

BAŞKENT ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ YÜKSEK LİSANS PROGRAMI

DERİN ÖĞRENME YÖNTEMİYLE KAPALI ALANLARDA
SUÇ DAVRANIŞLARININ TESPİTİNE YÖNELİK BİR UYGULAMA

YÜKSEK LİSANS TEZİ

HAZIRLAYAN
EMEL BAĞDATOĞLU

ANKARA - 2020

BAŞKENT ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ YÜKSEK LİSANS PROGRAMI

DERİN ÖĞRENME YÖNTEMİYLE KAPALI ALANLARDA
SUÇ DAVRANIŞLARININ TESPİTİNE YÖNELİK BİR UYGULAMA

YÜKSEK LİSANS TEZİ

HAZIRLAYAN
EMEL BAĞDATOĞLU

TEZ DANIŞMANI
PROF. DR. MURAT PAŞA UYSAL

ANKARA - 2020

BAŞKENT ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: 27 / 08 / 2020

Öğrencinin Adı, Soyadı: Emel BAĞDATOĞLU

Öğrencinin Numarası: 21810212

Anabilim Dalı: Yönetim Bilişim Sistemleri

Programı: Yönetim Bilişim Sistemleri

Danışmanın Unvanı/Adı, Soyadı: Prof. Dr. Murat Paşa UYSAL

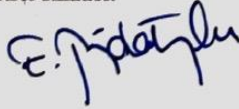
Tez Başlığı: Derin Öğrenme Yöntemiyle Kapalı Alanlarda Video Görüntülerinden Suç Tespiti Üzerine Bir Uygulama

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 116 sayfalık kısmına ilişkin, 09/08/2020 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezin benzerlik oranı % 9'dur. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

"Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını" inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:



ONAY

Tarih: 27 / 08 / 2020

Prof. Dr. Murat Paşa UYSAL



TEŞEKKÜR

Tez çalışmam boyunca akademik ve manevi desteğini sürekli hissettiren danışman hocam sayın Prof. Dr. Murat Paşa UYSAL’a; veri seti hazırlamadaki yardımlarından dolayı Servet YILMAZ, Mustafa Mert KESKİN ve Ayşegül ÖZDEMİR’e; kod yazımındaki yardımları, literatür araştırmaları sırasında yol göstermesi, tez yazım sürecinde tüm sevgisi ve sabrıyla yanımda olan sevgili eşim Muhammed YILMAZ’a; her zaman yanımda olan ve uygun çalışma ortamımı hazırlayan annem Ülkü KANGAL’a ve aileme sonsuz teşekkürlerimi sunarım.

Emel BAĞDATOĞLU

Ankara, 2020

ÖZET

Suç ve suça ilişkin olaylar doğası gereği, geçmişte insan ve toplum hayatı için her zaman önemli olgular içerdiği gibi gelecekte de önemli etkilerinin olacağı açıkça gözlenmektedir. Hukuki yaptırımlara ek olarak sınırlı düzeydeki teknolojik uygulamalar suça ilgili olayları belirli oranlarda azaltsa da beklenen gereksinimleri karşılamaktan uzak olduğunu söylemek mümkündür. Bunun en önemli nedenlerinden birisi, suç teşkil eden olayların gerçekleştikten sonra asenkron ve manuel olarak incelenmeleri, teknolojik araçların ise çoğu zaman bu olaylara ait verinin kaydı ve işlenmesi için kullanılması olduğu söylenebilir. Dolayısıyla, olayları gerçek zamanlı izleyecek, otomatik biçimde analiz ederek suça ilişkin davranışları anında tespit edecek akıllı teknolojik çözümlerin suç davranışlarını azaltacağı gibi suçun caydırıcılığını da artıracığı değerlendirilmektedir.

Bu kapsamdaki çalışmanın temel amacı, kapalı alanlarda güvenlik kameralarından (CCTV) elde edilen video görüntülerini kullanarak suç davranışlarını gerçek zamanlı belirlemeye yönelik akıllı bir çözüm üretmektir. Araştırma, Tasarım Bilimi Araştırma Yöntemi (Design Science Research) çerçevesinde derin öğrenme yöntem, teknik ve araçları kullanılarak bütünsel bir yapıda yürütülmüştür. Araştırmanın kuramsal temelleri, suç ve suç tespit yöntemleri, yapay zeka, yapay sinir ağları, derin öğrenme ile görüntü işleme bilgi alanlarıdır. Çalışmanın ana çıktısını, gerçekleştirilen derin öğrenme çözümü ile onun içerdiği özgün modeller ve algoritmalar oluşturmaktadır. Araştırma probleminin çözümü için suç davranışları belirlenmiştir. Bunlar kavga, koşma, silah doğrultma, atlama, el kaldırma, yerde uzanma, eylemsizlik ve diğer olmak üzere sekiz alt kategoriye ayrılarak, güvenlik kameralarından alınmış 617 gerçek video görüntüsünden oluşan Güvenlik Kamerası Suç Veri Seti oluşturulmuştur. Çalışma sonucunda gözetimsiz öğrenme modellerinin gözetimli öğrenme modellerine nazaran suç tespitinde daha etkili olduğu bulunmuştur. Elde edilen bulgular ve gözlemler doğrultusunda, derin öğrenme ile geliştirilen çözümün araştırmanın amaç ve hedeflerini karşıladığını söylemek mümkündür. Ancak, araştırma sınırlılıkları dikkate alındığında ise gelecekteki çalışmalarla iyileştirilmesi gereken yönlerinin bulunduğunu da belirtmek gerekmektedir.

Anahtar Kelimeler: Derin Öğrenme, Suç Tespiti, Yapay Zeka, Anomali Tespiti, Görüntü İşleme

ABSTRACT

Due to their nature, crime and criminal events have always been important for human and social life in the past, and it is clearly observed that they would have important effects in the future. In addition to legal sanctions, it is possible to say that limited technological applications reduce the crime-related incidents to a certain extent, however, they are far from meeting the expected requirements. One of the most important reasons for this is that criminal events are examined asynchronously and manually after they occur, and technological tools are often used for recording and processing the data belonging to these events. Therefore, it is considered that smart technological solutions that would monitor the events in real time and automatically analyze the criminal behaviors may reduce crimes and also increase the deterrence for crime.

The main purpose of the study is, therefore, to produce a smart solution for detecting criminal behavior in real time by using video images obtained from security cameras (CCTV) in closed areas. An integrated research is carried out using deep learning methods, techniques and tools within the framework of design science research method. The theoretical foundations of the research are crime and crime detection methods, artificial intelligence, artificial neural networks, deep learning and image processing knowledge areas. The proposed deep learning solution and its original models and algorithms constitute the main outputs of this study. Additionally, criminal behaviors have been determined for the solution of the research problem. These are fighting, running, aiming a gun, jumping, raising hands, lying on the ground, inaction positions and others. Security Camera Crime Data Set consisting of 617 real video images taken from security cameras has also been created. As a result of the study, it is found that unsupervised learning models are more effective in crime detection than supervised learning models. In line with the findings and research observations, it is possible to say that the solution, developed using deep learning methods, meets the aims and objectives of the research. However, considering the limitations of the research, it should be noted that there are some aspects that need improvement with future studies.

Keywords: Deep Learning, Crime Detection, Artificial Intelligence, Anomaly Detection, Image Processing

İÇİNDEKİLER

TEŞEKKÜR.....	i
ÖZET.....	ii
ABSTRACT	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	viii
ŞEKİLLER LİSTESİ.....	ix
KISALTMALAR VE SİMGELER LİSTESİ.....	xi
1. GİRİŞ.....	1
1.1. Problem ve Motivasyon.....	1
1.2. Önerilen Çözüm ve Katkıları.....	2
1.3. Tez Taslağı	3
2. KURAMSAL ÇERÇEVE.....	5
2.1. Suç Kavramı ve Suç Analizi	5
2.1.1. Suç analizi.....	5
2.1.2. Suç tespiti ve suçun önlenmesinde güvenlik kameralarının rolü	6
2.2. Yapay Zeka	7
2.3. Makine Öğrenmesi	8
2.3.1. Öğrenme yöntemleri	8
2.3.1.1. Gözetimli öğrenme.....	8
2.3.1.2. Yarı kontrollü öğrenme.....	9
2.3.1.3. Gözetimsiz öğrenme.....	9
2.3.2. Hata fonksiyonları	9
2.3.2.1. Ortalama kare hatası.....	11
2.3.2.2. Ortalama mutlak hatası	11
2.3.2.3. Çapraz entropi	12
2.3.3. Metrikler	12
2.3.3.1. Karmaşıklık matrisi	13
2.3.3.2. Doğruluk	14
2.3.3.3. Kesinlik.....	15
2.3.3.4. Hassasiyet	15

2.3.3.5.	F1-skoru	16
2.3.3.6.	AUC-ROC eğrisi.....	16
2.3.4.	Model geçerleme	18
2.3.4.1.	Genelleştirme	18
2.3.4.2.	Erken durdurma	20
2.3.4.3.	Çapraz geçerleme	21
2.3.4.4.	Model geçerleme süreci.....	22
2.3.5.	Geri yayılım.....	23
2.3.5.1.	Gradyan iniş	24
2.3.5.2.	Momentum.....	26
2.3.5.3.	Eniyileştiriciler.....	26
2.4.	Derin Öğrenme.....	27
2.4.1.	Yapay sinir ağları.....	29
2.4.1.1.	Perceptron.....	29
2.4.1.2.	Çok katmanlı perceptron.....	30
2.4.1.3.	Tamamen bağlı sinir ağı	31
2.4.2.	Aktivasyon fonksiyonları	32
2.4.2.1.	Adım fonksiyonu.....	32
2.4.2.2.	İşaret fonksiyonu.....	33
2.4.2.3.	Sigmoid fonksiyonu.....	33
2.4.2.4.	Tanh	35
2.4.2.5.	Doğrusal fonksiyon.....	35
2.4.2.6.	ReLU fonksiyonu	36
2.4.2.7.	Softmax fonksiyonu.....	37
2.4.3.	Evrişimsel sinir ağı.....	38
2.4.3.1.	Konvolüsyonel katman.....	39
2.4.3.2.	Havuz katmanı	40
2.4.3.3.	Düğüm atma	41
2.4.3.4.	Yığın normalizasyonu	42
2.4.3.5.	Tamamen bağlı katman	42
2.4.4.	Özkodlayıcılar	42
2.4.5.	Özyinelemeli sinir ağı.....	44
2.4.5.1.	LSTM.....	47
2.4.5.2.	ConvLSTM	49

2.5.	Video Verisi ve Video Anomali	50
2.5.1.	Görüntü verisi	51
2.5.1.1.	Renk modeli ve RGB	51
2.5.1.2.	Video verisi	52
2.5.1.3.	Görsel veri ön işleme	52
2.5.2.	Video anomali türleri	53
2.5.3.	Derin öğrenme ile videoda anomali tespiti.....	54
3.	LİTERATÜR TARAMASI	56
3.1.	Veri Setleri.....	57
3.1.1.	Anormal insan davranışı ile ilgili veri setleri.....	57
3.1.2.	Kavga tespiti ile ilgili veri setleri	60
3.2.	Derin Öğrenme Modelleri	62
3.2.1.	Anomali tespiti	62
3.2.2.	Video görüntülerinden anormal insan davranışı tespiti	64
3.2.2.1.	Konvolüsyonel özkodlayıcı (CAE)	65
3.2.2.2.	RCNN-AE.....	66
3.2.2.3.	ConvLSTM-AE.....	67
3.2.2.4.	Two-Stream CNN-AE	69
3.2.3.	Video görüntülerinden kavga tespiti	70
3.2.3.1.	CNN ve LSTM hibriti modeller	71
3.2.3.2.	Öğrenme aktarımı ile geliştirilen modeller	72
4.	YÖNTEM	78
4.1.	Araştırma Modeli.....	79
4.2.	Tasarım, Geliştirme ve Uygulama.....	80
4.2.1.	Aşama 1: Problem analizi ve ihtiyaç tanımlama.....	82
4.2.2.	Aşama 2: Derin öğrenme platformunun hazırlanması.....	83
4.2.2.1.	Veri seti için video toplanması	83
4.2.2.2.	Derin öğrenme platformunun hazırlanması ve kütüphanelerin belirlenmesi.....	83
4.2.2.3.	Verilerin işlenmesi ve etiketlenmesi için platformun hazırlanması, kütüphanelerin belirlenmesi	85
4.2.2.4.	Problemin değerlendirme ölçütlerinin ve çözüm hedeflerinin belirlenmesi.....	86
4.2.3.	Aşama 3: Veri seti oluşturma	87
4.2.4.	Aşama 4: Derin öğrenme model ve yöntemlerinin geliştirilmesi	95

4.2.4.1.	Girdi formatı seçimi	96
4.2.4.2.	Çoklu girdi ile eğitme	98
4.2.4.3.	Gözetimsiz eğitme modeli	101
4.2.4.4.	Gözetimli eğitme modelleri	104
4.3.	Geçerlik ve Güvenirlik	108
4.4.	Çalışmanın Sınırlılıkları	108
5.	TARTIŞMA	110
5.1.	Tek Girdi ve Çoklu Girdi Yöntemlerinin Performanslarının Karşılaştırılması	110
5.2.	Gözetimsiz Öğrenme Modeli Performansını Geliştirme	111
5.2.1.	Uzaklık hesaplamadan önce havuz katmanı kullanma	112
5.2.2.	Eğitirken suç verisini kullanma	112
5.3.	Gözetimli ve Gözetimsiz Öğrenme Modellerin Performanslarının Karşılaştırılması	113
6.	SONUÇ VE ÖNERİLER	115
EKLER		117
KAYNAKLAR		132

TABLÖLAR LİSTESİ

Tablo 1: Karmaşıklık Matrisi Örneği	14
Tablo 2: Başlıca Anormal İnsan Davranışı Algılama Veri Setleri	59
Tablo 3: Başlıca Kavga Algılama Veri Setleri	62
Tablo 4: AE Tabanlı Başlıca Anomali Tespiti Modellerinin Çeşitli Veri Setlerindeki Performansları (AUC)	70
Tablo 5: Başlıca Kavga Tespiti Modellerinin Çeşitli Veri Setlerindeki Performansları (Doğruluk)	77
Tablo 6: Tasarım, Geliştirme ve Uygulama Adımları	81
Tablo 7: Güvenlik Kamerası Suç Verilerinin Gruplara Göre Dağılımı.....	90
Tablo 8: Güvenlik Kamerası Suç Veri Seti ile Başlıca AİDA ve Kavga Veri Setlerinin Karşılaştırılması.....	95
Tablo 9: Eşit Aralıklı Örnekleme (EAÖ) ve Dinamik Örnekleme (DÖ) ile Girdi Oluşturma Yöntemlerinin Performans Karşılaştırması (Doğruluk)	110
Tablo 10: Çoklu Girdi ile Eğitilen Modellerin Girdi Çerçeve Sayısına ve Karar Verme Algoritmasına Göre Performansları (Doğruluk)	111
Tablo 11: Gözetimli ve Gözetimsiz Öğrenme Modellerin Anomali Tespiti Performansı (AUC)	113
Tablo 12: Gözetimli ve Gözetimsiz Öğrenme Modellerin Suç Tespiti Performansı	114
Tablo 13: Gözetimsiz Öğrenme Modellerin Suç Tespiti Performansı	115
Tablo 14: Eşit Aralıklı Örnekleme ile Bir Videodan Bir Girdi Oluşturularak Eğitilen ve Test	117
Tablo 15: Dinamik Örnekleme ile Bir Videodan Bir Girdi Oluşturularak Eğitilen ve Test Edilen C3D Modelinin Girdi Çerçeve Sayısına Göre Suç Tespiti Performansı.....	117
Tablo 16: Girdi Çerçeve Sayısı 10 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı	118
Tablo 17: Girdi Çerçeve Sayısı 12 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı	118
Tablo 18: Girdi Çerçeve Sayısı 15 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı	119
Tablo 19: Girdi Çerçeve Sayısı 20 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı	119
Tablo 20: C3D MLP Tümeleştirme Modelinin Anomali Tespiti Performansı	120
Tablo 21: C3D LSTM Tümeleştirme Modelinin Anomali Tespiti Performansı.....	120
Tablo 22: ConvLSTM-AE Modelinin Anomali Tespiti Performansı	121
Tablo 23: Suç Verisi ile de Eğitilen ConvLSTM-AE Modelinin Anomali Tespiti Performansı.....	121

ŞEKİLLER LİSTESİ

Şekil 1: Akıllı Algoritmalar Hiyerarşisi	7
Şekil 2: Makine Öğrenmesi Modeli Eğitim Akış Şeması	11
Şekil 3: Karmaşıklık Matrisi	13
Şekil 4: İdeal Model için AUC-ROC Eğrisi	17
Şekil 5: AUC-ROC Eğrisi	17
Şekil 6: Eğitim Başarısına göre Model Karar Sınırı	19
Şekil 7: Seçilen Modelin Karmaşıklık ve Gerçek Başarı İlişkisi	20
Şekil 8: Eğitim Epok Sayısı ile Gerçek Başarı İlişkisi	21
Şekil 9: 10-Fold Çapraz Geçerleme ile Eğitim Örneği	22
Şekil 10: Model Geçerleme Süreci	23
Şekil 11: Gradyan İniş ile Eğitim Örneği	24
Şekil 12: ANN (Soldaki Model) ve DNN (Sağdaki Model) Örneği	28
Şekil 13: Perceptron Hücresi Yapısı	30
Şekil 14: İki Gizli Katmanlı İkili Sınıflandırıcı MLP Örneği	31
Şekil 15: Adım Fonksiyonu	33
Şekil 16: İşaret Fonksiyonu	33
Şekil 17: Sigmoid Fonksiyonu	34
Şekil 18: Hiperbolik Tanjant (Tanh) Fonksiyonu	35
Şekil 19: Doğrusal Fonksiyon	36
Şekil 20: ReLu Fonksiyonu	36
Şekil 21: 10 Sınıflı Bire Bir Kodlama Örneği	37
Şekil 22: Evrimsel Sinir Ağı Örneği	39
Şekil 23: Konvolüsyon Örneği	40
Şekil 24: Maksimum ve Ortalama Havuz Katmanı Örneği	41
Şekil 25: Özkodlayıcı Örneği	43
Şekil 26: CAE Örneği	44
Şekil 27: Gürültü Temizleme için AE Eğitimi Örneği	44
Şekil 28: RNN Hücresinin Çıktısının Zamansal Gösterimi	45
Şekil 29: Klasik RNN Hücresi Yapısı	45
Şekil 30: Çeşitli RNN Modelleri	46
Şekil 31: LSTM Hücresi Yapısı	47
Şekil 32: LSTM Hücresinin Girdi, Çıktı ve Unutma Kapıları	48
Şekil 33: ConvLSTM Hücresi Yapısı	49
Şekil 34: Gri Tonlamalı Görüntü ve Piksel Örneği	51
Şekil 35: RGB Görüntü Serisi (Video) Örneği	52
Şekil 36: Video Görüntülerinden Gözetimsiz Öğrenme ile Anomali Tespiti Yöntemi	64
Şekil 37: CAE Modeli (Ribeiro vd., 2018)	65
Şekil 38: RCNN-AE Modelinin Zamansal Gösterimi (Luo vd., 2017)	67
Şekil 39: C3D-AE Modeli (Hasan vd., 2016)	67
Şekil 40: ConvLSTM-AE Modeli (Chong ve Tay, 2017)	68
Şekil 41: Two-Stream CNN Modeli (Simonyan ve Zisserman, 2014)	69
Şekil 42: CNN Modeli (Tay vd., 2019)	71
Şekil 43: Two-Stream CNN Modeli (Cheng vd., 2020)	72
Şekil 44: Yüz Tanıma Problemi Öznitelikleri	73
Şekil 45: Aktarmalı Öğrenme Süreci Gösterimi	74

Şekil 46: Aktarmalı Öğrenme ile Nesne Tanıma Yapan CNN Modeli Örneği	75
Şekil 47: Kavga Tanıma Yapan Two-Stream CNN Modeli (Majhi vd., 2019).....	76
Şekil 48: Tasarım Bilimi Araştırma Yöntem ve Modeli (Uysal, 2017)	79
Şekil 49: Güvenlik Kamerası Suç Veri Seti Anomalileri Örneği	93
Şekil 50: Gözetimsiz Öğrenme Süreci Akış Şeması	96
Şekil 51: MLP (soldaki, Yeşil Katman: ReLu), ve LSTM (sağdaki, N: Girdi Boyutu) FCNN Modelleri.....	101
Şekil 52: Suç Tespiti için ConvLSTM-AE Modeli (Yeşil Katman: Tanh Katmanı)	104
Şekil 53: C3D Modeli (Yeşil Katman: Tanh Katmanı, Gri Katman: ReLu Katmanı)	107
Şekil 54: Kdenlive Ekran Görüntüsü.....	122
Şekil 55: Güvenlik Kamera Suç Veri Seti Anomali Sınıfları Analizi - 1. Kısım	122
Şekil 56: Güvenlik Kamera Suç Veri Seti Anomali Sınıfları Analizi - 2. Kısım	123
Şekil 57: Güvenlik Kamera Suç Veri Seti Anomali Sınıfları Analizi - 3. Kısım	123
Şekil 58: Güvenlik Kamera Suç Veri Seti Suç Verileri Analizi.....	124
Şekil 59: Güvenlik Kamera Suç Veri Seti Normal ve Suç Sınıfları Analizi	124
Şekil 60: Güvenlik Kamera Suç Veri Seti Çerçeveleri Örneği	125
Şekil 61: Derin Öğrenme Modeli Eğitimde Kullanılan Kütüphaneler	126
Şekil 62: Gözetimli Derin Öğrenme Modeli Keras Kodu	126
Şekil 63: Gözetimli Derin Öğrenme Modeli	127
Şekil 64: Gözetimsiz Derin Öğrenme Modeli Keras Kodu	128
Şekil 65: Gözetimsiz Derin Öğrenme Modeli	128
Şekil 66: Derin Öğrenme Modeli Eğitim Kodu.....	129
Şekil 67: Derin Öğrenme Modeli Testte Kullanılan Kütüphaneler	130
Şekil 68: Derin Öğrenme Modeli Test Kodu.....	130
Şekil 69: Derin Öğrenme Modeli Test Metodu	131

