonksiyonun en dikkat çekici özelliği, çalışma mantığının biyolojik sinir hücrelerine benzemesidir. Giriş değeri negatif olduğunda, çıktı sıfır olur. Pozitif bir değerle karşılaştığında ise herhangi bir dönüşüm yapılmaz, değer doğrudan aktarılır. Tanh ve sigmoid gibi fonksiyonlar daha karmaşık hesaplamalar gerektirirken, ReLU yalnızca işaret kontrolü yapar. Bu da hesaplama süresini ciddi ölçüde azaltır. ReLU, güçlü bir tahmin kabiliyetine sahiptir, ancak nöron ölümü gibi dezavantajlar da barındırır. Farklı ReLU türlerinin bir araya getirilmesiyle, başka fonksiyonlara da yakınsama mümkündür; bu sebeple çok katmanlı ağ yapılarında etkili şekilde kullanılabilir [106].



Şekil 3.15: ReLu Fonksiyonu.

f. Sızıntı (Leaky ReLu) Fonksiyonu:

Leaky ReLU, "ölü nöron" problemine çözüm olarak geliştirilmiştir. Negatif girişlerde sıfır yerine çok küçük negatif bir değer döndürerek nöronların tamamen devre dışı kalmasını engeller. Bu özelliği sayesinde öğrenmeyi sürdürebilir. Diğer fonksiyonlara göre hesaplama süresi artsa da genel anlamda oldukça hızlı çalışan bir fonksiyondur [109].



Şekil 3.16: Sızıntı (Leaky ReLu) Fonksiyonu.

g. Softmax Fonksiyonu:

Softmax, sigmoid fonksiyonuna oldukça benzer bir yapı sergiler. Özellikle çok sınıflı problemler için, derin öğrenme modellerinin çıkış katmanlarında yaygın biçimde kullanılır. 0 ile 1 arasında değerler üretir ve her sınıfa ait olasılığı yansıtarak, hangi sınıfa daha çok ait olduğunu belirler. Bu da çıktının olasılıksal yorumlanmasına olanak tanır [109].

Aktivasyon fonksiyonlarının seçiminde temel soru, hangi fonksiyonun belirli bir problem için daha uygun olduğudur. Geleneksel olarak sıkça tercih edilen sigmoid fonksiyonu, günümüzde yerini ReLU (Rectified Linear Unit) fonksiyonuna bırakmıştır. Özellikle çok sınıflı sınıflandırma problemlerinde, aktivasyon çıktısının daha geniş bir aralıkta yayılabilmesi sebebiyle hiperbolik tanjant (tanh) fonksiyonu kullanılabilir. Öte yandan, modelin öğrenme sürecinin görece yavaş ilerlemesinde bir sakınca yoksa, sigmoid fonksiyonu da hala geçerli bir seçenek olarak değerlendirilebilir. Daha derin yapay sinir ağı yapılarında, işlemsel verimlilik göz önünde bulundurularak genellikle ReLU fonksiyonu tercih edilmektedir. Ancak ReLU'nun gradyanların sıfırlanması (vanishing gradient) gibi bir sorunu bulunduğundan, bu duruma çözüm olarak Leaky ReLU fonksiyonunun kullanılması önerilmektedir [110].

Aktivasyon fonksiyonu seçiminde dikkate alınması gereken ölçütler şu şekilde özetlenebilir:

- a. Modelin hızlı ve kararlı biçimde yakınsama sağlaması,
- b. ReLU'nun yüksek işlem hızı sağlaması, ancak gradyan kaybı problemini göz önünde bulundurmak gerekliliği,
- c. Gradyan sıfırlanması sorununun azaltılması için Leaky ReLU gibi alternatiflerin değerlendirilmesi,
- d. Derin öğrenme modelleri için başlangıç noktası olarak ReLU'nun uygunluğu,
- e. Çıkış katmanlarında genellikle Softmax fonksiyonunun tercih edilmesi.

Sonuç olarak, aktivasyon fonksiyonunun seçimi, geliştirilecek modele özgü deneysel süreçlerle belirlenmeli ve bu süreçte farklı fonksiyonların performansları karşılaştırılmalıdır.

AKTİVASYON FONKSİYON	DENKLEM	ARALIK
Doğrusal Fonksiyon	$f(x) = x$	$(-\infty, \infty)$
Basamak Fonksiyonu	$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ 1 & \text{ için } x \geq 0 \end{cases}$	$\{0, 1\}$
Sigmoid Fonksiyon	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$(0, 1)$
Hiperbolik Tanjant Fonksiyonu	$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$(-1, 1)$
ReLU	$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ x & \text{ için } x \geq 0 \end{cases}$	$[0, \infty)$
Leaky (Sızıntı) ReLU	$f(x) = \begin{cases} 0.01 & \text{ için } x < 0 \\ x & \text{ için } x \geq 0 \end{cases}$	$(-\infty, \infty)$
Swish Fonksiyonu	$f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0 & \text{ için } f(x) = x \\ \beta \rightarrow \infty & \text{ için } f(x) = 2\max(0, x) \end{cases}$	$(-\infty, \infty)$

Şekil 3.17: Aktivasyon Fonksiyonları [110].

Sinir Ağı Öğrenme Yöntemleri

Şu ana kadar yapılan işlemler ileri yayılım olarak adlandırılır.

Öğrenme süreci, daha önce de ifade edildiği gibi, hata fonksiyonunun minimize edilmesi amacıyla ağ parametrelerinin güncellenmesi üzerine kuruludur ve bu güncellemeler, geri yayılım algoritması aracılığıyla gerçekleştirilir. Bu bağlamda, gradyan inişi algoritması, ilk optimizasyon yöntemlerinden biri olarak literatürde yer almaktadır. Ancak veri setlerinin karmaşıklığı ve örnekleme sayısındaki artış, global ya da lokal minimumların bulunmasını

zorlaştırmakta ve hesaplama yükünü artırmaktadır. Bu doğrultuda, literatürde zaman içerisinde pek çok alternatif optimizasyon yöntemi geliştirilmiştir [111].

Stokastik gradyan inişi (SGD), klasik gradyan inişi algoritmasının bir varyantı olarak, her iterasyonda rastgele seçilmiş örnekler üzerinden parametre güncellemeleri yaparak çalışır. Bu yöntem, düşük hesaplama maliyeti avantajı sunmakla birlikte, parametre güncellemelerindeki varyansı artırabilir ve öğrenme sürecinde dalgalanmalara neden olabilir. Bu olumsuzlukları bertaraf etmek amacıyla geliştirilen Momentum algoritması, geçmiş adımlarda elde edilen güncellemeleri dikkate alan bir parametre aracılığıyla, SGD'nin dalgalanma etkisini azaltarak daha istikrarlı bir öğrenme süreci sağlar [111].

Önceki jenerasyon optimizasyon algoritmaları, sabit bir öğrenme oranı kullanmaları nedeniyle düşük gradyan büyüklüklerinde yetersiz performans sergilemektedir. Bu nedenle AdaGrad, RMSProp, Adam ve AdaDelta gibi algoritmalar, öğrenme oranını dinamik olarak ayarlayabilen parametrelerle bu sınırlamayı aşmayı hedeflemiştir. Söz konusu algoritmalar, geçmiş gradyan değerlerinin üstel hareketli ortalamalarını dikkate alarak öğrenme oranını optimize etmekte ve bu sayede AdaGrad'ın karşılaştığı öğrenme oranı azalması problemini önemli ölçüde azaltmaktadır [112].

Başlıca bilmemiz gereken bir diğer terimler ise şunlardır:

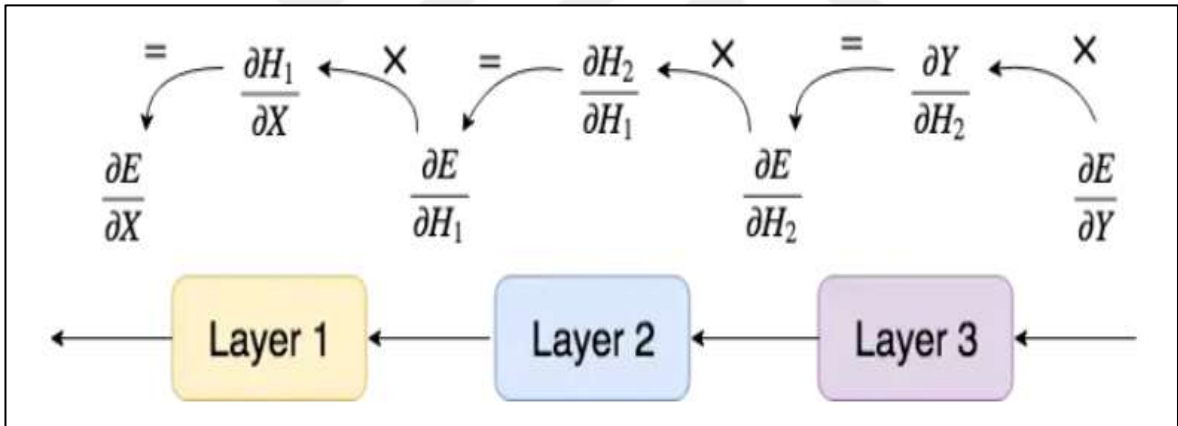
- a. Öğrenme Oranı (Learning Rate): Geri yayılım sonucu elde edilen hata bilgisinin ne kadarının ağırlıklar üzerinde etkili olacağını belirleyen, 0 ile 1 arasında bir katsayıdır. Eğer bu oran çok küçük seçilirse, modelin öğrenmesi yavaşlar; çok büyük seçilirse model dengesizleşebilir ve hedefe ulaşamayabilir [113].
- b. Yineleme sayısı (epoch): Her adımda, belirlenen öğrenme oranı doğrultusunda ağırlıklar güncellenir ve tekrar hata hesaplanır. Bu işlemin toplam kaç defa yapılacağı, modelin performansını etkiler. Yetersiz sayıda tekrar yapılırsa model öğrenemez; aşırı tekrarlar ise aşırı uyum (overfitting) problemine yol açarak genelleme yeteneğini düşürebilir [113].
- c. Ağırlık Matrisinin Başlangıç Değeri: Bir modelin doğru sonuçlar üretmesi, ağırlık değerlerinin uygun biçimde ayarlanmasına bağlıdır. Bu sebeple, başlangıçta ağırlıklara verilecek değerler büyük önem taşır. Genellikle $[0, 1]$ aralığında rastgele değerler

kullanılır. Tüm ağırlıkların başlangıçta 0 veya 1 olması, girdinin etkisini modelin ayırt edememesine yol açacağından doğru bir yöntem değildir [113].

Öğrenme yöntemlerini detaylıca ele alalım:

a. Backpropagation (Geri Yayılım): Modelin çıktısında oluşan hatalar, geriye doğru iletilerek ağıın parametreleri güncellenir. Her bağlantı için hata değeri hesaplanır ve bu bilgilerle ağı optimize etme süreci gerçekleştirilir. Girdi katmanından itibaren, ağırlıklar ve aktivasyon fonksiyonları kullanılarak çıktıya ulaşılırken meydana gelen hatalar hesaplanır. Bu hataların sadece çıkış katmanında kalması yerine, ara katmanlara da bildirilmesi gerekir. Aksi halde nöronlar ağırlıklarını düzeltemez ve hata azaltılamaz. Bu sürecin, yani önceki katmanların hatadan haberdar edilip kendilerini ayarlama mekanizmasının adı geri yayılımdır.

Geri yayılım ayrıca ağırlıkların değiştirilmesinin tüm ağıdaki maliyet fonksiyonunu nasıl etkilediğini görmeye de yardımcı olur [112].



Şekil 3.18: Backpropagation (Geri Yayılım).

b. Stokastik Gradient Descent (SGD): Bu yöntem, her bir eğitim örneği üzerinden yapılan gradyan inişi yaklaşımıyla ağı günceller. Böylece modelin öğrenme süreci ilerler. Parametreler, her örnek üzerinde işlem yapılırken güncellenmektedir. Genellikle her eğitim döngüsünde (epoch), veriler rastgele sıralanır. Bu teknik, özellikle büyük veri kümeleriyle çalışırken daha az bellek tüketir ve klasik BGD yöntemine kıyasla çok daha hızlıdır. Ancak her veri örneğiyle ağı güncellemesi nedeniyle model, hedefe doğru ilerlerken oldukça düzensiz ve gürültülü adımlar atabilir. Bu da yakınsama sürecini dengesizleştirebilir [112].

c. Momentum: Bu teknik, önceki adımlardan alınan ivme sayesinde daha hızlı ve kararlı bir öğrenme süreci sağlar. Yapay sinir ağlarında, hedef fonksiyon üzerinde çalışırken önceki adımdaki gradyan bilgisiyle birlikte yeni güncelleme yapılır. Bu, λ ile gösterilen momentum katsayısı üzerinden ağırlıklandırılır. Ancak bu durum zaman zaman modelin küresel minimuma ulaşmak yerine yerel minimumda takılı kalmasına sebep olabilir. Bu, özellikle hedef fonksiyon dışbükey değilse yaygın bir durumdur [112].

Momentum katsayısı (λ), 0 ile 1 arasında tutulur. Düşük değerler, yerel minimumda takılma ihtimalini artırırken, yüksek değerler modelin daha hızlı yakınsamasını sağlar. Ancak yüksek momentum değeri, modelin küresel minimumu atlmasına da sebep olabilir. Gradyanların yönü sürekli değiştiğinde, doğru momentum değeri öğrenme sürecinin daha dengeli ilerlemesine katkı sağlar [112].

d. AdaGrad: Her bir parametre için bağımsız olarak öğrenme oranını ayarlayan bir optimizasyon algoritmasıdır. Zamanla öğrenme oranını azaltarak, sık güncellenen parametrelerde daha küçük adımlar atılmasını sağlar [112].

e. RMSprop: AdaGrad algoritmasının geliştirilmiş bir versiyonudur. Öğrenme oranının daha dengeli kalmasını sağlar ve zamanla aşırı küçülmesini engeller [112].

f. Adam (Adaptive Moment Estimation): Momentum ve adaptif öğrenme oranlarını birleştiren etkili bir optimizasyon algoritmasıdır. Adam, günümüzde derin öğrenme modellerinde yaygın olarak kullanılan bir yaklaşımdır. İkinci mertebeden türevleri temsil eden Hessian matrisine dayanan prensipler içerir. Derin sinir ağları için tasarlanmış olan bu yöntem, düşük hesaplama maliyeti ve yüksek bellek verimliliği gibi avantajlar sunar. Adam algoritması, her parametre için ayrı adaptif bir öğrenme oranı hesaplar. RMSprop'un

kare gradyan ortalaması yaklaşımını ve Momentum'un hareketli ortalama yapısını birleştirir [112].

g. Learning Rate Scheduling: Eğitim süreci ilerledikçe, öğrenme oranını adım adım azaltmak gibi dinamik bir ayarlama mekanizmasıdır. Bu, daha stabil bir öğrenme süreci sağlar [112].

h. Early Stopping: Eğitim sırasında doğrulama verisindeki hata yükselmeye başlarsa, süreci durdurarak modelin aşırı öğrenmesini (overfitting) önlemeye yardımcı olur [113].

İleri Beslemeli Yapay Sinir Ağları

İleri beslemeli yapay sinir ağları, nöronların giriş katmanından çıkış katmanına kadar tek yönlü olarak sıralandığı bir yapıdır. Aynı katmanda yer alan nöronlar arasında doğrudan bir bağlantı bulunmaz. Her bir nöron yalnızca bir sonraki veya bir önceki katmandaki nöronlarla bağlantılıdır. Girdi verileri doğrudan gizli katmanlara (ara katmanlara) iletilir; burada işlenen veriler çıkış katmanına aktarılır ve nihai çıktılar elde edilir. Süreç, yalnızca girişten çıkışa doğru tek yönlü ilerler ve burada sonlanır [114].

Geri Beslemeli Yapay Sinir Ağları

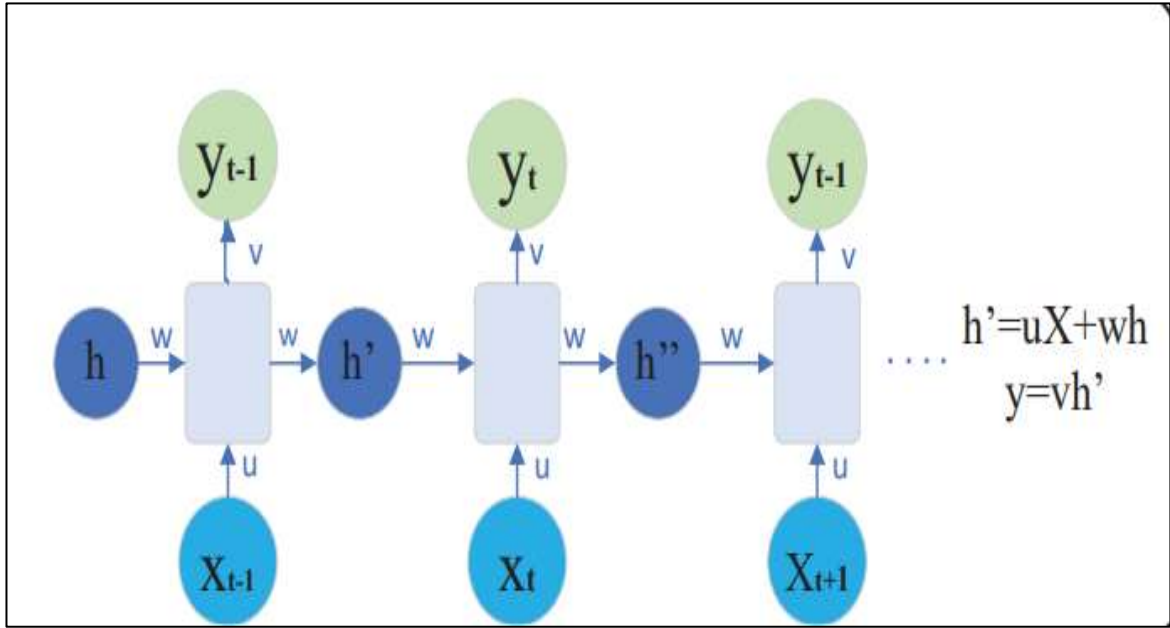
Geri beslemeli yapay sinir ağları, ara ve çıkış katmanlarından elde edilen çıktıların, giriş katmanına ya da önceki ara katmanlara geri gönderilebildiği yapılar olarak tanımlanır. Bu yapılarda bir nöronun çıktısı yalnızca kendisinden sonraki katmana değil, aynı zamanda aynı katman içinde veya bir önceki katmana da girdi olabilir. Böylelikle veri akışı sadece ileri yönde değil, geri yönde de gerçekleşir. Bu çift yönlü aktarım, doğrusal olmayan dinamik bir yapı oluşturur. Geri besleme bağlantılarının düzenlenme biçimine göre, aynı sinir ağı farklı davranışlar sergileyebilir. Bu yapı, özellikle giriş verilerinin doğrudan etkilenmesini sağladığı için tahminleme gibi zaman bağımlı problemlerde sıkça tercih edilir [114].

3.5.1.2 Tekrarlayan sinir ağları

Jeffrey Elman tarafından geliştirilen tekrarlayan sinir ağları (RNN), önceki çıktıyı bir sonraki girdiye entegre ederek döngüsel bir yapı oluşturan ve bu süreçte sıralı bilgiyi dikkate alan öğrenme ve tahmin modelleridir. Bu yapılar, bilgilerin döngü içerisinde kalıcılığını sağlayarak, ihtiyaç duyulduğunda tekrar kullanılmasına olanak tanır [115]. RNN'ler, klasik ileri beslemeli sinir ağlarından farklı olarak kendi giriş belleklerini kullanarak veri dizilerinin

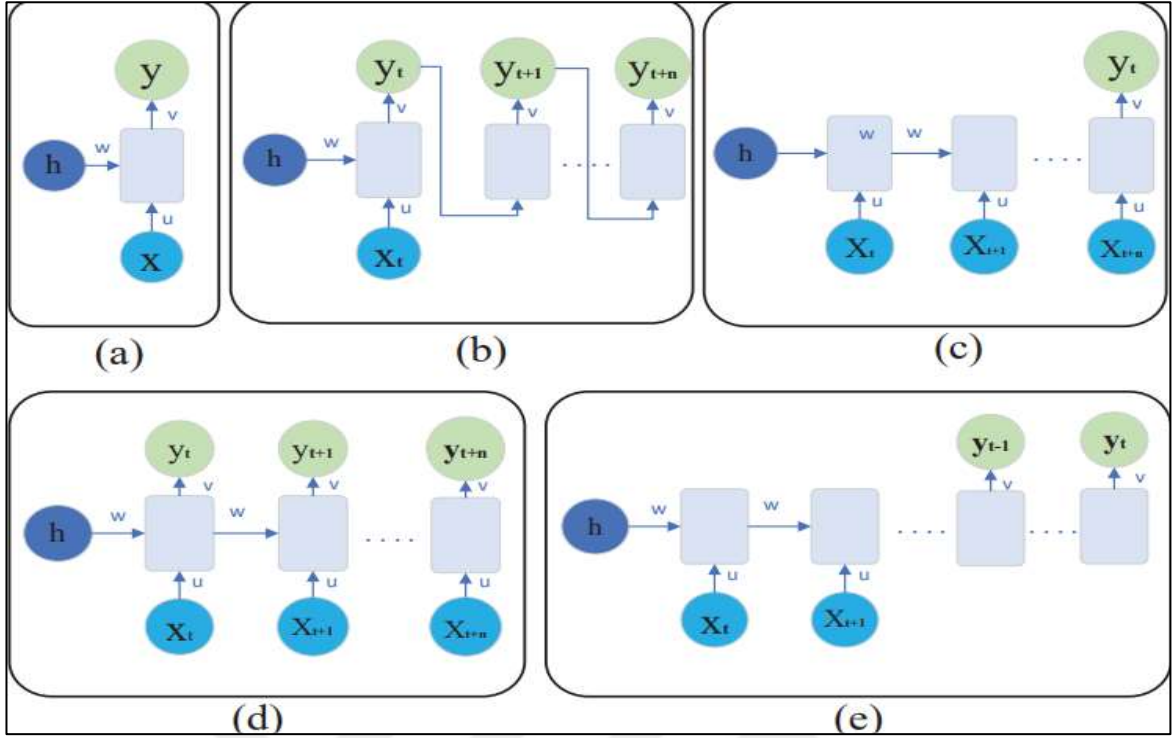
sırasal yapısını işleyebilirler. Bu modelin temel amacı, sıralı gelen veriler arasında ilişki kurarak bu ilişkileri öğrenmektir. Geleneksel sinir ağlarında girişler bağımsız kabul edilirken, RNN'lerde her bir verinin çıktısı, önceki girdiye ve onun çıktısına bağlıdır.

RNN algoritmasının işleyiş biçimi gösterilmiştir. Burada herhangi bir t zamanındaki $\mathbf{h}$, gizli katmandaki aktivasyonu; $\mathbf{x}$, girdi verisini ve $\mathbf{y}$, çıktıyı temsil etmektedir. Ayrıca, önceki h değeri, bir sonraki zaman adımında giriş olarak kullanılır.



Şekil 3.19: RNN Algoritmasının İşleyiş Biçimi.

Döngüsel yapılarından dolayı RNN'ler zamanla gelişen olaylar arasındaki bağlantıları kurma konusunda başarılıdır. Bu sıralı ilişkilerin analiz ve sınıflandırılmasındaki yetenekleri, bu ağları oldukça kullanışlı kılar. RNN modeli; metin çevirileri, otomatik altyazı oluşturma, video gürültü giderme, uzun belgelerin özetlenmesi, el yazısı sahibinin belirlenmesi, ses tanıma gibi çok çeşitli alanlarda başarıyla uygulanmaktadır [115]. RNN algoritmasının beş farklı çalışma yapısı bulunmaktadır: (a) Tek giriş-tek çıkış, (b) Tek giriş-çok çıkış, (c) Çok giriş-çok çıkış, (d) Çoklu sıralı giriş-çoklu sıralı çıkış, (e) Çoklu senkronize giriş-çoklu senkronize çıkış.

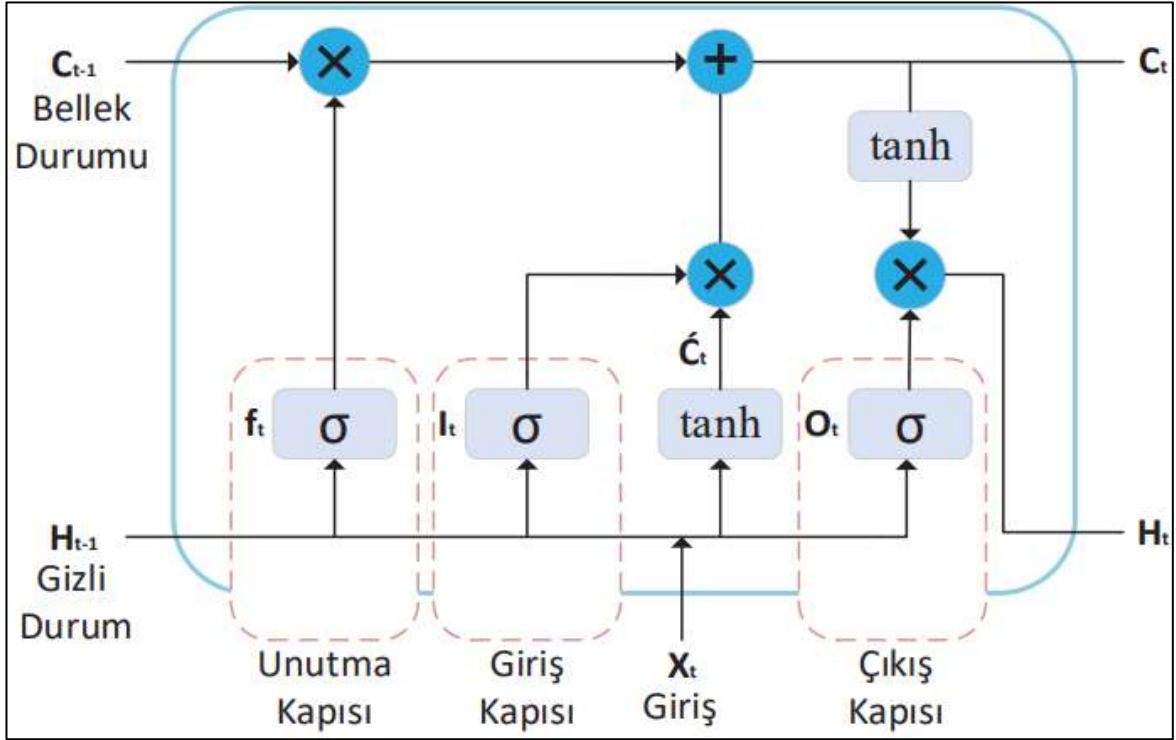


Şekil 3.20: RNN Algoritmasının Beş Farklı Çalışma Yapısı Gösterilmiştir.