KISALTMALAR VE SİMGELER LİSTESİ

- Adam:** Adaptif Moment Tahmin Yöntemi (Adaptive Moment Estimation Method)
- AE:** Özkodlayıcı (Autoencoder)
- AI:** Yapay Zeka (Artificial Intelligence)
- AED:** Anormal Olay Algılama (Abnormal Event Detection)
- AİDA:** Anormal İnsan Davranışı Algılama
- ANN:** Yapay Sinir Ağı (Artificial Neural Network)
- AUC:** Eğri Altındaki Alan (Area Under The Curve)
- CAE:** Konvolüsyonel Özkodlayıcı (Convolutional Autoencoder)
- CCTV:** Kapalı Devre Televizyon (Closed-Circuit TeleVision)
- CE:** Çapraz Entropi (Cross Entropy)
- CNN:** Evrimsel Sinir Ağı (Convolutional Neural Network)
- DL:** Derin Öğrenme (Deep Learning)
- DNN:** Derin Sinir Ağı (Deep Neural Network)
- FCNN:** Tamamen Bağlı Sinir Ağı (Fully Connected Neural Network)
- FN:** Yanlış Negatif (False Negative)
- FP:** Yanlış Pozitif (False Positive)
- FPR:** Yanlış Pozitif Oranı (False Positive Rate)
- GPU:** Grafik İşlem Birimi (Graphic Processing Unit)
- HAR:** İnsan Davranışı Tanıma (Human Action Recognition)
- LSTM:** Uzun Ömürlü Kısa Dönemli Bellek (Long Short-term Memory)
- MAE:** Ortalama Mutlak Hata (Mean Absolute Error)
- ML:** Makine Öğrenmesi (Machine Learning)
- MLP:** Çok Katmanlı Perceptron (Multi-Layer Perceptron)
- MSE:** Ortalama Kare Hatası (Mean Squared Error)
- RAM:** Rastgele Erişimli Bellek (Random-Access Memory)
- ReLU:** Doğrusal Doğrultmaç Ünitesi (Rectified Linear Unit)
- RGB:** Kırmızı Yeşil Mavi (Red Green Blue)
- RNN:** Özyinelemeli Sinir Ağı (Recurrent Neural Network)
- ROC:** Alıcı İşletim Karakteristiği (Receiver Operating Characteristics)
- SGD:** Rastgele Gradyan İnişi (Stochastic Gradient Descent)

$\tanh$: Hiperbolik Tanjant (Hyperbolic Tangent)

TBAY: Tasarım Bilimi Araştırma Yöntemi

TN: Doğru Negatif (True Negative)

TP: Doğru Pozitif (True Positive)

TPR: Doğru Pozitif Oranı (True Positive Rate)

Σ : Toplam

n : Veri setindeki eleman sayısı

x_i : Veri setindeki i 'ninci girdi

y_i : Veri setindeki i 'ninci girdinin etiketi (veriye karşılık gelen parametrenin gerçek değeri)

$\hat{y}_i$: Modelin veri setindeki i 'ninci girdiye karşılık ürettiği çıktı (parametrenin tahmini değeri)

w : Yapay sinir ağının ağırlıkları

η : Öğrenme oranı

γ : Momentum Sabiti

J : Hata Fonksiyonu

∇ : Gradyan

θ : Değeri hesaplanan parametre

ϵ : Oldukça küçük pozitif reel sayı

1. GİRİŞ

Mağazalar (retail) ve ofisler, bulundukları sektör itibarıyla para alışverişinin yoğun olduğu ve müşteri giriş çıkışlarının yoğun olduğu yerlerdir (POPCENTER, 2007). Özellikle değerli metallerin ve lüks segment ürünlerinin satıldığı mağazalarda çeşitli hırsızlık ve suç teşkil eden olaylar sıkça gerçekleşebilmektedir (Jewelry Crime Statistics, 2018). Bulgular, kapalı mekanlardaki kapalı devre televizyon (Closed-Circuit Television, CCTV) sistemlerinin yaygınlaşmasıyla işlenen suçlarda tutarlı bir düşüş olduğunu göstermektedir (Piza vd., 2019). Ancak bu düşüşe rağmen, sadece Amerika Birleşik Devletleri'nde (ABD) değerli metal satan mağazalarda 2018 yılında, hırsızlık ve soygun nedeniyle 53.40 milyon dolarlık kayıp meydana gelmiştir (2018-JSA Annual Crime Report, 2018). Bu durumlar göz önünde tutulduğunda gözetleme ve alarm sistemlerinin yeteri kadar caydırıcı olmadığını söylemek mümkündür. Bu sistemlerin yetersiz kalma nedenlerinden birisi olarak hırsızlık, yankesicilik gibi olayların fark edildikten sonra görüntülerin manuel olarak incelenmesi gösterilebilir. Bu kapsamda, video görüntüleri aracılığıyla suç içeren davranışların gerçek zamanlı analiz edilerek, suçun gerçekleşmesi durumunda ilgililere ve yetkililere haber verilmesinin; olayların azaltılması ve caydırıcılığın artırılmasında önemli katkıları olabileceği düşünülmektedir. Literatür incelendiğinde söz konusu probleme yönelik çalışmaların sınırlı olduğu gözlenmektedir.

Çalışmanın temel amacı, kapalı alanlarda güvenlik kameralarından elde edilen video görüntülerini kullanarak suç davranışlarını gerçek zamanlı belirlemeye yönelik bir çözüm geliştirmektir. Bu kapsamda araştırmada gerçekleştirilen uygulamalar, elde edilen bulgular ve sonuçlar raporun sonraki bölümlerinde sunulmaktadır.

1.1. Problem ve Motivasyon

Kentsel suç tüm toplumlarda yaygın bir sorundur (Clemente ve Kleiman, 1977). ABD Adalet Bakanlığı Toplum Odaklı Polislik Hizmetleri Bürosu'nun yayınladığı rapora göre en yaygın suçların işlendiği on bir çevre bulunmaktadır. Bu çalışmada ise sadece tek bir çevrede gerçekleşen suçlar araştırmaya konu olmuştur. Nakit akışının yaşandığı müşterilerin yaya olarak veya arabayla giriş çıkış yapabildiği mağazalar (retail) (POPCENTER, 2007) araştırma için seçilen suç çevresidir.

1.2. Önerilen Çözüm ve Katkıları

Güvenlik kameraları görüntülerinin insanlar tarafından izlenerek suç takibi yapılması yerine, derin öğrenme ile geliştirilen çözümle video görüntülerinden suç tespiti yapılmasının daha etkili olabileceği düşünülmektedir. Ancak, bu raporun yazıldığı zaman dilimi içerisinde, literatürde suçları genel olarak tespit eden algoritmaların sınırlı sayıda olduğu, bu araştırmanın amaç ve hedeflerini karşılayabilecek algoritmanın ise bulunmadığı gözlenmiştir. Bu sorunun çözümüne yönelik ilk aşamada kapsamlı bir suç veri seti oluşturulmuştur. Bir sonraki adımda, derin öğrenme ile sadece çok özel senaryolarda suç değil, günlük hayattaki suçları tespit edecek bir yöntem geliştirilerek literatüre kazandırılmıştır.

Suçun çok kapsamlı bir tanımı bulunmaktadır. Tüm suçları tespit edebilmek için veri seti ve algoritma geliştirilmesi uzun soluklu bir çalışma gerektirir. Bu yüzden problemi düzgün bir çerçevede ele alacak aşamalı bir yaklaşım sergilemek gerekmektedir. Literatürde araştırma problemiyle doğrudan ilişkili ve detaylı bir çalışma olmadığından, ilk adımda olası davranış ve olay sayısı sınırlı bir ortam için suç tespiti yapmanın daha verimli ve etkili bir yaklaşım olacağı değerlendirilmiştir. Bu amaçla derin öğrenme ile kapalı ortamlardaki güvenlik kameralarından şiddet içeren suç tespiti yapan modeller geliştirilmiştir. Geliştirilen modeli uygulayan algoritmanın sadece güvenlik kameraları görüntülerinden suç tespiti yapması yeterli olabileceği araştırma kabulleri arasındadır. Dolayısıyla veri setinin sadece CCTV görüntülerinden oluşturulmasına karar verilmiştir. Bir ortamdaki birden çok kamera görüntüsü analiz edilerek daha iyi çıkarımlar yapılabilir, ancak başlangıçta çoklu görüntü ile çalışmak algoritma geliştirmeyi zorlaştıracak gibi önemli ölçüde kaynak kullanımını da gerektirmektedir. Bu yüzden veri setindeki her veri, sadece bir açıdan alınan kamera görüntülerinden oluşturulmuş ve kapalı ortamlarda gerçekleşen suçları tespit etmede kullanılması hedeflenmiştir. Pratikte insan tespiti algoritması kullanılarak, sadece insan içeren görüntüler analiz edilerek suç tespiti yapılabileceğinden, araştırmadaki veri seti insan içeren görüntülerden oluşmaktadır. Topluluk analizi farklı ve uyarılma gerektiren özel yöntem ve teknikler gerektirdiğinden, söz konusu durumu içeren videolar veri seti dışında bırakılmıştır. Kuyumcu mağazalarından değerli eşya aşırma gibi suçlar çok uzun bir zaman dilimine yayılabildiğinden veya o suçları işaret edecek delillerin tespiti zor olabileceğinden, sadece şiddet içeren suçları tespit etmek amaçlanmıştır. Bu sınırlamalar olsa bile çok sayıda farklı senaryo içeren veri seti ve bu veri seti ile eğitilmiş çeşitli suçların tespitini yapan bir model geliştirilmiştir. Bu çalışma ile literatüre aşağıdaki katkıların yapıldığını söylemek mümkündür:

- Suç davranışlarına yönelik farklı anomali sınıfları geliştirilmiştir.
- Çeşitli anomaliler bir veri seti çatısı altında toplanmış, bu anomalileri karşılıklı değerlendirme imkânı sunulmuştur
- Kavga, silah doğrultma, koşma gibi farklı unsurlar içeren kapsamlı ve büyük bir suç veri seti oluşturulmuştur
- Suç tespiti için anomali tespiti modelinin, sınıflandırma modelinden daha iyi sonuç verdiği gösterilmiştir.
- Derin öğrenme ile suç tespitinin başarıyla yapılabileceği gösterilmiştir.

1.3. Tez Taslağı

Tez çalışmasının geri kalan bölümleri şu şekilde organize edilmektedir:

- **2. bölümde** araştırmanın kuramsal çerçevesi 5 ana bölümde anlatılmaktadır. Bu bölümler:
 - Suç Kavramı ve Analizi: Suç, suç türleri ve unsurları açıklanmaktadır. Suç tespiti yöntemleri ve suç tespitinde güvenlik kameralarının rolünden bahsedilmektedir.
 - Yapay Zeka: Yapay zeka açıklanmaktadır.
 - Makine öğrenmesi: Makine öğrenmesi (Machine Learning, ML), ML modeli performansı değerlendirme yöntemleri ve ML modeli eğitme yöntemleri açıklanmaktadır. ML modeli geliştirilirken ve eğitilirken nelere dikkat edilmesi gerektiğinden bahsedilmektedir.
 - Derin öğrenme: Çok katmanlı perceptron (Multi-Layer Perceptron, MLP), evrimsel sinir ağı (Convolutional Neural Network, CNN), uzun ömürlü kısa dönemli bellek (Long Short-term Memory, LSTM), özkodlayıcı (Autoencoder, AE) gibi başlıca derin öğrenme modelleri detaylı olarak anlatılmaktadır.
 - Video Verisi ve Video Anomali: Video verilerinin bilgisayarda depolanma ve işleme yöntemi anlatılmaktadır. Video anomali açıklanmakta ve derin öğrenme ile tespit edilen video anomalilerinden bahsedilmektedir.
- **3. bölümde** literatürdeki video görüntülerinden anormal insan davranışı tespit etme ve suç tespit etme ile ilgili çalışmalardan bahsedilmektedir. Konu ile ilgili veri setleri ve veri setleri ile geliştirilen modeller açıklanmaktadır, bu modellerin performanslarının değerlendirilmesi ve karşılaştırılması yapılmaktadır.
- **4. bölümde** araştırmada uygulanan metodoloji anlatılmaktadır. Araştırma süreci

Tasarım Bilimi Araştırma Yöntemi (TBAY) kapsamında dört ayrı aşamada gerçekleştirilmiştir. Ayrıca, oluşturulan güvenlik kameraları suç veri seti, veri setini oluşturan anormal davranışlar ve videolar açıklanmıştır. Veri seti kullanılarak geliştirilen modeller detaylı şekilde anlatılmıştır. Gerçekleştirilen testler listelenmektedir.

- **5. bölümde** elde edilen bulgular yorumlanmaktadır. Modellerin performansları ve çeşitli yöntemler karşılaştırılıp tartışılmaktadır.
- **6. bölümde** Araştırmanın sonuçları, literatüre kazandırdıkları ve gelecekte yapılması planlanan çalışmalar ele alınmaktadır.



2. KURAMSAL ÇERÇEVE

Bu araştırmanın kapsamı kapalı ortamda gerçekleşen ve şiddet içeren suç davranışlarını, video görüntüleri kullanarak derin öğrenme yöntemleriyle tespit etmektir. Çalışmanın dayandığı kuramsal çerçeve ve ilgili bilgi alanları aşağıdaki başlıklarda sunulmuştur.

2.1. Suç Kavramı ve Suç Analizi

Suç, kanunlar tarafından suç olarak düzenlenen eylemlerin fiilen gerçekleştirilmesidir. Suçun işlenmesi hukuki bir yaptırımın oluşmasına yol açmaktadır. Çünkü kanunla korunan ve ihlal edilmemesi yasalarla belirlenen bir emir ihlal olunmuştur (Soyaslan, 1998). ABD Adalet Bakanlığı Toplum Odaklı Polislik Hizmetleri Bürosu'nun yayınladığı rapora göre en yaygın suçların işlendiği 11 çevre vardır ve bunlar; ikamet, dinlenme-eğlenme yerleri, ofisler, mağazalar, sanayi, tarım, eğitim ve insan hizmetleri, kamu yolları, ulaşım, açık geçiş alanları başlıkları altında tanımlanmaktadır (POPCENTER, 2007).

Suçu iki ayrı başlık altında inceleyebiliriz: Suç analizi ve suçla mücadele. Suç analizi ile suçun toplumsal ve ekonomik etkileri; suçlu, toplum ve kolluk kuvvetleri arasındaki etkileşim incelenip değerlendirilebilir. Suçla mücadele kapsamında ise, suçun tespiti, suça müdahale, caydırıcılık ve nihayetinde suçun gerçekleşmesinin önlenmesi amaçlanır (Karakehya, 2009). Suçun topluma olan maliyetlerini asgariye indirmek, suçla mücadelede büyük önem arz etmektedir.

2.1.1. Suç analizi

Bireylerin çıkarları doğrultusunda gerçekleştirdiği tutumlarla birlikte sosyal bir toplum halinde yaşamının getirdiği çeşitli çelişkiler ve uyumsuzluklar nedeni ile suç ilk çağlardan itibaren var olmuştur ve var olmaya da devam edecektir (Dağ, 2003). Suçun insan doğası, psikolojik ve sosyolojik koşullar nedeniyle öngörülemez olduğu düşünülmektedir. Mutlak şekilde rastgele olmamakla birlikte, belirli bir mekân veya zamanda tutarlı bir şekilde gerçekleşmemektedir (Yu vd., 2011). Suçun ekonomik maliyetleri incelendiğinde özellikle üç kesimin bu maliyetlere maruz kaldığı yapılan araştırmalarda ifade edilmektedir (Ünver,

2008). Suç işlemeye eğilim gösterenler caydırıcı ve cezalandırıcı maliyetlere, işlenen suçta mağdur olanlar doğrudan maliyetlere sahiplerdir. Devletler ve hükümetler için en temel maliyet ise güvenlik maliyetleridir (Gibbons, 2004). Bu nedenle suçun analizi ve/veya tespiti başta devletler, kolluk kuvvetleri ve istihbarat birimleri olmak üzere birçok farklı paydaşı ilgilendiren önemli bir güvenlik ve ekonomi problemidir. Bu maliyetleri asgari düzeye indirebilmek ve suçun gerçekleşme nedenlerini anlayabilmek için suç analizinin yapılması gerekmektedir.

Suç analizi, suç ve verdiği her türlü rahatsızlık sorunlarının yanı sıra, polisin cezai kaygı, suç ve düzensizliğin azaltılması, suçun önlenmesi ve değerlendirilmesinde yardımcı olmak için sosyo-demografik, mekânsal ve zamansal faktörler dahil olmak üzere polisle ilgili diğer konuların sistematik olarak incelenmesidir (Brown, 1998). Geçmişte suç analizi ve tespiti için geleneksel yöntemlerin kullanılması yeterli olsa da gelişen teknoloji ve değişen çevresel koşullar nedeni ile multidisipliner yaklaşımların ve güncel teknolojilerin kullanılması önem arz etmektedir.

2.1.2. Suç tespiti ve suçun önlenmesinde güvenlik kameralarının rolü

Çoğu ülkede suçun tespiti polisin sorumluluğundadır, ancak özel kolluk kuvvetleri de belirli suç türlerinin keşfinden sorumlu olabilir. Türkiye’de buna örnek olarak Millî İstihbarat Teşkilâtı ve Gümrükler Muhafaza Genel Müdürlüğü gösterilebilir (Çolak, 1998). Suç tespiti üç ayrı aşamaya ayrılır: Bir suçun işlendiğinin keşfi, bir şüphelinin tanımlanması ve şüpheliyi mahkemeye çıkarmak için yeterli kanıtların toplanması (Encyclopedia Britannica). Birçok suç, polis dışındaki kişiler (ör. mağdurlar veya tanıklar) tarafından tespit edilir ve bildirilir. Bu durum suçun işlendiği zaman ve bu suçun keşfinin arasında zaman kaybına neden olmaktadır ve suç tespitinde diğer aşamalara geçişi zorlaştırmaktadır.

Rasyonel seçim modeli kapsamında polis, tutuklanma ve ceza olasılığını artırarak suçu azaltır (Welsh vd., 2009). Ancak bunu başarmanın tek yolu daha büyük bir polis gücüne sahip olmak değildir. Güvenlik kamerası sistemlerinin kullanımıyla tespit ve cezalandırma süreçlerini hızlandırarak suçu gerçekleştirecek kişi caydırılabilir. Son yıllarda güvenlik kameralarının hızla yaygınlaşması sonucunda, birçok araştırmacı gözetim kameralarının kullanımının suçu azaltıp azaltmadığını araştırmaktadır. Kriminologlar Brandon Welsh ve David Farrington, güvenlik kameralarının suç üzerindeki etkisi üzerine mevcut 44 çalışmayı gözden geçirerek kanıtların karışık olduğunu bildirmiştir: Mevcut çalışmaların 15’inde kameraların suçu önemli ölçüde azalttığı, 3’ünde kameraların suçu önemli ölçüde artırdığı

ve 23'ünde ise kameraların suçun üzerinde önemli bir etkisinin olmadığı belirtilmektedir (Welsh vd., 2009). Fakat bu çalışmaların sınırlamalarını da belirtilmektedir: Örneğin bir otoparka yeni güvenlik kameraları kurduğunda, bu değişikliğe ek güvenlik personeli ve gelişmiş sokak aydınlatması eşlik etmektedir. Sonuç olarak, güvenlik kameralarının etkisini diğer suç önleme tedbirlerinin etkilerinden ayırmak zorlaşmaktadır.

Güvenlik kameralarının etkin olmama nedenleri arasında kameraların her anının insan kaynakları ile manuel kontrol edilmesi gelmektedir. Bazı yer ve bölgelerde, monitörlere yansıtılan kamera görüntüleri bir güvenlik görevlisi tarafından izlense de dikkat dağınık unsurlar nedeni ile tüm görüntüleri sağlıklı izleme ve inceleme imkanları ortadan kalkmaktadır. Bir suç gerçekleştikten sonra gelen ihbar üzerine varsa gün ve saat bazında inceleme yapılarak, zaman belirtimi yoksa tüm görüntüler izlenerek suç tespiti ve suçlunun yakalanması gerçekleştirilmeye çalışılmaktadır. Bu durum da caydırıcılığı azaltmaya ve zaman kaybına yol açmaktadır.

2.2. Yapay Zeka

Bir algoritmanın rasyonel karar vermesi veya bir algoritmanın kontrol ettiği robotun rasyonel davranması yeteneğine yapay zeka (Artificial Intelligence, AI) denir (Wolfgang, 2011). Yapay zeka, kısaca insan gibi çıkarımlar yapma ve davranma yeteneğidir. Klasik algoritmalar girdiden, önceden belirtilen parametreler çerçevesinde hesaplamalar ve arama yaparak, bir sonuca ulaşmaya çalışırlar. Aksi takdirde hata verirler. Akıllı algoritmalar girdi ve içinde bulunduğu koşullar sonucu olası en doğru sonucu verir. Bu açıdan klasik algoritmalara göre daha esnek ve girdideki küçük değişikliklere karşı daha dirençlidir (Nilsson, 2014).



Şekil 1: Akıllı Algoritmalar Hiyerarşisi

2.3. Makine Öğrenmesi

Yapay zekanın çok büyük bir araştırma ve uygulama alanı ile çeşitli alt dalları bulunmaktadır. Alt dallarından biri de makine öğrenmesidir (Machine Learning, ML). Makine öğrenmesinde önceden belirlenen model, veriden kendiliğinden zamanla çıkarım yapmayı öğrenir. Sonrasında yeni bir girdi verildiğinde, o girdiyi önceden hiç öğrenmediyse bile doğru sonucu verir. Bu özelliği sayesinde ML, standart AI yöntemlerinden çok daha başarılı sonuçlar vermektedir (Alpaydin, 2020). Makine öğrenmesinde problem ve onun çözüm uzayı çok karmaşık olmasına rağmen model basit şekilde ifade edilebilir. Çözümü doğrudan bulmak zordur, ancak modele girdi verilir ve çıktıyı doğru hesaplaması beklenir. Eğer model çıktıyı yanlış verirse, parametreler çıktıyı doğru sonuca yaklaştıracak şekilde belirli ölçüde değiştirilebilir. Bu işlem veri setindeki tüm veriler için ve tüm veri seti için defalarca gerçekleşir. Bu süreç sonunda ML modeli problemi öğrenmiş (eğitilmiş) olur (Kubat, 2017). ML'nin en önemli gücü, problem ne kadar karmaşık olursa olsun, tıpkı bir insanın zamanla tecrübe kazanması gibi problemin çözümüne yönelik doğru çıkarımları ve sınıflamaları yapabilmeyi veriden öğrenebilmesidir.

Makine öğrenmesi ve onun alt dalı olan derin öğrenme ve uygulamalarının anlaşılması için önemli ML terminolojisi sonraki bölümlerde açıklanmaktadır.

2.3.1. Öğrenme yöntemleri

Makine öğrenmesinde verilerin ve problemin tipine göre birden çok öğrenme yöntemi bulunmaktadır. Bu öğrenme yöntemlerinden en çok kullanılanları gözetimli öğrenme, yarı kontrollü öğrenme ve gözetimsiz öğrenmedir. Bu durumlar için ayrı ayrı geliştirilen teknikler alt başlıklarda açıklanmaktadır.

2.3.1.1. Gözetimli öğrenme

Makine öğrenmesi problemlerinin çoğunda gözetimli öğrenme (Supervised Learning) yöntemleri kullanılmaktadır. Gözetimli öğrenme girdilerin x , çıktıların ise y olarak gösterildiği bir problemde bir $f(x) = y$ fonksiyonu bulma işlemi olarak düşünülebilir. Burada x 'ler veri üzerinden elde edilen değerler, y ise çıktıda eşleştirilecek olan ya da elde edilecek değerler olarak düşünülmektedir (Shalev-Shwartz ve Ben-David, 2014). Buradaki amaç olabilecek en iyi $f(x) = y$ eşitliğini bulmaktır, fakat bu her zaman mümkün olmayacağı için genellikle bir hata payı oluşmaktadır.

Bu öğrenme yöntemine gözetimli öğrenme denilmektedir çünkü burada model bir öğretmenin gözetimiyle eğitiliyormuş gibi düşünülebilir. Bu yöntemde eğitim verilerine karşılık gelen çıktılar önceden tek tek belirlenir (etiketlenir). Model eğitilirken girdiye karşılık verdiği cevabın doğru veya yanlış olduğu geri bildirimi ile sürekli kendini güncelleyerek geliştirir.

2.3.1.2. Yarı kontrollü öğrenme

Yarı kontrollü öğrenme (Semi-Supervised Learning) yönteminde verilerin tamamı bir etikete sahip değildir. Bazı girdilere karşılık gelen çıktılar etiketlenmiş iken, bazı girdiler ise etiketlenmemiştir (Michie, 1994). Yarı kontrollü öğrenmede amaç genel olarak var olan etiketleri kullanarak eksik olan etiketleri tamamlamaktır. Bu süreç sonunda verinin tamamı eğitim için kullanılır.

Birçok makine öğrenmesi problemi bu alt başlıkta incelenebilir çünkü genellikle makine öğrenmesi problemlerinde verilerin etiketlenmesi zordur. Veri sayısı milyonlara ulaşınca verilerin tamamını etiketlemek yerine bu tarz bir yaklaşım ile eğitmek daha verimli olabilmektedir (Goodfellow vd., 2016).

2.3.1.3. Gözetimsiz öğrenme

Gözetimsiz öğrenme (Unsupervised learning) yönteminde herhangi bir veri etiketi bulunmamaktadır. Dolayısıyla verinin hangi gruba ait olduğu bilinmemektedir. Bu öğrenme yönteminde amaç verinin dağılımından hangi verilerin bir grup oluşturabileceğini düşünerek eğitim yapmaktır (Goodfellow vd., 2016). Gözetimsiz öğrenme olarak geçmektedir çünkü verilerin etiketi yoktur. Gözetimli öğrenimdeki gibi kesin cevap veya bir öğretmen yoktur. Problemin türüne göre farklı gözetimsiz öğrenme yöntemleri bulunmaktadır.

2.3.2. Hata fonksiyonları

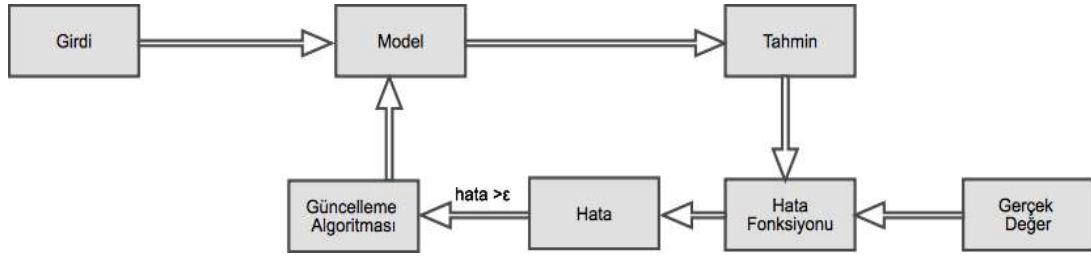
Veriden analiz sonucu yapılan çıkarımlar genel olarak iki kategoriye ayrılır: Bunlar regresyon ve sınıflandırma (classification) problemleridir. Sınıflandırma probleminde girdinin önceden belirlenen kategorilerden hangisine ait olduğu belirlenmeye çalışılır (Goodfellow vd., 2016). Örneğin bir suç tanıma yazılımında soygun, kundaklama, cinayet gibi farklı suç türleri önceden belirlenebilir. Sonrasında bir suçun hangi kategoriye ait olduğu

belirlenmeye çalışılır. Burada N tane farklı sınıf varsa, sınıflar 0, 1, 2, ..., N-1 şeklinde tam sayılarla etiketlenir. Suç tespiti problemi ‘suç yok’ ve ‘suç var’ gibi iki sınıflı (ikili, binary) sınıflandırma problemi olarak düşünülebilir. Bu durumda sınıf etiketleri 0 ve 1 olur.

Regresyon problemi, girdiye ait bir parametrenin değerinin makina öğrenme ile bulunmasıdır (Goodfellow vd., 2016). Parametre gerçek (real) ve sürekli (continuous) değer alabilir. Bir evin satış fiyatı tahmini, bir filmin gişe hasılatı tahmini, ertesi günün ortalama sıcaklığı tahmini regresyon problemine örnek olarak verilebilir. Burada çıktı, olası sonsuz değer alabilir. Dolayısıyla model çıktısının doğru değerle aynı olmasını kontrol etmek gerçekçi olmayabilir. Modelin çıktısı, gerçek değere ne kadar yakınsa o kadar iyi olduğu sonucuna varılabilir. Bu yakınlık farklı matematiksel formüllerle tanımlanabilir. Bahsedilen yakınlık değerini veren formüle hata fonksiyonu (loss function) denir (Zhang vd., 2019). Bu değere hata da denebilir.

Makine öğrenmesi modeli tasarlandıktan sonra, belirlenmesi gereken parametrelere rastgele ilklendirme (initialization) yapılır. Model eğitilir. Eğitilirken amaç, eldeki veriler için modelin hata fonksiyonu sonucu verdiği değeri minimize etmektir. Böylece, sonrasında gerçek hayattaki verilerle kullanılınca da hata fonksiyonu sonucunun düşük olması, yani modelin gerçek değer veya gerçek değere yakın, sonuç vermesi beklenir (Goodfellow vd., 2016). Burada hata fonksiyonuna göre problemi çözen bir model geliştirildiği ve her bir problemin tanımlanması birbirinden farklı olduğundan dolayı eğitim esnasında kullanılacak hata fonksiyonları farklı olabilmektedir. Makine öğrenmesi modelinin eğitme algoritması şu şekilde açıklanabilir:

1. Modelin parametreleri rastgele ilklendirilir.
2. Verilerden önce çıktı sonra hata hesaplanır.
3. Hata miktarını daha az verecek şekilde parametreler güncellenir.
4. Hata miktarı değişimi ihmal edilecek kadar az olana kadar 2. ve 3. adımlar tekrar edilir.
5. Model veriler dışında test ve gerçek ortamda kullanılır. Bu ortamdaki hata modelin gerçek performansı olarak değerlendirilir.



Şekil 2: Makine Öğrenmesi Modeli Eğitim Akış Şeması

En yaygın hata fonksiyonları şöyledir:

2.3.2.1. Ortalama kare hatası

Ortalama kare hata (Mean Square Error, MSE) makine öğrenmesi problemlerinde en çok kullanılan hata fonksiyon tipidir. Burada hata değerlendirilirken gerçek değerler ve tahmin edilen değerler arasındaki farkların kareleri ortalaması hesaplanır (Bhardwaj vd., 2018). Farkın karesi alınarak bir sonucun gerçek değerden uzaklaştıkça daha çok cezalandırılması sağlanmaktadır. MSE şu şekilde hesaplanır.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Regresyon problemlerinde modelin performansı MSE ile ölçülebilir. Burada çıktılar, gerçek değere ne kadar yakınsa MSE de o kadar küçük olur. MSE hesaplandıktan sonra modelin parametrelerini hata miktarını azaltacak şekilde güncellemek için türev almak gerekmektedir. Bu yüzden hata fonksiyonunun türevlenebilir olması gerekmektedir (Zhang vd., 2019). MSE bunu sağlamaktadır.

Hata durumunu kare ile cezalandırmak genellikle işe yaramaktadır, fakat bazen kare ceza ağır olabilir. Bu duruma karşılık türev alması ve hesaplaması basit bir fonksiyon olduğu için yönlü türevlerin (gradient) hesaplanması diğer yöntemlere göre daha kolay hata tipidir.

2.3.2.2. Ortalama mutlak hatası

Ortalama mutlak hata (Mean Absolute Error, MAE) gerçek değer ile tahmin edilen değer arasındaki hatanın mutlak değerlerinin ortalamasıdır (Bhardwaj vd., 2018). Dolayısıyla iki değer arasındaki uzaklık anlamına gelmektedir. Bu cezalandırma kare kadar

güçlü değildir. Bu yüzden az cezalandırma gerektiren regresyon problemlerinde MSE yerine MAE tercih edilmektedir. Ortalama Mutlak Hata şu şekilde hesaplanır:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Mutlak hatayı hesaplamak için kareye göre daha çok hesaplama gerekmektedir. Ayrıca MAE'nin türevini hesaplamak MSE'ye göre daha zahmetlidir (Zhang vd., 2019). Bu yüzden araştırmalarda modelin performans ölçümünde MSE daha yaygın kullanılmaktadır.

2.3.2.3. Çapraz entropi

Çapraz entropi (Cross Entropy, CE), sınırlandırma problemlerinde modelin performansını değerlendirmek için hata fonksiyonu olarak sıkça kullanılmaktadır (Zhang vd., 2019). İki kategorili sınıflandırma problemi için çapraz entropi şu şekilde hesaplanır:

$$CE = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

Bir veri kümesinin bir elemanı için CE hesaplanması şu şekilde açıklanabilir. Eğer tahmin edilen değer 1 ise $y_i \log(\hat{y}_i)$ ifadesi 0 olur. Gerçek değer 1 ise $(1 - y_i)$ ifadesinden dolayı CE 0 olur. Gerçek değer 0 ise yani tahmin yanlışsa CE $\log(1 - \hat{y}_i)$ değerini alacaktır. Eğer tahmin edilen değer 0 ise $(1 - y_i) \log(1 - \hat{y}_i)$ ifadesi 0 olur. Gerçek değerde 0 ise $y_i \log(\hat{y}_i)$ ifadesinden dolayı CE 0 olur. Gerçek değer 1 ise yani tahmin yanlışsa CE $\log(\hat{y}_i)$ değerini alır (Bhardwaj vd., 2018).

Logaritma fonksiyonu yüksek hata payları için aşırı artacak bir fonksiyondur, dolayısıyla sınıflandırma problemleri için iyi sonuç vermektedir. Mükemmel bir modelin kayıp fonksiyonu genellikle 0 olmalıdır. Sınıflandırması yanlış yapılan girdi sayısı arttıkça, bununla orantılı olarak CE hatası da artar.

2.3.3. Metrikler

Hata fonksiyonları modeli optimize etmek için kullanılır. Eğitilen modelin

performansının değerlendirilmesi için de çeşitli metrikler kullanılır. Eğitim sonrasında oluşan model üzerinde yalnızca bir hata fonksiyonuna bakıp değerlendirmek genellikle yanlıştır. Bunun sebebi ise eğer modeller veriyi ezberlerse (memorize) hata fonksiyonu genellikle ek bir bilgi sunmaz. Farklı metrikler farklı bilgileri ifade eder. Bu yüzden model değerlendirilirken, uygulamanın amacına göre farklı metrikler kullanılır. Başlıca metrikler şöyledir:

2.3.3.1. Karmaşıklık matrisi

	GERÇEKLEŞEN: EVET	GERÇEKLEŞEN: HAYIR
TAHMİN: EVET	TP	FP
TAHMİN: HAYIR	FN	TN

Şekil 3: Karmaşıklık Matrisi

Makine öğrenmesi ve özellikle istatistiksel sınıflandırma problemlerinde kullanılan, hata matrisi olarak da bilinen karmaşıklık matrisi (Confusion Matrix), test verisi ile tahmin değerlerinin karşılaştırılmasını ve oluşturulan modelin performansının ölçülmesini sağlar (Zheng, 2015). Şekil 3’de görüldüğü gibi karmaşıklık matrisi dört değerden oluşur. TP (Gerçek Pozitif, True Positive) ve TN (Gerçek Negatif, True Negative) değerleri doğru değer sayısını verir. FP (Yanlış Pozitif, False Positive) ve FN (Yanlış Negatif, False Negative) değerleri yanlış değer sayısını verir. Yanıtı öğrenilmek istenen bir soru için;

- **TP:** Gerçek değer evet olduğunda eğitilen modelin evet cevabı vermesi durumu
- **TN:** Gerçek değer hayır olduğunda eğitilen modelin hayır cevabı vermesi durumu
- **FP:** Gerçek değer evet olduğunda eğitilen modelin hayır cevabı vermesi durumu
- **FN:** Gerçek değer hayır olduğunda eğitilen modelin evet cevabı vermesi durumu

Bu ölçütte FP ve FN değerleri ne kadar küçük olursa, bir başka deyişle TP ve TN değerleri ne kadar büyükse model o kadar başarılı bir model olmaktadır. Örnek olarak

videoda suç olup olmadığı yanıtını arayalım. 85 test verisi ile Tablo 1’deki değerler elde edilmiş olsun. Bu tablodan şu çıkarımlar yapılabilir. Toplamda 40 normal, 45 anormal veri bulunmaktadır. Normal verilerden 10 tanesi, suç verilerinin 5 tanesi yanlış tahmin edilmektedir. Karmaşıklık matrisi bu şekilde, verilerin ne kadarının hangi şekilde tahmin edildiğini göstermektedir.

	NORMAL: HAYIR	NORMAL: EVET	Toplam
SUÇ: HAYIR	TN=30	FP=10	40
SUÇ: EVET	FN=5	TP=40	45
Toplam	35	50	85

Tablo 1: Karmaşıklık Matrisi Örneği

2.3.3.2. Doğruluk

Sınıflandırma problemlerindeki doğruluk (Accuracy), modelin doğru tahmin sayısının toplam tahmin sayısına oranıdır (Zheng, 2015). Doğruluk formülü şöyledir:

$$\text{Doğruluk} = \frac{TP + TN}{TP + FP + FN + TN}$$

Payda doğru tahminler (Gerçek Pozitifler ve Gerçek Negatifler) ve paydada algoritma tarafından yapılan tüm tahminler (doğru ve yanlış olanlar) bulunmaktadır. Doğruluk [0, 1] arası değer alır. Modelin performansı arttıkça doğruluk 1’e yaklaşır. Modelin performansı azaldıkça doğruluk 0’a yaklaşır.

Sınıflara ait veri miktarı birbirine yakın ise yani sınıflardaki veriler dengeli iken doğruluk iyi bir performans ölçüsüdür. Çünkü doğruluk tüm yanlış tahminlere eşit önem vermektedir. Örnek olarak insan tanıma yapan bir modele girdi olarak verilen görüntülerin %55’inde insan varsa ve %45’inde insan yoksa doğruluk metriği ile modelin performansı doğru şekilde değerlendirilebilir. Bir diğer örnekte 100 görüntülü suç tespiti veri setinde sadece 5 görüntüde suç olsun. Diyelim ki modelimiz çok kötü ve her görüntüyü ‘suç yok’ olarak tahmin ediyor. Bunu yaparken, 95 suç olmayan görüntüyü doğru ve 5 suç olan görüntüyü ‘suç yok’ olarak sınıflandırmıştır. 0.95 olan doğruluk değeri ile değerlendirildiğinde modelin performansının iyi olduğu sonucuna varılabilir ama aslında

model suç içeren hiçbir görüntüyü doğru tespit etmemiştir.

2.3.3.3. Kesinlik

Suç tespiti problemi üzerinden açıklayacak olursak, kesinlik (Precision) ‘suç var’ diye tahmin edilen görüntülerdeki gerçek suç oranını verir (Zheng, 2015). Kesinlik formülü şöyledir:

$$Kesinlik = \frac{TP}{TP + FP}$$

Payda doğru tahmin edilen suçlar (Gerçek Pozitifler) ve paydada tüm suç tahminleri (Gerçek ve Yanlış Pozitifler) bulunmaktadır. Kesinlik [0, 1] arası değer alır. Modelin performansı arttıkça kesinlik 1’e yaklaşır. Modelin performansı azaldıkça kesinlik 0’a yaklaşır.

Kesinlik, modelin yaptığı pozitif tahminlerin başarısının bir ölçüsüdür. Örneğin 100 görüntülü suç tespiti veri setinde sadece 5 görüntüde suç olsun. Diyelim ki modelimiz çok kötü ve her görüntüyü ‘suç var’ olarak tahmin ediyor. Bu durumda TP 5, FP ise 95 olur. Kesinlik değeri 0,05 olarak modelin kötü performansını doğru şekilde yansıtabilir.

2.3.3.4. Hassasiyet

Suç tespiti problemi üzerinden açıklayacak olursak, Hassasiyet (Recall) gerçekte suç olan görüntülerin ne kadarının algoritma tarafından doğru tespit edildiğini söyleyen bir ölçüdür (Zheng, 2015). Hassasiyet formülü şöyledir:

$$Hassasiyet = \frac{TP}{TP + FN}$$

Payda doğru tahmin edilen suç görüntüleri (Gerçek Pozitifler) ve paydada tüm suç görüntüleri (Gerçek Pozitifler ve Yanlış Negatifler) bulunmaktadır. Hassasiyet [0, 1] arası değer alır. Modelin performansı arttıkça hassasiyet 1’e yaklaşır. Modelin performansı azaldıkça hassasiyet 0’a yaklaşır.

Hassasiyet, modelin suç içeren görüntülerdeki performansının bir ölçüsüdür. Örneğin

100 görüntülü suç tespiti veri setinde sadece 5 görüntüde suç olsun. Diyelim ki modelimiz çok kötü ve her görüntüyü ‘suç var’ olarak tahmin ediyor. Bu durumda TP 5, FN ise 0 olur. Hassasiyet değeri 1.0 olarak modelin kötü performansını yansıtmaz. Modelimiz her görüntü için ‘suç yok’ tahmininde bulunsaydı TP 0, FN ise 5 olacaktı. Bu durumda hassasiyet değeri 0.0 ile modelin kötü performansını doğru şekilde yansıtacaktır.

2.3.3.5. F1-skoru

F1-skoru, yanlış pozitif ve yanlış negatif hatalarının yakın olduğu derecede önemli olduğu durumlarda performans değerlendirmesi için kullanılır. F1-skoru şu şekilde hesaplanır:

$$F1 - skoru = 2 \times \frac{kesinlik \times hassasiyet}{kesinlik + hassasiyet}$$

Sistemin genel performansını değerlendirmek için kesinlik ve hassasiyet değerlerine bakılabilir. F1-skoru [0, 1] arası değer alır. Modelin performansı arttıkça F1-skoru 1’e yaklaşır. Modelin performansı azaldıkça ise F1-skoru 0’a yaklaşır. F1-skoru genel performansı vererek bir değerlendirme yapılmasını sağlarken, kesinlik ve hassasiyet sistemin nasıl geliştirileceği hakkında bir fikir sunmaktadır (Zheng, 2015).

2.3.3.6. AUC-ROC eğrisi

ROC bir olasılık eğrisi iken eğri altındaki alan (Area Under The Curve, AUC) sınıflar arası farkın ne kadar iyi tespit edildiğini gösteren eğridir (Zheng, 2015). AUC [0, 1] arası değer alır. Modelin performansı arttıkça AUC 1’e yaklaşır. Modelin performansı azaldıkça AUC 0’a yaklaşır. AUC değerinin yüksek olması daha keskin ve iyi bir tahmin yapıldığını yani sınıfların birbirinden iyi şekilde ayrıştırıldığını gösterir.