3.5.1.3 Uzun kısa süreli bellek

Uzun Kısa Süreli Bellek (LSTM), Sepp Hochreiter ve Jürgen Schmidhuber tarafından 1997 yılında, geleneksel RNN modellerinin bazı yetersizliklerini gidermek amacıyla geliştirilmiştir [116]. RNN modellerinde, zaman içerisindeki ardışık veriler arasında oluşan kopukluklar, bir sonraki adımın tahmin edilmesini güçleştirmekteydi. LSTM, bu sorunun üstesinden gelmek için, klasik RNN hücrelerinin yerine bellek blokları önererek, özellikle kaybolan ve patlayan gradyan problemlerini çözmeyi hedeflemiştir. Bu modeller, bilgi depolamak için hücre durumunu kullanır ve bu yönüyle RNN'lerden önemli bir ayrım gösterir. LSTM, geçmiş verileri hatırlayabilir ve bunları güncel bilgilerle ilişkilendirerek karar verme sürecinde kullanılabilir [115]. LSTM algoritması, geçmişteki bilgileri hafızada tutarak özellikle bellek gerektiren uygulamalarda tercih edilir. RNN tabanlı bir model olan LSTM, ileri beslemeli sinir ağlarından farklı olarak geriye doğru bağlantılara sahiptir ve rastgele zaman aralıklarında verileri hatırlayabilir. Bu özelliği sayesinde sadece görüntü

işleme değil, aynı zamanda ses, video gibi ardışık ve karmaşık veri türlerinin işlenmesinde de oldukça başarılıdır.



Şekil 3.21: Herhangi t Anında Genel LSTM Hücresinin Diyagramı.

LSTM hücresindeki yapılar; **Giriş Kapısı (I)** verinin girişine izin verirken, **Unutma Kapısı (f)** önceki hücreden gelen bilginin ne kadarının korunacağını belirler. **Aday Bellek** yeni bilgilerin öğrenilmesini desteklerken, **Çıkış Kapısı (O)** ise son çıktının üretilmesini sağlar. Bu kapılar, $[0,1]$ aralığında çıktı üretmek için sigmoid fonksiyonunu; $[-1,1]$ aralığında ise tanh fonksiyonunu kullanır. LSTM; görüntü alt yazı üretimi, sessiz videolara otomatik ses ekleme, cümlelerden kelime tahmini, düzensiz dillerin öğrenilmesi, robotik kalp ameliyatları gibi çeşitli alanlarda başarıyla kullanılmaktadır [116].

3.5.1.4 U-Net

Hedeflenen anatomik yapıların segmentasyonu, tıbbi görüntüleme alanında temel adımlardan biri olarak sıkça uygulanmaktadır [117]. Görüntü segmentasyonu hem nesnenin algılanmasını hem de ilgili nesnenin uzamsal sınırlarının piksel ya da voksel düzeyinde olasılık haritaları ile belirlenmesini içerir. Önceki bölümlerde belirtildiği gibi, CNN mimarileri genel görüntü sınıflandırması ve nesne tespiti için etkili olmakla birlikte, piksel bazlı olasılık tahmini için her zaman en uygun seçenek olmayabilir. Ronneberger ve çalışma

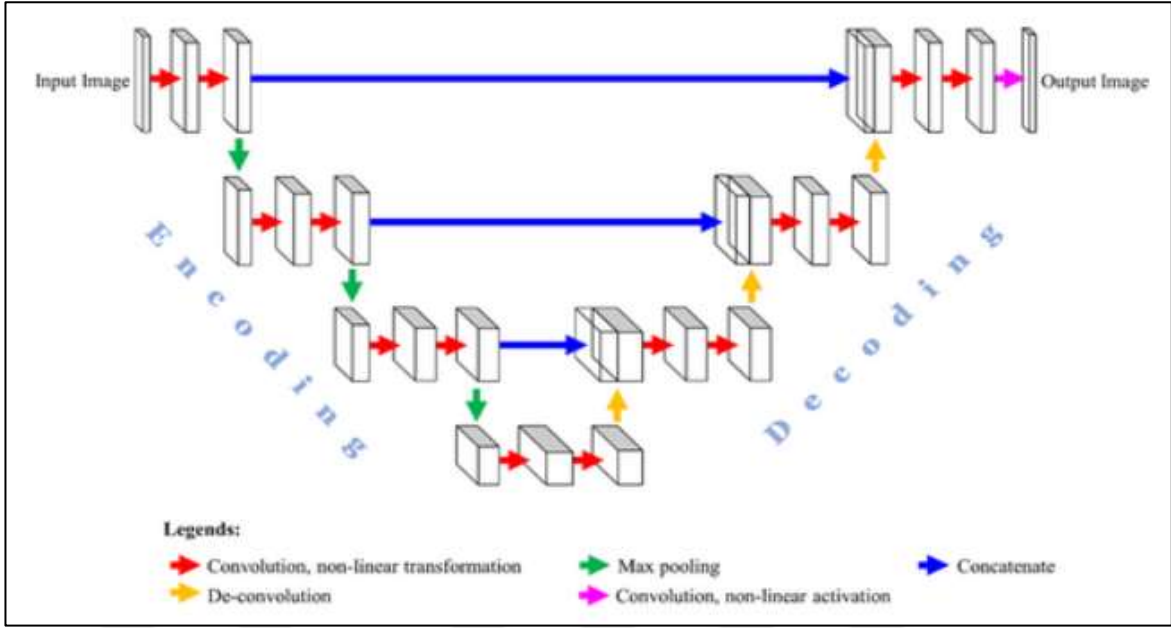
arkadaşları, bu amaca yönelik olarak 2B ve 3B U-net mimarilerini sunmuş ve bu modeller piksel ile voksel düzeyinde olasılık haritalaması için ıgır açıcı olarak kabul edilmiştir [118].

U-net'e benzer şekilde geliştirilen diğ er yapılar arasında V-net, E-net ve SegNet yer almakta olup, tüm bu ağlar benzer şekilde simetrik bir kodlayıcı-kod çözücü mimarisini benimsemektedir. U-net ve varyasyonları, biyomedikal görüntü segmentasyonunda derin öğrenme temelli yaklaşımlar arasında en sık tercih edilen ağ yapılarından biri haline gelmiştir [114]. U şeklinde tasarlanmış bu mimari, sol tarafta daralan (kodlayıcı) ve sağ tarafta simetrik şekilde genişleyen (kod çözücü) yol olmak üzere iki ana bölümden oluşur ve çok ölçekli kodlanmış özellikler üzerinden piksel düzeyinde sınıflama yapılmasına olanak tanır. Kodlayıcı yol, klasik CNN yapısına benzer şekilde çalışır; her katmanda özellik çıkarımı yapılır ve bu özellikler, havuzlama işlemleriyle daha yüksek seviyeli gösterimlere dönüştürülür. En alt katmanda havuzlama uygulanmaz ve buradan elde edilen özellikler, doğrudan kod çözücü yola aktarılır [118].

Kod çözücü yol ise üç temel adımdan oluşur:

- a. Bir önceki katmandan gelenlerin yukarı örneklenmesi,
- b. Eşdeğer çözünürlükteki kodlayıcı katmandan gelen özelliklerle birleştirilmesi,
- c. Birleştirilen haritanın evrişim işlemiyle işlenmesi.

Bu işlemlerde kullanılan evrişim çekirdekleri eğitim süreci boyunca öğrenilir. Kodlayıcı ve kod çözücü yollar arasında yapılan bu birleştirme, gradyanların daha serbest şekilde akmasına izin vererek, her ölçekteki özelliklerin öğrenimini kolaylaştırır ve aynı zamanda kaybolan gradyan sorununu hafifletir [118]. Son katmanda ise çok sayıda özellik haritası birleştirilerek her bir piksel için nesneye ait olasılık değeri hesaplanır. U-net mimarisi, çok ölçekli ve veri odaklı özellik öğrenimini mümkün kılarak segmentasyon problemlerine etkili çözümler sunar. Ronneberger ve ekibinin orijinal makalesinde, bu yöntem 2B elektron mikroskopi görüntülerinde nöronal yapıların segmentasyonunda kullanılmış ve kayan pencereli CNN yaklaşımına kıyasla daha yüksek başarı sağlamıştır [118]. U-net'in kodlama yolu farklı ölçeklerde özellik haritaları üretirken, kod çözme yolu kodlayıcı yolu boyunca hesaplanan çok ölçekli özelliklere dayalı olarak orijinal görüntü çözünürlüğünde segmentasyon çıktısını üretir.



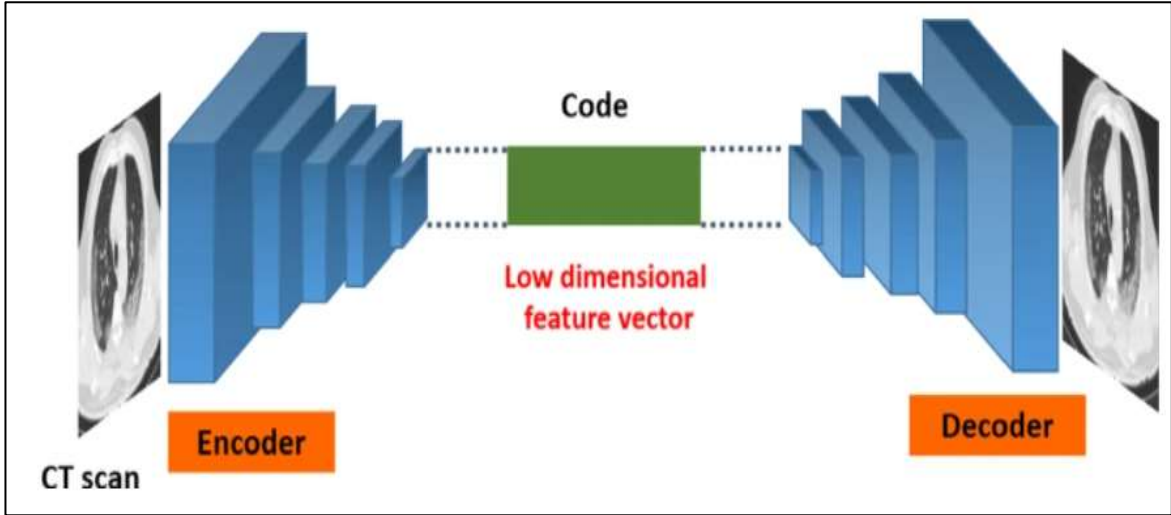
Şekil 3.22: U-Net Mimarisindeki Ana Ağ Bileşenleri.

3.5.1.5 Otomatik kodlayıcılar

Derin otomatik kodlayıcılar (AE), denetimsiz öğrenme yöntemlerinde yaygın şekilde kullanılan ve giriş katmanındaki verileri çıkış katmanında yeniden oluşturmaya çalışan, literatürde Diablo ağı olarak da bilinen özel bir sinir ağı mimarisidir [119].

AE yapısı iki ana aşamadan oluşur: kodlayıcı (encoder) ve kod çözücü (decoder). İlk aşamada, giriş verileri daha düşük boyutlu bir alana yansıtılırken, bu temsilde anlamlı özelliklerin korunması sağlanır. İkinci aşamada ise, bu sıkıştırılmış temsilden özgün verilere yakın bir yeniden yapılandırma yapılır. Şekil 2.36’da oto kodlayıcının bu iki temel bölümünü görsel olarak sunmaktadır [119].

Zatürre teşhisinde bu tür bir denetimsiz sınıflandırma yaklaşımının avantajı, gözetimli yöntemlerde olduğu gibi büyük miktarda etiketlenmiş veriye olan ihtiyacı ortadan kaldırmasıdır; bu da teşhis sürecini hızlandırarak ölüm riskinin azaltılmasına katkıda bulunabilir [120]. Derin bir oto kodlayıcının ana amacı, yüksek boyutlu veriyi daha küçük bir boyuta indirgerken bilgi kaybını en aza indirmektir. Bu tür yapılar, sadece boyut indirgeme için değil, aynı zamanda veri sıkıştırma, gürültüden arındırma [121] ve denetimli modellerde öznetelik çıkarımı için de kullanılabilir.



Şekil 3.23: Otomatik Kodlayıcılar (Autoencoders- AE).

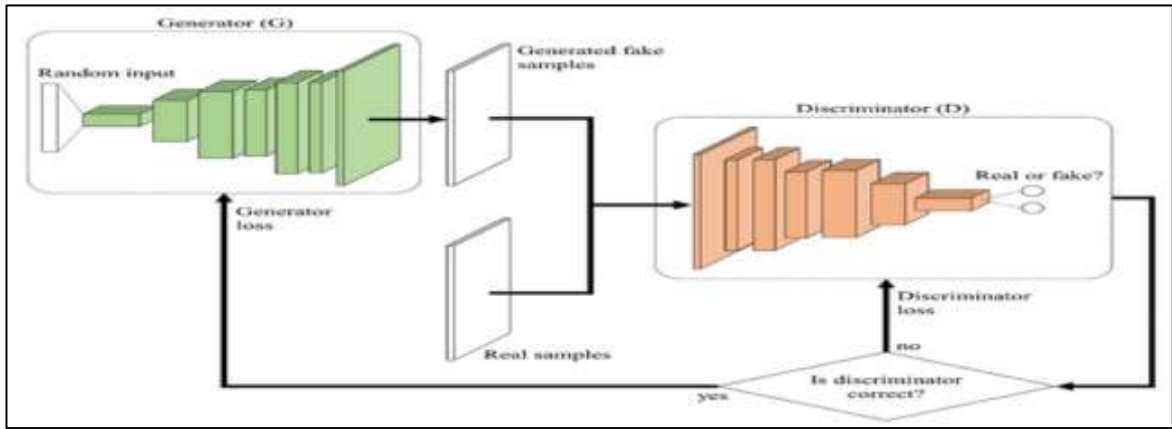
3.5.1.6 Üretken çatışmalı ağlar

Üretken Düşmanca Ağlar (Generative Adversarial Networks – GANs), derin öğrenme mimarilerinde jeneratör ve ayıricıdan oluşan çift yapısıyla öne çıkan modellerdir. Bu yöntem, Goodfellow ve arkadaşları tarafından düşmanca bir öğrenme süreciyle üretici modeller oluşturmak amacıyla tanıtılmıştır [122].

GAN'ların iki temel bileşeni olan jeneratör (G) ve ayıricı (D), birlikte çalışır: jeneratör, eğitim verisine benzer yeni görüntüler üretmeyi öğrenirken; ayıricı, üretilen bu görüntülerin gerçek mi sahte mi olduğunu ayırt etmeyi öğrenir. Bu süreç, sıfır toplamlı bir rekabet oyunu olarak işler; jeneratör, öyle gerçekçi görüntüler üretir ki sonunda ayıricı bu görüntülerin sahte olup olmadığını ayırt edemez hale gelir ve böylece jeneratör, eğitim verisinin dağılımına yakın örnekler üretmiş olur [123].

Jeneratör, genellikle sabit boyutlu rastgele bir girdi vektörü alır ve GU-net'in genişleyen yoluna benzer şekilde, bu girdi üzerinden bir görüntü üretir – ancak atlamalı bağlantılar olmaksızın. Diğer yandan ayıricı (D), genellikle bir görüntüyü giriş olarak alır ve bunun gerçek mi sahte mi olduğunu belirleyen ikili bir sınıflandırma gerçekleştirir. Her iki bileşen de hata geri yayılım yöntemiyle birlikte eğitilir [124]. Örneğin ayıricı, jeneratörün ürettiği bir sahte görüntüyü başarıyla tanırsa bu başarının etkisi jeneratöre hata olarak aktarılır. Aksi durumda ayıricı bir ceza ile karşılaşır. GAN'lar; metinden görsele dönüşüm, süper

özünürlük ve görüntü çevirme gibi birçok alanda başarılı bir şekilde uygulanmıştır. Bazı popüler GAN türevleri şunlardır: Koşullu GAN, CycleGAN, Stil Tabanlı GAN, En Küçük Kareler GAN ve DiscoGAN. Radford ve arkadaşları, giriş ve çıkış verileri arasında doğrudan eşleştirme gerektirmeyen gözetimsiz bir GAN yapısı sunmuştur. Bu tür GAN modelleri, düşük ve yüksek çözünürlüklü görüntüler arasında doğrudan ilişki kurmanın zor olduğu tıbbi görüntü işleme uygulamalarında da etkilidir [125]. Ayrıca GAN'lar, önceden eğitilmiş modellerle birlikte hibrit yapılar oluşturarak eğitim aşamasında destekleyici bileşen olarak da kullanılabilir.

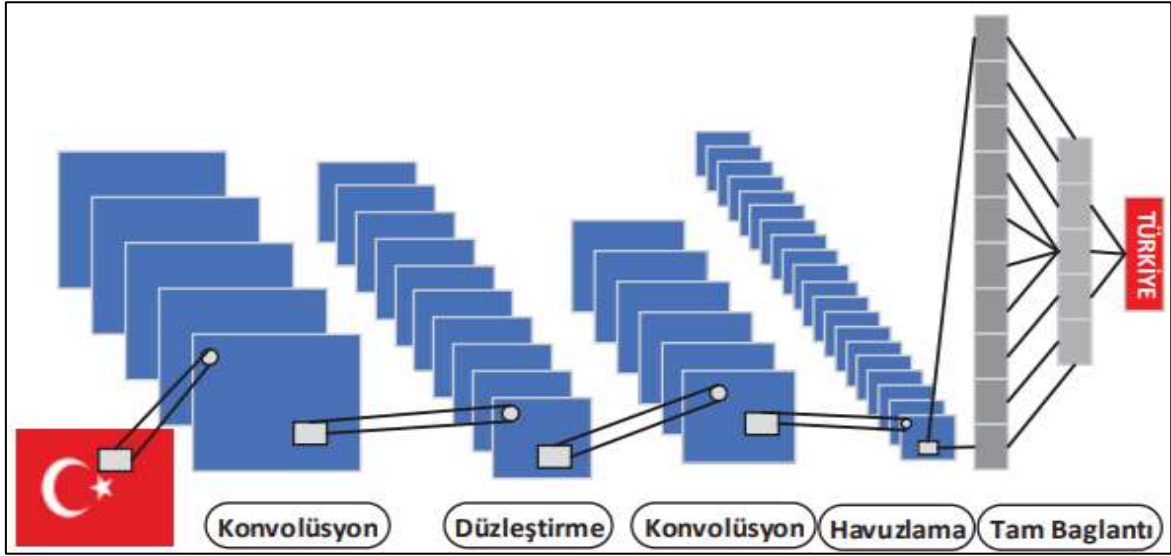


Şekil 3.24: Üretken Çatışmalı Ağlar (GANs).

3.5.1.7 Evrişimli sinir ağları

Konvolüsyonel Sinir Ağları (Convolutional Neural Networks- CNN), temel olarak görselleri sınıflandırmak ve içeriklerine göre nesneleri veya objeleri benzer özellikleri doğrultusunda gruplamak amacıyla kullanılan çok katmanlı ileri beslemeli bir yapay sinir ağı türüdür. Bu ağı modeli ilk olarak 1968 yılında Hubel ve Wiesel tarafından, hayvanların görsel sistemleri temel alınarak geliştirilmiştir [126]. Yapay sinir ağları arasında en yaygın kullanılan yöntemlerden biridir. CNN'in seleflerine göre en büyük avantajı, gerekli özellikleri insan müdahalesine ihtiyaç duymadan otomatik olarak öğrenebilmesidir.

CNN mimarisi, konvolüsyon (convolution), aktivasyon (rectified linear unit), havuzlama (pooling) ve tam bağlantı (fully connected) katmanlarından oluşur. Bu katmanlar, elde

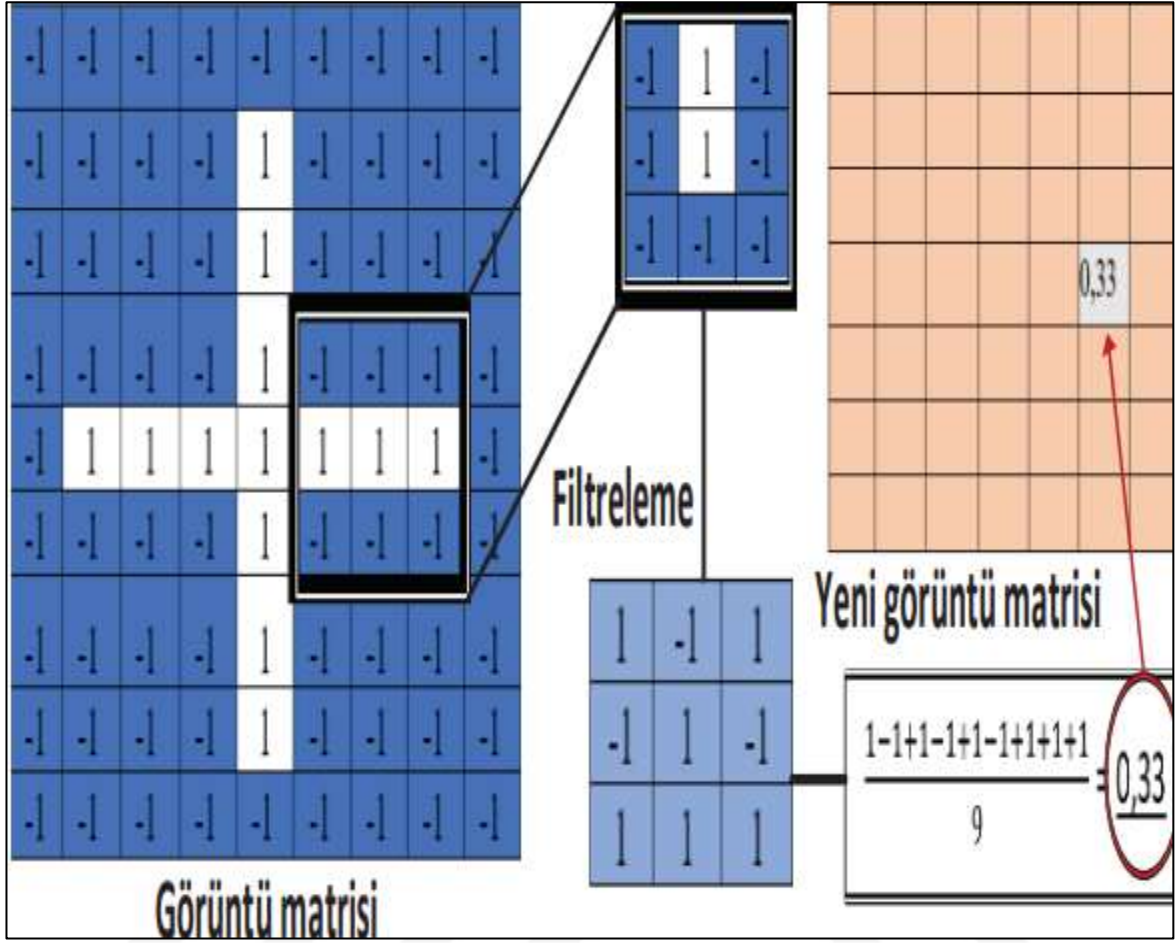


Şekil 3.25: Konvolüsyonel Sinir Ağı Katmanları [31].

edilmek istenen görüntü kalitesine göre farklı sayıda ve boyutta düzenlenebilir. Şekil 2.36’te bu katmanların sıralaması gösterilmiştir [127].

Konvolüsyonel sinir ağıları, çok sayıda filtreleme ve öznitelik çıkarımı gerçekleştirdiği için görsellerdeki farklı detayları tespit edip ayırmada oldukça etkilidir. Görüntülerde yer alan nesnelerin türü (örneğin; insan, araç, hayvan, bitki vb.) kolayca tanımlanabilir [127].

İlk aşama olan konvolüsyon katmanında, Şekil 3.28’de gösterildiği gibi, $9 \times 9 \times 3$ boyutundaki bir görüntü matrisi üzerinde $3 \times 3 \times 3$ ’lük bir filtreleme uygulanır ve sonuç olarak $7 \times 7 \times 3$ boyutlarında yeni bir çıktı matrisi elde edilir. Bu işlem sırasında, filtre matrisi görüntü matrisi üzerinde satır ve sütun bazında hareket ettirilerek çarpım işlemi yapılır ve elde edilen değerlerin ortalaması alınarak sonuç matrise işlenir [128]. Bu değerler filtre türüne ve veri boyutuna göre farklılık gösterebilir. Konvolüsyon katmanında uygulanabilecek yöntemlerden birisi olan Conv2D ise şöyle açıklanabilir; bu katman, 2B uzamsal (veya zamansal) bir boyut (yükseklik ve genişlik) üzerinde katman girişiyle evrilen ve çıktıların bir tensörünü üreten bir evrişim çekirdeği oluşturur. use_bias True ise, bir önyargı vektörü oluşturulur ve çıktılarına eklenir. Son olarak, değilse activation, None çıktılarına da uygulanır.

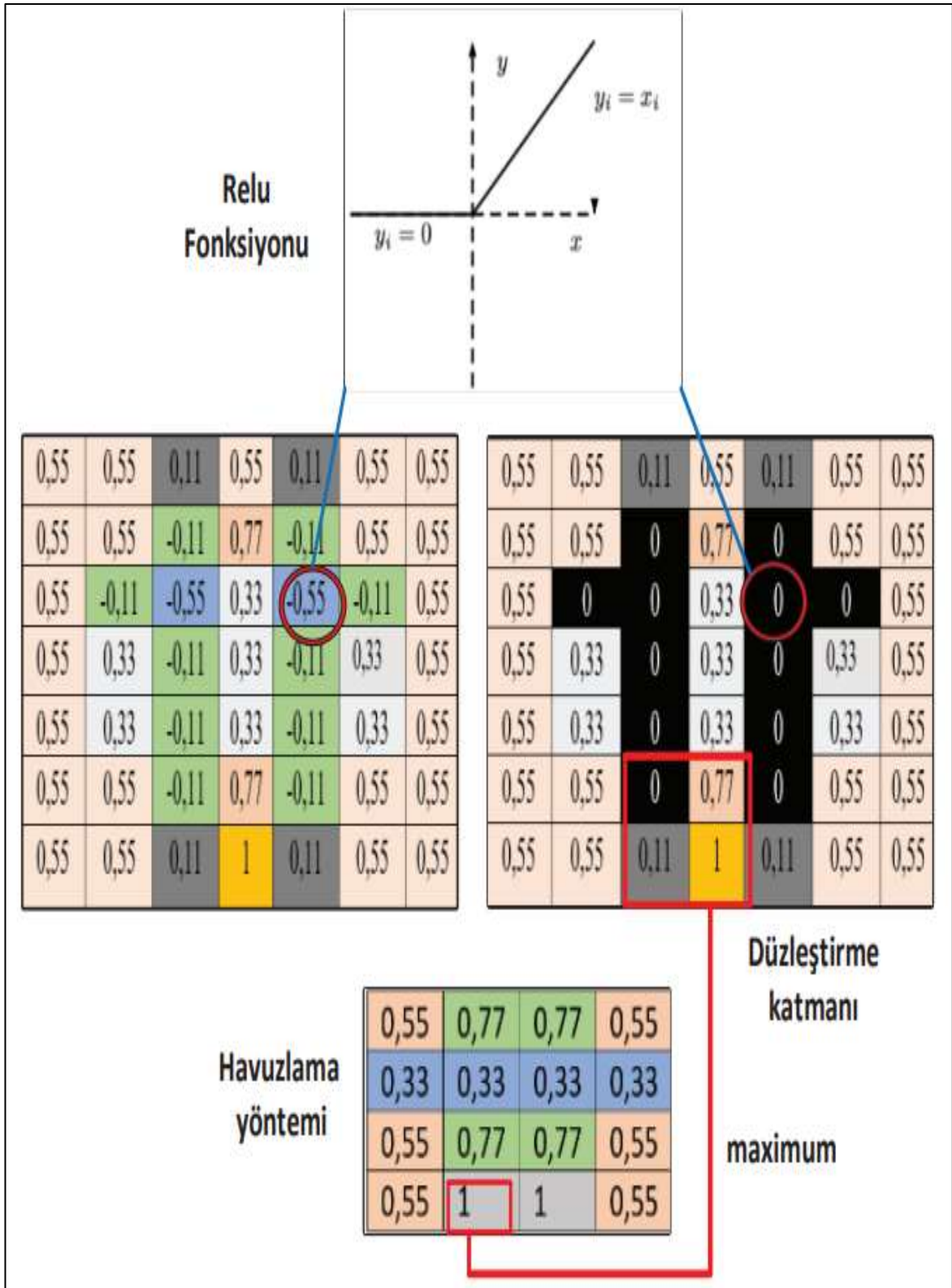


Şekil 3.26: Konvolüsyonel Katmanı İşlem Basamakları.