Bir model için gerçek pozitif oranı (True Positive Rate, TPR) ve yanlış pozitif oranı (False Positive Rate, FPR) şu şekilde tanımlanmaktadır.

$$TPR = \frac{TP}{TP + FN}$$

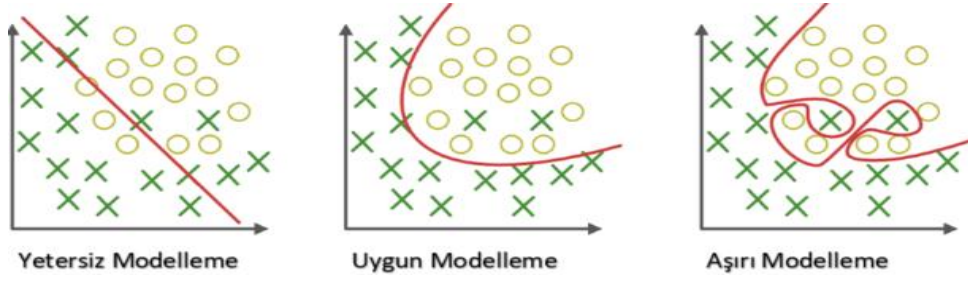
2.3.4. Model geerleme

Makine ğrenmesi problemlerinde olabildiğince genel ve her girdiyi tahmin eden bir sistem tasarlanması gerekmektedir. Bu sistemin genelleştirme (generalization) yeteneğini göstermektedir (Goodfellow vd., 2016). Teoride olan birtakım varsayımlar, verinin parametreleri ve dağılımı bilindiğinde, olabildiğince güçlü (robust) bir sistem ile tasarlanabilir fakat pratikte bu kadar kolay değildir. Gerçek hayat problemlerinde bir takım bilinmeyen özellikleri olacağından dolayı mükemmel bir sistem tasarlanamaz. Eğitilen modele verilen veriler problemin tüm bilgilerini içermediğinden dolayı, model her türlü senaryoyu ğrenemez. Örneğın, sisteme verilen eğitim verisi ile test için verilen veri arasında ciddi bir fark olabilir ve bu durumda sistem iyi bir tahmin çıkarmayabilir. Bu gibi sorunları özerek, olabildiğince farklı veriler için doğru sonuç veren bir model eğitmek için eşitli yöntemler mevcuttur. Bu yöntemler kullanılarak veri seti için seçilen modelin en iyi genelleştirebilecek şekilde eğitilmesine model geerleme (Model Validation) denir (Zhang vd., 2019).

2.3.4.1. Genelleştirme

Makine ğrenmesi probleminde her girdi ok boyutlu uzayda bir nokta olarak ele alınabilir. Aslında model eğitilerek, modelin sınıflar arasındaki karar sınır hiper düzlemi (decision boundary hyperplane) ğrenmesi sağlanır (Haykin, 2009). Model bu şekilde noktaların yani girdilerin hangi sınıfa ait olduğunu çıkarabilmektedir. Modelin performansı ile veri setinden karar sınırı hakkında yaptığı doğru çıkarımlar orantılıdır.

Model, tüm veri seti ile eğitilebilir. Sonrasında merak edilen ise modelin veri seti üzerindeki performansı değildir. Modelin önceden hiç görmediğı veriler üzerindeki performansdır. Çünkü model gerekte farklı ortamlarda ve farklı koşullarda kullanılabilir. Modelin bu gibi önceden görmediğı verilerdeki performansına genelleştirme başarısı denir. Bu yüzden modelin performansı değeriendirilirken, model önceden hiçbir detayını ğrenmediğı verilerle test edilmelidir (Goodfellow vd., 2016). Eğitirken tüm veriyi kullanınca modelin genelleştirme performansını ölçmek imkânsız olmaktadır. Bu yüzden model eğitilmeden önce veri seti ikiye ayrılır: Eğitim Veri Seti (Training Dataset) ve Test Veri Seti (Test Dataset). Model, eğitilirken test veri setindeki hiçbir veri veya bilgidan faydalanmaz. Model eğitim veri seti ile eğitilince, test veri setindeki başarısı ölçülür. Test veri seti yeterince büyükse test verisindeki performansın genelleştirme performansını yansıtması beklenir.

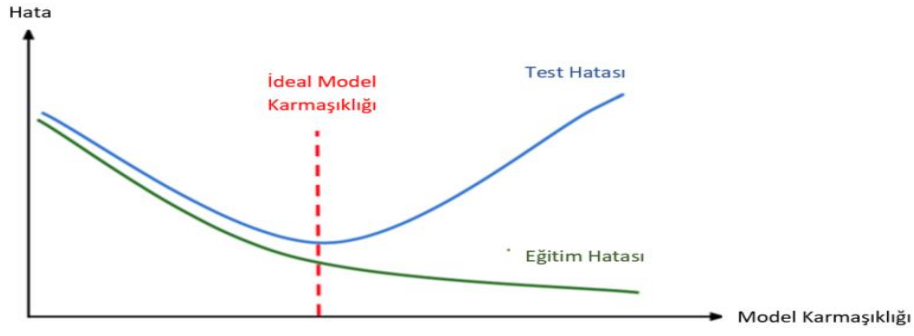


Şekil 6: Eğitim Başarısına göre Model Karar Sınırı

Problem için seçilen model, uygun şekilde ve karmaşıklıkta (complexity) tasarlanmalıdır. Sonrasında eğitim veri setinde uygun şekilde ve uygun miktarda eğitilmelidir. Bu gerçekleştiğinde model eğitim veri setinden çıkarılabilecek bilgileri alır. Bu durumda model eğitim veri seti üzerindeki en iyi performansını vermez ama eğitim veri seti üzerinde eğitildiğinde olabilecek en iyi genelleştirme performansını verir. Bir başka ifadeyle model eğitime eğitim veri setini çözen en iyi karar sınırını değil, problemi genel olarak çözen en iyi karar sınırını bulur (Haykin, 2009). Şekil 6'da bu durumun örneği verilmektedir.

Yetersiz modelleme (underfitting) genellikle modelin düzgün eğitilmemesi veya yeterince iyi veri olmamasından kaynaklanmaktadır. Yetersiz modelleme, modelin problemi yeterince iyi öğrenememesi veya modelin o problemi çözecek kadar karmaşık bir model olmamasından kaynaklanabilir. Bu durumda modelin tepkisi genel olarak basit kalır. Şekil 6'daki yetersiz modelleme buna uygun durumu göstermektedir. Eğer veriler uzayda şekildeki gibi dizilirse ve model yeterince güçlü değilse basit bir yaklaşım ile düşünecektir ve dolayısıyla hata payı yüksek olacaktır. Bir başka deyişle model yeterince karmaşık değilse öğrenme kapasitesi sınırlı kalır veya veri az ise model gerekli bilgileri çıkaramaz (Zhang vd., 2019). Bu durumlarda model iyi genelleştirme performansı gösteremez. Buna yetersiz modelleme denir.

Aşırı modelleme (overfitting) ise tasarladığımız model aşırı güçlü olduğundan dolayı gürültünün çok olması veya öğrenilmemesi gereken şeyleri dahi öğrenmesi durumudur. Eğer bir eğitim verisi doğrudan güçlü veya karmaşık bir sistemle öğretilirse algoritma veri setindeki problemle alakasız bilgiler de dahil her şeyi öğrenebilir, diğer bir ifadeyle veriyi ezberler ve dolayısıyla farklı bir veri setinde yeterince başarılı olamaz (Zhang vd., 2019). Bu durum aşırı modelleme olarak geçmektedir. Bu durumun önüne geçebilmek için veri seti sadece eğitim verisi ile eğitilmeli ve test verisi ayrı tutulmalıdır. Aksi takdirde model iyi performans gösteriyor gibi görünürken, aslında genelleştirme performansı kötüdür.



Şekil 7: Seçilen Modelin Karmaşıklık ve Gerçek Başarı İlişkisi

Şekil 7 model karmaşıklığı ile hata arasındaki ilişkiyi ortaya koymaktadır. Modelin karmaşıklığı arttıkça eğitim verisini iyi öğrendiği, yani eğitim verisi hatasının azaldığı, görülmektedir. Model yeterince karmaşık değilse (kırmızı çizginin solu) eğitim setinden öğrenilebilecek tüm bilgileri edinemez, eğitme ve test (genelleştirme) hatası yüksek olur. Bu durumda yetersiz modelleme gerçekleşir. Modelin karmaşıklığı arttıkça eğitim hatası sürekli azalır. Test hatası başlarda azalmaktadır ancak bir noktadan sonra aşırı modelleme gerçekleştiğinden beklenenin aksine test hatası artmaya başlar (kırmızı çizginin sağı). Bu yüzden en düşük genelleştirme hatasını elde etmek için uygun karmaşıklıkta model seçilmelidir (Goodfellow vd., 2016).

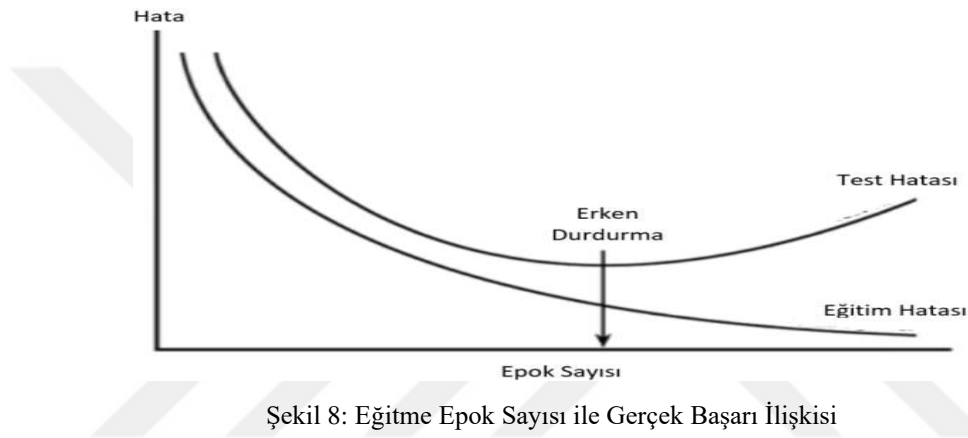
2.3.4.2. Erken durdurma

Model eğitilirken eğitim veri setindeki bir veya birkaç veri ile modelin parametreleri hata fonksiyonunu azaltacak şekilde güncellenir. Bu şekilde veri setindeki tüm verileri eğitim sürecinde bir tur kullanmaya bir epok (epoch) denir (Haykin, 2009). Her epokta problemin doğasına uygun olarak veriler aynı sırada veya rastgele sırada verilir.

Yeterli büyüklükte eğitim veri seti oluşturularak ve doğru karmaşıklıkta model seçerek yetersiz ve aşırı modellemenin önüne geçmeye çalışılır. Ancak modelin kaç epok eğitileceği de çok önemlidir. Epok sayısı az ise model gerekli çıkarımları yapamaz ve yetersiz modelleme olur. Epok sayısı çok ise model veri setini ezberlemeye başlar ve aşırı modelleme olur (Goodfellow vd., 2016). Bu yüzden iyi genelleştirme performansı için modeli uygun epok sayısı kadar eğitmek gerekir.

Model eğitilirken her epokta eğitim veri setindeki hatası azalır. Dolayısıyla hangi epokta durmak gerektiği eğitim veri setine bakılarak bilinemez. Bunun için test veri seti kullanılırsa, eğitme sırasında test veri seti kullanılmış olur. Bu durum test veri setindeki

performansın genelleştirme performansını yansıtmaz özelliğini kaybettirir. Bu yüzden eğitim veri seti 2'ye ayrılır: Eğitim veri seti ve geçerleme veri seti (validation dataset) (Shalev-Shwartz ve Ben-David, 2014). Modeli eğitmek için büyük bir epok sayısı belirlenir ve eğitime başlanır. Her epokta model eğitim veri seti ile eğitilir. Epok sonunda geçerleme veri setindeki hata hesaplanır. Geçerleme veri setinde hata artmaya başlayınca, modelin aşırı modelleme sürecine girdiği sonucuna varılır ve eğitim durdurulur (Aggarwal, 2018). Sonrasında test veri setindeki hata hesaplanarak genelleştirme performansı ölçülür. Bu şekilde hangi epokta eğitimin durdurulmasına karar verilmesine erken durdurma (Early Stopping) denir. Bu süreç Şekil 8'de gösterilmektedir.



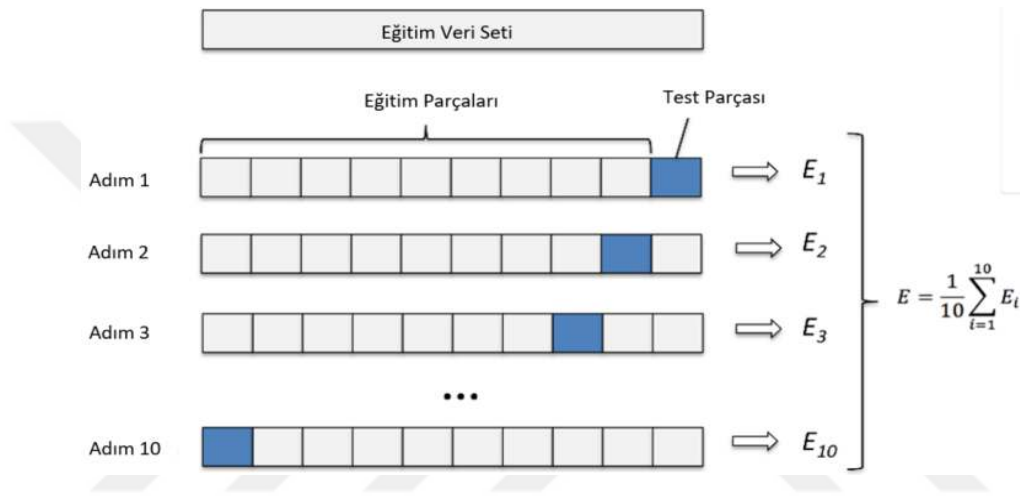
Erken durdurma eğitim verisinin hatası azalmaya devam ederken test verisinin hata payının arttığı zamanda sistemin eğitiminin bitirilmesi durumudur. Yani sistemin ezberlemesinin önüne geçmek için uygulanan bir algoritmadır. Özellikle büyük modellerde sistemi öğrenme gücü yüksek olduğundan dolayı sistemin ezberleme olasılığı artmaktadır. Burada amaç veri setinin tamamını ya da olmayan bir kısmını bile genelleştirebilen bir model elde etmek olduğu için bu durum ciddi önem taşımaktadır.

2.3.4.3. Çapraz geçerleme

Erken durdurma veri sayısı yüksek iken işe yarayacak bir algoritmadır. Çünkü eğitim verisinin bir kısmı geçerleme veri seti olarak ayrıldığından model daha az veri ile eğitilmiş olur. Eğer veri sayısı yeterince yüksek değil ise bu algoritma genellikle işe yaramaz. Bu durumda çapraz geçerleme (Cross-Validation) uygulanır (Shalev-Shwartz ve Ben-David, 2014). Çapraz geçerleme eğitim veri setinin k adet eşit parçaya (fold) bölünmesi ile olur. Bir epokta k tane eğitme yapılır. Her seferinde farklı bir parça seçilir. Diğer parçalarla eğitme

yapılır ve seçilen parça ile geçerleme yapılır. Bunu her k parça için ayrı ayrı yaptıktan sonra hataların ortalaması genel hata oranını verir. Hata oranı artmaya başlayınca eğitim durdurulur. Bu şekilde eğitim verisinin bir kısmından feragat etmeden uygun epok sayısı belirlenir.

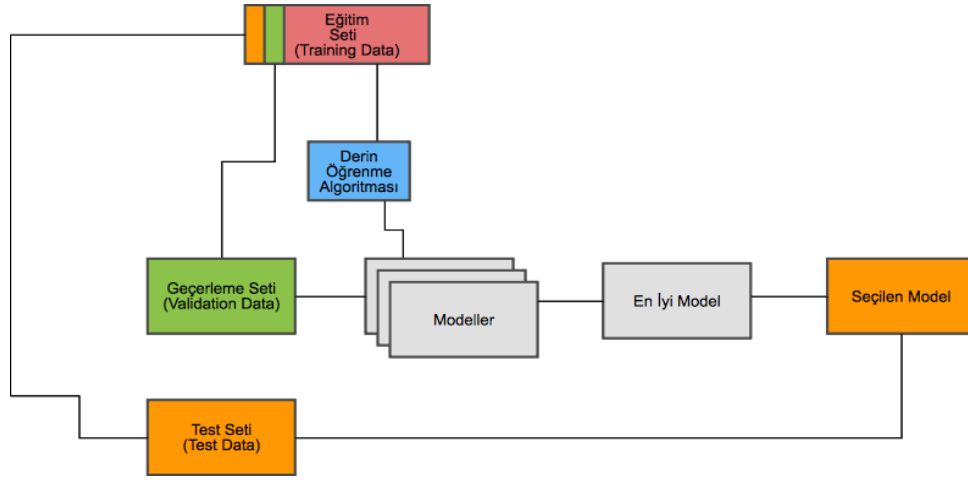
Şekil 9, çapraz geçerlemenin temsili gösterimidir. E_i i. adımdaki hatayı, E ise bir epok sonundaki ortalama hatayı vermektedir. Bu yöntemde k çok küçük seçilmemelidir. Literatürde k genellikle 5 veya 10 seçilmektedir. Çapraz geçerlemenin erken durdurmaya göre dezavantajı eğitime süresinin uzun olmasıdır.



Şekil 9: 10-Fold Çapraz Geçerleme ile Eğitim Örneği

2.3.4.4. Model geçerleme süreci

Probleme en uygun modeli seçip, en doğru şekilde eğitmek veri setinden mümkün olan en yüksek genelleştirme performansını elde etmek için çok önemlidir. Bu süreç Şekil 10’da gösterilmektedir. Bir problem için farklı modellerin iyi sonuç vermesi beklenebilir. Bu yüzden her model için bu süreç tekrarlanır.



Şekil 10: Model Geçerleme Süreci

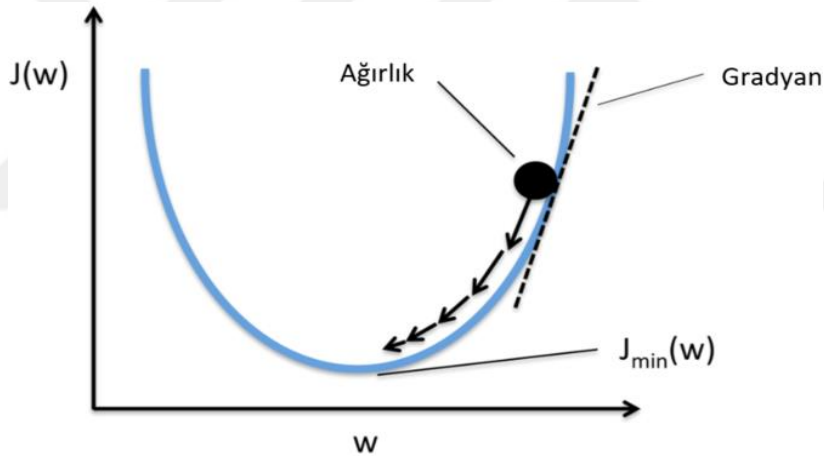
Bir model eğitim ve geçerleme veri setinde eğitilir. Bir model eğitilirken farklı karmaşıklıklarda (complexity) versiyonları eğitilir ve o model için en uygun karmaşıklık bulunur. Belirli bir karmaşıklıkta belirli bir model erken durdurma gibi yöntemler kullanılarak ideal miktarda eğitilir. Böylece modelin geçerleme veri seti sayesinde ideal karmaşıklığı bulunur ve ideal epokta eğitilir (Shalev-Shwartz ve Ben-David, 2014). Bu şekilde yetersiz ve aşırı modellemekten kaçınılması amaçlanır. Sonrasında test veri setindeki performansı o modelin gerçek performansı olur. Farklı modeller farklı problemlerde daha iyi sonuç verebilir. Dolayısıyla bu süreç çeşitli modeller için tekrarlanarak en iyi test performansı olan model pratikte kullanılır.

2.3.5. Geri yayılım

Bir makine öğrenmesi algoritmasını eğitirken fonksiyona girdi verildikten sonra çıktı ve hata elde edilir. Hata payını azaltmak için model hata ile geri beslenir. Fonksiyonun çalışmasının tersi yönünde geri dönüş ile fonksiyonun parametreleri güncellenir. Buna geri yayılım (backpropagation) denir (Skansi, 2018). Geri yayılım algoritması özellikle yapay sinir ağı üzerinde uygulanmaktadır. Burada hatanın minimum olduğu noktaya doğru gidişatı sağlamak için türev alınır. Türev analitik çözüm ile hesaplanamayacak kadar zor olabilir. Dolayısıyla iteratif bir yöntem ile minimum nokta aranır. Bu iteratif hesaplama esnasında model üzerindeki olası tüm parametreler sürekli güncellenir.

2.3.5.1. Gradyan iniş

Gradyan iniş (Gradient Descent) bir hata fonksiyonunu iteratif yöntemle minimum hale getirmeyi sağlayan bir algoritmadır. Gradyan vektörü parametrelerin en çok arttığı noktaları bilgi olarak sunmaktadır. Her iterasyonda fonksiyonun gradyanın tersi yönünde güncellenmesi ile sürekli değerinin azalması sağlanır (Zhang vd., 2019). Değişimin miktarını belirleyen parametreye öğrenme oranı (learning rate) denir. Bu durum çok boyutlu bir yüzeyde topun yaptığı salınım hareketine benzetilebilir. Topun zamanla aşağı hareket edeceği gibi, fonksiyon gradyanın tersi yönünde bir miktar güncellenerek fonksiyonun zamanla azalması sağlanır. Bu iteratif yöntemle her zaman yerel bir minimum nokta bulunur (Goodfellow vd., 2016). Bu durum Şekil 11’de gösterilmektedir. ML modelinin eğitim sırasında değerini veriden öğrendiği parametrelerine ağırlık (weight) denir. Bu parametreler w ile, modelin hatası ise $J(w)$ ile gösterilmektedir çünkü ağırlıklar güncellendikçe modelin hatası değişmektedir.



Şekil 11: Gradyan İniş ile Eğitim Örneği

Gradyan inişte öğrenme oranı eğer çok büyük ise gradyanlar büyük adımlar atarak ilerler ve minimum noktaya ulaşır. Fakat eğer minimum nokta aşılsa büyük öğrenme oranı ile çok fazla salınım hareketi yapılmaya başlanır ve en iyi ağırlık değerlerine ulaşılamayabilir. Eğer öğrenme oranı çok düşük olursa da gradyanlar çok yavaş bir şekilde ilerleyecektir ve minimum noktaya ulaşmak daha çok zaman alacaktır. Bu nedenle öğrenme oranı başta büyük seçilir ve dinamik olarak zamanla azaltılır (Goodfellow vd., 2016). Gradyan iniş bir geri yayılım aşamasında kullanılan veri miktarına göre üçe ayrılır:

Yığın gradyan iniş (Batch Gradient Descent) yönteminde veri setindeki tüm verilerden çıktılar ve hatalar hesaplanır. Sonrasında ortalama hatadan türev hesaplanarak

güncelleme yapılır (Zhang vd., 2019). Yani bir epokta tüm veriden tek bir adımda geri yayılım yapılır. Yığın gradyan iniş formülü şöyledir:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta)$$

Burada parametre çıktısındaki hatanın türevinin öğrenme oranı katı kadar azaltılarak, modelin ileride veriler için ortalamada daha az hataya neden olması sağlanır. Tüm veriden bir hesaplama yaptığı için düşük varyans ile güncelleme yapılarak hata fonksiyonunun yumuşak azalmasına olanak sağlar. Normalde bu tarz bir gradyan veri sayısı yüksek iken en düzgün çalışacak olan türdür fakat veri sayısı çok yüksek olduğu zaman hafızanın yetmemesi gibi sorunlar ortaya çıkabilir.

Stokastik gradyan iniş, yığın gradyan inişin dezavantajını gidermek için geliştirilmiştir. Yığın gradyan iniş için özellikle veri sayısı çok ise büyük bir hafıza gerekir. Bunu önlemek için her girdi için ayrı bir çıktı ve hata hesaplanarak bir geri yayılım yapılabilir (Zhang vd., 2019). Böylece büyük hafıza ihtiyacı ortadan kalkar. Bu yöntemle stokastik gradyan iniş (Stochastic Gradient Descent) denir ve formülü şöyledir:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x_i; y_i)$$

Her veri için ayrı güncelleme yapıldığından aykırı verilere daha hassastır. Varyans çok olabileceğinden hata fonksiyonunun azalması yumuşak olmayabilir. Bu hassasiyetten dolayı stokastik gradyan iniş salınım oluşturarak en uygun noktayı bulmayı sağlar. Bu yöntemin bir diğer dezavantajı bir epokta her veri için ayrı güncelleme yaptığından çok yavaş olmasıdır.

Mini-yığın gradyan iniş yaygın ve başarılı olarak kullanılan gradyan iniş yöntemidir. Stokastik gradyan inişte bir iterasyonda her veri için ayrı türevler hesaplanır ve güncellemeler yapılır. Bu eğitme sürecini ciddi derecede yavaşlatır. Ayrıca veri setine karşı aşırı hassastır. Yığın gradyan iniş ise büyük veri setlerinde hafıza sorunundan dolayı kullanılamaz. Bu sebeple makine öğrenmesi modelleri eğitilirken, bahsedilen 2 yöntemin ortasını bulan mini-yığın gradyan iniş (Mini-batch Gradient Descent) yöntemi genellikle tercih edilir (Zhang vd., 2019). Mini-yığın gradyan iniş formülü şöyledir.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x_{(i:i+k)}; y_{(i:i+k)})$$

Bu yöntemde veri her epokta eşit mini-yığınlarla bölünür. Formülde mini-yığın veri sayısı k 'dir. Her mini-yığındaki girdilerden ortalama hata hesaplanır ve türev ile parametre güncellemesi yapılır. Böylece fonksiyondaki yumuşak iniş olabildiğince korunurken model hızlı eğitilebilir.

2.3.5.2. Momentum

Öğrenme oranını seçmek her zaman kolay olmayabilir. Öğrenme oranı bazen çok küçük olabilir ve bu durumda modelin öğrenmesi çok yavaş olur. Öğrenme oranı bazen çok büyük olabilir ve bu durumda ise güncellemeler büyük adımlarla yapıldığından model minimum değer etrafında salınım yapabilir. Bunun önüne geçmek için fizikteki momentum mantığı uygulanır. Küçük bir öğrenme oranı ile başlanır ve momentum sürekli bu oranı büyütür. En son gradyan ters yöne döndüğünde momentum gücü düşürülür ve gradyan yeri bulmaya başlar (Buduma ve Locascio, 2017). Momentumlu gradyan iniş formülü şöyledir.

$$v_t = \gamma v_{(t-1)} + \eta \nabla_{\theta} J(\theta)$$
$$\theta = \theta - v_t$$

Bu yöntemde parametrenin değişme miktarı (v_t) hatanın gradyanının öğrenme oranı katı kadar önceki değişim miktarının (v_{t-1}) momentum sabiti katına da bağlıdır. Bu sayede parametrenin değişim yönü önceki adımdaki ile aynı ise güncelleme adımının büyük olması sağlanır ve güncelleme süreci hızlandırılmış olur. Parametrenin değişim yönü önceki adımdaki ile zıt yönde ise güncelleme adımının küçük olması sağlanır. Bu da minimum değer etrafında küçük adımlar atmaya olanak sağlar.

2.3.5.3. Eniyileştiriciler

Gradyan iniş algoritmalarını daha düzgün ve iyi çalışır hale getiren algoritmalara eniyileştiriciler (optimizers) adı verilmektedir (Buduma ve Locascio, 2017). Bunların her biri farklı bir yaklaşım ile çalışmaktadır. En temel eniyileştiriciler şunlardır:

AdaGrad güncelleme kuralında öğrenme oranını her adımda değiştirmektedir. Sürekli değişen parametreler için öğrenme oranı değişimini düşük tutar. Parametrelerin değiştiği

durumlarda ise öğrenme oranını artırır (Buduma ve Locascio, 2017). Öğrenme oranı standart başlangıç değeri 0.01'dir. AdaGrad'ın en önemli zayıflığı kare gradyanların paydada yapmış olduğu büyüme durumudur. Her bir terim eklendiğinde bu büyüme artmaktadır ve öğrenme oranı aşırı küçülmeye başlamaktadır.

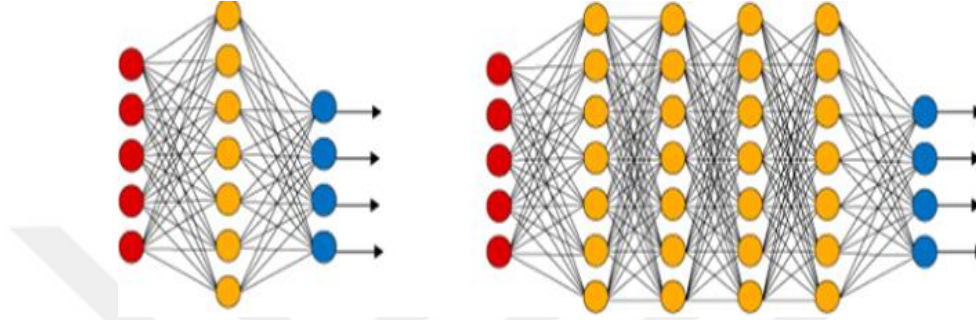
RMSProp öğrenme oranını gradyanların karesinin ortalamasına bölerek bir değişme miktarı bulur (Buduma ve Locascio, 2017). Burada öğrenme oranının 0.001 ile, momentum sabitinin 0.9 ile başlatılması tavsiye edilmektedir. Önceki gradyanların karelerini depolamak yerine, gradyanların toplamı tüm eski gradyanların azalan ortalaması şeklinde tanımlanmıştır.

Adam geçmiş gradyanların yalnızca ortalama miktarlarının karesine bakmak yerine aynı zamanda ikincil momentumların ortalamasına da bakarak bir güncelleme yaparak ilerlemektedir (Buduma ve Locascio, 2017). Adam ile, AdaGrad'ın öğrenme oranını her parametre için ayrıca güncelleme özelliğini ve RMSProp'un ikinci momentlerin ortalama alma özelliğini kullanarak gürültülü ve yoğun veri tipinde daha iyi çalışan bir optimizasyon mantığı ortaya çıkarmaktadır. Bu yüzden literatürde geri yayılımla makine öğrenmesi ve derin öğrenme modellerini eğitmek için Adam ve türevleri sıkça kullanılmaktadır.

2.4. Derin Öğrenme

Bilim insanları ML modelleri geliştirilirken milyarlarca yıllık evrim sonucu mükemmelleşen insan beyninin çalışma prensibinden esinlenmişlerdir. Nöronlardan esinlenerek çeşitli özelleşmiş yapay hücreler tasarlanmıştır. Bu hücreleri sırayla yan yana dizerek katmanlar (layer), katmanları da art arda dizerek yapay sinir ağları (Artificial Neural Network, ANN) oluşturulmaktadır (Buduma ve Locascio, 2017). ANN dolayısıyla ML'nin bir alt kümesini oluşturmaktadır. ML'ye benzer şekilde önceden oluşturulan model, veri seti ile eğitilerek optimize edilmektedir. Problem türlerine göre özel olarak geliştirilmiş yapay hücreler ve yapay sinir ağları bulunmaktadır ve ANN modelleri standart ML algoritmalarına göre daha iyi performans göstermektedirler (Daniel, 2013). ANN'ler katmanlardan oluşmaktadır. Her katman bir önceki katmanın çıktısını alarak çıkarım yapmakta, sonucunu sonraki katmana aktarmaktadır. Doğal olarak, verinin doğru analizi için problemin karmaşıklığı arttıkça modelin katman sayısı ve katmanlardaki hücre sayısı arttırılmalıdır. Bu da öğrenilmesi gerekli parametre sayısı, bununla birlikte modeli eğitmek için gerekli veri ve işlem sayısını arttırmaktadır. Katman sayısı çok olan ANN modellerine derin sinir ağı (Deep Neural Network, DNN) ve bu yöntemle algoritma geliştirmeye derin öğrenme (Deep

Learning, DL) denir (Zhang vd., 2019). Şekil 12’de gösterildiği üzere DL için akıllı algoritmaların en özelleşmiş grubu denebilir. Şekil 12’de ANN ve DNN farkı gösterilmektedir. Kırmızı katman girdiyi, mavi katmanın çıktısı modelin çıktısını ve sarı katman ara katmanı temsil etmektedir. Ara katmana, saklı katman (hidden layer) denir. Çizgiler bir önceki katmandaki hangi hücrenin çıktısının bir sonraki katmandaki hangi hücreye girdi olarak verildiğini göstermektedir.



Şekil 12: ANN (Soldaki Model) ve DNN (Sağdaki Model) Örneği

İnternetin yaygınlaşmasıyla neredeyse her konuda çok büyük miktarda veri bulunmaktadır. Dolayısıyla derin öğrenme için veri toplamak kolaylaşmıştır. Grafik işlemci birimleri (Graphics Processing Unit, GPU) ile bilgisayarlar hızlı işlem yapabilmektedir. GPU’ların ucuzlaması ve yaygınlaşması ile kişisel bilgisayarların işlem gücü önceki yıllara göre ciddi derecede artmıştır. Ayrıca, aşırı işlem gücü gereken çalışmaları kısa sürede bitirebilmek için ekonomik fiyata bulut hizmeti alınabilmektedir. Bu sebeple derin öğrenme sürecindeki işlemleri tamamlama süresi yıllardan aylara, aylardan günlere, günlerden saatlere kısalmıştır. Tüm bu gelişmeler DL modelleri geliştirmek için ideal bir ortam oluşturmuştur. DL modelleri çok katmanlı yapısı sayesinde ham veriden, düşük seviyeli bilgilerden başlayarak katman katman daha yüksek seviyeli çıkarımlar yaparak diğer yöntemlerin çözemediği problemlere cevap verebilmektedir (Goodfellow vd., 2016).

Derin öğrenme çok katmanlı yapısı sayesinde dağınık gösterimleri keşfetmeye olanak sağlar. Bu yüzden makine öğrenmesinin sadece bir alt dalı olmasına rağmen son yıllarda görüntü tanıma ve makine öğrenmesi alanlarını domine etmiştir (Schmidhuber, 2015; Guo vd., 2016). Derin öğrenme ile resim, video, ses ve yazı analizinde başta olmak üzere birçok alanla ilgili akıllı sistem geliştirmede önemli yol katedilmiştir (LeCun vd., 2015; Liu vd., 2017). Video görüntülerinden derin öğrenme ile görüntüyü yazıya çevirme (Li vd., 2018; Chen vd., 2020), işaret dili okuma (Browy vd., 2020; Camgoz vd., 2020), dudak okuma (Afouras vd., 2020; Lu ve Yan, 2020), davranış tanıma (Dai vd., 2020; Wu vd., 2020) ve

anormal durum tespiti (Nawaratne vd., 2018; Morais vd., 2019; Yao vd., 2020) gibi birçok uygulama başarı ile hayata geçirilmektedir.

Sonraki bölümlerde temel derin öğrenme hücreleri ve modelleri anlatılmaktadır.

2.4.1. Yapay sinir ağları

Yapay sinir ağları insan beynini yapay olarak modelleyen yöntemdir. Hücreler katmanları katmanlar modeli oluşturur. Her konuya özel hücre ve model ile o modeli çok iyi öğrenen yapılar elde edilebilir. Ayrıca katman sayısını gerektiğinde arttırarak istenilen derinlikte model elde edilir ve karmaşık problemleri öğrenen yapılar elde edilebilir. Bu yüzden klasik makine öğrenmesine göre çok güçlü ve esnek bir yapıdır.

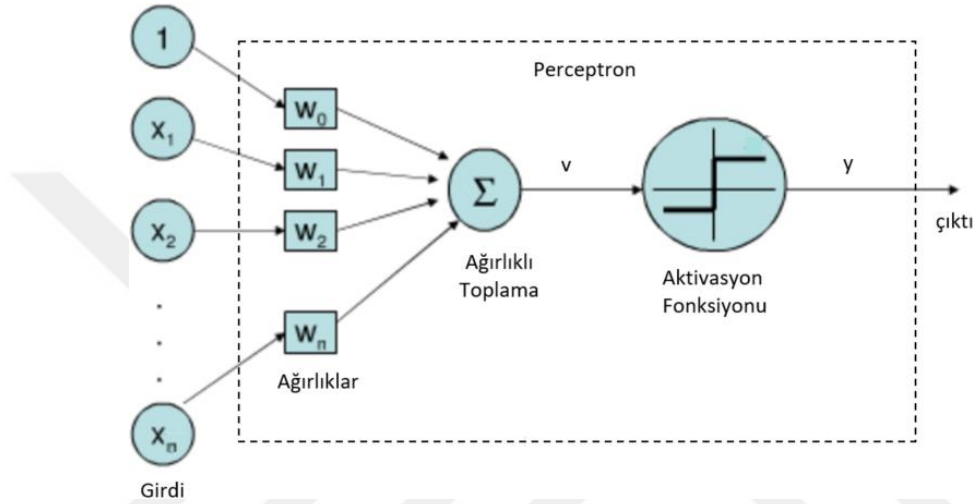
2.4.1.1. Perceptron

Perceptron en basit ve ilk yapay sinir ağı hücresidir. İnsandaki nöron yapısından ilham alınarak tasarlanmıştır. Bir nörona binlerce dendrit girer ama nöronun 1 tane akson çıkar. Nörondaki kimyasal enerji belirli bir eşiği geçerse aksonun elektriksel sinyal gönderilir. Benzer şekilde perceptrona da birden fazla girdi gelmektedir. Bu hücrenin girdilerinin belirli ağırlıklar ile çarpılmış halleri bu nöron üzerinde toplanmaktadır. Bu yapıda en son oluşan değer bir aktivasyon fonksiyonundan geçirilerek çıktı elde edilir (Haykin, 2009). Farklı amaca yönelik olarak farklı aktivasyon fonksiyonları kullanılabilir.

En basit aktivasyon fonksiyonu olarak, aynı beyin hücrelerinde olduğu gibi, bir eşik değeri düşünülebilir. Çıktıda oluşan değer bu aktivasyon fonksiyonundan geçince bir elemeden geçmiş olur ve çıktı ona göre bir değer alır (Shalev-Shwartz ve Ben-David, 2014). Örneğin, eşik değeri aşılmış bir nöronun 1 çıktı olarak üretilir, eşik değerinin altında kalma durumunda ise 0 dışarı çıktı olarak verilir. Burada 0 normal durum, 1 ise hücrenin sinyal gönderdiği durum olarak düşünülebilir. Perceptron matematiksel olarak şu şekilde ifade edilebilir:

$$v = \sum_{i=0}^n w_i x_i$$
$$y = f(v)$$

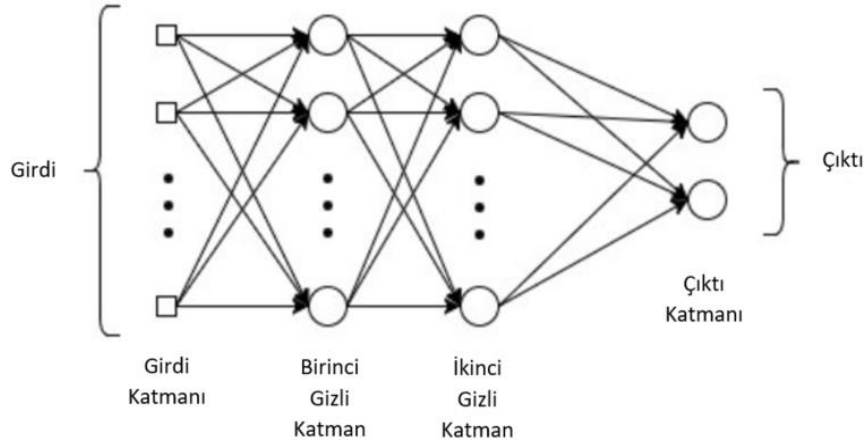
Burada girdi vektörünün (x) elemanları perceptronun ağırlıkları (w) ile çarpılarak toplanır. Ağırlıklar, perceptronun eğitim sırasında değerleri öğrenilen parametreleridir. Sonrasında elde edilen sonuç aktivasyon fonksiyonundan (f) geçirilerek çıktı (y) elde edilir (Shalev-Shwartz ve Ben-David, 2014). Perceptron hücresinin detaylı yapısı Şekil 13'te gösterilmektedir. Perceptron yapısı basit olsa bile birden çok perceptron ile oluşturulan yapay sinir ağları ile karmaşık problemler başarı şekilde çözülebilmektedir. Ayrıca çeşitli amaçlara yönelik çok sayıda farklı yapay sinir ağı hücresi bulunmaktadır.



Şekil 13: Perceptron Hücresi Yapısı

2.4.1.2. Çok katmanlı perceptron

Perceptron ile bazı basit problemler çözülebilmektedir, çünkü perceptron sadece bir yapay sinir ağı hücresidir. Lakin aynı beynin nöronlardan oluşması gibi çok sayıda perceptron kullanılarak yapay sinir ağı elde edilebilir. Perceptronlar birbirlerinden bağımsız bir şekilde alt alta dizilirlerse bir katman (layer) oluşur. Eğer bu katmanlar uç uca eklenirse çok katmanlı perceptron (Multilayer Perceptron, MLP) ortaya çıkmaktadır (Haykin, 2009). Bir katman girdi vektörünü alır ve çıktı vektörünü sonraki katmana aktarılır. Katmanın çıkışı, o katmandaki hücrelerin çıkışından oluşur. Her hücre diğerlerinden bağımsız olarak çıktı üretir ve bunların birleşimi çıktı vektörünü oluşturur. Bu yüzden bir katmanın çıktı büyüklüğü o katmandaki perceptron sayısına eşittir.



Şekil 14: İki Gizli Katmanlı İkili Sınıflandırıcı MLP Örneği

MLP'ye verilen girdi bir katman olarak düşünülmektedir (girdi katmanı). Büyüklüğü girdi vektörü kadar olup genellikle önceden işlenen girdiyi doğrudan sonraki katmana aktarır. Aradaki katmanlara gizli katman (hidden layer) denir. Gizli katman sayısı ve katmanların büyüklüğü uygulamadan uygulamaya değişebilir. Aktivasyon fonksiyonu olarak çeşitli fonksiyonlar kullanılabilir. Son katman ise çıktı katmanıdır. Çıktı katmanında regresyon problemlerinde doğrusal aktivasyon fonksiyonlu bir perceptron, sınıflandırma problemlerinde softmax fonksiyonlu sınıf sayısı kadar perceptron kullanılır (Zhang vd., 2019). Burada her nöron bir sınıfı temsil eder. Bu konu aktivasyon fonksiyonları anlatılırken detaylandırılacaktır. 2'den daha fazla gizli katmandan oluşan MLP modelleri derin öğrenme modeli olarak kabul edilmektedir. Şekil 14'te iki gizli katmanlı ikili sınıflandırma yapan MLP örneği verilmektedir

2.4.1.3. Tamamen bağlı sinir ağı

Normalde MLP'de bir katmandaki girdinin tamamı tüm hücrelere verilmek zorunda değildir. Girdi istenilen şekilde hücrelere verilebilir. Tamamen bağlı sinir ağında (Fully Connected Neural Network, FCNN) bir katmandaki girdi tamamen her hücreye bağlanılır (Zhang vd., 2019). FCNN'de hücre olarak perceptron dışında yapılar da kullanılabilir. FCNN genel yapısı sayesinde, girdi ile ilgili bir varsayım yapmadan kullanılma imkânı sağlar.

Bir problem çözümünde birden çok modelin birleşiminden hibrit bir ağ oluşturmak daha iyi sonuç verebilir. Bazı problemlerde önceki aşamalarda eğitilen model veya modellerden gelen çıktılar FCNN'ye verilir ve FCNN ile çıktı elde edilir (Zhang vd., 2019). Bu tip uygulamalarda önceki modelin çıktısı bir öznitelik vektörü olarak düşünülmektedir.

Yani problemin doğasından çıkarılan bilgiler kümesi. Sonrasında FCNN ile bu öz niteliklerden çıktı elde etme öğrenilir. Özellikle görüntü ve video başta olmak üzere birçok derin öğrenme modelinde bu yaklaşım kullanılmaktadır.