İkinci aşamada, düzleştirme (aktivasyon) işlemi gerçekleştirilir. Bu aşamada yaygın olarak kullanılan fonksiyon Relu (Rectified Linear Unit) fonksiyonudur. Bu işlemin amacı negatif değerleri sıfıra eşitleyerek karanlık alanların etkisini ortadan kaldırmaktır [128].

Üçüncü adım olan havuzlama katmanının görevi, konvolüsyon sonrası oluşan görüntü matrisinin boyutunu azaltarak modelin işlem süresini ve karmaşıklığını düşürmektir. En sık kullanılan yöntem, maksimum havuzlama (max pooling) olup, genellikle 2x2'lik matrisler kullanılarak en yüksek değer seçilir. Relu'dan sonra uygulanan bu katmanın temel hedefi, düşük aydınlatmalı yani önemsiz bölge etkilerini azaltmaktır. İstenilen netliğe ulaşamadığı durumlarda, havuzlama işlemi tekrar edilebilir. Havuzlama katmanında kullanılan yöntemlerden birisi ise MaxPooling2D yöntemidir [128]. Girişin her kanalı için bir giriş penceresi (boyutu ile tanımlanan) üzerindeki maksimum değeri alarak girişi mekansal

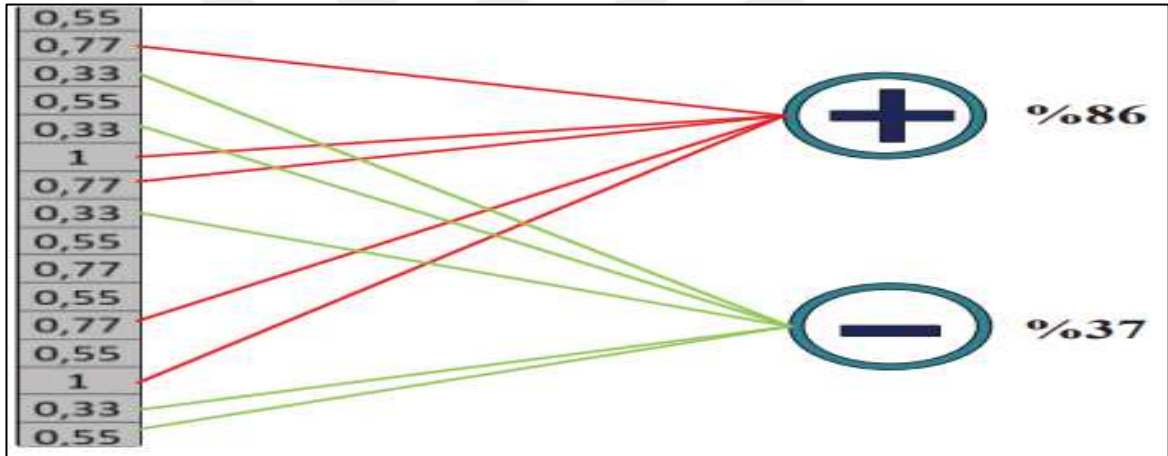
boyutları (yükseklik ve genişlik) boyunca aşağı örneklemeler . Pencere her boyut boyunca kaydırılır.



Şekil 3.27: Relu Fonksiyonu Kullanarak Düzleştirme Katmanı ve Maksimum Havuzlama Yöntemi.

Bahsetmemiz gereken bir diğer katman ise Flatten (Düzleştirme) Katmanıdır. Bu katmanın görevi basitçe, son ve en önemli katman olan Tam Bağımlı Katman'ın (Fully Connected Layer) girişindeki verileri hazırlamaktır. Genel olarak, sinir ağları, giriş verilerini tek boyutlu bir diziden alır. Bu sinir ağındaki veriler ise Evrişim ve Havuzlama katmanından gelen matrislerin tek boyutlu diziye çevrilmiş halidir. Matris halinde gelen girdiyi vektör haline dönüştürür [129].

Konvolüsyon, havuzlama ve düzleştirme işlemlerinin sayısı; kullanılan görsel verinin yapısına veya uygulama amacına bağlı olarak değişiklik gösterebilir. Bu işlemlerin temel amacı, tam bağlantılı katmana geçmeden önce tek boyutlu bir giriş vektörü oluşturmaktır.[129] Tam bağlantı katmanında, nesneye ait çıkarılan özelliklerin hangi sınıfa ait olduğu belirlenir. Tam bağlantı katmanında elde edilen vektördeki en yüksek değerlerin ortalaması doğruluk oranını, en düşük değerlerin ortalaması ise hata oranını ifade etmektedir.



Şekil 3.28: Tam Bağlantı Katmanında Elde Edilen Vektördeki En Yüksek Değerlerin Ortalaması Doğruluk Oranı.

Bu aşamaların ardından kayıp (loss) fonksiyonları devreye girer [130]. Önceki bölümlerde CNN mimarisinin çeşitli katmanlarına değinilmiş olup, sınıflandırma sonucunun elde edildiği çıktı katmanı, bu yapının son basamağını oluşturur. Eğitim süresince ortaya çıkan tahmin hatalarının ölçülmesi amacıyla, çıktı katmanında uygun bir kayıp fonksiyonu kullanılır. Bu hata, modelin öngördüğü değer ile gerçek değer (etiket) arasındaki farkı gösterir ve CNN'in öğrenme süreci bu farkı minimize etmek üzere optimize edilir [131].

Kayıp fonksiyonu, bu hatayı hesaplamak için iki parametreye ihtiyaç duyar: modelin tahmini çıktısı (öngörülen değer) ve gerçek çıktı (etiket). Problemin yapısına bağlı olarak farklı kayıp fonksiyonları tercih edilebilir. Aşağıda bazı temel kayıp fonksiyonları özetlenmektedir [132, 133]:

- a. Çapraz Entropi (Cross Entropy) veya Softmax Kayıp Fonksiyonu: Genellikle çok sınıflı sınıflandırma problemlerinde kullanılır ve log-kayıp fonksiyonu olarak da bilinir. Softmax aktivasyonu yardımıyla modelin çıktısını bir olasılık dağılımı şeklinde üretir.
- b. Öklid Kayıp Fonksiyonu: Regresyon problemlerinde yaygın olarak tercih edilir. Genellikle ortalama kare hata fonksiyonunun bir türevi olarak değerlendirilir.
- c. Menteşe Kayıp Fonksiyonu (Hinge Loss): Özellikle ikili sınıflandırma problemlerinde, maksimum marj ilkesine dayalı sınıflandırmalarda (örneğin SVM) kullanılır. Bu fonksiyon, hedef sınıflar arasındaki marjı maksimize etmeye odaklanır.

Günümüzde konvolüsyonel sinir ağı (CNN) modelleri, özellikle görüntü işleme alanında yaygın bir şekilde kullanılmaktadır. Bununla birlikte, ses ve video işleme, tıbbi görüntüleme ve hastalıkların teşhisi, otonom sistemlerin kontrolü ve görüntüdeki nesnelerin tanımlanması gibi çeşitli uygulama alanlarında da etkin rol oynamaktadırlar.

Bilgisayarla görme uygulamalarında CNN'lerin geleneksel yapay sinir ağlarına kıyasla tercih edilmesinin bazı temel nedenleri şunlardır [134]:

- a. CNN mimarisinde ağırlık paylaşımı, öğrenilecek parametre sayısını azaltarak modelin genelleme yeteneğini artırmakta ve aşırı öğrenmeyi (overfitting) engellemeye katkı sağlamaktadır.
- b. Özellik çıkarımı ile sınıflandırma işlemlerinin aynı anda öğrenilmesi, model çıktılarının hem düzenli hem de çıkarılan özelliklerle yüksek düzeyde ilişkili olmasını sağlar.
- c. CNN'ler, büyük ölçekli uygulamalarda diğer yapay sinir ağlarına kıyasla daha verimli ve uygulanabilir yapıdadır.

CNN modellerinde aşırı öğrenme, modelin eğitim verilerinde yüksek doğruluk göstermesine rağmen test verilerinde yetersiz performans sergilemesi durumunu ifade etmektedir. Bunun

zıttı olan yetersiz öğrenme ise modelin hem eğitim hem de test verilerinde düşük başarı göstermesi anlamına gelir. Modelin her iki veri türünde de başarılı performans göstermesi "tam öğrenme" olarak tanımlanır. Şekil 11'de bu üç durum grafiksel olarak gösterilmiştir. Aşırı öğrenmenin önüne geçmek amacıyla çeşitli düzenleme yöntemleri geliştirilmiştir [135]:

a. Dropout (Düğüm Bırakma): Genelleme yetisini artırmak amacıyla yaygın olarak kullanılan bu yöntem, her eğitim döngüsünde bazı nöronları rastgele devre dışı bırakır. Bu durum, farklı nöronların bilgi taşıma kapasitesini artırır ve modelin daha çeşitli özellikler öğrenmesini sağlar.

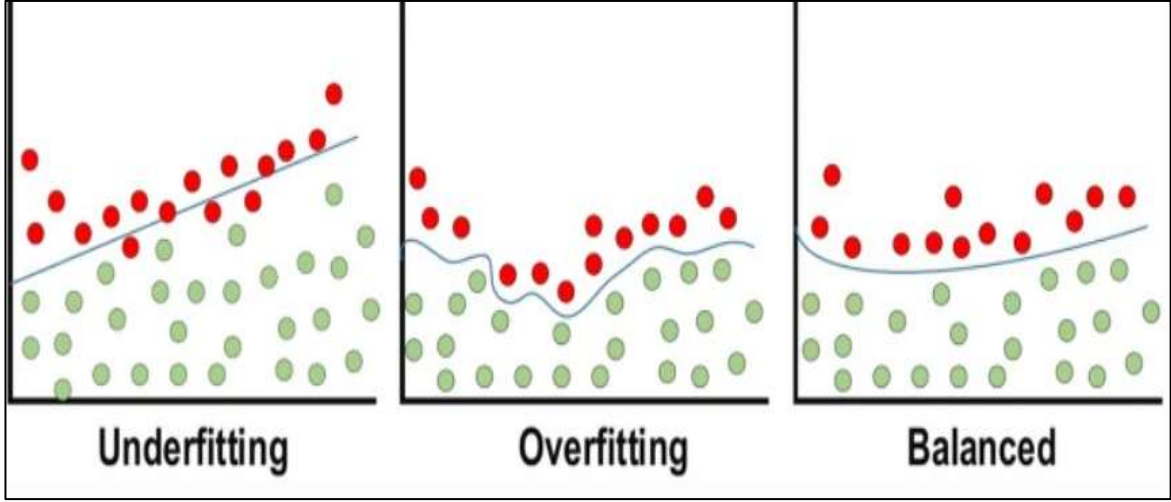
b. Ağırlık Bırakma: Dropout'a benzer şekilde, bu yöntem doğrudan nöron bağlantılarını geçici olarak devre dışı bırakır. Böylece farklı bağlantı yapılarına sahip alt ağlar eğitilerek genelleme yeteneği artırılır.

c. Veri Artırma: Eğitim verisi miktarını artırarak modeli daha çeşitli örneklerle eğitmek, aşırı öğrenmenin önlenmesinde etkili bir stratejidir. Bu doğrultuda, veri artırma teknikleri ile eğitim seti yapay olarak genişletilir.

d. Toplu Normalizasyon (Batch Normalization): Aktivasyon dağılımlarını normalize ederek, öğrenme sürecinin daha stabil hale getirilmesini sağlar. Bu teknik, katman çıktılarının ortalama ve standart sapmasını dengeleyerek eğitim sürecinin hızlandırılmasına yardımcı olur. Aynı zamanda içsel kovaryans kaymasını azaltarak modelin öğrenme süresini önemli ölçüde kısaltır.

Toplu normalizasyonun sağladığı başlıca avantajlar şu şekilde sıralanabilir [132, 135]:

- a. Kaybolan gradyan problemini önlemede etkilidir.
- b. Zayıf ağırlık başlangıcının etkilerini azaltır.
- c. Eğitim süresini kısaltır ve büyük veri kümeleriyle çalışırken avantaj sağlar.
- d. Hiperparametre hassasiyetini azaltır.
- e. Aşırı öğrenmeyi sınırlayıcı etkisiyle düzenleme görevini kısmen üstlenir.



Şekil 3.29: Model Uyum Örnekleri.

Tablo 3.2: Modellerin Sınırlamaları ve Potansiyel Çözümleri [136].

Model	Sınırlamalar	Potansiyel Çözümler
CNN	Yavaş hesaplama süresi, daha fazla parametre başlatma gereksinimi, nesne dönüşünü kodlayamama.	Göğüs röntgenleri ve BT taramaları gibi tıbbi görüntülerdeki karmaşık özellikleri yüksek doğruluk ve verimlilikle otomatik olarak çıkarma ve sınıflandırma yeteneğine sahiptir.
RNN	Gradyan kaybı nedeniyle öğrenme süreci yavaşlar, görüntülerin karmaşık özellikleriyle başa çıkamaz.	Zamana bağlı tıbbi verileri işleyerek zatürre'ye işaret eden uzun vadeli örüntüleri belirlemede etkilidir.
LSTM	Daha yüksek uzaysal karmaşıklık, aşırı uyum problemi.	Sıralı tıbbi verilerdeki uzun vadeli bağımlılıkları ve örüntüleri doğru şekilde yakalayıp zatürre tespiti ve analizinde başarılı sonuçlar elde edilmesini sağlar.

Tablo 3.3: Modellerin Sınırlamaları ve Potansiyel Çözümleri [136] (Tablonun Devamı).

GRU	Yavaş yakınsama hızı, düşük eğitim verimliliği, yüksek zaman karmaşıklığı, yetersiz uyum problemi.	Sıralı tıbbi verilerdeki zamansal bağımlılıkları ve örüntüleri etkili şekilde yakalayıp virüsün doğru şekilde tahmin ve teşhis edilmesini mümkün kılar.
CNN-RNN	Daha az özelliğe sahip küçük nesneleri işlerken düşük performans sergiler.	CNN tarafından çıkarılan görüntü özelliklerinin, LSTM ile yapılan sıralı veri analiziyle bütünleştirilmesi sayesinde sağlam ve doğru bir yöntem sunar.
Dikkat Mekanizması	Daha fazla parametre içerir, zaman karmaşıklığı yüksektir.	Tıbbi görüntülerde veya sıralı verilerde virüse işaret eden kritik özelliklere öncelik vererek doğruluğu artırır.
Kapsül Ağları	Hızlı hesaplama yapamaz, CNN'e kıyasla daha yüksek zaman ve uzay karmaşıklığına sahiptir.	Görüntü özellikleri arasındaki hiyerarşik ilişkileri etkili şekilde yakalayıp virüse ait ince örüntülerin hassas tanımlanmasını ve daha iyi genelleme yapılmasını sağlar.
Hibrit Modeller	Karmaşık yapıda olup, çalıştırılması maliyetlidir.	Farklı mimarilerin tamamlayıcı güçlü yönlerinden, sıralı veri işleminden ve ilgili özelliklere odaklanmadan faydalanarak tanısal doğruluk ve sağlamlığı artıran yüksek performans sağlar.

Transfer öğrenme yöntemi ve modelleri

Derin öğrenme modellerinin performansı, büyük ölçüde eğitim aşamasında kullanılan veri miktarına bağlıdır. Günümüzde internetin yaygınlaşmasıyla birlikte çeşitli alanlarda büyük veri setlerine erişim kolaylaşmış olsa da sağlık, uzay araştırmaları ve canlı organizmaların

izlenmesi gibi özel alanlarda hâlâ yeterli veri bulunmamaktadır. Bu tür durumlarda, sınırlı veri ile başarılı sonuçlar elde etmek amacıyla transfer öğrenme tekniği kullanılmaktadır. Makine öğrenmesinin bir alt disiplini olan transfer öğrenme, makinenin tıpkı insan gibi daha önce edinilen bilgileri yeni görevlerde kullanabilmesini sağlar. Bu yöntem, ağırlı sınıftan eğitilmesine kıyasla hem zaman hem de hesaplama maliyetlerini önemli ölçüde azaltmaktadır [137].

Transfer öğrenmeyi açıklamak için öğretmen-öğrenci ilişkisi etkili bir benzetmedir. Bu ilişki kapsamında, öğretmen (bilgi sahibi uzman), sahip olduğu bilgileri öğrenciye belirli bir düzen içerisinde aktarır [138]. Benzer biçimde, derin öğrenme modelleri büyük veri kümeleri üzerinde eğitilerek çeşitli özellikleri ve ağırlıkları öğrenir. Elde edilen bu ön bilgileri içeren ağırlı parametreleri, daha sonra benzer fakat daha az veri içeren başka bir problem üzerinde kullanılmak üzere transfer edilir. Böylece, yeni modelin eğitim süresi önemli ölçüde kısaltılır.

Transfer öğrenme en basit şekliyle, geniş kapsamlı bir veri kümesinde eğitilmiş modelin bilgi çıktılarının, daha küçük veri setlerine sahip farklı bir problem alanında yeniden kullanılmasını ifade eder [139]. Bu süreç genellikle dört temel aşamadan oluşmaktadır:

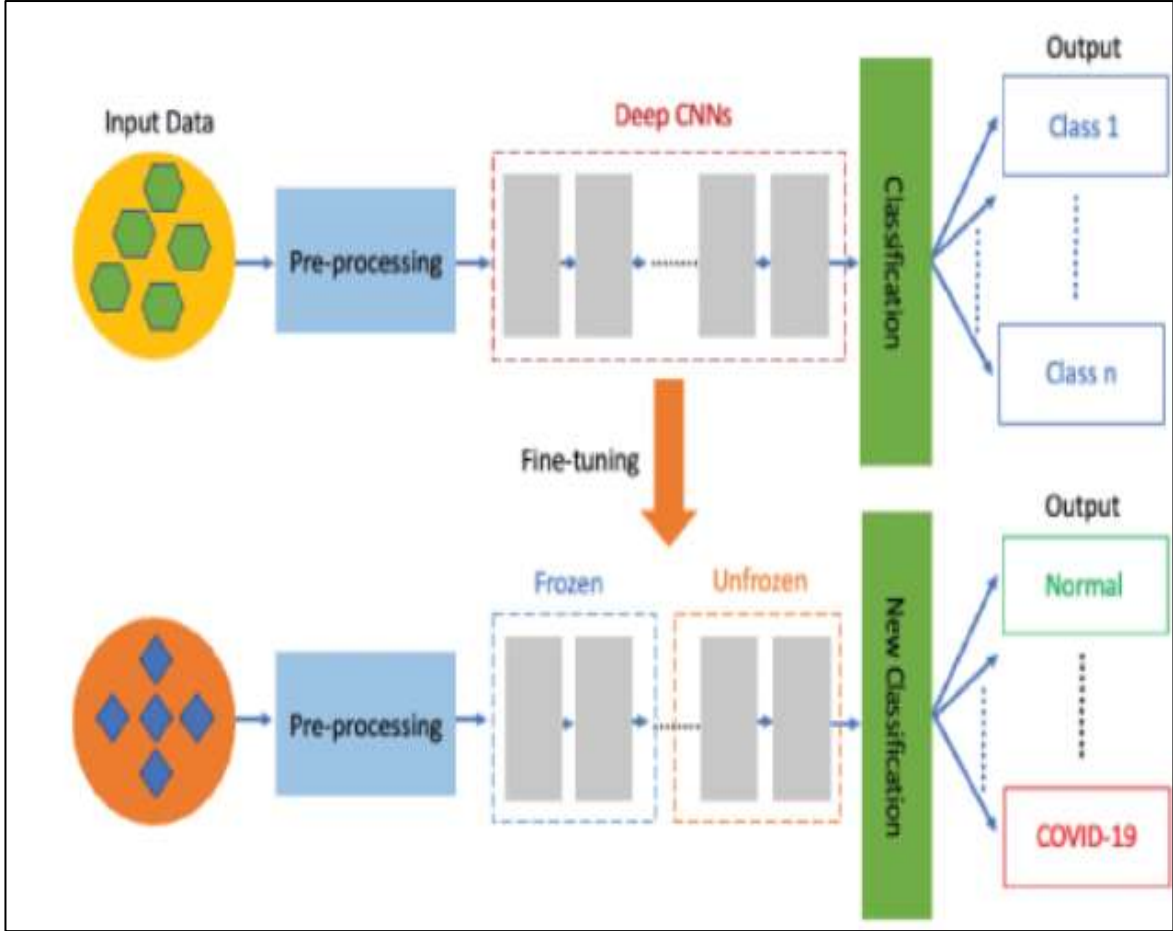
- a. Önceden eğitilmiş bir modelin katmanlarının seçilmesi,
- b. Bu katmanların bilgi kaybını önlemek amacıyla dondurulması,
- c. Dondurulan katmanların üzerine yeni, eğitilebilir katmanlar eklenmesi,
- d. Eklenen yeni katmanların, hedef veri kümesiyle eğitilmesi.

Transfer öğrenme yaklaşımları iki kategoriye ayrılabilir [139, 140]:

- a. İnce ayar tabanlı yöntemler (Fine Tuning): İnce ayar, belirli bir göreve yönelik derin öğrenme modelinin eğitilmesinde, başka bir alandan aktarılmış önceden öğrenilmiş ağırlıkların başlangıç noktası olarak kullanılması sürecidir. Bu yöntem, küçük etiketli veri kümeleri üzerinde başarılı modeller oluşturmak amacıyla, geniş ve etiketlenmiş veri kümeleriyle önceden eğitilmiş modellerden faydalanmaktadır.

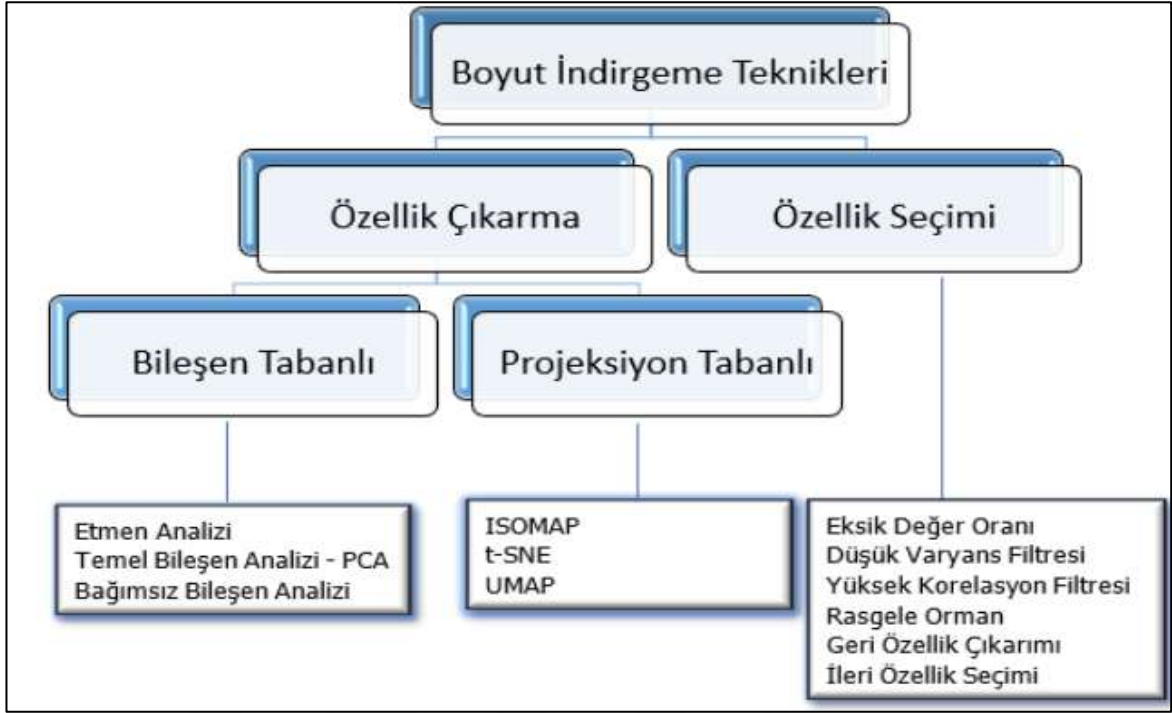
Derin evrişimli sinir ağı (DCNN) ilk olarak geniş kapsamlı bir veri kümesi üzerinde eğitilmektedir. DCNN'nin ilk katmanları daha genel nitelikteki özellikleri öğrenirken, son

katmanlar eğitim verisiyle daha özgül ilişkiler kurar. İnce ayar sürecinde, genellikle bu ilk genel katmanlar sabitlenir (dondurulur), ardından üst katmanlardan bazıları yeniden eğitmek üzere açılır ve yeni görev için uygun bir sınıflandırma katmanı modele entegre edilir. Dondurulmamış katmanlarla birlikte eklenen bu sınıflayıcı, hedef veri kümesi üzerinde sıfırdan eğitilir. Bu şekilde, mevcut bilgiler yeni problem alanına daha uygun hale getirilmiş olur.



Şekil 3.30: İnce Ayar Tabanlı Yöntemler (Fine Tuning).

b. Özellik Seçimi ve Özellik Çıkarma (Features Extraction): Her şeyden önce, boyut indirgeme sorunu gibi temel problemlere çözüm sunması nedeniyle, neden özellik çıkarma ve seçme işlemlerinin uygulandığını açıklamak gerekmektedir [141]. Peki, boyut indirgeme tam olarak nedir?



Şekil 3.31: Boyut İndirgeme Teknikleri.

Boyut indirgeme, yüksek boyutlu verilerin daha küçük bir temsiliyi oluşturmak suretiyle, makine öğrenmesi algoritmalarının daha az işlem süresi ve kaynak kullanımıyla çalışmasını sağlar. Bu da modelin doğruluk oranını artıran bir etkidir. Özellikle çok sayıda özelliğe sahip veri setlerinde modelin eğitimi ciddi zaman ve işlem gücü gerektirir. Yüksek boyutlu ortamlarda, örnekler genellikle uzayın uç noktalarına dağılır ve birbirlerinden uzaklaşır. Bu durum, algoritmaların öğrenmesini zorlaştırır ve "boyut laneti (curse of dimensionality)" olarak bilinir [141].

Boyut indirgeme işlemlerinin başlıca avantajları şunlardır:

- Aşırı öğrenmenin önüne geçerek modelin genelleme yeteneğini artırır.
- Verilerin daha etkili şekilde görselleştirilmesini sağlar.
- Özellikler arası çoklu doğrusal bağlantıyı ortadan kaldırır.
- Etmen analizine olanak tanır.
- Gürültüden arındırılmış daha temiz veriler elde edilmesine yardımcı olur.
- Görüntü sıkıştırma gibi alanlarda kullanılabilir.

g. Doğrusal olmayan verileri, doğrusal ayrılabilir forma dönüştürebilir.

Ancak bu işlemlerin bazı dezavantajları da vardır:

h. Boyut indirgeme sırasında bazı önemli veriler kaybolabilir.

i. Özellikle PCA gibi yöntemlerde hangi temel bileşenlerin dikkate alınacağı önceden bilinmeyebilir.

Özellik seçimi

Veri setindeki ilgili ve anlamlı özellikleri koruyup, ilgisiz veya gereksiz olanları dışarıda bırakma sürecidir [142]. Amaç, daha başarılı ve yalın bir model elde etmektir. Özellik seçimi üç ana başlık altında incelenebilir:

a. Filtreleme Yöntemleri: Veriler, istatistiksel ön değerlendirme yöntemleriyle filtrenir. En yaygın teknikler:

- i. Korelasyon analizi
- ii. Ki-kare (Chi-Square) testi
- iii. Varyans analizi (ANOVA)
- iv. Bilgi kazanımı (Information Gain)

b. Sarmalama Yöntemleri (Wrapper): Belirli bir makine öğrenmesi algoritması kullanılarak her bir özellik alt kümesinin model başarısına etkisi değerlendirilir. Bu yöntem, filtrelemeye kıyasla daha doğru sonuçlar verebilir fakat işlem karmaşıklığı daha yüksektir. Örnek yöntemler:

- i. İleri seçim
- ii. Geri eleme
- iii. Adım adım (çift taraflı) eleme

c. Gömülü Yöntemler: Bu teknikler, modelin eğitimi sırasında özelliklerin önem düzeyini değerlendirir. En sık kullanılan yöntemler:

- i. Lasso regresyon
- ii. Elastic Net
- iii. Ridge regresyon

Özellik çıkarımı

Yüksek boyutlu veri uzayını daha düşük boyutlu bir temsil ile ifade etme işlemidir. Bu yöntem, verideki tüm bilginin mümkün olduğunca korunarak daha az işlem gücü ile analiz yapılmasını sağlamak açısından oldukça etkilidir [143].

Yaygın olarak kullanılan bazı özellik çıkarım yöntemleri şunlardır:

- a. PCA (Temel Bileşen Analizi)
- b. LDA (Doğrusal Ayrım Analizi)
- c. Öz-PCA (Çekirdek PCA)
- d. QDA (İkinci Dereceden Ayrım Analizi)

Etmen (faktör) analizi

Gözlemlenen değişkenlerin altında yatan yapıları, yani gözlemlenemeyen gizli etmenleri keşfetmeyi amaçlayan keşifsel bir analiz yöntemidir. Bu analiz, değişkenler arasındaki ortak varyansı tek bir yapı altında toplayarak veri boyutunu azaltır. Genellikle pazarlama, reklamcılık, psikoloji ve finans gibi alanlarda tercih edilir [139].

Etmen analizi, istatistiksel doğrusal bir model temelinde çalışır. Değişkenlerdeki varyansı açıklamaya çalışır ve bu değişkenleri etmen adı verilen daha az sayıda bileşene indirger. Orijinal veriler, etmenlerin doğrusal birleşimi ve hata terimiyle ifade edilir. Bir etmen, birden fazla değişkenle ilişkilendirilebilir ve her biri toplam varyansın bir kısmını açıklar.

Bu analiz yöntemi, büyük veri setlerinde yer alan değişkenler arasındaki gizli ilişkileri belirlemeye yardımcı olur [141]. Farklı analizciler, aynı etmen çözümünden farklı yorumlar çıkarabilir; bu nedenle, etmenlerin tanımlanması ve adlandırılması alan bilgisi gerektirir.

Etmen analizinde varsayılan koşullar:

- a. Veri seti aykırı değer içermemelidir.

- b. Gözlem sayısı, çıkarılacak etmen sayısından fazla olmalıdır.
- c. Değişkenler arasında tam doğrusal bağımlılık bulunmamalıdır.
- d. Tüm değişkenler benzer varyansa sahip olmamalıdır.

Etmen analizi türleri:

- a. *Keşifsel Etmen Analizi (EFA)*: Gözlemler ile etmenler arasında doğrudan ilişki kurarak yapıyı ortaya çıkarır.
- b. *Onaylayıcı Etmen Analizi (CFA)*: Önceden oluşturulan varsayımları test etmek amacıyla kullanılır.

Etmen analizinin işleyiş adımları:

- a. *Etmen Çıkarımı*: PCA veya ortak etmen analizi gibi yöntemlerle etmen sayısı belirlenir.
- b. *Etmen Dönüştürme (Rotasyon)*: Daha anlamlı yorumlar elde etmek için etmenler, Varimax, Quartimax veya Promax gibi rotasyon teknikleriyle yeniden düzenlenir.

Etmen analizine başlamadan önce veri setinin bu yöntem için uygun olup olmadığını değerlendirmek gerekir. Bu amaçla iki temel test kullanılır [141]:

- a. *Bartlett's Test*: Değişkenlerin korelasyonlu olup olmadığını belirler. Test sonucunun anlamsız çıkması durumunda analiz uygulanmamalıdır.
- b. *Kaiser-Meyer-Olkin (KMO) Testi*: Veri setinin etmen analizine uygunluğunu ölçer. Değerler 0 ile 1 arasında değişir; 0.6'nın altındaki değerler yetersiz olarak kabul edilir.

Etmen sayısını belirleme yöntemleri:

- a. *Kaiser Kriteri*: Özdeğeri 1'den büyük olan etmenlerin seçilmesi gerektiğini ifade eder.
- b. *Grafiksel Yöntem (Scree Plot)*: Özdeğerlerin grafiksel olarak gösterimi yapılır ve eğrinin dirsek noktası dikkate alınarak etmen sayısı belirlenir.

Temel bileşen analizi (PCA)

İstatistiksel bir yöntem olarak, birbiriyle ilişkili olan çok sayıda özelliği, birbirinden bağımsız (korelasyonsuz) yeni özelliklere dönüştüren ortogonal bir dönüşüm tekniğidir. Bu dönüşüm sonucu ortaya çıkan yeni boyutlar, *temel bileşenler* olarak adlandırılır. PCA, veri

keşfi ve tahmine dayalı modelleme süreçlerinde sıklıkla başvuru alan bir boyut indirgeme yöntemidir [142].

PCA'nın temel hedefi, orijinal veri setindeki toplam varyansı en yüksek düzeyde temsil edebilecek optimal sayıda bileşenin belirlenmesidir. Çünkü varyans, sınıflar arasındaki ayrımı ifade ederken aynı zamanda boyut indirgemeyi de mümkün kılar. Bu yöntem, her bir özelliğin varyansını dikkate alarak çalışır.

PCA'nın uygulama alanları oldukça geniştir; görüntü işleme, öneri sistemleri (film tavsiyeleri vb.) ve iletişim kanallarında enerji dengesini sağlama gibi çeşitli uygulamalarda kullanılmaktadır [138].

PCA'dan en verimli şekilde faydalanmak için algoritmanın iki aşamalı biçimde kullanılması önerilir: ilki temel bileşen sayısının belirlenmesine, ikincisi ise dönüşümün gerçekleştirilmesine yöneliktir. Bu işlemler öncesinde, verinin uygun şekilde standartlaştırılması gerekmektedir. Uygulama adımları aşağıdaki gibidir:

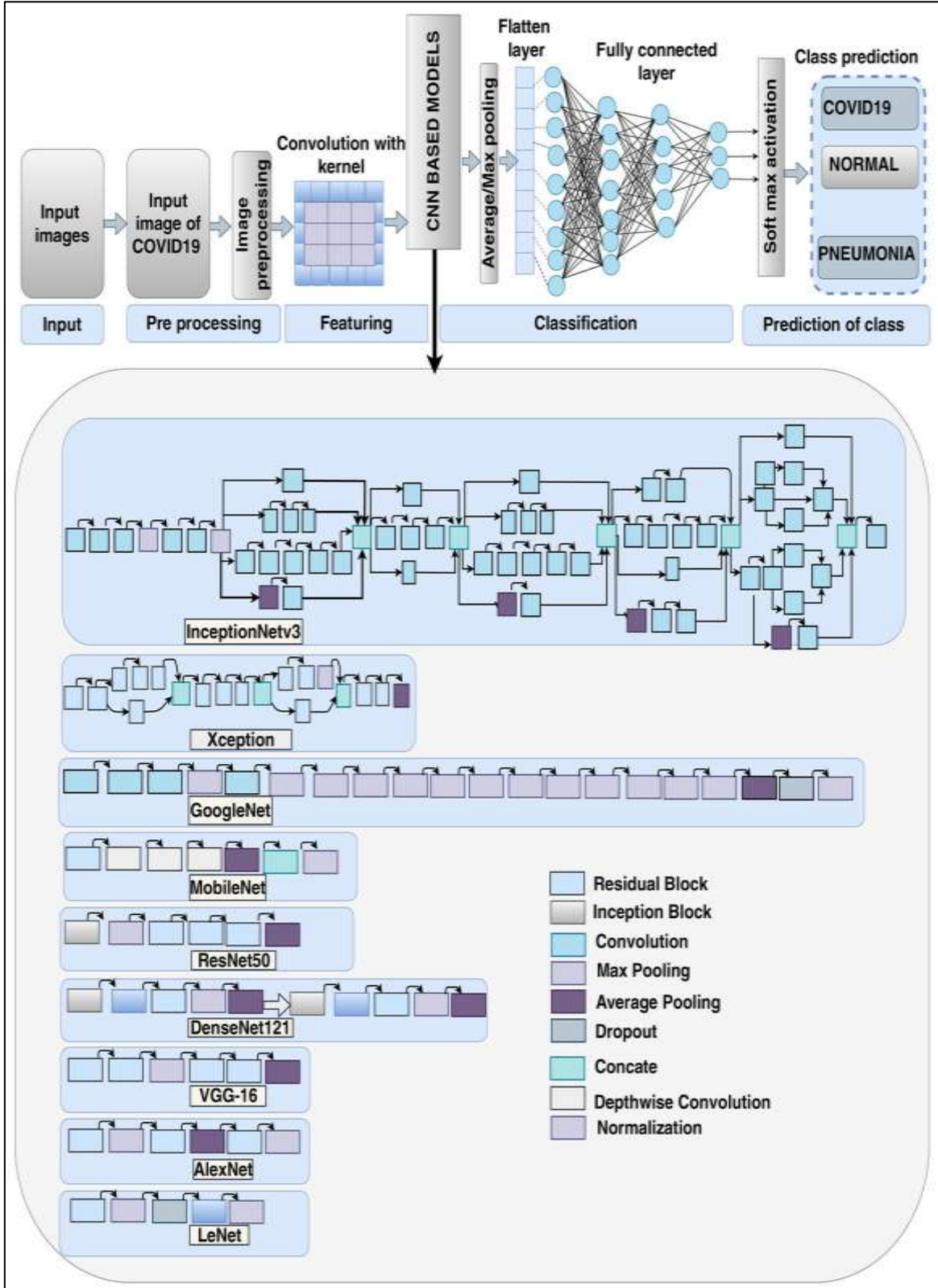
- a. **Özelliklerin Standartlaştırılması:** PCA yöntemi, özelliklerin değer aralıklarına duyarlı olduğundan, tüm özelliklerin aynı ölçekte olması sağlanmalıdır. Özellikler arasında ciddi varyans farkı yoksa ya da PCA kovaryans matrisi kullanılarak uygulanacaksa standartlaştırma zorunlu değildir, fakat her durumda uygulanması tavsiye edilir [139].
- b. **Temel Bileşenlerin Belirlenmesi:** PCA ilk olarak tüm özellikler üzerinde çalıştırılarak her bileşenin açıklayabildiği varyans hesaplanır. Bu işlem sonucunda optimum bileşen sayısı belirlenir. Bu aşamada veri dönüştürülmez; yalnızca varyans oranları değerlendirilir.
- c. **Veri Dönüşümünün Gerçekleştirilmesi:** Önceden belirlenen bileşen sayısına göre dönüşüm yapılır. Bunun için `n_components` parametresi belirlenerek PCA tekrar çalıştırılır. `pca.fit_transform()` komutu hem bileşenlerin hesaplanmasını hem de dönüşüm işlemini birlikte gerçekleştirir.

Etmen Analizi ve PCA Karşılaştırması [142]:

- a. PCA, toplam varyansı en iyi biçimde açıklayan bileşenleri bulmayı hedeflerken; Etmen Analizi, değişkenler arasındaki ortak varyansı açıklamayı amaçlar.
- b. PCA'da bileşenler birbirine ortogonaldır; Etmen Analizi'nde ise etmenlerin birbirinden bağımsız olması gerekmez.

- c. PCA bileşenleri doğrudan gözlenen verilerin bileşiminden elde edilirken, Etmen Analizi'nde gözlemler etmenlerin doğrusal bileşimidir.
- d. PCA bileşenleri genellikle yorumlamaya açık değildir; buna karşın, Etmen Analizi'ndeki etmenler yorumlanabilir ve anlamlı etiketler ile tanımlanabilir.
- e. PCA, temel olarak bir boyut indirgeme tekniğidir; Etmen Analizi ise gizli değişkenlerin analizi için kullanılan bir modelleme yaklaşımıdır.
- f. PCA gözlemsel bir yöntemdir; Etmen Analizi ise teorik modelleme içeren, açıklayıcı bir tekniktir.





Şekil 3.32: Transfer Öğrenme Yönteminin Operasyonel Diyagramı: InceptionNetV3, XceptionNet, GoogleNet, MobileNet, ResNet50, DenseNet121, VGG-16, AlexNet ve LeNet [144].

CNN Tabanlı Mimari Yapıları

- a. GoogleNet: 2014 ILSVRC yarışmasının kazananı olan GoogleNet (Inception-V1 olarak da bilinir) [145], yüksek doğruluk seviyelerine ulaşırken hesaplama maliyetini azaltmayı hedeflemiştir. Bu mimari, CNN yapısına çok ölçekli evrimsel dönüşümleri entegre ederek bir ilki gerçekleştirmiştir. Özellik çıkarımı sırasında birleştirme, ayrıştırma ve dönüştürme işlemleriyle çalışır. GoogleNet, her bloğun içerisinde çeşitli boyutlardaki evrim filtrelerinden seçim yapma imkânı tanıyan Inception modüllerini temel yapı taşı olarak kullanır.
- b. DenseNet: Huang ve arkadaşları tarafından 2017 yılında geliştirilen *Yoğun Evrimsel Ağ* (DenseNet), görsel nesne tanıma alanında yenilikçi bir yaklaşımdır. Bu model, ileri beslemeli bir yapı kullanarak her katmanı, ağ içindeki tüm diğer katmanlara doğrudan bağlamaktadır. Böylelikle önceki tüm katmanlardan gelen özellik haritaları sonraki katmanlara giriş olarak iletilir. Yapılan araştırmalar, ağın erken katmanları ile derin katmanları arasındaki doğrudan bağlantıların, modelin genel başarımını artırdığını ortaya koymuştur [145], [146]. Bu yoğun bağlantı yapısı sayesinde DenseNet, geleneksel CNN'lere göre daha az parametre ile çalışabilir; çünkü tekrar eden özellikleri yeniden öğrenmeye gerek duymaz. Ağ, sabit boyutlu özellik haritalarının yer aldığı bloklara ayrılırken, bu bloklar içerisindeki filtre sayısı değişkenlik gösterebilir. DenseNet, parametre sayısında ciddi azalma, öğrenilen özelliklerin yeniden kullanımı ve gradyan kaybının önlenmesi gibi avantajlar sağlar.
- c. AlexNet: Derin öğrenmenin başlangıç dönemlerinde CNN'ler genellikle el yazısı karakter tanıma gibi dar kapsamlı görevlerle sınırlıydı. Ancak AlexNet, görüntü tanıma ve sınıflandırma problemlerinde çığır açan sonuçlar elde etmiştir [139]. Krizhevsky ve çalışma arkadaşları tarafından önerilen bu mimari [140], derinlik artırımı ve farklı parametre optimizasyon tekniklerinin uygulanmasıyla CNN'lerin öğrenme yeteneğini kayda değer ölçüde geliştirmiştir.
- d. ZefNet: Zeiler ve Fergus tarafından 2013 yılında geliştirilen DeconvNet [139], çok katmanlı de-evrimsel yapısıyla tanıtılmış ve sonrasında ZefNet olarak evrilmiştir. Bu yöntem, ağın iç yapısını nicel olarak görselleştirmek amacıyla tasarlanmıştır. ZefNet, CNN içindeki nöron aktivasyonlarını anlamaya çalışarak eğitim sürecini izlemeye

olarak tanır. Aynı zamanda, modeldeki eksiklikleri veya darboğazları tespit etme amacıyla çıktıların analizini içerir.

- e. VGG (Görsel Geometri Grubu): CNN'lerin görüntü tanıma alanındaki başarısının ardından, Simonyan ve Zisserman, derin sinir ağları için basit ama etkili bir mimari yaklaşım geliştirmiştir. Bu yapı, Görsel Geometri Grubu (VGG) adıyla literatüre geçmiştir. VGG mimarisi, ZefNet [139] ve AlexNet gibi önceki modellerden farklı olarak, daha derin bir yapı sunmakta ve toplamda 19 katman içermektedir [140]. Bu derin yapı, ağıın öğrenme kapasitesini artırmakta ve daha karmaşık örüntüleri modellemesine olanak sağlamaktadır.
- f. ResNet (Artık Ağ): Szegedy ve arkadaşları tarafından önerilen ResNet mimarisi [140], VGGNet'e göre sekiz kat daha derin olacak şekilde tasarlanmıştır. Bu ağ, derin mimarilerde sıklıkla karşılaşılan kaybolan gradyan problemini ortadan kaldırmayı hedeflemiştir. ResNet, "artık bloklar" olarak adlandırılan modüller kullanır ve bu bloklar, girdiyi hem işlenmiş haliyle hem de doğrudan (değiştirilmeden) birleştirerek çıktıya dahil eder. Bu yaklaşım, katmanlar arasındaki geçişi kolaylaştırır ve öğrenmeyi daha kararlı hale getirir [139]. Ayrıca, baypas (skip) bağlantıları sayesinde birden fazla katman atlanabilir ve bu da mimarinin önemli bir özelliğidir. Ağ yapısı beş evrişim bloğundan oluşur ve SGD ile geri yayılım yöntemi kullanılarak eğitilebilir.
- g. Inception-V3: Szegedy ve çalışma arkadaşları [139], Inception-V3 mimarisini geliştirirken, ağıın derinliği artırılırken hesaplama yükünü minimuma indirmeyi amaçlamışlardır. Model, kalıntı bağlantılarını kullanarak eğitimin daha hızlı ve etkili şekilde gerçekleşmesini sağlamaktadır. Derin öğrenme modellerinin büyüklüğü arttıkça aşırı öğrenmeye daha yatkın hale gelmesi ve parametre sayısındaki artışın hesaplama gereksinimlerini de artırması temel bir sorundur. Inception-V3, bu sorunlara çözüm getirecek şekilde yapılandırılmıştır [140]. Böylece, modelin hem genişliği hem de derinliği artırılırken toplam işlem maliyeti sabit tutulabilir.
- h. MobileNet: MobileNet, Howard ve arkadaşları tarafından 2017 yılında geliştirilen ve derin sinir ağlarının temel yapı taşı olarak kullanılan, derinlik bazlı ayrılabilir evrişim katmanlarıyla oluşturulan kompakt bir CNN mimarisidir [137]. Bu mimari, özellikle düşük hesaplama gücü gereksinimi olan mobil ve gömülü cihazlar için tasarlanmıştır.

MobileNetV2, MobileNetV1'in geliştirilmiş versiyonudur ve verimli öğrenme için iki farklı blok tipi kullanır: tek adımlı artık (residual) bloklar ve boyut küçültme amacıyla kullanılan iki adımlı bloklar. Ayrıca, ağıın doğrusal katmanlar aracılığıyla çok fazla bilgi kaybı yaşamadan öğrenmesini sağlamak adına “doğrusal darboğaz” yapıları kullanılır. Bu darboğaz katmanları, modelin ara katmanlarındaki girdileri ve çıktıları daha verimli şekilde temsil eder. İç katmanlar, düşük seviyeli özellikleri (örneğin pikselleri), daha üst düzey tanımlayıcılara (örneğin görüntü kategorileri) dönüştürmeye yardımcı olur. Ek olarak, bu darboğazlar arasında yer alan kısa yol bağlantıları bilgi akışını güçlendirir. Bu çalışmada, önceden eğitilmiş bir MobileNetV2 modeli kullanılmıştır. Modelin zatiire sınıflandırması görevine uyarlanması için transfer öğrenmesi tekniğinden yararlanılmıştır.

- i. Highway Network (Otoyol Ağı): Sinir ağlarının derinliğini artırmak genellikle daha yüksek temsil gücü ve daha iyi performans anlamına gelse de bu durum eğitim sürecini zorlaştırabilir. Derin yapılar, özellikle düşük seviyeli katmanlarda gradyanların sönümlenmesi gibi problemlere neden olabilir. Bu sorunun çözümü amacıyla Srivastava ve çalışma arkadaşları, 2015 yılında “Highway Network” adlı yeni bir mimari önermiştir [135]. Bu yaklaşım, bilgi geçişini kolaylaştırmak için çapraz bağlantılara dayalıdır. Otoyol Ağı'ndaki bilgi akışı, katmanlar içindeki iki geçit (gate) yapısının koordinasyonu ile kontrol edilir; böylece bilgi engellenmeden derin katmanlara aktarılabilir.

3.5.2 Performans Değerlendirme Metrikleri

Sınıflandırma algoritmalarının başarı düzeyini ölçmek için yaygın olarak kullanılan yöntemlerden biri karışıklık matrisi (Confusion Matrix) ya da diğer adıyla hata matrisidir [147]. Karışıklık matrisi, anlaşılır yapısı ve uygulama kolaylığı nedeniyle özellikle kötü amaçlı yazılımların tespitinde sıkça tercih edilmektedir. Bu çalışmada da uygulanan sınıflandırma algoritmalarının doğruluklarını analiz etmek amacıyla karışıklık matrisinden yararlanılmıştır. Söz konusu matris, sınıflandırıcı tarafından tahmin edilen sonuçlarla gerçek etiketler arasındaki ilişkiyi inceleyerek modelin performansını ölçmeye yardımcı olmaktadır.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Şekil 3.33: Hata Matrisi.

- a. TP (Gerçek Pozitifler): Gerçek değeri 1 (evet) olan bir örneğin, model tarafından da 1 (evet) olarak doğru biçimde tahmin edilmesidir.
- b. TN (Gerçek Negatifler): Gerçek değeri 0 (hayır) olan bir örneğin, model tarafından yine doğru bir şekilde 0 (hayır) olarak tahmin edilmesidir.
- c. FP (Yanlış Pozitifler): Gerçek değeri 0 (hayır) olan bir örneğin, model tarafından hatalı biçimde 1 (evet) olarak sınıflandırılmasıdır.
- d. FN (Yanlış Negatifler): Gerçek değeri 1 (evet) olan bir örneğin, model tarafından hatalı olarak 0 (hayır) şeklinde sınıflandırılmasıdır.

Bu çalışmada yapılan deneyler kapsamında karışıklık matrisinden elde edilen veriler doğrultusunda, Doğruluk (Accuracy), Hassasiyet (Precision), Duyarlılık (Recall), F1-Skoru (F1-Score), ROC eğrisi altındaki alan (ROC Area Under the Curve – Roc_Auc) ve K-turlu çapraz doğrulama (Stratified K-Fold) gibi temel performans ölçütleri aşağıda açıklanmıştır.

3.5.2.1 Doğruluk

Doğruluk metriği, bir sınıflandırma modelinin tüm örnekleri ne ölçüde doğru sınıflandırabildiğini gösterir. Bu oran; doğru pozitif (TP) ve doğru negatif (TN) sınıfların toplamının, tüm örnek sayısına (TP+TN+FP+FN) bölünmesiyle hesaplanmaktadır [148].

Ne zaman kullanılır?

Doğruluk oranı, sınıfların dengeli dağıldığı veri setlerinde etkili bir ölçüttür. Ancak sınıf dengesizliğinin mevcut olduğu durumlarda yanıltıcı sonuçlar doğurabilir. Örneğin; kredi kartı sahteciliği gibi, yalnızca %1 oranında pozitif sınıfın bulunduğu bir veri setinde, tüm

örnekleri “geçerli” olarak tahmin eden bir model %99 doğruluk gösterebilir. Bu yüksek oran, gerçek anlamda güvenilir bir başarıyı yansıtmaz.

3.5.2.2 Hassasiyet

Hassasiyet, modelin pozitif sınıf olarak tahmin ettiği örnekler içerisinde, gerçekten pozitif olanların oranını ifade eder [149].

Ne zaman kullanılır?

Özellikle yanlış pozitif tahminlerin yüksek maliyetli olduğu senaryolarda tercih edilmelidir. Örneğin; kredi limit düşürme kararlarında, yanlış tahminler müşteri kaybına yol açabileceğinden, yüksek hassasiyet tercih edilir.

3.5.2.3 Duyarlılık

Duyarlılık, gerçekte pozitif olan örneklerin ne kadarının doğru şekilde tespit edilebildiğini gösterir [149].

Ne zaman kullanılır?

Hatalı negatif sınıflandırmaların kritik olduğu durumlarda önemlidir. Örneğin, COVID-19 testi pozitif olan bir bireyin yanlışlıkla negatif olarak sınıflandırılması, önemli sağlık risklerine yol açabilir. Bu nedenle, sağlık uygulamalarında genellikle duyarlılığın yüksek olması beklenir.

3.5.2.4 F1-ölçütü

F1 skoru, modelin hassasiyet ve duyarlılık değerlerinin harmonik ortalamasını alarak, bu iki ölçüt arasında denge kurar [148].