2.4.2. Aktivasyon fonksiyonları

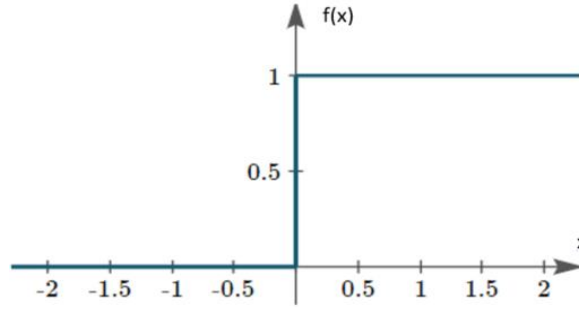
Aktivasyon fonksiyonları katmanlar arasındaki çıktıları değerlendiren ya da en son çıktıda karar veren bölgeyi sağlayan birer matematiksel fonksiyondur (Bhardwaj vd., 2018). Örneğin aktivasyon fonksiyonları katman çıktısının normalizasyonu sağlayabilir. Her aktivasyon fonksiyonun kendine göre bir kullanım alanı vardır. Aktivasyon fonksiyonları derin öğrenme modellerinin daha iyi öğrenmesini sağlamaktadır. Başlıca aktivasyon fonksiyonları aşağıda verilmektedir.

2.4.2.1. Adım fonksiyonu

Adım (Step) fonksiyonu girdi sıfırdan büyükse 1, aksi durumda 0 değeri alır (Bhardwaj vd., 2018). Adım fonksiyonu ile nöronun sinyal göndermesi simüle edilmiş olur. Adım fonksiyonu matematiksel olarak şu şekilde ifade edilebilir:

$$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

Tercih edilmese bile, çıktı katmanında adım fonksiyonlu bir yapay sinir hücresi ile ikili sınıflandırma yapılabilir. Bu yöntemde çıktı 0 ise ilk sınıf, 1 ise ikinci sınıf kararı verilir. Adım fonksiyonu pratikte kullanılmamaktadır. Genel olarak ara katmanlarda adım fonksiyonunun türevlenebilir versiyonu olan sigmoid fonksiyonu kullanılmaktadır.



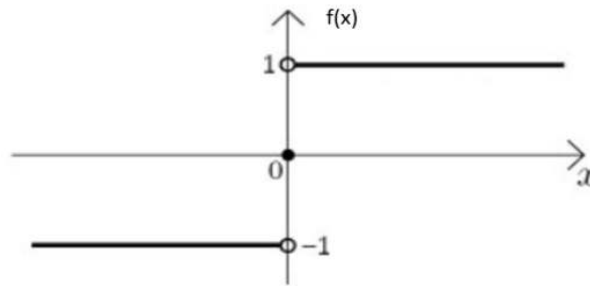
Şekil 15: Adım Fonksiyonu

2.4.2.2. İşaret fonksiyonu

İşaret (Sign) fonksiyonu ile adım fonksiyona benzer şekilde nöronun sinyal göndermesi simüle edilir. Adım fonksiyonundan farklı olarak girdi 0'dan küçükse çıktı -1 değerini alır (Bhardwaj vd., 2018). İşaret fonksiyonu şu şekilde ifade edilebilir:

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

İşaret fonksiyonunun en önemli özelliği negatif değerlere de önem vermesidir. Adım fonksiyonu gibi işaret fonksiyonu da pratikte kullanılmamaktadır. Genel olarak ara katmanlarda işaret fonksiyonunun türevlenebilir versiyonu olan tanh kullanılmaktadır.



Şekil 16: İşaret Fonksiyonu

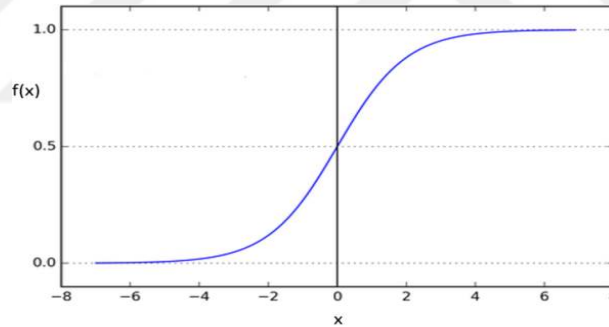
2.4.2.3. Sigmoid fonksiyonu

Adım fonksiyonu girdiye göre hassastır. Girdi vektöründeki ufak değişiklik çıktıyı tamamen değiştirebilir. Sigmoid fonksiyonu 0 ve 1 aralığında gerçel sayı değeri alır

(Bhardwaj vd., 2018). Bu sayede girdideki ufak deęişiklikler, çıktıda büyük deęişikliklere neden olmaz. Adım fonksiyonuna benzer şekilde 1'e yakın deęerler sinyal var, 0'a yakın deęerler sinyal yok olarak düşünülür ancak eğitim sürecinde fonksiyonun sürekli yapısı sayesinde hücre daha yavaş ve kararlı eğitilebilir. Fonksiyonun matematiksel formülü şu şekildedir:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid fonksiyonu sürekli yapısından dolayı bir nöronun eşik deęeri aşılnca sinyal göndermesi özelliğini tam olarak yansıtmamaktadır. Buna rağmen adım fonksiyonu yerine tercih edilmekte ve ara katmanlarda sıkça kullanılmaktadır. Bunun en önemli nedeni türevlenebilir olmasıdır. Bu sayede geri yayılım sürecinde hücre eğitilmekte yani ağırlıklar güncellenebilmektedir. Adım fonksiyonuna göre dezavantajı ise hesaplamaların daha maliyetli olmasıdır.



Şekil 17: Sigmoid Fonksiyonu

Çıktı katmanında sigmoid fonksiyonu kullanılarak ikili sınıflandırma yapılabilir. Bu durumda çıktı 0.5'ten küçükse ilk sınıf, aksi takdirde ikinci sınıf çıkarımı yapılır. Burada çıktıdaki sayı sadece girdinin sınıfını işaret etmez. Çıktı deęeri girdinin ikinci sınıfa ait olma olasılığını verir (Bhardwaj vd., 2018). Bu da sonuca ne kadar güvenilebileceğini işaret eder. Örneğin çıktı 0.1 veya 0.4 olması girdinin ilk sınıfta olmasını işaret eder ama çıktı 0.1 ise sonuç daha olasıdır. Ara katmanlarda ise çıktının 0-1 arası deęer alması çıktının normalizasyonunun yapılması olarak düşünülebilir. Böylece ara kademelerde bir hücrenin çok büyük veya çok küçük sayı üretmesi engellenerek çıktının sadece bir hücreye göre hareket etmesi önlenmiş olur.

2.4.2.4. Tanh

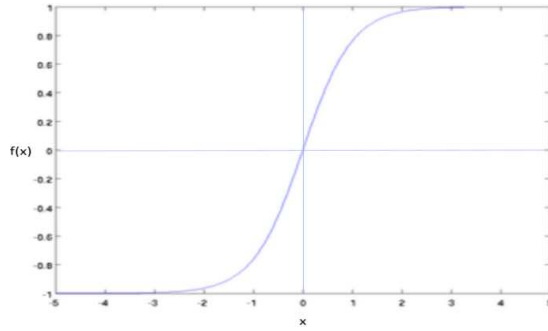
Hiperbolik tanjant (hyperbolic tangent, tanh) fonksiyonu, sigmoid fonksiyonunda olduğu gibi, işaret fonksiyonunun türevlenebilir versiyonudur (Bhardwaj vd., 2018). Formülü şu şekildedir:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

Ayrıca tanh, sigmoid fonksiyonu cinsinden şu şekilde ifade edilebilir:

$$\tanh(x) = 2 \operatorname{sigmoid}(2x) - 1$$

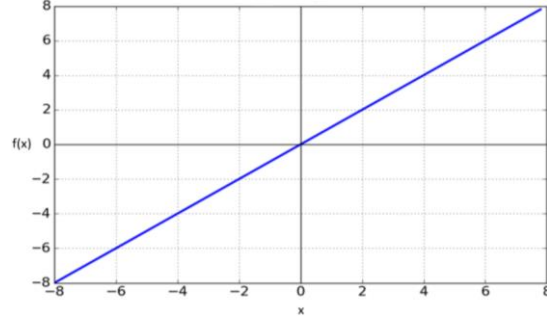
Sigmoid fonksiyonu için yapılan yorumlar tanh için de geçerlidir. Tek farkı çıktının, $[-1, 1]$ aralığında olması denilebilir. Bu durumda çıktı 0'dan büyük ve küçük diye iki gruba ayrılmış olarak düşünülebilir. Tanh ara katmanlarda sıkça kullanılmakta olup pratikte genel olarak sigmoid fonksiyonu yerine tercih edilmektedir.



Şekil 18: Hiperbolik Tanjant (Tanh) Fonksiyonu

2.4.2.5. Doğrusal fonksiyon

$f(x) = x$ şeklinde olan doğrusal (linear) fonksiyon girdiyi aynen çıktıya aktarır (Bhardwaj vd., 2018). Regresyon problemlerinde amaç sürekli (continuous) parametrenin değerini hesaplamaktır. O yüzden çıktı katmanında doğrusal aktivasyon fonksiyonlu bir hücre kullanılarak bu değer hesaplanmaya çalışılır.

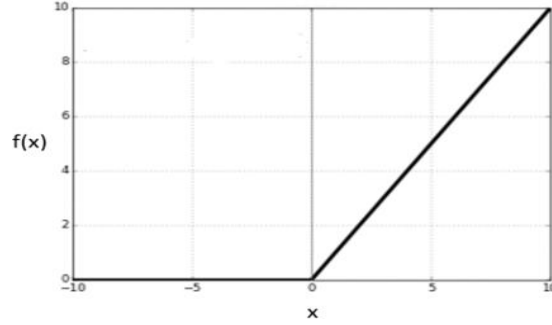


Şekil 19: Doğrusal Fonksiyon

Genellikle birkaç sorunu vardır. Geri besleme yapılacağı zaman türevi sabit bir rakam elde edilmektedir. Bu yüzden eğitilmesi pek mümkün değildir. Buna ek olarak son katmanın lineer olması diğer önceki tüm katmanları da etkilemektedir. Temeli lineer regresyon modeline dayandığı için karmaşık hesaplamaları yapmak pek mümkün olmamaktadır.

2.4.2.6. ReLu fonksiyonu

Doğrusal doğrultmaç ünitesi (Rectified Linear Unit, ReLu) doğrusal fonksiyonunun doğrusal olmayan (non-linear) versiyonu olarak düşünülebilir. Fonksiyon girdi 0'dan küçükse 0, girdi sıfırdan büyükse girdinin değerini alır (Bhardwaj vd., 2018).



Şekil 20: ReLu Fonksiyonu

Önemli bir dezavantajı bulunmaktadır. 0'dan küçük X değerlerini 0'a eşleştirmesidir. Bundan dolayı da 0'dan küçük değerler özelliklerini kaybetmektedir. Buna rağmen son yıllarda çeşitli uygulamalarda ara katmanlarda tanh yerine kullanıldığında sinir ağının performansını arttırdığı görülmektedir ve popülerliği gitgide artmaktadır (Bhardwaj vd., 2018). Bunun en önemli özelliği, şu ana kadar anlatılan özelliklerle perceptron ve aktivasyon fonksiyonları başta olmak üzere yapay sinir ağı unsurlarının doğrusal olmasıdır. Lakin çoğu zor problem doğası gereği doğrusal değildir. ReLu sayesinde yapay sinir ağına doğrusal

olmayan fonksiyon tanıtılabilmekte bu sayede model problemi daha iyi öğrenebilmektedir.

2.4.2.7. Softmax fonksiyonu

İkili sistemde N tane sınıf 0, 1, 2, ..., (N-1) sayıları ile ifade edilebilir. Örneğin N'nin 10 olduğu durumda çıktı katmanında aktivasyon fonksiyonu [0, 1] arasında değer alan 4 hücre ile sınıflar ifade edilebilir. 7 0111, 8 1000, 3 ise 0011 olarak ifade edilir. Sınıflandırmada genelde her sınıfın olasılığı, dolayısıyla iki sınıf arasındaki uzaklığın eşit olduğu varsayılır. Ancak bu yöntemde 7 olması gereken bir çıktı 3 ile karıştırıldığında 1 hücre değeri, 8 olarak karıştırıldığında ise 4 hücre değeri yanlış olmaktadır. Dolayısıyla sınıflar arası fark eşit ve doğru şekilde ifade edilmemiş olmaktadır.

Bu durum bire bir kodlama (One-hot Coding) ile çözülmektedir. Bu kodlama yönteminde N sınıfın hepsi N elemanlı vektör ile ifade edilmektedir. Vektörün tüm değerleri 0'dır. Sadece girdinin ait olduğu sınıf numarasına karşılık gelen elemanı 1'dir (Zhang vd., 2019). Bir başka ifade ile sınıfa ait vektörün tüm elemanları 0 (sönük), sadece sınıf numarasına karşılık gelen elemanı 1'dir (sıcak). Bu yüzden vektör one-hot ismini almaktadır. Şekil 21'de 10 sınıfın one-hot karşılığı görülmektedir. Bu vektörün en önemli avantajı her sınıf arasındaki farklı bit sayısının aynı (2) olmasıdır. Bu sayede sınıflar arası fark eşit şekilde ifade edilmektedir.

Sınıf Numarası	0	1	2	...	9
one-hot vektörü	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	...	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

Şekil 21: 10 Sınıflı Bire Bir Kodlama Örneği

Sigmoid fonksiyonu ile bir hücrenin çıktısı [0, 1] aralığında değer almaktadır. Bu sayede idealde 0 veya 1 değeri alması gereken iki sınıfın olasılığını vermektedir. Çoklu sınıflandırma probleminde ise ideal çıktıyı one-hot vektör ifade etmektedir. Çıktı katmanında sınıf sayısı kadar hücre ile softmax aktivasyon fonksiyonu kullanıldığında her hücrenin çıktısı pozitif ve hücrelerin çıktıları toplamı 1 olur. Bu durumda softmax

fonksiyonunun her elemanı o hücre numarasına karşılık gelen sınıfın olasılığını verir (Bhardwaj vd., 2018). Değeri en büyük olan hücreye karşılık olan sınıfın olasılığı en büyük olduğundan girdinin bu sınıfa ait olduğu yorumu yapılır. Bu yüzden sınıflandırma problemlerinde çıktı katmanında softmax fonksiyonu kullanılır. Softmax fonksiyonu bir hücreye değil, bir katmana uygulanır (Bhardwaj vd., 2018). Formülü şöyledir:

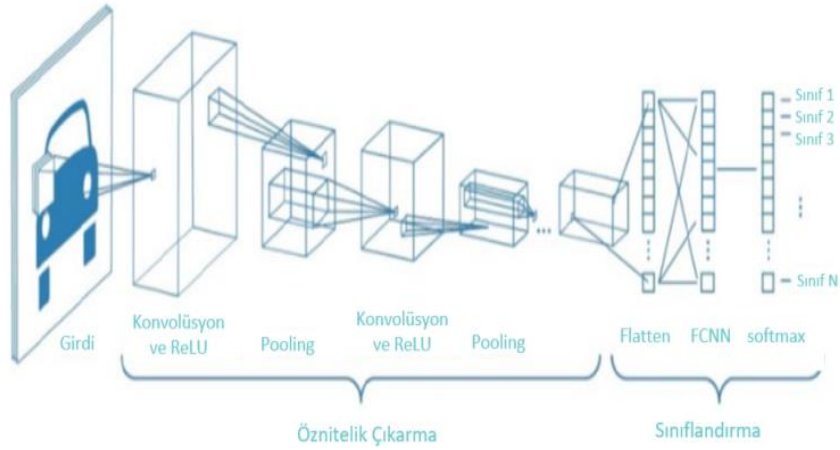
$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

Burada x_i , i. hücrenin aktivasyon fonksiyonu öncesindeki değerini ifade etmektedir.

2.4.3. Evrişimsel sinir ağı

Evrişimsel sinir ağı (Convolutional Neural Network, CNN) biyoloji ve matematiğin bir araya gelmesiyle ortaya çıkmıştır. İnsan gözünden alınan sinyallerin birtakım filtreler yardımıyla beyinde elendiği biyolojik olarak bilinmektedir. Dolayısıyla bu yaklaşımla bir algoritma ortaya konulmuştur. Burada problemin genel tanımı alınan bir görüntünün türünü belirlemek olarak düşünülebilir. İnsan düşünme yetisini kullanmadan dahi bu işlemi yapabilmektedir. Bu yetenek, doğru örüntü tanıma işidir. Dolayısıyla bir sınıflandırma problemi olarak düşünülebilir.

Burada kullanılacak olan CNN modellemesi iki boyutlu bir matris almaktadır. Eğer renk bilgisi de kullanılırsa 3 boyut üzerinden incelenebilir. CNN modellerin farklı versiyonları bulunmaktadır. Tek boyutlu, iki boyutlu ve üç boyutlu tüm veriler üzerinde uygulanabilir. Birçok filtreden, eleme işlemlerinden ve FCNN katmanlarından oluşmaktadır (Zhang vd., 2019). Bu modeller genellikle büyük olduğundan dolayı eğitilebilmesi için daha çok veri gerekmektedir ve eğitilmesi uzun sürmektedir. Ancak derin yapısı sayesinde görüntülerden problemin çözümü için gerekli bilgileri kademe kademe çıkarabilmektedir (Goodfellow vd., 2016). CNN bir derin öğrenme modeli olarak kabul edilmektedir.



Şekil 22: Evrimsel Sinir Ağı Örneği

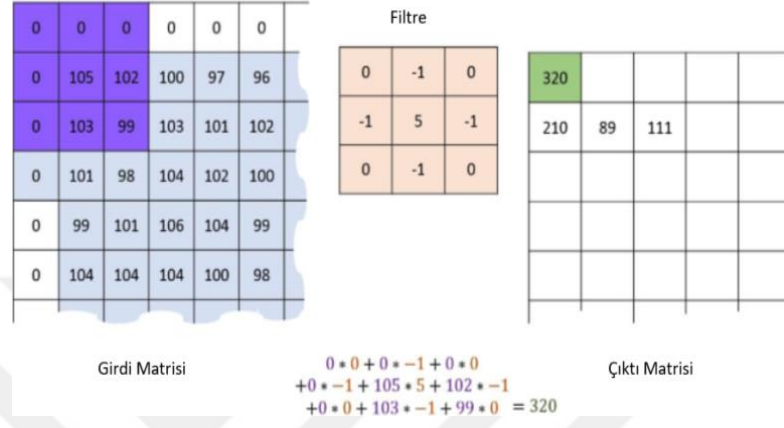
Şekil 22 temsili olarak evrimsel yapay sinir ağının açık modelini göstermektedir. Burada ilk katmanlar evrişim kısımlarıdır. Bu ilk katmanlar genellikle daha az detay içeren öğeleri algılarken, model derinleşmeye başladıkça daha detaylı özellikleri algılamaya başlamaktadır. Modelin en son birkaç katmanı olarak FCNN kullanılır. FCNN öncesi katmanlar görüntüden öznitelik çıkaran, FCNN kısmı ise problemin çözümünü öğrenmeyi sağlayan katmanlardır.

Öznitelik, problemin çözümü için değerli olan ve girdiden çıkarılan bilgilerdir. Örneğin, hastalık tespiti probleminde özniteliklerden biri kandaki antikor sayısı olabilir, yüz tanıma probleminde bir öznitelik göz ile burun arasındaki uzaklık olabilir. Bu gibi özniteliklerin bazıları matematiksel olarak hesaplanabilir. Lakin görüntüden suç tespiti gibi karmaşık problemlerde CNN gibi derin öğrenme modelleri ham girdiden öznitelikleri kendisi çıkarabilir. Genelde bunların tam olarak neye karşılık geldiği bizim tarafımızdan anlaşılamayabilir ama FCNN'ye girdi olarak verilerek problemin cevabına ulaşmada önemli rol oynayabilirler. CNN'yi oluşturan katmanlar sonraki bölümlerde anlatılmaktadır.

2.4.3.1. Konvolüsyonel katman

Konvolüsyon bir matematiksel ifadedir ve vektör veya matris üzerindeki verilerin birçok farklı şekilde değişimini sağlar. Girdi ile filtre (filter, kernel) konvolüsyon operasyonundan geçirilerek çıktı hesaplanır (Zhang vd., 2019). CNN'nin çıkış noktası iki veya daha çok boyutlu matrislerde konvolüsyon yapmak ve böylece görüntülerden anlamlı bilgiler çıkarabilmektir. Bunu konvolüsyonel katman (Convolution Layer) sağlar. CNN'de konvolüsyon işlemlerinin amacı doğru bilgileri çıkarmaktır. Bunların belirli ağırlığı

olmaktadır. Örneğin insanın yüzünü tespit edebilen bir filtre ile arabanın tekerleğini tespit eden filtrelerin katsayıları birbirlerinden farklı olacaktır. Dolayısıyla eğitilmesi gereken şeylerden bir tanesi de bu katmanların ağırlıklarının değerleridir. Bir konvolüsyonel katmanda farklı boyutlarda ve sayıda filtreler olabilir. Bu filtrelerin değerleri geri yayılım ile eğitilir.



Şekil 23: Konvolüsyon Örneği

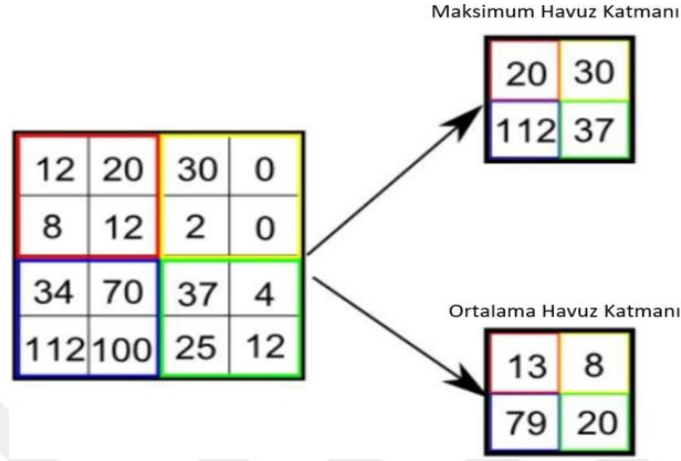
Şekil 23’de konvolüsyon örneği verilmektedir. Filtre Girdi üzerinde bir bölgeye getirilir. Elemanlar çarpılıp toplanarak çıktı elde edilir. Filtre sürekli yana ve aşağı kaydırılarak çıktı matrisi oluşturulur (Goodfellow vd., 2016). Çıktı matrisinin büyüklüğü ve değerini filtrenin büyüklüğü ve her adımda filtreyi kaydırma miktarı (stride) belirler (Zhang vd., 2019). Örneğin filtre birer birer kaydırılırsa çıktı büyüklüğü yaklaşık olarak girdininkiyle aynı olur ama filtre ikişer ikişer kaydırılırsa çıktı matrisi büyüklüğü yaklaşık olarak girdininkinin yarısı olur.