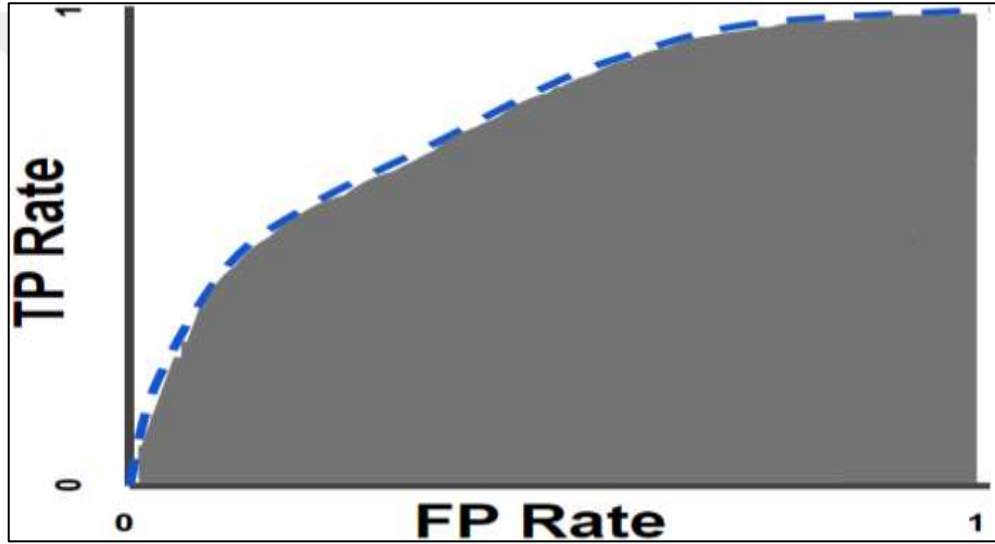
Ne zaman kullanılır?

Hassasiyet ve duyarlılığın birlikte önemli olduğu durumlarda, örneğin sahte işlemlerin hem isabetli hem de kapsamlı bir şekilde yakalanmasının istendiği kredi kartı dolandırıcılığı senaryolarında kullanılır. Yüksek F1 skoru, modelin genel başarımını temsil eder.

3.5.2.5 ROC eğrisi altındaki alan

Sınıflandırma modellerinin performansı, ROC eğrisi altında kalan alanın büyüklüğü ile ölçülmektedir. ROC (Receiver Operating Characteristic) eğrisi, iki boyutlu bir grafik olup, y ekseninde yer alan Gerçek Pozitif Oranı (True Positive Rate- TPR) ile x eksenindeki Yanlış Pozitif Oranı (False Positive Rate- FPR) arasındaki ilişkiyi göstermektedir [148].

ROC eğrisi, modelin farklı sınıflandırma eşik değerlerine göre TPR ve FPR'yi nasıl değiştirdiğini gösterir. Sınıflandırma eşiği düşürüldükçe, daha fazla veri pozitif sınıfa dahil edilir; bu durum hem Gerçek Pozitif hem de Yanlış Pozitif oranlarının artmasına neden olur. Aşağıda gösterilen şekil, tipik bir ROC eğrisini temsil etmektedir.



Şekil 3.34: ROC Eğrisi.

3.5.2.6 K-turlu çapraz doğrulama

Makine öğrenmesi algoritmalarının hata oranlarını daha doğru değerlendirebilmek adına, veri seti genellikle eğitim ve test kümeleri olarak ayrılmaktadır. Bu ayrımın yapılmasında çeşitli yöntemler kullanılmakla birlikte, bu çalışmada Python'un Scikit-learn kütüphanesinin sunduğu “Stratified K-Fold” yöntemi tercih edilmiştir. Literatürde yaygın olarak kullanılan 10 turlu çapraz doğrulama yöntemi bu çalışmada da uygulanmıştır [150].

Bu yöntem, veri setini eşit büyüklükte k alt kümeye ayırır. İlk turda ilk alt küme test için ayrılırken, kalan k-1 alt küme modelin eğitimi için kullanılır. İkinci turda bu işlem, ikinci alt

küme test verisi olarak seçilerek tekrarlanır. Bu süreç k kez devam eder ve her küme bir kez test verisi olarak kullanılır [147].

Çapraz doğrulama yöntemi, modelin başarımının yalnızca belirli bir eğitim verisine bağımlı olmadığını göstermek ve daha genel bir performans ölçümü elde etmek amacıyla kullanılır. Bu yöntem sayesinde, eğitim verileri değişse bile modelin tutarlı sonuçlar üretip üretmediği değerlendirilebilmektedir [148].

3.5.3 Kullanılan Yöntemler

3.5.3.1 Veri ön işleme

Veri ön işleme aşaması, modelin doğruluğunu ve genel performansını artırmak için oldukça kritik bir adımdır. Bu süreçte aşağıdaki işlemler gerçekleştirilmiştir:

- a. Eksik veya Hatalı Verilerin Temizlenmesi: Veri setinde eksik veya hatalı değerlerin bulunmaması için gerekli temizlik işlemleri yapılmıştır. Bu adım, modelin eğitimi sırasında herhangi bir veri kaybı veya hatalı sonuçlar alınmasını engellemektedir.
- b. Görsellerin Yeniden Boyutlandırılması ve Normalize Edilmesi: Görseller, modelin gereksinimlerine uygun hale getirilmek için 224x224 piksel boyutlarına yeniden boyutlandırılmıştır. Ayrıca, piksel değerleri [0, 1] aralığına normalize edilmiştir. Bu işlem, modelin daha hızlı ve doğru öğrenmesini sağlamaktadır, çünkü büyük farklılıklar gösteren piksel değerleri, öğrenme sürecinde zorluklara yol açabilir.
- c. Kontrast Artırma ve Gürültü Giderme İşlemleri: Görseller üzerinde kontrast artırma veya gürültü giderme işlemleri yapılmamıştır. Bunun yerine, görsellerin doğal yapıları ve özellikleri bozulmadan doğrudan kullanılması tercih edilmiştir.
- d. Veri Artırma (Data Augmentation): Modelin daha genellenebilir hale gelmesi amacıyla, eğitim veri setinde görüntü döndürme, ölçekleme ve parlaklık ayarlama gibi veri artırma teknikleri kullanılmamıştır. Bu tür veri artırma yöntemleri, modelin overfitting (aşırı uyum) riskini azaltmak için genellikle tercih edilse de bu çalışmada kullanılan modelin yeterli veriye sahip olduğu düşünülerek bu adım atlanmıştır.

3.5.3.2 Özellik çıkarımı ve seçimi

- a. Görsellerden Çıkarılan Özellikler: Bu çalışmada, özellik çıkarımı doğrudan Konvolüsyonel Sinir Ağı (CNN) katmanları tarafından yapılmıştır. CNN yapıları, görsel verilerden önemli görsel özellikleri (kenarlar, dokular, desenler) otomatik olarak öğrenebilecek şekilde tasarlanmıştır. Görüntülerin her katmanda özellik çıkarımı yapılır ve bu özellikler sonraki katmanlarda daha soyut seviyelerde işlenir.
- b. Özellik Seçimi Yöntemleri: Özellik seçimi için geleneksel yöntemler (örneğin: PCA veya LDA) kullanılmamıştır. Bunun yerine, derin öğrenme mimarisi olan CNN, özellik seçimi için gerekli tüm işlemleri içeren end-to-end bir öğrenme süreci sağlamaktadır. CNN katmanları, özelliklerin önemini öğrenir ve gereksiz özelliklerin ağda işlenmesine gerek kalmaz.
- c. Özelliklerin Model Üzerindeki Etkisi: Derin öğrenme modelinde özellikler, görsel verilerden otomatik olarak çıkarılmaktadır. Özelliklerin doğru seçilmesi ve modelin bu özellikler üzerinde derinlemesine öğrenme yapması, modelin genel performansını doğrudan etkilemektedir. Bu çalışma, CNN'in görsellerden etkili özellikler çıkararak modelin başarısını artırabileceğini göstermektedir.

3.5.3.3 Modelin eğitimi ve hiperparametreler

Modelin eğitimi, aşağıdaki hiperparametreler kullanılarak gerçekleştirilmiştir:

- a. Optimizasyon Algoritması: Modelin eğitiminde Adamax optimizasyon algoritması kullanılmıştır. Bu algoritma, öğrenme oranını dinamik olarak ayarlayarak daha hızlı ve etkili bir eğitim süreci sağlar.
- b. Kayıp Fonksiyonu: Categorical Cross-Entropy kayıp fonksiyonu kullanılarak çok sınıflı sınıflandırma yapılmıştır. Bu fonksiyon, modelin sınıflar arasındaki hataları minimize etmeye çalışır.
- c. Eğitim ve Doğrulama Veri Seti Oranı: Veri seti, %80 eğitim, %10 doğrulama ve %10 test verisi olarak üçe ayrılmıştır. Bu oran, modelin doğru genelleme yapabilmesi için uygun bir veri ayırma oranıdır.

- d. Epoch Sayısı, Öğrenme Oranı, Batch Size: Modelin eğitim süreci 13 epoch boyunca devam etmiştir. Batch size değeri 4 olarak belirlenmiş ve her batch'te 4 görüntü işlenmiştir. Öğrenme oranı ise 0.001 olarak ayarlanmıştır.

3.5.3.4 Derin öğrenme modeli

Bu çalışmada, zatürre tespiti için Konvolüsyonel Sinir Ağı (CNN) mimarisi kullanılmaktadır. Aşağıda kullanılan modelin yapı taşları detaylı olarak açıklanmıştır:

- a. CNN Mimarisi: Kullanılan modelde, bir dizi konvolüsyonel (Conv2D) ve havuzlama (MaxPooling2D) katmanı bulunmaktadır. Konvolüsyonel katmanlar, görsellerdeki temel özellikleri öğrenirken, havuzlama katmanları bu özelliklerin boyutlarını küçültür ve daha soyut bir temsile indirger. Modelin derinliği arttıkça, model daha karmaşık ve soyut özellikleri öğrenmektedir.
- b. Katman Türleri: Kullanılan katmanlar, sırasıyla şunlardır:
- i. Conv2D: Görsellerin temel özelliklerini öğrenmek için kullanılan konvolüsyonel katmanlar. Her biri activation="relu" ile çalışmaktadır.
 - ii. MaxPooling2D: Özellik haritalarının boyutlarını küçültmek için kullanılan havuzlama katmanları.
 - iii. Flatten: Çok boyutlu çıktıları tek boyutlu hale getirir.
 - iv. Dense (Tam Bağlantılı) Katmanlar: Modelin öğrenmesini ve karar verme süreçlerini gerçekleştiren katmanlar. Son sınıflandırma işlemi için kullanılmış; son katmanda softmax aktivasyon fonksiyonu uygulanarak sınıf olasılıkları hesaplanır.
- c. Topluluk Öğrenme ve Transfer Öğrenme: Bu modelde topluluk öğrenme veya transfer öğrenme yöntemleri kullanılmamıştır. Model sıfırdan eğitilmiştir ve yalnızca mevcut veri setiyle optimizasyon yapılmıştır.
- d. Modelin Giriş-Çıkış Yapıları ve Aktivasyon Fonksiyonları: Modelin girdi boyutu 224x224 piksellik ve 3 renk kanallı görsellerdir. Son katmanda softmax aktivasyon fonksiyonu kullanılarak, her iki sınıf (NORMAL ve PNEUMONIA) için olasılık değerleri tahmin edilmiştir.

3.5.3.5 Performans ve değerlendirme kriterleri

Modelin performansı, doğrulama ve test veri setlerinde yapılan değerlendirmeler ile ölçülmüştür. Performans kriterleri şu şekilde belirlenmiştir:

- a. Kayıp ve Doğruluk (Loss & Accuracy): Eğitim, doğrulama ve test veri setleri üzerinde `model.evaluate()` metodu ile kayıp ve doğruluk değerleri hesaplanmıştır. Modelin tahminlerinin doğruluğu, accuracy metriği ile değerlendirilmiştir. Eğitim, doğrulama ve test veri setlerinde elde edilen doğruluk oranları modelin genel performansını yansıtmaktadır.
- b. Karışıklık Matrisi (Confusion Matrix): Modelin tahminleri ile gerçek etiketler karşılaştırılarak `confusion_matrix` fonksiyonu ile karışıklık matrisi oluşturulmuş ve yorumlanmıştır. Modelin doğruluğunu daha ayrıntılı analiz etmek için karışıklık matrisi kullanılmıştır. Bu matris, doğru ve yanlış sınıflandırmalar arasındaki ilişkiyi görselleştirir ve modelin hangi sınıflarda hatalı tahminler yaptığını gösterir.
- c. Precision, Recall ve F1-Score: Modelin başarımını değerlendirmek amacıyla `classification_report` fonksiyonu ile Modelin precision, recall ve F1-score değerleri, sınıflandırma başarısını daha derinlemesine analiz etmek için hesaplanmıştır.
- d. ROC Eğrisi ve AUC Skoru: ROC eğrisi çizilmiş ve altında kalan alan (AUC) hesaplanarak modelin sınıfları ayırt etme kabiliyeti görselleştirilmiştir. Bu, modelin pozitif ve negatif sınıfları ayırt etme gücünü görselleştirir.

3.6 YAŞANAN VE YAŞANABİLECEK ZORLUKLAR

Bu bölümde derin öğrenme ile göğüs röntgeni görüntülerinden zatürre tespitinde araştırmacıların karşılaştıkları bazı zorlukları ele alıyoruz.

3.6.1 Veri Zorlukları

Tablo 3.4: Veri Zorlukları [151-156].

Zorluk	Açıklama
Sınıf Dengesizliği	Veri kümesindeki zatürre vakalarının azlığı, modelin azınlık sınıfı üzerinde düşük performans göstermesine neden olur. Yeniden örnekleme (örneğin SMOTE, Borderline SMOTE) ve maliyet fonksiyonlarına ağırlık ekleme bu sorunu azaltmak için kullanılır.
Veri Kalitesi ve Etiketleme	Etiketli radyografi görüntülerinin yetersizliği, özellikle yüksek kaliteli ve doğru etiketli örneklerin azlığı, model eğitiminin güvenilirliğini düşürür. Ayrıca, etiketleme süreci uzmanlık gerektirdiğinden zaman alıcı ve maliyetlidir.
Veri Artırma Yöntemleri	Ters çevirme, döndürme gibi teknikler ile sınırlı veri artırılır. Ayrıca MoEx gibi gelişmiş yöntemler de kullanılmıştır. Transfer öğrenme ile önceden eğitilmiş modellerden faydalanılarak daha az veriyle daha iyi sonuçlar elde edilmiştir.
Alternatif Öğrenme Yöntemleri	Az atışlı ve sıfır atışlı öğrenme gibi yaklaşımlar, veri yetersizliği problemini azaltmak için etkili yöntemler olarak önerilmektedir.
Verilerin Yorumlanabilirliği	Derin öğrenme (DL) teknikleri çoğu zaman "kara kutu" olarak değerlendirilse de aslında belirli yöntemlerle yorumlanabilir özellikler taşımaktadır. DL modellerinin özellikle biyoenformatik gibi alanlarda yorumlanabilirliğini sağlama ihtiyacı giderek artmaktadır. Model yorumlanabilirliği, uygulandığı bağlama göre farklı anlamlar kazanabilmekte ve bu nedenle açıklanabilir yapay zekâ (XAI) uygulamaları, tıbbi karar destek sistemlerinin güvenilirliğini artırmada kilit rol oynamaktadır.

3.6.2 Model Zorlukları

Tablo 3.5: Model Zorlukları [151-156].

Zorluk Alanı	Açıklama
Veri Miktarı Gereksinimi	Derin öğrenme modelleri, yüksek doğruluk oranlarına ulaşmak için geniş veri kümelerine ihtiyaç duyar. Alternatif yöntemlere kıyasla daha fazla veri gereksinimi, özellikle sağlık verileri gibi erişimi sınırlı alanlarda önemli bir sınırlılık oluşturur.
Hesaplama ve Uzay Karmaşıklığı	Büyük veri kümeleri ve derin mimariler yüksek hesaplama gücü ve bellek kapasitesi gerektirir. Bu durum, eğitim sürecinin uzun sürmesine, dağıtımda gecikmelere ve özellikle kaynak kısıtlı klinik ortamlarda gerçek zamanlı uygulamaların zorlaşmasına neden olur. Model yorumlaması ve genelleştirme zorlukları da önemli kısıtlamalardır.
Yorumlanabilirlik Güçlüğü	Derin öğrenme modellerinin karar alma süreçleri genellikle şeffaf değildir. Yorumlanabilirliği artırmak adına genellikle CNN tabanlı açıklayıcı yöntemler kullanılmaktadır; ancak bu yöntemler sınırlı etkililik gösterebilir.
Kara Kutu Problemi	Derin öğrenme modellerinin içsel işleyişinin kullanıcı tarafından anlaşılamaması, özellikle sağlık gibi kritik alanlarda model çıktısına olan güveni azaltır. Bu sorun, model geliştirme ve hata ayıklama süreçlerini de zorlaştırır.
Aşırı Uyum (Overfitting)	Modelin eğitim verisine aşırı uyum sağlaması, yeni ve görülmemiş veriler üzerinde düşük doğruluk sergilemesine neden olur. Bu durum genellikle küçük veri setleri ve aşırı karmaşık modellerden kaynaklanır. Düzenleme (regularization) teknikleri bu sorunu azaltmaya yönelik kullanılır.

Tablo 3.5: Model Zorlukları [151-156] (Devamı).

Bağlamsal Anlayış Eksikliği	Derin öğrenme modelleri, hastalıkla ilişkili özellikleri tanımlayabilir; ancak bu özelliklerin görüntüdeki genel bağlam içerisindeki önemini kavrayamaz. Bu durum, yanlış sınıflandırmalara ve açıklama zorluklarına neden olabilir. Çok modlu veriler ve transfer öğrenme bu eksikliği gidermek için önerilen stratejilerdir.
Felaketsel Unutma	Bu durum, öğrenilmiş bilgilere müdahale edilerek yeni bilgilerin mevcut bir derin öğrenme (DL) modeline entegre edilmesi süreci olarak tanımlanır. Örneğin, 1000 farklı çiçek türünü sınıflandırmak üzere eğitilmiş bir model düşünelim; bu modele yeni bir çiçek türü eklendiğinde ve sadece bu sınıfla model ince ayarlandığında, önceki sınıflar üzerindeki performans kayda değer biçimde düşebilir.
Kaybolan Gradyan Problemi	Yapay sinir ağlarında (YSA), geri yayılım ve gradyan tabanlı öğrenme teknikleri kullanılırken sıklıkla "kaybolan gradyan" problemiyle karşılaşılır. Eğitim sürecinde, her ağırlık değeri hata fonksiyonunun kısmi türevi doğrultusunda güncellenir; ancak gradyanların çok küçük olması durumunda bu güncelleme gerçekleşmeyebilir. Bu durum, ağırlıkların öğrenmeyi tamamen durdurmasına neden olabilir.
Patlayan Gradyan Problemi	Kaybolan gradyan sorununun zıt yönünde yer alan bir diğer önemli problem, "aşırı büyüyen gradyan" (exploding gradient) problemidir. Bu durumda, geri yayılım süreci sırasında hata gradyanlarının aşırı derecede büyümesi söz konusu olur. Bu durum, ağırlıkların kontrolsüz ve büyük güncellemelere neden olarak modelin kararsız hale gelmesine ve öğrenme sürecinin bozulmasına yol açar.

Tablo 3.5: Model Zorlukları [151-156] (Devamı).

Yetersiz Tanımlama Problemi	Modelin öğrenme sürecinde aynı eğitim performansına sahip farklı parametre kombinasyonlarının, dağıtımda farklı sonuçlara yol açabilmesi nedeniyle ciddi bir belirsizlik oluşturur. Bu sorunun üstesinden gelmek için geliştirilen yaklaşımlardan biri, modelin gerçek dünya koşullarındaki dayanıklılığını değerlendirmek amacıyla "stres testleri" uygulanmasıdır.
-----------------------------	--

3.6.3 Klinik Zorlukları

Tablo 3.6: Klinik Zorluklar [151-156].

Zorluk Alanı	Açıklama
Karmaşık Anatomi ve Görüntüleme Zorlukları	Akciğer yapısı oldukça dallı ve karmaşıktır. İç yapılarının polimorfik olması ve dokuların benzer gri tonlarında görüntülenmesi, çeşitli akciğer hasarlarının örtüşmesine neden olabilir. Yaş, ilaç etkileri, bağışıklık durumu gibi faktörler semptomların benzerliğine yol açarak tanıyı zorlaştırır.
Klinik Özelliklerin Benzerliği	Zatürre ile toplum kökenli pnömoni, histoplazmoz, eozinofilik pnömoni gibi diğer hastalıklar arasında klinik benzerlikler vardır. BT görüntülerinin düşük ayırt ediciliği tanıda hata payını artırır.
Yardımcı Klinik Bilgi Gereksinimi	Zatürre tanısı için yalnızca BT görüntüleri yeterli olmayıp, hastanın temas, seyahat geçmişi ve semptomlarının da değerlendirilmesi gereklidir.

Tablo 3.7: Klinik Zorluklar [151-156] (Devamı).

Belirsiz Semptomlar ve Görünmeyen Bulgular	Hafif zatürre vakaları belirgin semptomlar göstermeyebilir ve görüntülerde fark edilmesi zordur. Bu durum düşük doğru pozitif oranı (TPR) ve yüksek yanlış negatif oranı (FNR) ile sonuçlanabilir. Yanlış negatiflerin önlenmesi için duyarlılığın artırılması gerekirken, bu durum özgüllüğün azalmasına neden olabilir . Görüntüleme bulgularına ek olarak, yanlış pozitifleri dışlamak için diğer test ve klinik bilgilerle destekleme zorunluluğu vardır.
--	---

3.6.4 Etik ve Gizlilik Zorlukları

Tablo 3.8: Etik ve Gizlilik Zorlukları [151-156].

Zorluk Alanı	Açıklama
Etik Onam ve Anonimlik	Zatürre tespiti için YZ eğitimi sırasında hasta verilerinin kullanımı, bilgilendirilmiş onam ve anonimlik süreçlerini zorunlu kılmaktadır. GDPR ve HIPAA gibi yasal düzenlemelere uyum gereklidir.
Görüntü Gizliliği	Zatürreye dair görüntülerin güvenli şekilde şifrelenmesi ve yönetilmesi, veri mahremiyeti açısından temel bir gerekliliktir.
Bireysel Gizlilik ve Kamu Sağlığı Dengesi	Zatürre bağlamında halk sağlığı öncelikleri, bireysel gizlilik haklarının önüne geçebilir; bu da etik tartışmaları beraberinde getirir.
Veri Paylaşımı ve Üçüncü Taraflar	Zatürre sürecinde hastaneler tarafından toplanan veriler üçüncü taraflarla paylaşılırken, anonimlik korunmasına rağmen gizlilik açıklarına dair endişeler sürmektedir.

3.6.5 Analiz ve Yorumlama Zorlukları

Tablo 3.9: Analiz ve Yorumlama Zorlukları [151-156].

Zorluk Alanı	Açıklama
Veri Üretimi	Hasta verilerinin manuel elde edilmesi zordur; resmi izinler ve kurum desteği gereklidir.
Veri Erişimi ve Etiketleme	Kamuya açık ve özel veri kümeleri mevcut; etiketleme, artırma ve dengeleme bilgisi gerektirir.
Veri Artırma Zorlukları	GAN'lar gibi yöntemlerle yeni veri üretimi karmaşıktır, veri önyargıları aktarılabilir.
Gürültülü Veriler	Gürültü, sınıflandırma doğruluğunu azaltır ve tahmin kalitesini düşürür.
Görüntü Kalitesi ve Formatı	3B veriler genelde uygun değildir, 2B'ye dönüştürülmelidir. Kalitesiz görüntüler yanlış sınıflamaya neden olabilir.
Veri Dengesizliği	Dengesiz sınıf dağılımı, sınıflandırmayı zorlaştırır.
Veri Ön İşleme Sorunları	Eksik veri, format tutarsızlığı, yanlış türler gibi sorunlar analiz kalitesini etkiler.
Uzmanlık Gereksinimi	Model seçimi ve optimizasyonu uzmanlık ister; deneyimsiz kullanıcılar zorlanabilir.
Yorumlama Girişimleri	Isı haritaları ve dikkat mekanizmaları kullanılsa da, zatürreyi diğer akciğer hastalıklarından ayırmadaki yeterliliği sınırlıdır.

4. ARAŞTIRMA BULGULARI

4.1 MODEL YAPISI

Keras ile bir Konvolüsyonel Sinir Ağı (CNN) modeli tanımlar, derler ve özetini yazdırır. Model, zehirli tespiti için sınıflandırma yapar. Girdi boyutları 224x224 piksel ve 3 renk kanalından oluşan görüntülerdir. class_count ile sınıf sayısı belirlenir. Model, Adamax optimizasyon algoritması ve categorical_crossentropy kayıp fonksiyonu ile derlenir. accuracy metriği, doğruluğu ölçer. model.summary() ise modelin özetini verir. Modelin yapısını tanımlar ve eğitim sürecinde kullanılacak yapılandırmaları ayarlar. Sequential model, katmanların sırasıyla eklenmesini sağlar. İlk katmanlar, görüntülerden özellikleri çıkaran konvolüsyonel katmanlardır (Conv2D). Ardından, MaxPooling2D katmanı, özellik haritalarının boyutlarını küçültür. Flatten katmanı, çok boyutlu verileri tek boyutlu hale getirir. Dense katmanları, modelin öğrenmesini sağlayan tam bağlı katmanlardır. Son katman, sınıf sayısı kadar nöron içerir ve softmax aktivasyon fonksiyonu kullanarak sınıf olasılıkları üretir.

Model: "sequential"		
Layer (type)	Output shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	896
conv2d_1 (Conv2D)	(None, 224, 224, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 112, 112, 32)	9,248
conv2d_3 (Conv2D)	(None, 112, 112, 32)	9,248
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 32)	0
conv2d_4 (Conv2D)	(None, 56, 56, 32)	9,248
conv2d_5 (Conv2D)	(None, 56, 56, 32)	9,248
conv2d_6 (Conv2D)	(None, 56, 56, 32)	9,248
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 32)	0
conv2d_7 (Conv2D)	(None, 28, 28, 32)	9,248
conv2d_8 (Conv2D)	(None, 28, 28, 32)	9,248
conv2d_9 (Conv2D)	(None, 28, 28, 32)	9,248
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_10 (Conv2D)	(None, 14, 14, 32)	9,248
conv2d_11 (Conv2D)	(None, 14, 14, 32)	9,248
conv2d_12 (Conv2D)	(None, 14, 14, 32)	9,248
max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 32)	0
flatten (Flatten)	(None, 1568)	0
dense (Dense)	(None, 256)	401,664
dense_1 (Dense)	(None, 64)	16,448
dense_2 (Dense)	(None, 2)	130
Total params: 530,114 (2.02 MB)		
Trainable params: 530,114 (2.02 MB)		
Non-trainable params: 0 (0.00 B)		

Şekil 4.1: Model Yapısı ve Çıktısı İki Veri Kümesi İçin Aynıdır.

4.2 MODELİN EĞİTİLMESİ

Modelin eğitimini başlatır. epochs değeri 13 olarak ayarlanır, bu da modelin 13 kez tüm eğitim verisinden öğrenmesi anlamına gelir. model.fit fonksiyonu, eğitim verisiyle (train_gen) modeli eğitir, her epoch sonrası doğrulama verisiyle (valid_gen) modelin başarımını test eder. verbose=1, eğitim sürecinin her epoch'ta ilerlemesini gösterir. history, eğitim ve doğrulama sonuçlarını kaydeder.

Epoch 1/13	1043/1043	988s	942ms/step	- accuracy: 0.8472	- loss: 0.3698	- val_accuracy: 0.9540	- val_loss: 0.1269
Epoch 2/13	1043/1043	914s	876ms/step	- accuracy: 0.9585	- loss: 0.1122	- val_accuracy: 0.9636	- val_loss: 0.1016
Epoch 3/13	1043/1043	934s	895ms/step	- accuracy: 0.9652	- loss: 0.0912	- val_accuracy: 0.9195	- val_loss: 0.1930
Epoch 4/13	1043/1043	952s	912ms/step	- accuracy: 0.9658	- loss: 0.0923	- val_accuracy: 0.9693	- val_loss: 0.1043
Epoch 5/13	1043/1043	926s	888ms/step	- accuracy: 0.9759	- loss: 0.0600	- val_accuracy: 0.9636	- val_loss: 0.0900
Epoch 6/13	1043/1043	920s	882ms/step	- accuracy: 0.9821	- loss: 0.0496	- val_accuracy: 0.9655	- val_loss: 0.1187
Epoch 7/13	1043/1043	923s	885ms/step	- accuracy: 0.9830	- loss: 0.0406	- val_accuracy: 0.9732	- val_loss: 0.0752
Epoch 8/13	1043/1043	920s	882ms/step	- accuracy: 0.9857	- loss: 0.0413	- val_accuracy: 0.9559	- val_loss: 0.1784
Epoch 9/13	1043/1043	918s	879ms/step	- accuracy: 0.9912	- loss: 0.0323	- val_accuracy: 0.9674	- val_loss: 0.1100
Epoch 10/13	1043/1043	913s	875ms/step	- accuracy: 0.9895	- loss: 0.0240	- val_accuracy: 0.9713	- val_loss: 0.1029
Epoch 11/13	1043/1043	913s	875ms/step	- accuracy: 0.9936	- loss: 0.0183	- val_accuracy: 0.9674	- val_loss: 0.1109
Epoch 12/13	1043/1043	924s	886ms/step	- accuracy: 0.9968	- loss: 0.0098	- val_accuracy: 0.9598	- val_loss: 0.1489
Epoch 13/13	1043/1043	927s	889ms/step	- accuracy: 0.9939	- loss: 0.0174	- val_accuracy: 0.9674	- val_loss: 0.1984

Şekil 4.3: Birinci Veri Seti İçin Modelin Eğitilmesi Sonucu Ortaya Çıkan Çıktısı.

Epoch 1/13	1269/1269	1235s	968ms/step	- accuracy: 0.8390	- loss: 0.3595	- val_accuracy: 0.9543	- val_loss: 0.1316
Epoch 2/13	1269/1269	1038s	818ms/step	- accuracy: 0.9491	- loss: 0.1460	- val_accuracy: 0.9574	- val_loss: 0.1182
Epoch 3/13	1269/1269	1043s	822ms/step	- accuracy: 0.9571	- loss: 0.1203	- val_accuracy: 0.9748	- val_loss: 0.0968
Epoch 4/13	1269/1269	1039s	819ms/step	- accuracy: 0.9707	- loss: 0.0963	- val_accuracy: 0.9685	- val_loss: 0.0852
Epoch 5/13	1269/1269	1034s	815ms/step	- accuracy: 0.9726	- loss: 0.0840	- val_accuracy: 0.9716	- val_loss: 0.0745
Epoch 6/13	1269/1269	1059s	835ms/step	- accuracy: 0.9759	- loss: 0.0683	- val_accuracy: 0.9685	- val_loss: 0.0775
Epoch 7/13	1269/1269	1045s	823ms/step	- accuracy: 0.9763	- loss: 0.0660	- val_accuracy: 0.9811	- val_loss: 0.0522
Epoch 8/13	1269/1269	1089s	813ms/step	- accuracy: 0.9816	- loss: 0.0467	- val_accuracy: 0.9890	- val_loss: 0.0415
Epoch 9/13	1269/1269	1068s	842ms/step	- accuracy: 0.9872	- loss: 0.0326	- val_accuracy: 0.9732	- val_loss: 0.0648
Epoch 10/13	1269/1269	1096s	864ms/step	- accuracy: 0.9911	- loss: 0.0306	- val_accuracy: 0.9795	- val_loss: 0.0644
Epoch 11/13	1269/1269	1099s	866ms/step	- accuracy: 0.9907	- loss: 0.0260	- val_accuracy: 0.9826	- val_loss: 0.0495
Epoch 12/13	1269/1269	1076s	848ms/step	- accuracy: 0.9921	- loss: 0.0248	- val_accuracy: 0.9558	- val_loss: 0.1729
Epoch 13/13	1269/1269	1073s	846ms/step	- accuracy: 0.9954	- loss: 0.0158	- val_accuracy: 0.9763	- val_loss: 0.1034

Şekil 4.2: İkinci Veri Seti İçin Modelin Eğitilmesi Sonucu Ortaya Çıkan Çıktısı.

4.3 EĞİTİM GEÇMİŞİNİ GÖRSELLEŞTİRME

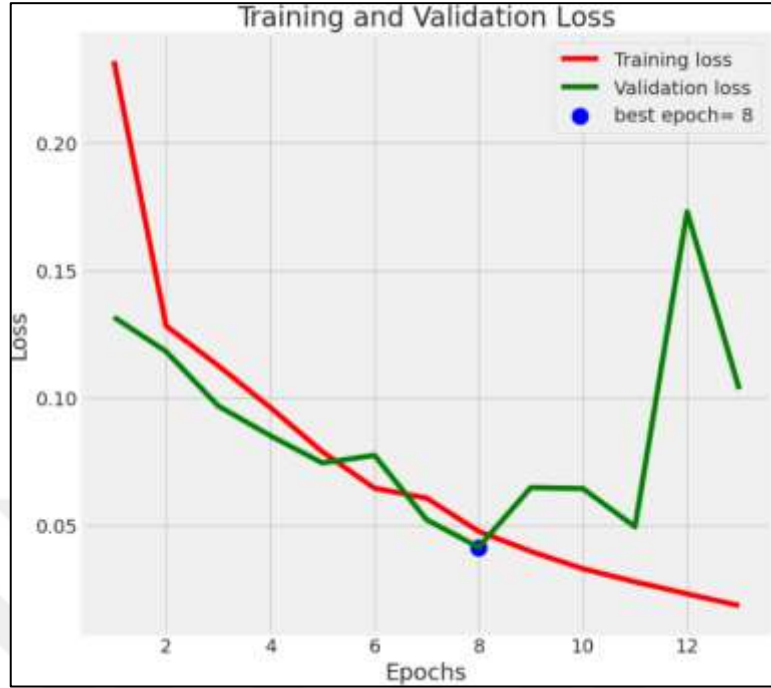
Modelin eğitim geçmişini görselleştirir. Eğitim doğruluğu ve kaybı ile doğrulama doğruluğu ve kaybı değerlerini çıkararak, her epoch için grafikler oluşturur. En iyi epoch'ları (en düşük doğrulama kaybı ve en yüksek doğrulama doğruluğu) belirler ve bunları grafiklerde vurgular. İki alt grafik (kayıbı ve doğruluğu) oluşturur, her birine eğitim ve doğrulama verileri ile en iyi epoch noktalarını ekler. Bu, modelin eğitim sürecini analiz etmeyi sağlar.



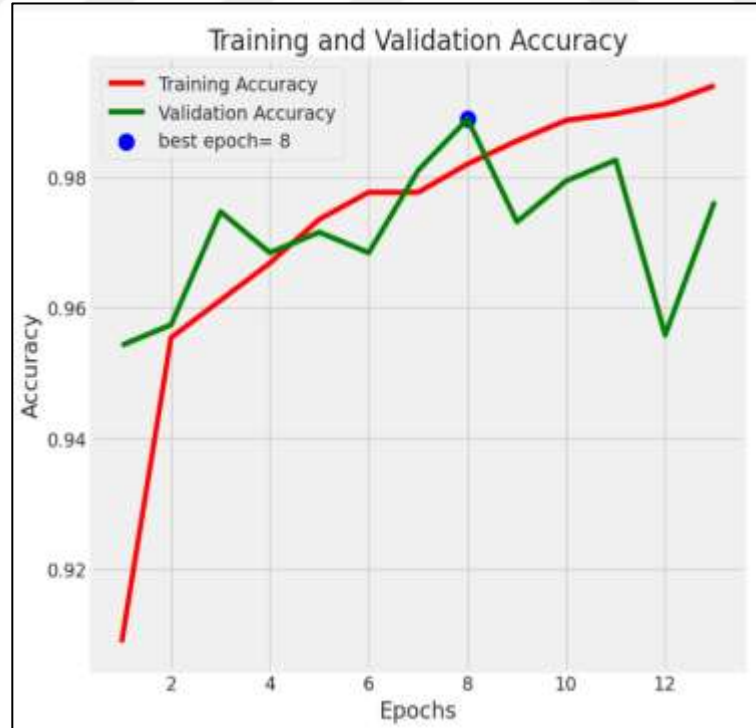
Şekil 4.4: Birinci Veri Seti İçin Eğitim ve Doğrulama Kaybı.



Şekil 4.5: Birinci Veri Seti İçin Eğitim ve Doğrulama Doğruluğu.



Şekil 4.6: İkinci Veri Seti İçin Eğitim ve Doğrulama Kaybı.



Şekil 4.7: İkinci Veri Seti İçin Eğitim ve Doğrulama Doğruluğu.

4.4 MODEL DEĞERLENDİRME

Eğitim, doğrulama ve test verileri üzerinde modelin performansını değerlendirir. model.evaluate() fonksiyonu ile her bir veri kümesi için kayıp ve doğruluk değerleri hesaplanır. Ardından bu sonuçlar, kayıp ve doğruluk değerleriyle birlikte ekrana yazdırılır. Bu işlem, modelin yeni verilere nasıl genelleme yaptığını değerlendirmeyi sağlar.

```
1043/1043 ————— 234s 224ms/step - accuracy: 0.9977 - loss: 0.0055
131/131 ————— 29s 223ms/step - accuracy: 0.9653 - loss: 0.2009
131/131 ————— 30s 228ms/step - accuracy: 0.9802 - loss: 0.0788
Train Loss: 0.007677972316741943
Train Accuracy: 0.9976030588150024
-----
Validation Loss: 0.19839701056480408
Validation Accuracy: 0.967432975769043
-----
Test Loss: 0.10645490884780884
Test Accuracy: 0.975095808506012
```

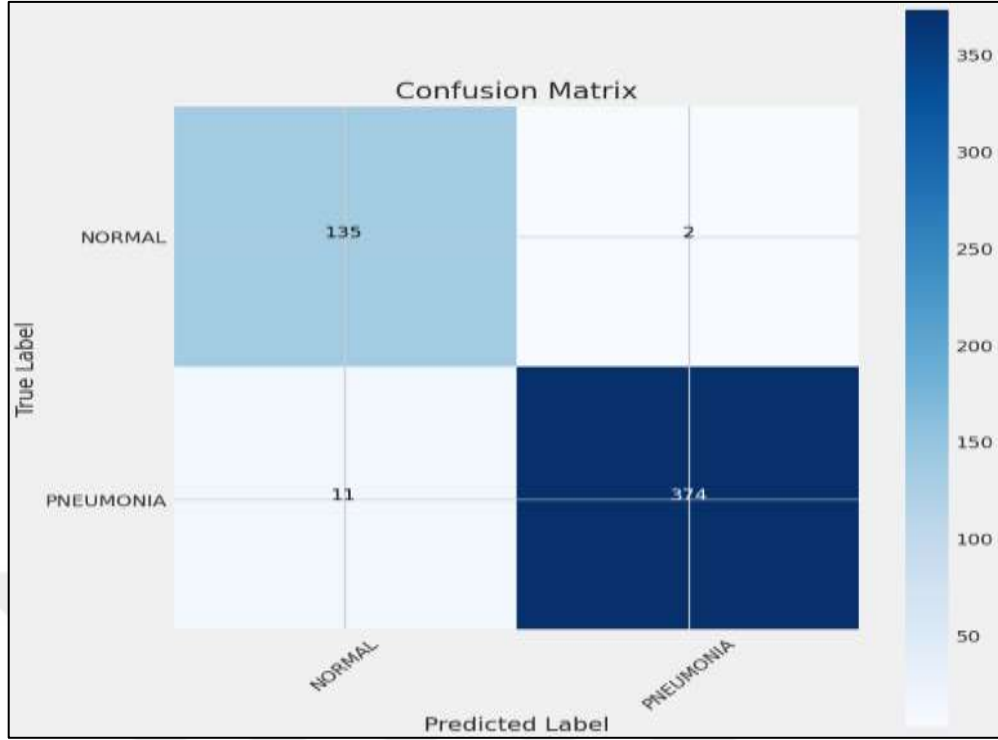
Şekil 4.9: Birinci Veri Seti İçin Alınan Sonuçlar.

```
1269/1269 ————— 265s 209ms/step - accuracy: 0.9955 - loss: 0.0108
159/159 ————— 34s 211ms/step - accuracy: 0.9787 - loss: 0.0953
159/159 ————— 126s 795ms/step - accuracy: 0.9966 - loss: 0.0309
Train Loss: 0.010264618322253227
Train Accuracy: 0.9962546825408936
-----
Validation Loss: 0.103374183177948
Validation Accuracy: 0.9763407111167908
-----
Test Loss: 0.04083247482776642
Test Accuracy: 0.995275616645813
```

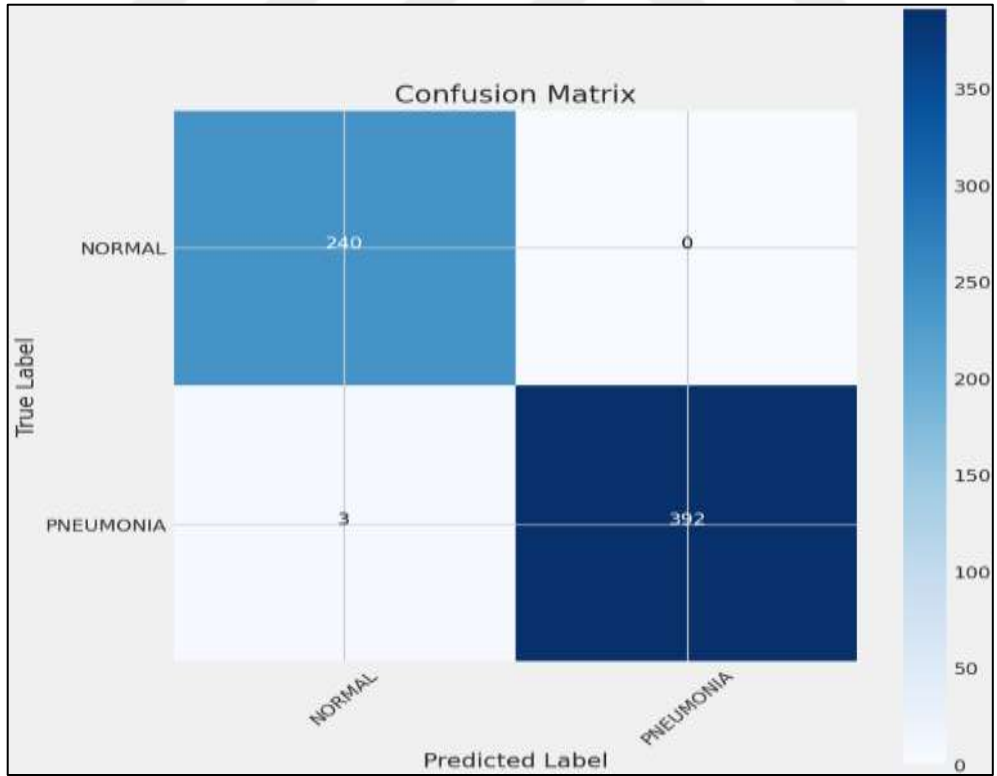
Şekil 4.8: İkinci Veri Seti İçin Alınan Sonuçlar.

4.5 KARIŞIKLIK MATRİSİ

Modelin tahmin performansını değerlendiren bir karışıklık matrisini (confusion matrix) hesaplar ve görselleştirir. Gerçek ve tahmin edilen etiketler karşılaştırılır, matris oluşturulur ve sınıflar arasındaki hatalar renkli bir şekilde görselleştirilir.



Şekil 4.10: Birinci Veri Seti İçin Karışıklık Matrisi Sonuçları.



Şekil 4.11: İkinci Veri Seti İçin Karışıklık Matrisi Sonucu.

4.6 SINIFLANDIRMA RAPORU

Modelin test verisi üzerindeki performansını değerlendiren bir sınıflandırma raporu oluşturur. Gerçek etiketler (test_gen.classes) ve tahmin edilen etiketler (y_pred) arasındaki doğruluk, precision, recall, F1 skoru gibi metrikleri hesaplar ve rapor olarak yazdırır.

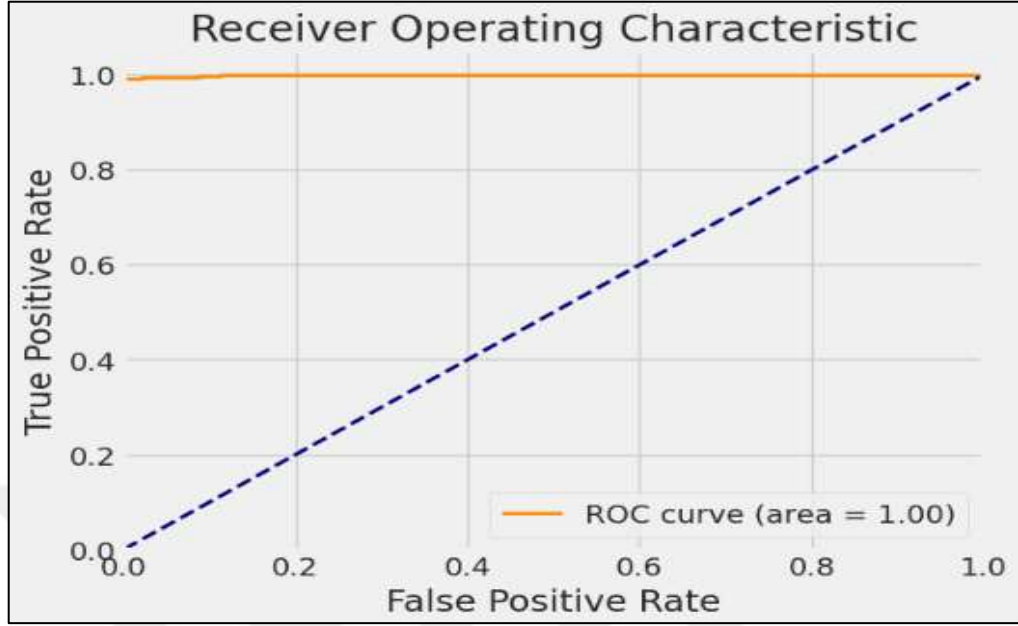
	precision	recall	f1-score	support
NORMAL	0.92	0.99	0.95	137
PNEUMONIA	0.99	0.97	0.98	385
accuracy			0.98	522
macro avg	0.96	0.98	0.97	522
weighted avg	0.98	0.98	0.98	522

Şekil 4.12: Birinci Veri Seti Sonucu.

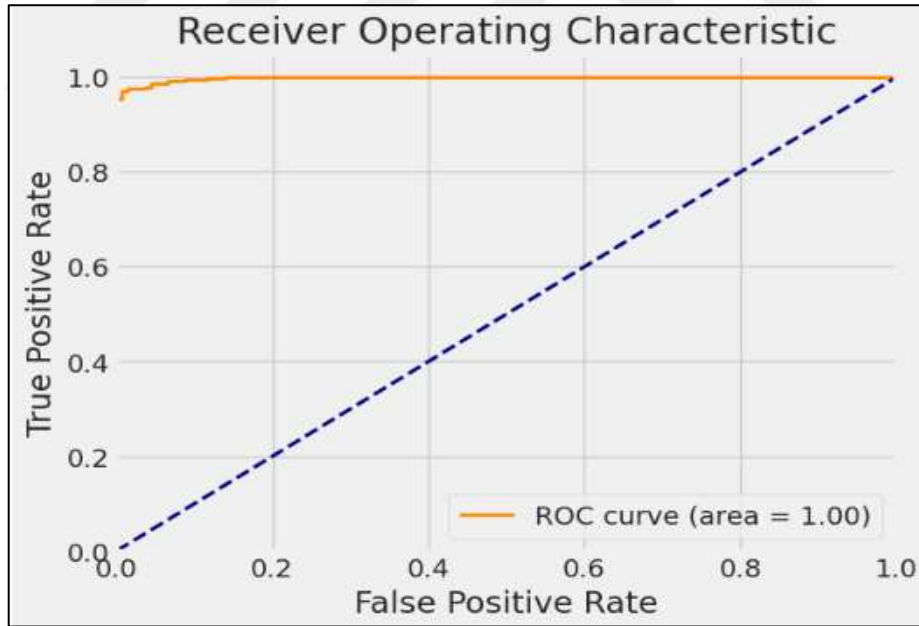
	precision	recall	f1-score	support
NORMAL	0.99	1.00	0.99	240
PNEUMONIA	1.00	0.99	1.00	395
accuracy			1.00	635
macro avg	0.99	1.00	0.99	635
weighted avg	1.00	1.00	1.00	635

Şekil 4.13: İkinci Veri Seti Sonucu.

4.7 ROC EĞRISI DEĞERLENDİRME

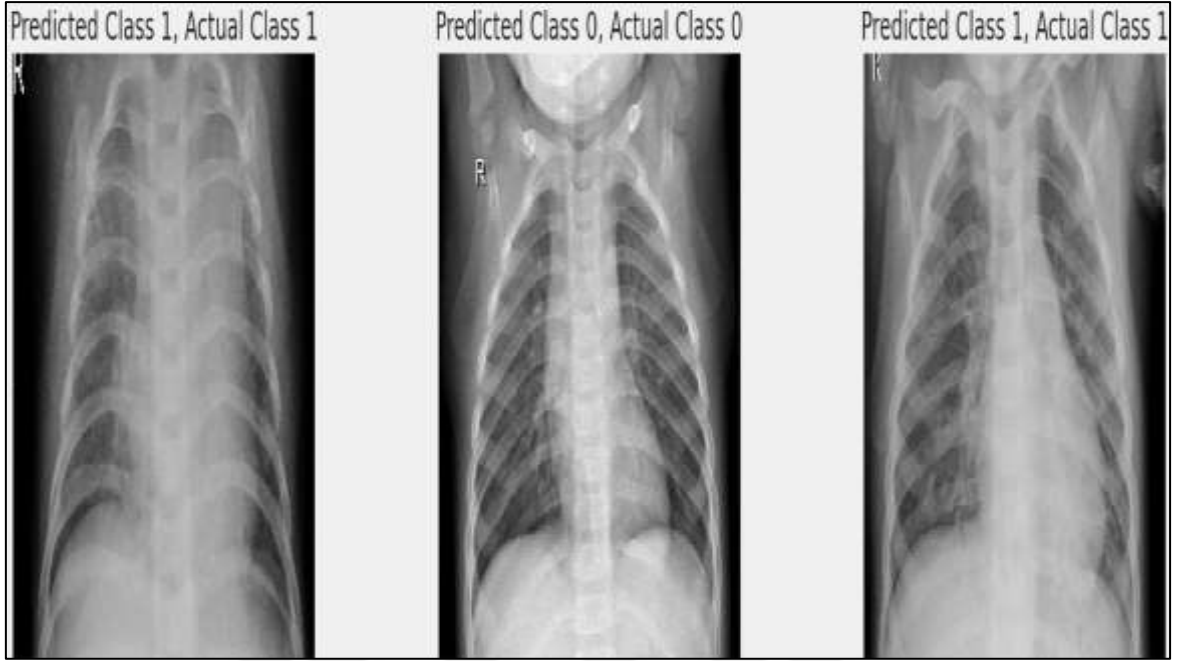


Şekil 4.14: İkinci Veri Seti İçin ROC Eğrisi Grafiği.



Şekil 4.15: Birinci Veri Seti İçin ROC Eğrisi Grafiği.

4.8 DOĞRU VE YANLIŞ SINIFLANDIRILMIŞ GÖRÜNTÜLER



Şekil 4.16: Birinci Veri Seti İçin Yapılan Deneydeki Doğru Tahminlerden Birkaçı.



Şekil 4.17: Birinci Veri Seti İçin Yapılan Deneydeki Yanlış Tahminlerden Birkaçı.



Şekil 4.18: İkinci Veri Seti İçin Yapılan Deneydeki Doğru Tahminlerden Birkacı.



Şekil 4.19: İkinci Veri Seti İçin Yapılan Deneydeki Yanlış Tahminler.

5. TARTIŞMA VE SONUÇ

5.1 KARŞILAŞTIRMALI ANALİZ VE TARTIŞMA

Zatürrenin erken teşhisi, uygun tedavi stratejilerinin belirlenmesi ve potansiyel komplikasyonların önlenmesi açısından kritik öneme sahiptir. Bu bağlamda, göğüs radyografileri tanı sürecinde sıklıkla kullanılmakla birlikte, görüntülerin yorumlanmasında uzman deneyimine olan bağımlılık ve sınıflar arası belirsizlikler tanısal doğruluğu olumsuz etkileyebilmektedir. Derin öğrenme (DL) yöntemleri, özellikle görüntü tabanlı analizlerde sunduğu yüksek doğruluk oranları nedeniyle tıbbi görüntüleme alanında giderek daha fazla tercih edilmektedir. Veri odaklı özellik çıkarımı gerçekleştiren farklı DL mimarileri, tanı süreçlerinde dikkate değer katkılar sağlamaktadır.

DL temelli modellerin geliştirilmesi genellikle yüksek hesaplama gücü ve zaman gerektirir. Bu nedenle, mevcut çalışmaların başarımlarının ayrıntılı biçimde raporlanması, benzer projelerin yinelenmesini engellemekte ve daha verimli çözümlere ulaşılmasına katkı sunmaktadır. Bu bağlamda, geliştirilen yapay zekâ sistemlerinin klinik karar destek aracı olarak kullanılması, özellikle uzman yetersizliğinin bulunduğu ve hızlı müdahalenin gerekli olduğu ortamlarda büyük avantaj sağlayabilir.

Geniş veri kümeleriyle eğitilmiş DL algoritmalarının daha yüksek genelleştirilebilirliğe sahip olması beklenmektedir. Ancak çok kurumlu veri setlerinin dış performans üzerindeki etkisi her zaman belirleyici olmamaktadır. DL yöntemlerinin sunduğu katkılar; tanı doğruluğunun artışı, uzaktan sağlık hizmetleri, iş yükünün azaltılması ve etik veri kullanımı gibi başlıklar altında toplanabilir. Bununla birlikte, DL sistemlerinin hekimlerin yerini alması değil, onlara destek olması gerektiği vurgulanmalıdır.