Ayrıca girdinin etrafına Şekil 23’deki gibi bir veya birkaç kat sayılar dizilebilir. Buna padding denir (Zhang vd., 2019). Şekil 23’de sıfır (zero) padding kullanılmaktadır. Padding yapılırken komşuyla aynı değeri atamaya aynı (same) padding denir. Bu başlıca yöntemler dışında farklı padding yöntemleri vardır. Bunlar, çıktının kenarlarında keskin değişimleri önleyerek daha düzgün geçiş sağlamada veya çıktı boyutunu ayarlama kullanılır.

2.4.3.2. Havuz katmanı

Konvolüsyonel katmanda parametreleri öğrenilen filtreler yer alır. Bunun dışında görevleri önceden belirlenen hazır filtreler kullanmak da gerekebilir. Bu tür filtrelerin kullanıldığı katmanlara havuz katmanı (Pooling Layer) denir (Zhang vd., 2019). Örneğin

maksimum havuz katmanı (Max Pooling) girdilerden en büyüğünü, ortalama havuz katmanı (Average Pooling) girdinin ortalamasını verir (Zhang vd., 2019). Şekil 24'te bu 2x2'lik havuz katmanına verilen 4x4'lük girdinin sonucu gösterilmektedir.



Şekil 24: Maksimum ve Ortalama Havuz Katmanı Örneği

Görüntüler büyük matrisler olduğundan dolayı tümüyle işlenmesi zordur. Bu gereken işlem gücünü azaltmak için katmanlar ilerledikçe ve düşük seviye çıkarımlar yapıldıkça havuz katmanları kullanılmaktadır. Bu da gerekli olan hesaplamada büyük bir kazanç sağlar. Havuz katmanı birkaç konvolüsyonel katmandan sonra genellikle yapılmaktadır.

2.4.3.3. Düşüm atma

Düşüm atma (dropout) sadece CNN'ye özel bir durum değildir ama CNN'lerde sıkça kullanılmaktadır (Bonaccorso, 2017). Bu genellikle modelde çok fazla nöron bulunması durumunda ortaya çıkan bir sorunu çözmek için gerekmektedir. Problemi öğrenmek için çok hücreli ve çok sayıda katman kullanmak gerekebilir. Bu durumda modelin karmaşıklığı artacaktır. Bu problemin çözümü için gerekebilir ama her girdi için sonuç hesaplarken tüm hücreleri kullanmak iyi sonuç vermeyebilir. Bu durum farklı duyularımızdan gelen uyarılar değerlendirilirken beynimizin farklı bölgelerinin yoğun şekilde kullanılması durumu ile özdeşleştirilebilir.

Düşüm atma ile önceki katmandaki her hücre her geri yayılım iterasyonunda belirlenen olasılıkla rastgele kapatılır. Böylece o katmanda sadece seçilen hücreler ile çıktı hesaplanır ve o hücreler geri yayılım ile güncellenirler (Bonaccorso, 2017). Bu şekilde girdinin etkisi daha küçük bir gruba daha yoğun olarak yansır. Düşüm atma işlemi ile nöron

sayısı her iterasyonda farklı sayıya düşmektedir. Normalde eğitilmesi gerekenden daha az nöron eğitilerek daha hızlı ve verimli bir eğitim gerçekleştirilebilir.

2.4.3.4. Yığın normalizasyonu

Yığın normalizasyonu (Batch Normalization) katmanı da CNN'ye özel olmamakla birlikte son yıllarda kullanımı yaygınlaşmaktadır. Yüksek sayı değerleri ile işlem yaparken sistemin optimal noktayı bulması zaman almaktadır. Dolayısıyla yalnızca girdi katmanında normalizasyon yapmak yetersiz geleceğinden dolayı katmanlar arasında da normalizasyona gerek duyulmaktadır. Bu katman önceki katmanın çıktısını normalizasyondan geçirerek sonraki katmana iletir (Zhang vd., 2019). Böylece modelin girdideki aykırı durumlara hassaslığını azaltarak daha güçlü bir model elde etmeyi sağlar.

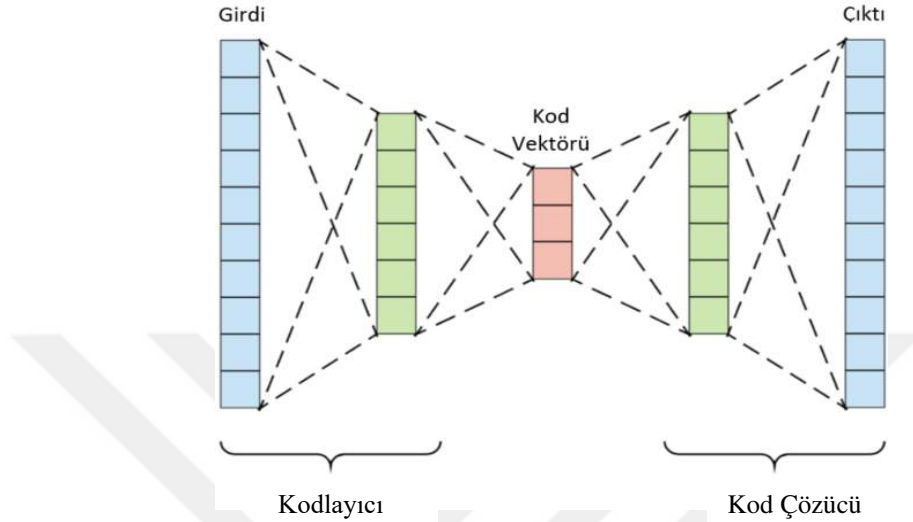
2.4.3.5. Tamamen bağlı katman

Tamamen bağlı katman bu modelin öğrenme kısmını gerçekleştiren katman olarak düşünülebilir. Konvolüsyon ve havuz işlemlerinden sonra model, girdiyi parametre sayısı çok daha küçük bir matrise indirger. Sonrasında elde edilen 2 veya 3 boyutlu matris, tek bir vektör haline getirilir. Bu işleme Flatten denir (Zhang vd., 2019). Sonra vektör, sınıflandırma veya regresyon problemi için tasarlanan FCNN'ye verilerek problemin sonucuna ulaşılır. FCNN hücresi olarak uygulamaya göre perceptron veya farklı hücreler kullanılabilir.

2.4.4. Özkodlayıcılar

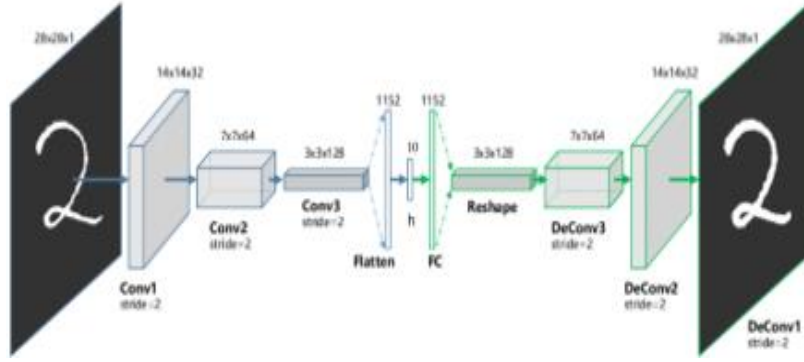
Özkodlayıcılar (Autoencoders, AE) genellikle resim, ses gibi dosyaların sıkıştırılmasını sağlayan sıkıştırma algoritmaları gibi çalışmaktadır. AE kodlayıcı (encoder) ve kod çözücü (decoder) olmak üzere genellikle simetrik iki bölümden oluşur (Goodfellow vd., 2016). Kodlayıcı bölümünde girdi her katmanda kendinden daha küçük bir vektöre dönüştürülerek devam eder. Kod çözücü bölümünde elde edilen kod vektörü yeniden katman katman büyütülerek orijinal boyuta getirilir. Şekil 25'te bu durum gösterilmektedir. Etiket olarak genelde orijinal girdi verilir. Eğitim sonunda sadece kodlayıcı kısmı kullanılır. Böylece girdiden daha küçük boyutlu bir vektör elde edilmiş olur. Böylece sistemin zamanla girdinin temsilini öğrenmesi beklenmektedir (Goodfellow vd., 2016). Bu sayede girdinin boyutunu azaltma, bir amaca yönelik girdiden öznelik çıkarma, dil ve benzeri

uygulamalarda kelimelerin vektörel ifadesi (word embedding) gibi uygulamalar başarı ile yapılabilmektedir. AE bu özelliği sayesinde gözetimsiz derin öğrenme modelidir. Kod çözücünün çıktısının FCNN'ye verilmesi gibi hibrit modellerle sınıflandırma gibi uygulamalarda da kullanılabilmektedir.



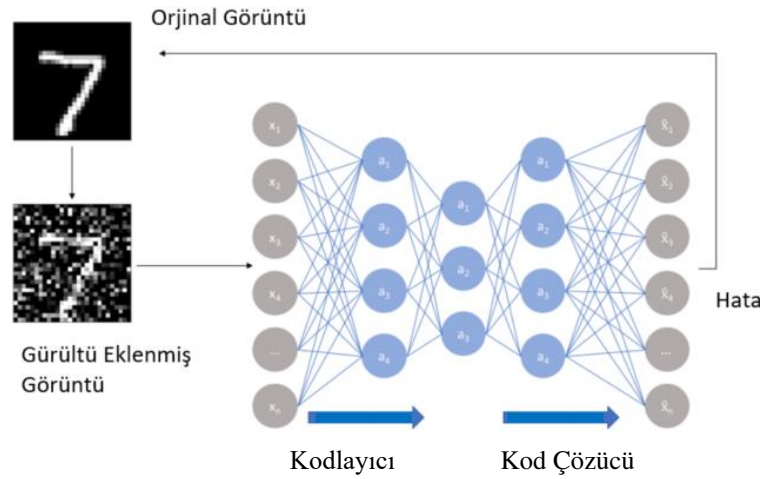
Şekil 25: Özkodlayıcı Örneği

AE yapı olarak esnektir ve probleme göre farklı tür katmanlar içerebilir. Örneğin vektörel girdi için perceptron katmanlarından oluşan AE, görüntü girdileri için konvolüsyonel katmanlardan oluşan AE tasarlanabilir. Şekil 26'da el yazısı ile yazılmış rakamların temsilini öğrenen konvolüsyonel katmanlardan oluşan AE örneği verilmektedir. Bu tür modellere kısaca konvolüsyonel özkodlayıcı (CAE) denilmektedir (Sewak vd., 2018). CAE'nin kod çözücü katmanında kodlayıcı katmandaki operasyonların tam tersi yapılmaktadır ve bu yüzden CNN'de olmayan yapılar da barındırmaktadır. Örneğin, konvolüsyonu geri almak için ters konvolüsyon (deconvolution), havuz katmanı yerine yukarı örnekleme (upsampling) katmanları bulunmaktadır. Sistemin öğrenmesi beklenen şey burada girdi fonksiyonu tekrar üretmek olacağı için genelde etiket olarak da girdi fonksiyonu verilmektedir. Teknik olarak etiket kullanılsa bile girdinin bir sınıf veya parametre bilgisi önceden belirlenip kullanılmadığı için ve kullanım tarzı bakımından gözetimsiz öğrenme modelidir. Buradaki hata fonksiyonu gerçek girdi ile üretilen girdi arasındaki fark sonucunda ortaya çıkmaktadır.



Şekil 26: CAE Örneği

AE'lerin birçok uygulama alanı bulunmaktadır. Boyut azaltma, öznitelik çıkarma, kelime kodlama, veri sıkıştırma, görüntü iyileştirme, gürültü temizleme ve anormal olay tespiti bunlardan başlıcalarıdır. AE'ye girdi olarak gürültü barındıran veri, etiket olarak gürültüsü temizlenmiş veri verilirse ses vb. verilerde gürültü temizleyen bir model geliştirilmiş olur (Sewak vd., 2018). AE ile Şekil 27'de olduğu gibi görüntü iyileştirme de yapılabilir. Bu modele aynı görüntünün kötü versiyonu girdi, iyi versiyonu çıktı olarak verilmektedir. Bu sayede AE kodlayıcı aşamasında görüntünün temsili çıkarıp, kod çözücü aşamasında olması gereken iyi görüntüyü geri elde edebilmektedir.

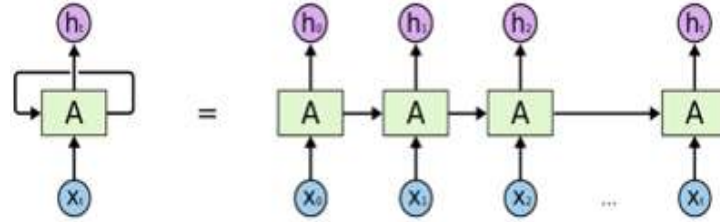


Şekil 27: Gürültü Temizleme için AE Eğitimi Örneği

2.4.5. Özyinelemeli sinir ağı

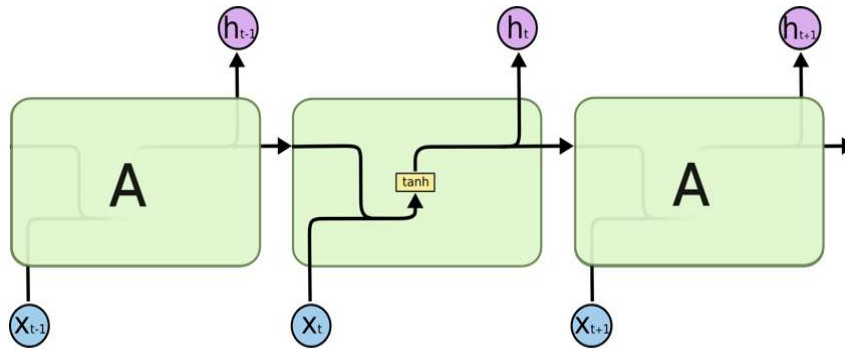
CNN ve MLP gibi yapılardaki en büyük eksik, bir hafıza hücresinin bulunmamasıdır. Yani eğer problemde bazı bilgiler hafıza gerektiriyor ise bu modeller problemlere bir çözüm getirmeyecektir. Bu hafıza sorununu çözmek için özyinelemeli sinir ağı (Recurrent Neural

Network, RNN) fikri ortaya çıkmıştır (Goodfellow vd., 2016). Burada her girdi farklı zamanlardaki bilgiyi temsil etmektedir. Bu bilgiler birbirine aktarılarak devam etmektedir. Her nöronun bir girdi kısmı, çıktı ve bilgiyi aktarmayı sağlayan ek bir bağlantısı vardır (Mandic ve Chambers, 2001). Bu bağlantılar sayesinde zamanlar arasındaki bilgi aktarımı gerçekleşerek her bir nöron için ayrı ayrı çıktı üretilmektedir.



Şekil 28: RNN Hücresinin Çıktısının Zamansal Gösterimi

Şekil 28’de bir RNN hücresinden oluşan A modeli gösterilmektedir. Eşitliğin solunda modelin gerçek fiziksel hali, eşitliğin sağ tarafında ise model çalışırken zamanla çıktı üretmesi anlatılmaktadır. Her adımda x girdisi alan hücre h çıktısını üretmektedir. Model ilk zaman diliminde x_0 girdisine karşılık h_0 çıktısını verilmektedir. Sonraki zaman diliminde x_1 girdisi ve bir önceki adımdan gelen değer sonucu ile h_1 çıktısı üretilmektedir ve yeni bilgi sonunda tutulan parametre güncellenmektedir (Mandic ve Chambers, 2001). Her zaman diliminde girdi ve geçmiş girdiler sonucu hesaplanan değer ile çıktılar elde edilerek, her adımda sadece o anki girdiye göre değil ayrıca geçmiş girdiler de göz önünde bulundurularak çıktı üretilmektedir (Goodfellow vd., 2016).



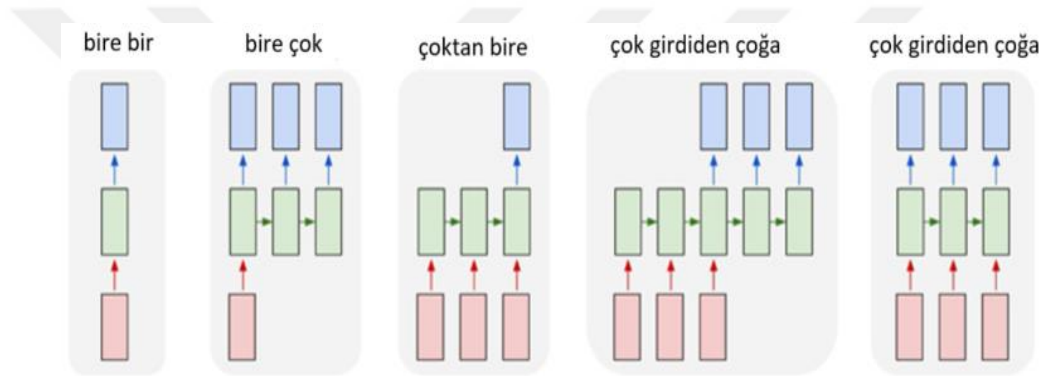
Şekil 29: Klasik RNN Hücresi Yapısı

Şekil 29’da gösterildiği gibi RNN’de çıktı hesaplanırken, bir önceki zaman dilimindeki çıktı değeri de göz önünde bulundurularak geçmiş bilgisi o anki zaman dilimine aktarılmış olur. Burada bir önceki gelen bilgi ve o anki bilgi belirli bir ağırlıklar ile

güncellenerek bir tanjant fonksiyonundan geçirildikten sonra bir sonraki hücreye aktarılmaktadır. Bu durum,

$$h_t = \tanh(Ux_t + Wh_{t-1})$$

formülü ile daha net görülmektedir. Burada x_t t-anındaki girdi vektörünü, h_t ise t-anındaki çıktı vektörünü temsil etmektedir. U ve W ise hücreye ait matrislerdir. Bu matrisin elemanlarının değerleri eğitme sürecinin sonunda belirlenir. Bu şekilde t-anında x_t ve h_{t-1} değerlerinden h_t çıktısı hesaplanır. Böylece anlık çıktı hesaplanırken geçmiş girdiler de değerlendirilmiş olur.



Şekil 30: Çeşitli RNN Modelleri

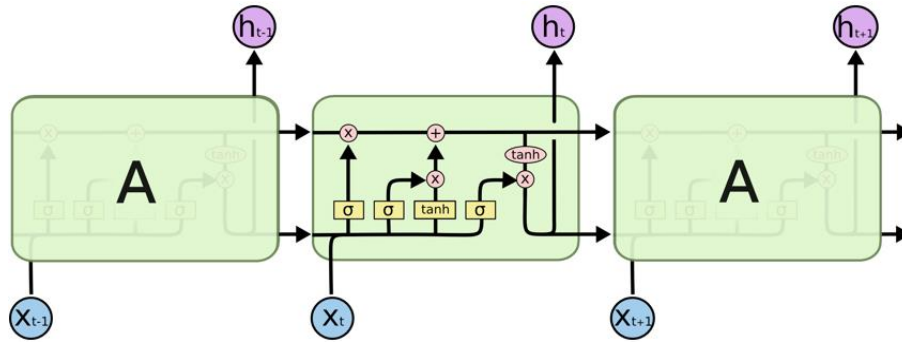
Şekil 30'daki bire bir (one to one), bire çok (one to many), çok girdiden bire (many to one), çok girdiden çok çıktıya (many to many) yapılarının ifadesi girdilerin hangi şekilde olabileceğini ifade etmektedir. Örneğin yalnızca 1 girdisi olan bir modelin birden çok üreteceği bir çıktı olabilir. Ya da ikiye ikiye artmakta olan bir problemde bazı nöronların girişleri boş bırakılarak problem çözdürülebilir. Bu esnek yapısı sayesinde farklı boyuttaki girdiler aynı modele sorunsuzca verilebilir.

Klasik RNN hücresinde geçmişten gelen değer sürekli matrisle çarpılarak yenilenmektedir. Dolayısıyla bir zaman dilimindeki girdi sonucu hesaplanan değer her yeni zaman diliminde etkisini kaybetmektedir. Bir başka ifade ile sonraki girdilerin çıktısındaki etkisi, öncekilere göre her zaman daha çok olmaktadır. Ancak, bazı uygulamalarda önceki zaman dilimindeki girdi, sonrakinden daha kritik olabilir. RNN'de bu durumu değerlendirecek bir mekanizma bulunmamaktadır (Goodfellow vd., 2016). Ayrıca zamanla etkisi azalsa bile her çıktıda geçmiş tüm girdilerin etkisi bulunmaktadır. Özyinelemeli sinir ağının en büyük problemi uzun süreli geçmişe bir bağlılık vardır. Bazen çok geçmişte kalan

bilgilere ihtiyaç duyulmamaktadır. Bu tarz problemler için, gerektiğinde geçmiş bilgiyi unutacak uygun bir yapısı yoktur. RNN'nin geçmiş önemli girdileri daha çok, önemsiz bilgileri ise daha az göz önünde bulundurmayı sağlayacak yapısı bulunmamaktadır. Bu yüzden dil çevirisi gibi birçok doğal dil işleme, finans gibi önceki bilgileri hatırlamayı gerektiren ama önemli girdinin zaman diliminin de değiştiği uygulamalarda yetersiz kalmaktadır (Mandic ve Chambers, 2001).