Mevcut literatür incelendiğinde, kullanılan veri kümelerinin çoğunun dengesiz dağılıma sahip olduğu ve yalnızca az sayıda çalışmada örnek dengeleme tekniklerine başvurulduğu görülmektedir. Bu durum, model güvenilirliğini düşürmekte ve klinik uygulamalarda yetersizliğe yol açmaktadır. Geniş çaplı ve dengeli veri kümeleri oluşturmak ideal çözüm olarak görülse de etik ve yasal sınırlamalar bu süreci zorlaştırmaktadır. Bu noktada, GAN tabanlı görüntü sentezi yöntemleri, geleneksel tekniklere kıyasla azınlık sınıflarına yönelik daha gerçekçi örnekler üretebilmeleri nedeniyle dikkat çekmektedir.

Pnömoni tanısı amacıyla geliştirilen çalışmaların büyük çoğunluğu DL algoritmalarına dayansa da klasik makine öğrenmesi (ML) yaklaşımlarının da başarılı sonuçlar sunduğu raporlanmıştır. Özellikle hesaplama maliyeti düşük olan bu yöntemler, sınırlı kaynaklarda avantaj sağlayabilmektedir. Literatürde, örnek olarak Sousa ve arkadaşlarının çalışmaları öne çıkmakta, ancak Hashmi ve Chouhan gibi araştırmacıların güçlü donanım kullanılarak elde ettikleri sonuçlar da dikkate değerdir.

Tablo 5.1: Diğer Araştırmacıların Çalışma Sonuçları ve Kendi Sonuçlarım.

Yazar	Yıl	Yöntem	Veri Seti	Kesinlik Değeri (Accuracy)
Brunese vd. [157]	2020	VGG-16	6523 CXR	97%
Panwar vd. [158]	2020	VGG-19 + GradCAM	2482 CT + 6382 CXR	95.61%
Mahmud vd. [159]	2020	customized CNN (CovXNet)	6161 CXR	97.4%
Ouchicha vd. [160]	2020	customized CNN (CVDNet)	2905 CXR	96.7%
Wang vd. [161]	2020	3D-ResNet	4697 CXR	93.3%
Choudhury vd. [162]	2020	DenseNet201	3487 CXR	97.94%
Ren vd. [163]	2020	CNN + Bayesian Network	35389 CXR	87%
Arias vd. [164]	2020	CNN	79500 CXR	91.5%
Sakib vd. [165]	2020	customized CNN (DL-CRC)	5367 CXR	94%
Ozturk vd. [166]	2020	YOLO via DarkNet	1000 CXR	87–98%

Tablo 5.1: Diğer Araştırmacıların Çalışma Sonuçları ve Kendi Sonuçlarım (Devamı).

Alhudjaif vd. [167]	2021	DenseNet-201	1218 CXR	94.96%
Nikolaou vd. [168]	2021	EfficientNet models	15153 CXR	95%
Das vd. [169]	2021	CNN + transfer learning	1004 CXR	95%
Munusamy vd. [170]	2021	FractalCovNet	473 CT + 11934 CXR	92–98%
Joshi vd. [171]	2021	DarkNet-53	6884 CXR	97.11%
Singh and Tripathi [172]	2022	Quaternion CNN	5856 CXR	93.75%
Dash and Mohapatra [173]	2022	CNN + transfer learning	1272 CXR	97.12%
Rahib H.Abiyey vd. [174]	2018	-	1000	92.4%
Okeke Stephen vd. [175]	2019	-	5856	93.73%
Cohen vd. [176]	2019	-	5232	92.8%
Rajaraman vd. [177]	2019	VGG-16	5856	96.2%
M.Togacar vd. [178]	2019	-	5849	96.84%
Saraiva vd. [179]	2020	-	5840	94.4%
Ayan vd. [180]	2019	-	5856	84.5%
Rahman vd. [181]	2020	-	5247	98.0%

Tablo 5.1: Diğer Araştırmacıların Çalışma Sonuçları ve Kendi Sonuçlarım (Devamı).

Vikash vd. [182]	2020	-	5232	96.39%
Li vd. [183]	2020	2D+3D CNN	4356 CT	96%
Mei vd. [184]	2020	3D CNN	905 CT	92%
Ni vd. [185]	2020	3D UNet CNN	19291 CT	96%
Rajpurkar vd. [186]	2017	2D DenseNet	ChestX-ray14	73.45%–93.71%
Chen vd. [187]	2020	UNet++	106 CT	98.85%
Harmon vd. [188]	2020	3D DenseNet	2617 CT	90.8%
Liu et vd. [189]	2020	3D VGG16	1224 CT	94.6%
Zheng vd. [190]	2020	3D CNN	630 CT	95.9%
Shukla vd. [191]	2020	2D ResNet	852 CT	93.02%
Ravi et vd. [192]	2023	-	Pediatric	98%
Sarada N. vd.[193]	2021	ANN	Sasoo Hospital	92%
Yi P vd. [194]	2020	CNN (ResNet-50)	NIH-IU-MSH	93%
M. Gour vd.[195]	2022	CNN + modified dropout	5856 CXR	97.4%
G.M.M.Alshmrani vd.[196]	2023	-	-	96.48%
MdNahiduzzaman vd. [197]	2023	-	-	95.33%
Tez Sonucu(1)	2025	DenseNet	5856 CXR	97.5%
Tez Sonucu(2)	2025	DenseNet	8551CXR	99.5%

Sonuçlardaki çeşitliliğin yalnızca veri kümesi özelliklerinden kaynaklanmadığı, aynı zamanda model farklılıkları ve görüntü ön işleme tekniklerinin de önemli bir rol oynadığı değerlendirilmektedir. Farklı yapay sinir ağı mimarilerinin kullanılması ve veriler üzerinde uygulanan çeşitli görüntü işleme yöntemleri, araştırmalar arasında ortaya çıkan performans farklılıklarının başlıca nedenleri arasında yer almaktadır.

Bu tez hakkında yorum yapacak olursak eğer, değerlendirme süreci, kamuya açık iki farklı göğüs röntgeni veri kümesi üzerinde gerçekleştirilmiştir. İlk veri kümesi, toplamda 5.856 göğüs röntgeni görüntüsünden oluşurken; ikinci veri kümesi, 8.551 adet göğüs röntgeni içermektedir. Her iki veri kümesine aynı derin öğrenme modeli uygulanmış olmasına rağmen, elde edilen sonuçlar farklılık göstermiştir. Bu farklılığın temel nedenlerinden biri, ilk veri kümesindeki zatürre ve normal göğüs röntgeni sayıları arasındaki belirgin dengesizlik olarak değerlendirilmektedir. Buna karşılık, ikinci veri kümesinde sınıf dağılımı tam olarak eşit olmasa da daha dengeli bir yapı sergilemekte olup, bu durumun modelin öğrenme sürecine daha dengeli katkı sağladığı düşünülmektedir.

5.2 SONUÇ

DL tabanlı sistemler, tıbbi karar destek süreçlerinde etkin bir araç olarak değerlendirilmektedir. Ancak bu potansiyelin tam anlamıyla kullanılabilmesi için, iyi etiketlenmiş büyük veri kümeleri, klinik uzmanların aktif gözetimi ve etik değerlendirme süreçleri ile desteklenmeleri gerekmektedir. Geliştirilen modellerin pandemi gibi küresel sağlık krizlerinde de etkili şekilde kullanılabilecek düzeyde olduğu görülmektedir.

Çalışma, CXR görüntülerinden pnömoni teşhisine yönelik mevcut yöntemleri kapsamlı biçimde incelemiş; algoritmaların etkinliği, hesaplama karmaşıklığı ve uygulanabilirliğini değerlendirmiştir. Literatürde yaygın olarak kullanılan veri kümeleri (MIMIC CXR, ChestX-ray14, Pediatric Chest X-ray) farklı avantaj ve sınırlılıklarla birlikte değerlendirilmiş, özellikle sınıf dengesizliği problemi ön plana çıkarılmıştır. Bu durumu gidermek amacıyla GAN tabanlı veri artırma yöntemleri önemli bir çözüm olarak önerilmektedir.

DL yöntemleri, geleneksel tekniklerin sınırlamalarını aşarak otomatik özellik çıkarımı ve yüksek sınıflandırma doğruluğu sunmaktadır. Model birleştirme ve topluluk öğrenmesi gibi

yaklaşımlar DL performansını daha da artırabilir. Ancak yüksek hesaplama gücü gereksinimi, açıklanabilirlik eksikliği ve veri mahremiyeti gibi dezavantajlar göz önünde bulundurulmalıdır. DL sistemlerinin klinik entegrasyonu hâlâ zorluklarla karşılaşmakta, bu nedenle ortak veri işleme prosedürleri ve değerlendirme çerçevelerine olan ihtiyaç giderek artmaktadır.

5.3 ARAŞTIRMA BOŞLUKLARI VE GELECEK KAPSAMLARI

Derin öğrenme (DL) tabanlı tıbbi görüntü sınıflandırma sistemlerinin performansını artırmak amacıyla gelecekte görüntü iyileştirme ve veri artırma tekniklerinin daha etkin bir biçimde kullanılması önerilmektedir. Mevcut literatürde genellikle yalnızca frontal radyografiler tercih edilmiştir. Ancak, lateral görüntülerin de modele entegre edilmesi, tanı doğruluğunu artıracak önemli bir adımdır. Ayrıca, pediatrik akciğer anatomisi ile yetişkinlerde görülen deformite içeren patolojik yapıların çoğu çalışmada yeterince dikkate alınmaması, model genellenebilirliğini sınırlayan bir diğer faktördür. Bu nedenle, farklı yaş grupları ve anatomik varyasyonlar üzerinde çalışan veri kümelerinin oluşturulması önem arz etmektedir.

DL modellerinin yalnızca görüntü verisi ile sınırlı kalmaması, yaş, cinsiyet, medikal geçmiş gibi görüntü dışı klinik parametrelerin de modele entegre edilmesi, sınıflandırma doğruluğunu artıracak potansiyel katkılardan biridir. Öte yandan, makine öğrenmesi tabanlı çalışmalarda veri bölünme stratejilerinin çoğu zaman metodolojik olarak şeffaf şekilde raporlanmaması ve genelleme kapasitesinin bağımsız veri kümeleriyle test edilmemesi, literatürde dikkat çeken metodolojik eksiklikler arasında yer almaktadır.

Bir diğer yaygın sınırlılık ise sınıflandırmanın çoğunlukla ikili (binary) olarak gerçekleştirilmesidir. Çok sınıflı sınıflandırma senaryolarının yeterince çalışılmamış olması, gerçek dünya klinik uygulamalarıyla uyumlu sistemlerin geliştirilmesini zorlaştırmaktadır. Ayrıca, DL modellerinin "kara kutu" doğası, karar mekanizmalarının açıklanabilirliğini sınırlamakta ve bu durum, özellikle klinik ortamlarda kullanıcı güvenini azaltmaktadır. Bu bağlamda, açıklanabilir yapay zekâ (XAI) yaklaşımlarının geliştirilmesi ve klinik uygulamalarla bütünleştirilmesi, öncelikli araştırma alanlarından biri olmalıdır.

COVID-19 bağlamında geliştirilen sistemlerde de benzer sorunlar gözlemlenmektedir. Mevcut çalışmalar genellikle farklı kaynaklardan elde edilen, standardize edilmemiş veri kümeleri kullanılarak yürütülmektedir. Bu durum, elde edilen sonuçların karşılaştırılabilirliğini ve genellenebilirliğini kısıtlamaktadır. Bu nedenle, farklı coğrafyalardan ve demografik gruplardan elde edilmiş, ortak bir formatta yapılandırılmış ve kamuya açık hale getirilmiş kıyaslama veri kümelerinin oluşturulması gerekmektedir. Ayrıca, çoğu çalışma X-ray görüntüleri üzerinde odaklanmışken, literatürde göğüs BT'sinin tanı doğruluğu açısından daha üstün olduğu belirtilmektedir. Bununla birlikte, ultrason taramaları, özellikle hamile bireyler için radyasyonsuz bir alternatif olarak öne çıkmaktadır. Bu nedenle, BT ve ultrason görüntülerinin de kapsandığı zenginleştirilmiş veri kümelerinin oluşturulması ve GAN gibi üretken modellerin bu süreçte kullanılması önerilmektedir.

Veri anotasyonunun zahmetli ve zaman alıcı bir süreç olması nedeniyle, denetimli öğrenme yöntemlerine duyulan yüksek bağımlılık da model geliştirmeyi sınırlayan önemli bir etkidir. Bu sorunun aşılması adına, gözetimsiz ve yarı gözetimli öğrenme yaklaşımlarının yanı sıra, birkaç örnekle öğrenme (few-shot learning) ve tek örnekle öğrenme (one-shot learning) gibi meta öğrenme tabanlı yöntemler, sınırlı veri ortamlarında güçlü bir alternatif olarak değerlendirilmektedir.

Ayrıca, transfer öğrenme ve alan uyarlama gibi yöntemlerin, modellerin farklı kaynaklardan gelen verilerle başarılı şekilde uyum sağlamasında etkili olduğu gösterilmiştir. Derin pekiştirmeli öğrenme (DRL), az ve sıfır örnekle öğrenme (few/zero-shot learning) teknikleri de veri yetersizliği sorununun çözümünde ön plana çıkmaktadır. Gelecekte, görüntü kalitesini artırmaya yönelik ön işleme adımlarının yanı sıra segmentasyon destekli sınıflandırma, model sadeleştirme ve mobil cihazlara uygunlaştırma gibi uygulamalara yönelik araştırmaların artırılması beklenmektedir.

Son olarak, disiplinler arası iş birliklerinin teşvik edilmesi, kullanıcı dostu ve klinisyenlerin pratikte kullanabileceği sistemlerin geliştirilmesi, farklı demografik gruplar üzerindeki model performansının değerlendirilmesi ve etik ilkelerin gözetilmesi, DL tabanlı tıbbi karar destek sistemlerinin sürdürülebilirliğini sağlamak açısından kritik önemdedir.

REFERANSLAR

- [1] Johns Hopkins Medicine, “Pneumonia,” [Çevrimiçi]. Erişim adresi: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/pneumonia>
- [2] S. Johnson ve D. Wells, “Viral Pnömoni: Belirtiler, Risk Faktörleri ve Daha Fazlası,” *Healthline*, [Çevrimiçi]. Erişim adresi: <https://www.healthline.com/health/viral-pneumonia>
- [3] Utah Üniversitesi Sağlık, “Zatürre Ölüm Nedenlerinin İlk 10'unda Yer Alıyor,” 2016. [Çevrimiçi]. Erişim adresi: https://healthcare.utah.edu/the-scope/shows.php?shows=0_riw4wti7 (Erişim tarihi: 31 Aralık 2019)
- [4] Dünya Sağlık Örgütü (DSÖ), “Zatürre Çocuklarda Ölümün Önde Gelen Nedenidir,” 2011. [Çevrimiçi]. Erişim adresi: https://www.who.int/maternal_child_adolescent/news_events/news/2011/pneumonia/en (Erişim tarihi: 31 Aralık 2019)
- [5] I. Rudan, L. Tomaskovic, C. Boschi-Pinto ve H. Campbell, “Beş yaş altı çocuklarda klinik pnömoni insidansının küresel tahmini,” *Bulletin of the World Health Organization*, cilt 82, ss. 895–903, 2004.
- [6] RadiologyInfo.org, “Zatürre,” [Çevrimiçi]. Erişim adresi: <https://www.radiologyinfo.org/en/info.cfm?pg=pneumonia>
- [7] Birleşmiş Milletler, “Birleşmiş Milletler Atom Radyasyonunun Etkileri Bilimsel Komitesi (UNSCEAR), İyonlaştırıcı Radyasyonun Kaynakları ve Etkileri Hakkında 2008 Raporu,” 2008. [Çevrimiçi]. Erişim adresi: http://www.unscear.org/docs/publications/2008/UNSCEAR_2008_Annex-A-CORR.pdf
- [8] “Radiologic and nuclear medicine studies in the United States and worldwide: Frequency, radiation dose, and comparison with other radiation sources—1950–2007,” *Radiology*, cilt 253, sy. 2, ss. 520-531, 2009. doi: 10.1148/radiol.2532082010

- [9] S. Raoof, D. Feigin, A. Sung, S. Raoof, L. Irugulpati ve E. C. Rosenow, “Interpretation of plain chest roentgenogram,” *Chest*, cilt 141, sy. 2, ss. 545-558, 2012. doi: 10.1378/chest.10-1302
- [10] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghighi, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren ve A. Y. Ng, “CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison,” *AAAI Conf. Artif. Intell.*, cilt 33, ss. 590–597, 2019.
- [11] L. G. Quekel, A. G. Kessels, R. Goei ve J. M. van Engelshoven, “Detection of lung cancer on the chest radiograph: a study on observer performance,” *Eur. J. Radiol.*, cilt 39, sy. 2, ss. 111–116, 2001, doi: 10.1016/s0720-048x(01)00301-1.
- [12] Y. Balabanova, R. Coker, I. Fedorin, S. Zakharova, S. Plavinskij, N. Krukov, R. Atun ve F. Drobniowski, “Variability in interpretation of chest radiographs among Russian clinicians and implications for screening programmes: observational study,” *BMJ*, cilt 331, sy. 7513, ss. 379–382, 2005, doi: 10.1136/bmj.331.7513.379.
- [13] M. Young, “Interobserver variability in the interpretation of chest roentgenograms of patients with possible pneumonia,” *Arch. Intern. Med.*, cilt 154, sy. 23, s. 2729, 1994, doi: 10.1001/archinte.1994.00420230122014.
- [14] G. S. Lodwick, T. E. Keats ve J. P. Dorst, “The coding of roentgen images for computer analysis as applied to lung cancer,” *Radiology*, cilt 81, sy. 2, ss. 185–200, 1963, doi: 10.1148/81.2.185.
- [15] H. C. Becker, W. J. Nettleton, P. H. Meyers, J. W. Sweeney ve C. M. Nice, “Digital computer determination of a medical diagnostic index directly from chest x-ray images,” *IEEE Trans. Biomed. Eng.*, cilt BME-11, sy. 3, ss. 67–72, 1964, doi: 10.1109/tbme.1964.4502309.
- [16] P. H. Meyers, C. M. Nice, H. C. Becker, W. J. Nettleton, J. W. Sweeney ve G. R. Meckstroth, “Automated computer analysis of radiographic images,” *Radiology*, cilt 83, sy. 6, ss. 1029–1034, 1964, doi: 10.1148/83.6.1029.

- [17] R. P. Kruger, J. R. Townes, D. L. Hall, S. J. Dwyer ve G. S. Lodwick, “Automated radiographic diagnosis via feature extraction and classification of cardiac size and shape descriptors,” *IEEE Trans. Biomed. Eng.*, cilt BME-19, sy. 3, ss. 174–186, 1972, doi: 10.1109/tbme.1972.324115.
- [18] J.-I. Toriwaki, Y. Suenaga, T. Negoro ve T. Fukumura, “Pattern recognition of chest X-ray images,” *Comput. Graph. Image Process.*, cilt 2, sy. 3–4, ss. 252–271, 1973, doi: 10.1016/0146-664x(73)90005-1.
- [19] S. Lal, S. Rehman, J. Shah, T. Meraj, H. Rauf, R. Damaševičius, vd., “Adversarial attack and defence through adversarial training and feature fusion for diabetic retinopathy recognition,” *Sensors*, cilt 21, s. 3922, 2021, pmid:34200216.
- [20] H. Rauf, M. Lali, M. Khan, S. Kadry, H. Alolaiyan, A. Razaq, vd., “Time series forecasting of COVID-19 transmission in Asia Pacific countries using deep neural networks,” *Pers. Ubiquitous Comput.*, ss. 1–18, 2021, pmid:33456433.
- [21] J. Deng, W. Dong, R. Socher, L. Li, K. Li ve L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” *Proc. 2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, ss. 248–255, 2009.
- [22] S. Dalhoumi, G. Dray, J. Montmain, G. Derosière ve S. Perrey, “An adaptive accuracy-weighted ensemble for inter-subjects classification in brain-computer interfacing,” *2015 7th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, ss. 126–129, 2015.
- [23] “<https://doi.org/10.1371/journal.pone.0256630.g002>” [Çevrimiçi]. Erişim adresi: <https://doi.org/10.1371/journal.pone.0256630.g002>
- [24] “Chest X-Ray Images (Pneumonia) | Kaggle,” [Çevrimiçi]. Erişim adresi: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia/data>. [Erişim tarihi: 25-Kas-2018].
- [25] T. Chandra ve K. Verma, “Pneumonia detection on chest X-Ray using machine learning paradigm,” *Proc. 3rd Int. Conf. Comput. Vis. Image Process.*, ss. 21–33, 2020.

- [26] K. Kuo, P. Talley, C. Huang ve L. Cheng, “Predicting hospital-acquired pneumonia among schizophrenic patients: a machine learning approach,” *BMC Med. Inform. Decis. Mak.*, cilt 19, ss. 1–8, 2019, pmid:30866913.
- [27] H. Yue, Q. Yu, C. Liu, Y. Huang, Z. Jiang, C. Shao, vd., “Machine learning-based CT radiomics method for predicting hospital stay in patients with pneumonia associated with SARS-CoV-2 infection: a multicenter study,” *Ann. Transl. Med.*, cilt 8, 2020, pmid:32793703.
- [28] H. Sharma, J. Jain, P. Bansal ve S. Gupta, “Feature extraction and classification of chest x-ray images using CNN to detect pneumonia,” *2020 10th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, ss. 227–231, 2020.
- [29] O. Stephen, M. Sain, U. Maduh ve D. Jeong, “An efficient deep learning approach to pneumonia classification in healthcare,” *J. Healthc. Eng.*, cilt 2019, 2019, pmid:31049186.
- [30] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, vd., “CheXNet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” *arXiv preprint*, arXiv:1711.05225, 2017.
- [31] T. Tuncer, F. Ozyurt, S. Dogan ve A. Subasi, “A novel COVID-19 and pneumonia classification method based on F-transform,” *Chemom. Intell. Lab. Syst.*, cilt 210, s. 104256, 2021, pmid:33531722.
- [32] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri ve R. M. Summers, “ChestX-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” *arXiv preprint*, arXiv:1705.02315, 2017.
- [33] L. Yao, E. Poblentz, D. Dagunts, B. Covington, D. Bernard ve K. Lyman, “Learning to diagnose from scratch by exploiting dependencies among labels,” *arXiv preprint*, arXiv:1710.10501, 2017.
- [34] “<https://doi.org/10.1371/journal.pone.0256630.t001>” [Çevrimiçi]. Erişim adresi: <https://doi.org/10.1371/journal.pone.0256630.t001>

- [35] J. Wang, Y. M. Bao, Y. F. Wen, H. B. Lu, H. Luo, Y. F. Xiang, X. M. Li, C. Liu, D. H. Qian, "Prior-Attention Residual Learning for More Discriminative COVID-19 Screening in CT Images," *IEEE Trans. Med. Imag.*, cilt 39, sy. 8, ss. 2572–2583, 2020, doi: 10.1109/TMI.2020.2994908.
- [36] Q. Yang, Q. Liu, H. Xu, H. Lu, S. Liu ve H. Li, "Imaging of coronavirus disease 2019: A Chinese expert consensus statement," *Eur. J. Radiol.*, cilt 127, s. 109008, 2020.
- [37] Z. Zheng, Z. Yao, K. Wu ve J. Zheng, "The diagnosis of pandemic coronavirus pneumonia: A review of radiology examination and laboratory test," *J. Clin. Virol.*, cilt 128, s. 104396, 2020.
- [38] World Health Organization, *Use of chest imaging in COVID-19: A rapid advice guide* [Çevrimiçi]. 11 Haz. 2020 tarihinde güncellendi. Erişim: Kas. 2020. Erişim adresi: <https://www.who.int/publications/i/item/use-of-chest-imaging-in-covid-19>
- [39] N. Stogiannos, D. Fotopoulos, N. Woznitza ve C. Malamateniou, "COVID-19 in the radiology department: What radiographers need to know," *Radiography (Lond).*, cilt 26, sy. 3, ss. 254–263, 2020.
- [40] J. Shuja, E. Alanazi, W. Alasmay ve A. Alashaikh, "COVID-19 open source data sets: A comprehensive survey," *Appl. Intell. (Dordr.)*, cilt 51, sy. 3, ss. 1296–1325, 2021.
- [41] D. Buonsenso, D. Pata ve A. Chiaretti, "COVID-19 outbreak: Less stethoscope, more ultrasound," *Lancet Respir. Med.*, cilt 8, sy. 5, s. e27, 2020.
- [42] M. Shen, Y. Zhou, J. Ye, A. A. Al-Maskri, Y. Kang, S. Zeng ve S. Cai, "Recent advances and perspectives of nucleic acid detection for coronavirus," *J. Pharm. Anal.*, cilt 10, sy. 2, ss. 97–101, 2020.
- [43] M. Lipsitch, S. Perlman ve M. K. Waldor, "Testing COVID-19 therapies to prevent progression of mild disease," *Lancet Infect. Dis.*, cilt 20, sy. 12, s. 1367, 2020.
- [44] A. Tahamtan ve A. Ardebili, "Real-time RT-PCR in COVID-19 detection: Issues affecting the results," *Expert Rev. Mol. Diagn.*, cilt 20, sy. 5, ss. 453–454, 2020.

- [45] P. B. van Kasteren, B. van der Veer, S. van den Brink, L. Wijsman, J. de Jonge, A. van den Brandt vd., “Comparison of seven commercial RT-PCR diagnostic kits for COVID-19,” *J. Clin. Virol.*, cilt 128, s. 104412, 2020.
- [46] M. Li, “Chest CT features and their role in COVID-19,” *Radiol. Infect. Dis.*, cilt 7, sy. 2, ss. 51–54, 2020.
- [47] N. Bonadia, A. Carnicelli, A. Piano, D. Buonsenso, E. Gilardi, C. Kadhim vd., “Lung ultrasound findings are associated with mortality and need for intensive care admission in COVID-19 patients evaluated in the emergency department,” *Ultrasound Med. Biol.*, cilt 46, sy. 11, ss. 2927–2937, 2020.
- [48] H. Kasban, M. El-Bendary ve D. Salama, “A comparative study of medical imaging techniques,” *Int. J. Intell. Inf. Syst.*, cilt 4, sy. 2, ss. 37–58, 2015.
- [49] W. C. Röntgen, “On a new kind of rays,” *Science*, cilt 3, sy. 59, ss. 227–231, 1896.
- [50] H. Mukherjee, S. Ghosh, A. Dhar, S. M. Obaidullah, K. C. Santosh ve K. Roy, “Shallow convolutional neural network for COVID-19 outbreak screening using chest X-rays,” *Cogn. Comput.*, 2021. [Çevrimiçi]. Erişim adresi: <https://doi.org/10.1007/s12559-020-09775-9>
- [51] B. S. Morris, M. Maheshwari ve A. Chalwa, “Chest wall tuberculosis: A review of CT appearances,” *Br. J. Radiol.*, cilt 77, sy. 917, ss. 449–457, 2004. doi: 10.1259/bjr/82634045.
- [52] P. Berger, V. Perot, P. Desbarats, J. M. Tunon-de-Lara, R. Marthan ve F. Laurent, “Airway wall thickness in cigarette smokers: Quantitative thin-section CT assessment,” *Radiology*, cilt 235, sy. 3, ss. 1055–1064, 2005.
- [53] A. Jacobi, M. Chung, A. Bernheim ve C. Eber, “Portable chest x-ray in coronavirus disease-19 (COVID-19): a pictorial review,” *Clin. Imaging*, cilt 64, ss. 35–42, 2020.
- [54] A. Yılmaz ve U. Kaya, *Derin Öğrenme*, 3. Baskı, İstanbul: Kodlab, 2021.

- [55] J. McCarthy, "From here to human-level AI," *Artif. Intell.*, cilt 171, sy. 18, ss. 1174–1182, 2007, doi: 10.1016/j.artint.2007.10.009.
- [56] O. Acar, *Yapay Zeka: Fırsat mı Yoksa Tehdit mi?*, İstanbul: Kriter Yayınevi, 2020.
- [57] D. P. Schultz ve S. Ellen-Schultz, *Modern Psikoloji Tarihi*, Y. Aslay, Çev., İstanbul: Kaknüs Yayınları, 2007.
- [58] A. M. Turing, "Computing machinery and intelligence," *Mind*, cilt LIX, sy. 236, ss. 433–460, 1950. [Çevrimiçi]. Erişim: <https://academic.oup.com/mind/article/LIX/236/433/986238>
- [59] S. Russell ve P. Norvig, *Artificial Intelligence: A Modern Approach*, 2. Baskı, Upper Saddle River, NJ: Prentice Hall, 2003.
- [60] A. Doğan, *Yapay Zeka*, İstanbul: Kariyer Yayıncılık, 2002.
- [61] S. Dick, "Artificial intelligence," *Harvard Data Sci. Rev.*, cilt 1, sy. 1, ss. 1–8, 2019, doi: 10.1162/99608f92.92fe150c.
- [62] K. Öztürk ve M. E. Şahin, "Yapay sinir ağları ve yapay zekaya genel bir bakış," *Takvim-i Vekayi*, cilt 6, sy. 2, ss. 25–36, 2018.
- [63] E. H. Shortliffe, *MYCIN: A rule-based computer program for advising physicians regarding antimicrobial therapy selection*, 1974.
- [65] M. Hauskrecht ve H. Fraser, "Planning treatment of ischemic heart disease with partially observable Markov decision processes," *Artif. Intell. Med.*, cilt 18, sy. 3, ss. 221–244, 2000, doi: 10.1016/s0933-3657(99)00042-1.
- [66] M. Z. Alom *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, cilt 8, sy. 3, 292, 2019, doi: 10.3390/electronics8030292.
- [67] I. Guha *et al.*, "Deep learning based high-resolution reconstruction of trabecular bone microstructures from low-resolution CT scans using GAN-CIRCLE," in *Proc. SPIE Med. Imaging 2022: Biomed. Appl. Mol. Struct. Funct. Imaging*, 2020.

- [68] S. Hamidian, B. Sahiner, N. Petrick ve A. Pezeshk, “3D convolutional neural network for automatic detection of lung nodules in chest CT,” in *Proc. Med. Imaging 2017: Comput.-Aided Diagn.*, 2017.
- [69] J. P. Charbonnier *et al.*, “Improving airway segmentation in computed tomography using leak detection with convolutional networks,” *Med. Image Anal.*, cilt 36, ss. 52–60, 2017, doi: 10.1016/j.media.2016.11.001.
- [70] G. Litjens *et al.*, “A survey on deep learning in medical image analysis,” *Med. Image Anal.*, cilt 42, ss. 60–88, 2017, doi: 10.1016/j.media.2017.07.005.
- [71] F. K. Došilović, M. Brčić ve N. Hlupić, “Explainable artificial intelligence: A survey,” in *Proc. 2018 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, ss. 210–215, 2018.
- [72] B. G. Buchanan, “A (Very) Brief History of Artificial Intelligence,” *AI Mag.*, cilt 26, sy. 4, ss. 53–60, 2006.
- [73] L. N. Yasnitsky, A. A. Dumler ve F. M. Cherepanov, “Robot-Doctor: What Can It Be?,” in *Adv. Technol. in Robotics and Intelligent Syst.*, S. Misyurin, V. Arakelian ve A. Avetisyan, Eds. Cham: Springer, 2020, *Mechanisms and Machine Science*, cilt 80.
- [74] S. A. Popenici ve S. Kerr, “Exploring the impact of artificial intelligence on teaching and learning in higher education,” *Res. Pract. Technol. Enhanc. Learn.*, cilt 12, sy. 1, ss. 1–13, 2017, doi: 10.1186/s41039-017-0062-8.
- [75] M. J. Timms, “Letting artificial intelligence in education out of the box: educational cobots and smart classrooms,” *Int. J. Artif. Intell. Educ.*, cilt 26, sy. 2, ss. 701–712, 2016, doi: 10.1007/s40593-016-0095-y.
- [76] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [77] T. Mitchell, *Machine Learning*, New York: McGraw Hill, 1997, ISBN: 978-0-07-042807-2.

- [78] S. Bozinovski, “A self-learning system using secondary reinforcement,” in *Cybernetics and Systems Research*, R. Trappl, Ed., Amsterdam: North Holland, 1982, ss. 397–402, ISBN: 978-0-444-86488-8.
- [79] A. Coates, H. Lee ve A. Y. Ng, “An analysis of single-layer networks in unsupervised feature learning,” in *Proc. AISTATS*, 2011. [Çevrimiçi]. Erişim: https://www.cs.toronto.edu/~hinton/csc2515/readings/CoatesLeeNg_aistats_2011.pdf
- [80] Y. Bengio, *Learning Deep Architectures for AI*, Now Publishers Inc., ss. 1–3, 2009, ISBN: 978-1-60198-294-0.
- [81] E. Alpaydın, *Introduction to Machine Learning*, MIT Press, 2004.
- [82] T. Kohonen, *Self-Organization and Associative Memory*, 3. Baskı, Berlin: Springer-Verlag, 1989. [Çevrimiçi]. Erişim: <http://www.willamette.edu/~gorr/classes/cs449/Classification/perceptron.html>
- [83] O. T. Yıldız ve E. Alpaydın, “Univariate and Multivariate Decision Trees,” TAINN 2000.
- [84] M. Mohammed, M. B. Khan ve B. E. Bashier Mohammed, *Machine Learning: Algorithms and Applications*, CRC Press, 2016.
- [85] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research,” *SN Comput. Sci.*, Springer Nature, 2021, doi: 10.1007/s42979-021-00592-x.
- [86] M. Mehryar, R. Afshin ve T. Ameet, *Foundations of Machine Learning*, MIT Press, 2012, ISBN: 978-0-262-01825-8.
- [87] M. Mayo, “Top Data Science, Machine Learning Methods Used in 2018/2019,” *KDnuggets*, 2021. [Çevrimiçi]. Erişim: <https://www.kdnuggets.com/2019/04/top-data-science-machine-learning-methods-2018-2019.html>
- [88] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe ve S. Lloyd, “Quantum machine learning,” *Nature*, cilt 549, sy. 7671, ss. 195–202, 2017.

- [89] J. Brownlee, *Machine Learning Mastery*, 2014. [Çevrimiçi]. Erişim: <http://machinelearningmastery.com/discover-feature-engineering-howtoengineer-features-and-how-to-get-good-at-it>
- [90] A. Dey, "Machine learning algorithms: a review," *Int. J. Comput. Sci. Inf. Technol.*, cilt 7, sy. 3, ss. 1174–1179, 2016.
- [91] A. Conneau, D. Kiela, H. Schwenk, L. Barrault ve A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint*, arXiv:1705.02364, 2017.
- [92] A. E. Hassanien, A. Salam, A. Darwish, A. Salama ve A. Darwsih, "Artificial Intelligence Approach to Predict the COVID-19 Patient's Recovery," *EasyChair Preprint* № 3223, 2020. [Çevrimiçi]. Erişim: <http://www.egyptscience.net>
- [93] F. Shi *et al.*, "Large-scale screening of COVID-19 from community acquired pneumonia using infection size-aware classification," *arXiv preprint*, arXiv:2003.09860, 2020.
- [94] M. Pourhomayoun ve M. Shakibi, "Predicting mortality risk in patients with COVID-19 using artificial intelligence to help medical decision-making," *MedRxiv*, Şubat 2020, doi: 10.1101/2020.03.30.20047308.
- [95] Z. Meng *et al.*, "Development and utilization of an intelligent application for aiding COVID-19 diagnosis," *MedRxiv*, 2020, doi: 10.1101/2020.03.18.20035816.
- [96] J. Han, J. Pei and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam: Elsevier, 2011.
- [97] A. A. K. Javan *et al.*, "Medical images encryption based on adaptive-robust multi-mode synchronization of chen hyper-chaotic systems," *Sensors*, vol. 21, no. 11, p. 3925, 2021.
- [98] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

- [99] Google Colab, "<https://colab.research.google.com/?hl=tr>" [Eriřim: 18-Nis-2025].
- [100] Amazon Web Services, "What is Python?" <https://aws.amazon.com/tr/what-is/python/> [Eriřim: 18-Nis-2025].
- [101] D. Kermany, K. Zhang, and M. Goldbaum, "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification," *Mendeley Data*, V2, 2018. doi: 10.17632/rscbjbr9sj.2.
- [102] P. Mooney, "Chest X-ray Pneumonia Dataset," Kaggle. <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia> [Eriřim: 18-Nis-2025].
- [103] U. Sait *et al.*, "Curated Dataset for COVID-19 Posterior-Anterior Chest Radiography Images (X-Rays)," *Mendeley Data*, V3, 2021. doi: 10.17632/9xkhgts2s6.3.
- [104] F. Monceda, "Curated COVID-19 Chest X-ray Dataset," Kaggle. <https://www.kaggle.com/datasets/francismon/curated-covid19-chest-xray-dataset> [Eriřim: 18-Nis-2025].
- [105] A. Ng, *Machine Learning Yearning: Technical Strategy for AI Engineers in the Era of Deep Learning*, 2019. [Online]. Available: <https://www.mlyearning.org>
- [106] B. Van Essen, H. Kim, R. Pearce, K. Boakye and B. Chen, "Lbann: Livermore big artificial neural network HPC toolkit," in *Proc. Workshop on Machine Learning in High-Performance Computing Environments*, 2015, pp. 1–6.
- [107] M. C. Moe *et al.*, "Development of neural networks from single stem cells harvested from the adult human brain," *Neurosurgery*, vol. 56, pp. 1182-1190, 2005.
- [108] S. C. Leung and J. Fulcher, "Classification of user expertise level by neural networks," *Int. J. Neural Syst.*, vol. 8, pp. 155–171, 1997.
- [109] P. S. Heckerling, B. S. Gerber, T. G. Tape and R. S. Wigton, "Selection of predictor variables for pneumonia using neural networks and genetic algorithms," *Methods Inf. Med.*, vol. 44, pp. 89–97, 2005.

- [110] M. Çetin, A. Uğur and Ş. Bayzan, "İleri beslemeli yapay sinir ağlarında backpropagation (geriye yayılım) algoritmasının sezgisel yaklaşımı," *Akademik Bilişim Kongresi*, Pamukkale Üniversitesi, Denizli, Şubat 2006.
- [111] N. Tajbakhsh et al., "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1299–1312, May 2016.
- [112] M. Qjidaa et al., "Early detection of COVID19 by deep learning transfer model for populations in isolated rural areas," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Jun. 2020, pp. 1–5.
- [113] A. Shazia et al., "A comparative study of multiple neural network for detection of COVID-19 on chest X-ray," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, pp. 1–16, Dec. 2021.
- [114] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [115] A. S. Raamkumar, S. G. Tan and H. L. Wee, "Health belief model-based deep learning classifiers for classifying COVID-19 social media content to examine public behaviors towards physical distancing," *JMIR Public Health Surveill.*, vol. 6, no. 3, p. e20493, 2020.
- [116] S. A. Razia, J. Arokiaraj and A. Jaithunbi, "LSTM and RNN to predict COVID cases: lethality's and tests in GCC nations and India," *Int. J. Performability Eng.*, vol. 17, no. 3, pp. 299–306, 2021.
- [117] F. A. Saiz and I. Barandiaran, "COVID-19 detection in chest X-ray images using a deep learning approach," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 6, no. 2, pp. 1–4, 2020.
- [118] C. Shorten, T. M. Khoshgoftaar and B. Furht, "Deep learning applications for COVID-19," *J. Big Data*, vol. 8, no. 1, pp. 1–54, 2021.

- [119] M. Fu et al., "Deep learning-based recognizing COVID-19 and other common infectious diseases of the lung by chest CT scan images," *medRxiv*, 2020.
- [120] A. K. Jaiswal et al., "Covidpen: a novel covid-19 detection model using chest x-rays and CT scans," *medRxiv*, 2020. [Online]. Available: <https://doi.org/10.1101/2020.07.08.20149161>
- [121] S. Misra et al., "Multi-channel transfer learning of chest X-ray images for screening of COVID-19," *Electronics*, vol. 9, no. 9, p. 1388, 2020.
- [122] M. Loey and F. Smarandache, "Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning," *Symmetry*, vol. 12, no. 4, p. 651, 2020.
- [123] B. D. Goodwin, C. Jaskolski, C. Zhong and H. Asmani, "Intra-model variability in COVID-19 classification using chest X-ray images," *arXiv preprint arXiv:2005.02167*, 2020.
- [124] E. Littmann and H. Ritter, "Generalization abilities of cascade network architecture," in *Advances in Neural Information Processing Systems*, 1993, pp. 188–195.
- [125] N. S. Punna and S. Agarwal, "Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks," *arXiv preprint arXiv:2004.11676*, 2020.
- [126] L. Sun et al., "Adaptive feature selection guided deep forest for COVID-19 classification with chest CT," *arXiv preprint arXiv:2005.03264*, 2020.
- [127] C. Douarre et al., "Transfer learning from synthetic data applied to soil–root segmentation in x-ray tomography images," *J. Imaging*, vol. 4, p. 65, 2018.
- [128] M. I. Razzak, S. Naz and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," in *Classification in BioApps*, Cham, Switzerland: Springer, 2018, pp. 323–350.

- [129] M. M. Saeed, Z. Al Aghbari and M. Alsharidah, "Big data clustering techniques based on spark: a literature review," *PeerJ Comput. Sci.*, vol. 6, p. 321, 2020.
- [130] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [131] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. 2013 Conf. Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.
- [132] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artificial Intelligence and Statistics*, 2010, pp. 249–256.
- [133] N. Abraham and N. M. Khan, "A novel focal Tversky loss function with improved attention U-Net for lesion segmentation," in *Proc. IEEE 16th Int. Symp. Biomed. Imaging (ISBI)*, 2019, pp. 683–687.
- [134] R. H. Abiyev and M. K. S. Maaitah, "Deep convolutional neural networks for chest diseases detection," *J. Healthc. Eng.*, vol. 2018, Article ID 4168538, 2018. [Online]. Available: <https://doi.org/10.1155/2018/4168538>
- [135] A. M. N. Erdoğan, T. Öztürk and M. Talo, "Yeni bir evrimsel sinir ağı modeli kullanarak bilgisayarlı tomografi görüntülerinden akciğer kanseri tespiti," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 34, no. 2, pp. 795–802, 2022.
- [136] E. E. D. Hemdan, M. A. Shouman and M. E. Karar, "COVIDX-Net: A framework of deep learning classifiers to diagnose COVID-19 in X-ray images," *arXiv preprint arXiv:2003.11055v1*, 2020.
- [137] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra and J. M. Z. Maningo, "Object detection using convolutional neural networks," in *Proc. IEEE TENCON*, 2019, pp. 2023–2027. doi: 10.1109/TENCON.2018.8650517.

- [138] S. Sony, K. Dunphy, A. Sadhu and M. Capretz, "A systematic review of convolutional neural network-based structural condition assessment techniques," *Eng. Struct.*, vol. 226, p. 111347, 2021. doi: 10.1016/j.engstruct.2020.111347.
- [139] S. Huang, F. Lee, R. Miao *et al.*, "A deep convolutional neural network architecture for interstitial lung disease pattern classification," *Med. Biol. Eng. Comput.*, vol. 58, pp. 725–737, 2020. doi: 10.1007/s11517-019-02111-w.
- [140] S. V. Militante, N. V. Dionisio and B. G. Sibbaluca, "Pneumonia detection through adaptive deep learning models of convolutional neural networks," in *Proc. 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, Aug. 2020. doi: 10.1109/ICSGRC49013.2020.9232613.
- [141] E. Yahalomi, M. Chernofsky and M. Werman, "Detection of distal radius fractures trained by a small set of X-ray images and Faster R-CNN," in *Intelligent Computing*, 2019, pp. 971–981.
- [142] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Machine Learning (ICML)*, Jun. 2010, pp. 807–814.
- [143] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [144] S. Chakraborty, B. Murali and A. K. Mitra, "An efficient deep learning model to detect COVID-19 using chest X-ray images," *Int. J. Environ. Res. Public Health*, vol. 19, no. 4, p. 2013, 2022.
- [145] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, pp. 574–582, 2017.
- [146] O. Gozes *et al.*, "Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis," *arXiv preprint arXiv:2003.05037*, 2020.

- [147] G. M. M. Alshmrani *et al.*, "A deep learning architecture for multi-class lung diseases classification using chest X-ray (CXR) images," *Alex. Eng. J.*, vol. 64, pp. 923–935, 2023.
- [148] M. Hossin and M. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 1–11, 2015.
- [149] F. Provost and P. Domingos, "Tree induction for probability-based ranking," *Mach. Learn.*, vol. 52, no. 3, pp. 199–215, 2003.
- [150] H. Oi, R. Kawakami, and T. Nakamura, "Analysis of evaluation metrics with the distance between positive pairs and negative pairs in deep metric learning," in *Proc. 17th Int. Conf. Machine Vision and Applications (MVA)*, 2021, pp. 1–5. doi: 10.23919/MVA51890.2021.9511393.
- [151] I. De Falco, G. De Pietro, and G. Sannino, "Classification of COVID-19 chest X-ray images by means of an interpretable evolutionary rule-based approach," *Neural Comput. Appl.*, vol. 35, pp. 16061–16071, 2023. doi: 10.1007/s00521-021-06806-w.
- [152] M. Barish *et al.*, "External validation demonstrates limited clinical utility of the interpretable mortality prediction model for patients with COVID-19," *Nat. Mach. Intell.*, vol. 3, pp. 25–27, 2021. doi: 10.1038/s42256-020-00254-2.
- [153] I. D. Apostolopoulos, S. I. Aznaouridis, and M. A. Tzani, "Extracting possibly representative COVID-19 biomarkers from X-ray images with deep learning approach and image data related to pulmonary diseases," *J. Med. Biol. Eng.*, vol. 40, no. 3, pp. 1–10, 2020.
- [154] M. Roberts *et al.*, "Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans," *Nat. Mach. Intell.*, vol. 3, pp. 199–217, 2021.
- [155] A. Durafe and V. Patidar, "Securing the COVID patients' medical records using encrypted image steganography," in *ICT Systems and Sustainability, Lecture Notes in Networks and Systems*, vol. 321, Springer, Singapore, 2022. doi: 10.1007/978-981-16-5987-4_43.

- [156] S. S. Long, C. G. Prober and M. Fischer, Eds., *Index*, 5th ed., Elsevier, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780323401814003030>
- [157] L. Brunese, F. Mercaldo, A. Reginelli and A. Santone, "Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays," *Comput. Meth. Prog. Biomed.*, vol. 196, p. 105608, 2020, doi: 10.1016/j.cmpb.2020.105608.
- [158] H. Panwar et al., "A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images," *Chaos, Solitons & Fractals*, vol. 140, p. 110190, 2020, doi: 10.1016/j.chaos.2020.110190.
- [159] T. Mahmud, M. A. Rahman and S. A. Fattah, "CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images," *Comput. Biol. Med.*, vol. 122, p. 103869, 2020, doi: 10.1016/j.combiomed.2020.103869.
- [160] C. Ouchicha, O. Ammor and M. Meknassi, "CVDNet: A novel deep learning architecture for detection of coronavirus (COVID-19) from chest X-ray images," *Chaos, Solitons & Fractals*, vol. 140, p. 110245, 2020, doi: 10.1016/j.chaos.2020.110245.
- [161] C. Wang, R. Pan, X. Wan, Y. Tan, L. Xu, C. S. Ho, and R. C. Ho, "Immediate psychological responses and associated factors during the initial stage of the 2019 coronavirus disease (COVID-19) epidemic among the general population in China," *Int. J. Environ. Res. Public Health*, vol. 17, no. 5, p. 1729, Mar. 2020, doi: 10.3390/ijerph17051729.
- [162] M. E. H. Chowdhury et al., "Can AI Help in Screening Viral and COVID-19 Pneumonia?," *IEEE Access*, vol. 8, pp. 132665–132676, 2020, doi: 10.1109/ACCESS.2020.3010287.
- [163] H. Ren et al., "Interpretable Pneumonia Detection by Combining Deep Learning and Explainable Models With Multisource Data," *IEEE Access*, vol. 9, pp. 95872–95883, 2021, doi: 10.1109/ACCESS.2021.3090215.

- [164] J. D. Arias-Londoño et al., "Artificial Intelligence Applied to Chest X-Ray Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach," *IEEE Access*, vol. 8, pp. 226811–226827, 2020.
- [165] S. Sakib et al., "DL-CRC: Deep Learning-Based Chest Radiograph Classification for COVID-19 Detection: A Novel Approach," *IEEE Access*, vol. 8, pp. 171575–171589, 2020.
- [166] T. Ozturk et al., "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Comput. Biol. Med.*, vol. 121, p. 103792, 2020.
- [167] A. Alhudhaif, K. Polat and O. Karaman, "Determination of COVID-19 pneumonia based on generalized convolutional neural network model from chest X-ray images," *Expert Syst. Appl.*, vol. 180, p. 115141, 2021.
- [168] V. Nikolaou et al., "COVID-19 diagnosis from chest x-rays: developing a simple, fast, and accurate neural network," *Health Inf. Sci. Syst.*, vol. 9, p. 36, 2021.
- [169] A. K. Das et al., "Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network," *Pattern Anal. Appl.*, vol. 24, pp. 1111–1123, 2021.
- [170] H. Munusamy et al., "FractalCovNet architecture for COVID-19 chest X-ray image classification and CT-scan image segmentation," *Biocybern. Biomed. Eng.*, vol. 41, pp. 1025–1039, 2021.
- [171] R. C. Joshi et al., "A deep learning-based COVID-19 automatic diagnostic framework using chest X-ray images," *Biocybern. Biomed. Eng.*, vol. 41, pp. 239–251, 2021.
- [172] S. Singh and B. K. Tripathi, "Pneumonia classification using quaternion deep learning," *Multimed. Tools Appl.*, vol. 81, no. 2, pp. 1743–1764, 2022.
- [173] A. K. Dash and P. Mohapatra, "A fine-tuned deep convolutional neural network for chest radiography image classification on COVID-19 cases," *Multimed. Tools Appl.*, 2021, doi: 10.1007/s11042-021-11388-9.

- [174] R. H. Abiyev and M. K. S. Ma'aitah, "Deep convolutional neural networks for chest diseases detection," *J. Healthc. Eng.*, vol. 2018, Article ID 4168538, 2018.
- [175] O. Stephen et al., "An efficient deep learning approach to pneumonia classification in healthcare," *J. Healthc. Eng.*, vol. 2019, Article ID 4180949, 2019.
- [176] J. P. Cohen, P. Bertin and V. Frappier, "Chester: A Web Delivered Locally Computed Chest X-Ray Disease Prediction System," *arXiv preprint arXiv:1901.11210*, 2019.
- [177] S. Rajaraman et al., "Visualization and interpretation of convolutional neural network predictions in detecting pneumonia in pediatric chest radiographs," *Appl. Sci.*, vol. 8, p. 1715, 2018.
- [178] M. Toğaçar, B. Ergen and Z. Cömert, "A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models," *IRBM*, vol. 40, no. 5, 2019.
- [179] A. Saraiva et al., "Models of Learning to Classify X-ray Images for the Detection of Pneumonia using Neural Networks," 2019. [Online]. Available: <https://www.semanticscholar.org/paper/Models-of-Learning-to-Classify-X-ray-Images-for-the-Saraiva-Santos/0b8f202505b3d49c42fd45d86eca5dbd0b76fded>
- [180] E. Ayan and H. M. Ünver, "Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning," in *Proc. 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, Istanbul, Turkey, 2019, pp. 1–5.
- [181] T. Rahman et al., "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray," *Appl. Sci.*, vol. 10, p. 3233, 2020.
- [182] V. Chouhan et al., "A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images," *Appl. Sci.*, vol. 10, p. 559, 2020.
- [183] L. Li et al., "Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT," *Radiology*, vol. 296, Article 200905, 2020.

- [184] X. Mei et al., "Artificial Intelligence-enabled rapid diagnosis of patients with COVID-19," *Nat. Med.*, vol. 26, no. 8, pp. 1224–1228, 2020.
- [185] Q. Ni et al., "A deep learning approach to characterize 2019 coronavirus disease (COVID-19) pneumonia in chest CT images," *Eur. Radiol.*, vol. 30, no. 12, pp. 6517–6527, 2020.
- [186] P. Rajpurkar et al., "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.
- [187] J. Chen et al., "Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography," *Sci. Rep.*, vol. 10, p. 19196, 2020.
- [188] S. Harmon et al., "Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets," *Nat. Commun.*, vol. 11, p. 4080, 2020.
- [189] B. Liu et al., "Online COVID-19 diagnosis with chest CT images: lesion-attention deep neural networks," *medRxiv*, 2020.
- [190] C. Zheng et al., "Deep learning-based detection for COVID-19 from chest CT using weak label," *medRxiv*, 2020.
- [191] Y. Pathak et al., "Deep transfer learning based classification model for COVID-19 disease," *Ingénierie et Recherche Biomédicale*, 2020.
- [192] V. Ravi, V. Acharya and M. Alazab, "A multichannel EfficientNet deep learning-based stacking ensemble approach for lung disease detection using chest X-ray images," *Cluster Comput.*, vol. 26, pp. 1181–1203, 2023.
- [193] N. Sarada and K. Rao, "A neural network architecture using separable neural networks for the identification of pneumonia in digital chest radiographs," *IJeC*, vol. 17, pp. 89–100, 2021, doi: 10.4018/IJeC.2021010106.

- [194] P. Yi, T. Kim and C. Lin, "Generalizability of deep learning tuberculosis classifier to COVID-19 chest radiographs: new tricks for an old algorithm?," *J. Thorac. Imaging.*, vol. 35, pp. W102–W104, 2020, doi: 10.1097/RTI.0000000000000532.
- [195] M. Gour and S. Jain, "Automated COVID-19 detection from X-ray and CT images with stacked ensemble convolutional neural network," *Biocybern. Biomed. Eng.*, vol. 42, no. 1, pp. 27–41, 2022, doi: 10.1016/j.bbe.2021.12.001.
- [196] G. M. M. Alshmrani et al., "A deep learning architecture for multi-class lung diseases classification using chest X-ray (CXR) images," *Alex. Eng. J.*, vol. 64, pp. 923–935, 2023.
- [197] Nahiduzzaman et al., "Parallel CNN-ELM: A multiclass classification of chest X-ray images to identify seventeen lung diseases including COVID-19," *Expert Syst. Appl.*, vol. 229, p. 120528, 2023.