2.4.5.1. LSTM

RNN'nin en büyük eksikliği girdileri önemine göre değerlendirememesidir. Değeri ne olursa olsun belirli bir zaman dilimi önceden gelen girdinin çıktı hesaplanırken aynı ağırlığı bulunmaktadır. RNN geçmişe çok bağlı bir yapıda tasarlandığından dolayı bazı problemleri çözmesi çok zor hale geliyor. Buna çözüm olarak ise bilgilerin hücreler arasında aktarılmasını engelleyecek mekanizmaları olan uzun ömürlü kısa dönemli bellek (Long Short-term Memory, LSTM) adlı RNN hücresi ortaya çıkmıştır (Mandic ve Chambers, 2001). LSTM hücreleri gerekirse aktarım yapabilir, gerekirse aktarım yapmayı durdurabilir. Böylelikle hafızadaki bilgi sadece gerekli görüldüğünde değerlendirilir ve daha esnek bir yapı ortaya çıkar. Bu özelliğinden dolayı RNN modellerinde hücre olarak çok sıklıkla LSTM hücresi kullanılmaktadır. Bundan dolayı literatürde hücre olarak LSTM kullanılan RNN modeli demek yerine kısaca LSTM modeli denilmektedir.



Şekil 31: LSTM Hücresi Yapısı

Şekil 31'den anlaşılacağı gibi bu yapı RNN'den daha karmaşık bir aktarım yapmaktadır. Burada her hücre arasında iki aktarım vardır. Bu şekilde geçmişteki bilgi hem ileriye aktarılmaktadır hem de istenirse belirli bir hücrenin hafızasından silinebilmektedir. Buradaki ilk sigmoid katmanı bilgiyi unutup unutmayacağını tutmaktadır. Girdinin tanh

fonksiyonu ile birleştiği yerdeki sigmoid fonksiyonu hangi değerlerin güncelleneceğine karar vermektedir. Tanh ve sigmoid fonksiyonları birleşerek bir güncelleme sağlamaktadır (Mandic ve Chambers, 2001). LSTM yapısı matematiksel olarak,

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

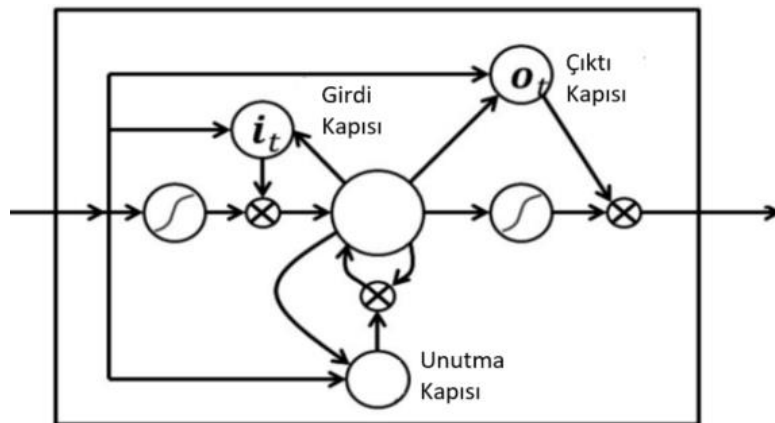
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

$$h_t = o_t \circ \sigma_h(c_t)$$

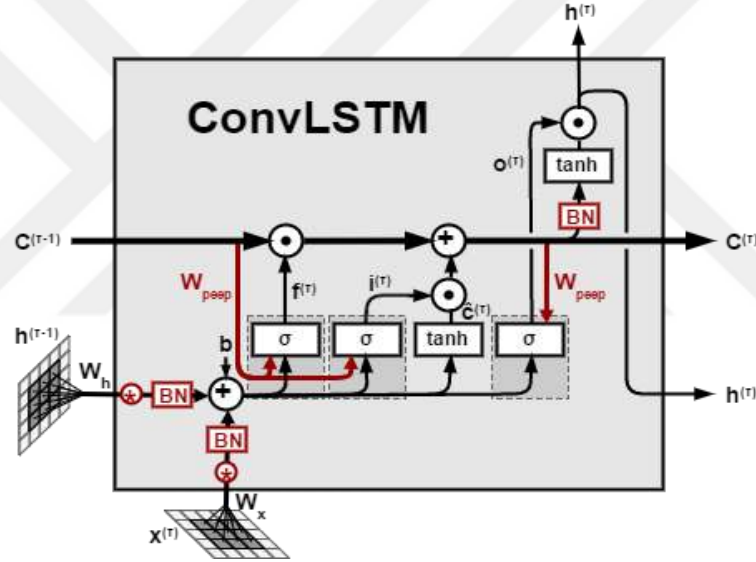
formülleri ile ifade edilebilir. Burada x_t t-anındaki girdi vektörünü, h_t ise t-anındaki çıktı vektörünü temsil etmektedir. LSTM hücresi 3 temel yapıdan oluşmaktadır: Girdi Kapısı (Input Gate), Çıktı Kapısı (Output Gate) ve Unutma Kapısı (Forget Gate) (Goodfellow vd., 2016). Hücrede, eğitime sırasında değerleri belirlenen, her kapıda birer U, W matrisi ve b vektörü bulunmaktadır. Girdi kapısı ile girdiden hücredeki parametreler güncellenmektedir. Çıktı kapısı ile bir sonraki hücreye gerektiğinde bilgi aktarılmakta, gerektiğinde aktarılmaması sağlanmaktadır. Unutma kapısı ile gerektiğinde hafıza temizlenerek geçmiş bilgilerin unutulması sağlanmaktadır. Bu durum Şekil 32’de gösterilmektedir.



Şekil 32: LSTM Hücresinin Girdi, Çıktı ve Unutma Kapıları

2.4.5.2. ConvLSTM

Derin öğrenmenin gücü ham veriden direk güçlü çıkarımlar yapabilmesidir. LSTM girdi ve çıktısı vektör olabilmektedir. Bu yüzden matris verileri arasındaki zamansal ilişki LSTM kullanılarak incelenemez. Bu yüzden girdi olarak matris alıp çıktı olarak matris üreten, Şekil 33’de gösterilen, ConvLSTM hücresi tasarlanmıştır (Xingjian vd., 2015). CNN’de olduğu gibi matrislerden bilgi çıkarımı konvolüsyon ile başarılı şekilde yapılmaktadır. Bu yüzden ConvLSTM hücresi konvolüsyon operasyonu içermektedir. Bu yüzden konvolüsyon ve LSTM operasyonlarının başarılı şekilde birleşimi olarak ConvLSTM adını almıştır. ConvLSTM hücresi içeren konvolüsyonel sinir ağlarına kısaca ConvLSTM modeli de denilmektedir. ConvLSTM ile matrisler arası zamansal ilişki analiz edilerek başarılı çıkarımlar yapılabilmektedir.



Şekil 33: ConvLSTM Hücresi Yapısı

ConvLSTM yapısı matematiksel olarak,

$$\begin{aligned}
 i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
 C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
 H_t &= o_t \circ \tanh(C_t)
 \end{aligned}$$

şeklinde ifade edilebilir. Burada X_t t-anındaki girdi matrisini, H_t ve C_t ise t-anındaki çıktı matrislerini temsil etmektedir. Önceki LSTM'den farklı olarak bir yerine iki çıktı bir sonraki zaman dilimine aktarılmaktadır. LSTM'de olduğu gibi birbirinden bağımsız girdi, çıktı, unutma kapıları yer almaktadır. LSTM'deki matris vektör çarpımları yerine iki matris arasındaki konvolüsyon operasyonu bulunmaktadır.

2.5. Video Verisi ve Video Anomali

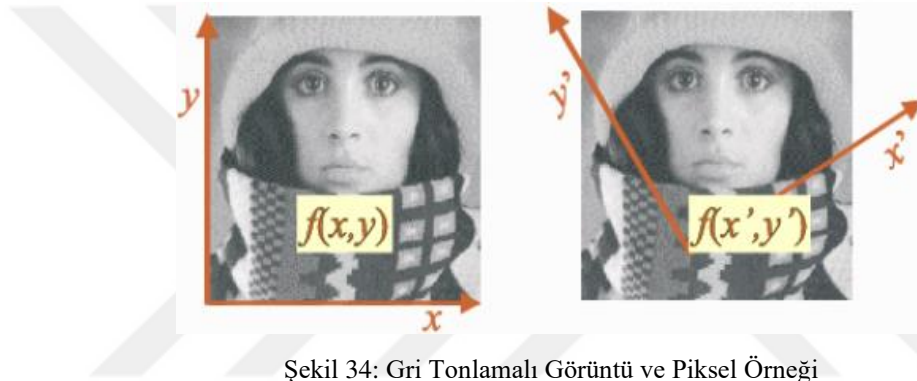
Bilgisayarla görü (Computer Vision) bilgisayarlara dünyayı görsel olarak anlamayı amaçlayan yapay zeka ve bilgisayar bilimi alanıdır (Szeliski, 2010). Görüntü verilerinden en iyi şekilde yararlanmak için bilgisayarların bir görüntüyü (image) “görmesi” ve içeriği anlaması gerekir. Görüntü işleme, dijital sinyal işleme türüdür ve içeriği bir şekilde basitleştirerek veya geliştirerek mevcut bir görüntüden yeni bir görüntü oluşturma işlemidir (Jensen, 1996).

Bizim için çok kolay olan çıkarımlar bilgisayar için çok zor olabilir. Bunun sebeplerinden bir tanesi de insanların görsel algısının ve beynin nasıl çalıştığı hakkında yeterince bir bilgi sahibi olmamasıdır. Görüntü işleme problemlerinin zor olmasının diğer bir sebebi ise gerçek hayat problemlerinin daha karmaşık olmasından kaynaklanmaktadır. Yansıma miktarı, ortamın ışıklandırılması ve kameranın çözünürlük kalitesi gibi etkenler problemin zorluğunu artırmaktadır (Szeliski, 2010). Doğru bir görüntü işleme sistemi sonsuz sayıda görüntüyü, senaryoyu görebilmeli ve onlardan anlamlar çıkarabilmelidir, fakat pratikte bu kadar kolay gerçekleşmemektedir. Bilgisayar her şeyi ikili (binary) sayılar olarak görmektedir. Bu, bizim için zor olan bazı problemleri algoritmaların kolayca çözmelerine olanak verirken, bizim için çok kolay olan bazı problemlerin ise çözülememelerine neden olur.

Anormal davranış tespiti, hedeflerin olağandışı davranışını veya durum geçişini tespit etmek için kullanılır. Gerçek dünyada, normal ve anormal davranış kalıpları belirli bir konuya bağlı olarak değişir. Ortamdaki normal davranış, farklı ortamlarda veya durumlarda anormal olabilir. Bu nedenle, anormal davranış algılama modeli belirli bir konuya göre özelleştirilmelidir. Bu açıdan bakıldığında suç tespiti bir anomali tespiti uygulamasıdır. Örneğin suç konseptinde, bir kişinin kendi evine girmesi normal bir durum iken, sahibi olmadığı eve girmesi ise anormal bir durumdur. Bu yüzden güvenlik kameralarında suç tespiti problemi teknik olarak video verilerinde anomali tespiti problemine denktir.

2.5.1. Görüntü verisi

Görüntü en basit haliyle iki boyutlu bir matris olarak düşünülebilir. Bu matris 2 boyutlu bir düzlemde x ve y koordinatları üzerinde değerler alabilen bir $f(x,y)$ fonksiyon olarak tanımlanabilir. $f(x,y)$ fonksiyonun her x,y değerleri için aldığı değerler yoğunluk (intensity) olarak tanımlanabilir (Jensen, 1996). Burada bahsedilen görüntüye gri tonlamalı (grayscale) görüntü denir. Her bir x,y ikilisinin gösterdiği noktaya da piksel denir. Şekil 34 üzerindeki resimlerde bunu açık bir şekilde görmekteyiz. Her piksel değeri 0-255 arası tam sayı değeri alır. 0 hiç ışık olmaması, 255 ise ışığın en yoğun olduğu değerdir. Bu tam sayı değerleri ışık oranını verir (Forsyth ve Ponce, 2002).



Şekil 34: Gri Tonlamalı Görüntü ve Piksel Örneği

İnsanlar için renk, insan deneyiminin duygusal, estetik ve kültürel yönleriyle derinden bağlantılı görsel bir özelliktir. Fizik bağlamında, renk sadece elektromanyetik radyasyonun farklı dalga boylarına atıfta bulunmak için uygun bir yoldur. Tamamen fiziksel alanda, renk gerçekten mevcut değildir (Backhaus vd., 2011).

2.5.1.1. Renk modeli ve RGB

İnsan bilişinin dalga boyuna dayalı renkli görüntüleri formüle etmesinin kısmen psikolojik yolu, bir makinede çoğaltılabilecek bir şey değildir. Bunun yerine makineler, insan görsel algısının doğası gereği matematiksel bir yaklaşımla tanımlanan bir renk modeli kullanır.

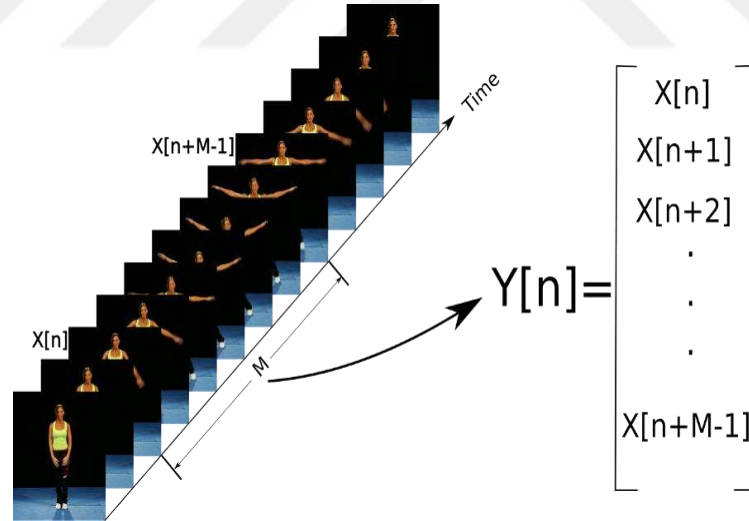
Renk için üç kanal gerekir ve bu kanallar için RGB modeli kullanılır (Jensen, 1996). RGB kırmızı (red), yeşil (green), mavi (blue) anlamına gelir. Geniş bir renk yelpazesi üretmek için değişen oranlarda RGB eklenir. Renkli görüntüler üç boyutlu bir matristir.

Örneğin bir mor rengi; 33 birim kırmızı yönde, 39 birim yeşil yönde ve 127 birim mavi yönde uzanan bir vektörle elde edilir.

2.5.1.2. Video verisi

Video bir görüntü serisidir ve saniye başına düşen çerçeve sayılarından (frame per second/fps) oluşmaktadır (Szeliski, 2010). Her saniye eşit fps'lik zaman dilimlerine bölünür ve her zaman diliminde bir resim ekrana yansıtılır. Böylece bir süreklilik algısı oluşur. En yaygın fps değeri 30'dur. Dolayısıyla video dört boyutlu (4. boyut: zaman, çerçeve sayısı) bir matris olarak düşünülebilir ve bu şekilde algoritmalara girdi olarak verilebilir. Veya üç boyutlu matris serisi olarak düşünülüp modellere bu şekilde verilebilir. Şekil 35'de, dört boyutlu bir video matrisi örneği verilmektedir. Bu videoda belirli bir aralıktaki görüntüler örnek uzayına alınmakta ve daha sonra tamamı bir matris üzerinde birleştirilmektedir.

Görüntü işleme ve makine öğrenmesi problemlerinde elde edilen bu matris üzerinden işlemler yapılmaktadır. Bu işlemler, video üzerinden sağlanabilecek bilginin büyük bir miktarını tutmayı sağlamaktadır.



Şekil 35: RGB Görüntü Serisi (Video) Örneği

2.5.1.3. Görsel veri ön işleme

Video bir önceki bölümde bahsedildiği gibi görüntü çerçevelerinden oluşmaktadır. Bu çerçeveler makine öğrenmesi algoritmalarına verilmeden önce ön işlemeden geçirilirler. Böylece modelin daha iyi öğrenmesi amaçlanır (Szeliski, 2010). Bunlara örnek olarak

normalizasyon verilebilir. Bu işlemde matris veya dizideki sayılar arasındaki oranlar korunarak $[0, 1]$ aralığına ölçeklendirilirler (Bonaccorso, 2017). Bu uygulama neredeyse bütün makine ve derin öğrenme yöntemlerinde uygulanmaktadır.

Sistemin öğrenmesini kolaylaştırabilecek diğer bir yöntem ise öznitelik çıkarmadır. Öznitelikler girdiden matematiksel veya benzeri yöntemlerle elde edilen çıkarımlardır. Bazı problemlerde, yapay sinir ağı modeline girdi olarak özniteliklerin verilmesi ile daha iyi sonuçlar elde edilebilir. Derin öğrenmenin gücü çok katmanlı yapısı ile veriden öznitelikleri kendisinin çıkarmasıdır. Bu şekildeki çıkarımlar klasik öznitelik çıkarma yöntemlerine göre daha güçlüdür. Derin öğrenme ile görüntü veya video işlemede de öznitelik çıkarma yerine girdinin doğrudan verilmesi daha iyi sonuçlar vermektedir (Jiang vd., 2018; O'Mahony vd., 2019).

2.5.2. Video anomali türleri

Suç görüntüleri günlük davranışlar arasından kavga etme, koşma, yerde yatma, el kaldırma, silah doğrultma, kesici alet kullanma gibi anormal durumların bir veya birkaçını içermektedir. Görüntülerden davranış tanıma, anomali tespiti gibi analizler için literatürdeki en başarılı yöntemler genellikle derin öğrenme yöntemleridir (Kiran vd., 2018). Derin öğrenme algoritması geliştirmek için öncelikle probleme özel veri seti oluşturmak gerekmektedir. Literatürde koşma, kavga, kırmızı ışıktaki geçme gibi anormal durumlarla ilgili veri setleri bulunmaktadır. Ancak genel olarak suç tespiti ile ilgili işlenmiş veri seti bulunmamaktadır. Araştırma sürecinde ilk aşamada suç veri seti oluşturulmuştur. Bu veri setinde suçu işaret eden farklı anormal durum verileri bulunmaktadır. Veri seti ile literatürde videodan anormal durum tespitinde başarılı olan bazı derin öğrenme modelleri eğitilerek, bu yöntemler birbiriyle karşılaştırılmış ve değerlendirilmiştir. Bu çalışma ile pratikte şiddet içeren suçların tespit edilebileceği gösterilerek yeni bir veri seti ve yöntemi literatüre kazandırılmıştır.

Video anomali, videoda bulunmaması gereken bir nesne, davranış veya olayı ifade edebilir. Anomalinin genel bir belirtisi yoktur çünkü konuya özeldir. Bir konu kapsamında normal olarak tanımlanan süreç dışında kalan an veya durumlar anormal olarak ifade edilir (Zhu vd., 2012). Dolayısıyla sürecin tamamına göre sadece çok küçük bir zaman bölümü anomali içermektedir. En sık görülen video anomalileri şunlardır:

1. **Sadece görünüm anomalileri**, bir sahnede alışılmadık nesne ortaya çıkışı olarak düşünülebilir. Bir yaya geçidindeki bisikletçiler veya bir yolda duran büyük bir kaya

- örnek olarak verilebilir. Bu anormallikleri tespit etmek için yalnızca tek bir video çerçevesinin (frame) kısmi bir bölgesinin incelenmesi yeterlidir.
2. **Sadece kısa süreli hareket anomalileri**, bir sahnede olağandışı nesne hareketi olarak düşünülebilir. Örnek olarak, yabancı elçiliklerin çevresinde rastgele dolaşan bir kişi gösterilebilir. Bu anormallikleri tespit etmek genellikle videonun kısmi bir bölgesinin kısa bir süre boyunca incelenmesini gerektirir. Bu ek özelliğe sahip oldukları için, sadece kısa süreli hareket anomalilere lokal anomaliler de denilebilir.
 3. **Uzun süreli yörünge anomalileri**, bir sahnede olağandışı nesne yörüngesi olarak düşünülebilir. Örnek olarak, kaldırımda zikzak şeklinde yürüyen kişiler gösterilebilir. Yörünge anormalliklerini tespit etmek, daha uzun video bölümlerinin incelenmesini gerektirir.
 4. **Grup anomalileri**, bir sahnede olağandışı nesne etkileşimi olarak düşünülebilir. Örnek olarak, yürüyen bantta yürüyen bir grup kişi gösterilebilir. Grup anormalliklerinin saptanması, videonun iki veya daha fazla bölgesi arasındaki ilişkinin analiz edilmesini gerektirir.
 5. **Gün saati anomalileri** daha önce tartışılan konuma bağlı anormalliklere çok benzerdir, ilgili çerçeve mekansal yerine zamansaldır. Bu faaliyetleri anormal yapan şey, gerçekleştikleri zamandır. Örnek olarak, sabahın erken saatlerinde insanların sinemaya gitmesi verilebilir. Genellikle, bu anormallikleri tespit etmek, günün farklı zamanları için farklı bir normallik modeli kullanmayı gerektirir.

2.5.3. Derin öğrenme ile videoda anomali tespiti

Video görüntüsünden alınan çerçeve (frame) serisini analiz ederek çıkarılan örüntülerden insan hareketini sınıflama sürecine insan davranışı tanıma (Human Action Recognition, HAR) denir. Birçok alanda olduğu gibi HAR'da da derin öğrenme sayesinde büyük gelişmeler olmuştur, çünkü derin öğrenme modeli videodan davranış tanıma için gerekli bilgileri ön işleme (pre-processing) yapmadan kademe kademe öğrenilebilmektedir (Poppe, 2010; Wu vd., 2017). HAR bir sınıflandırma (classification) problemidir. Bir davranış olduğu bilinir ve önceden belirlenen kategorilerden (sınıf) birine ait olduğu varsayılır. Sonrasında model her sınıfın olma olasılığını hesaplar ve en olası sınıfa ait davranışın gerçekleştiği sonucuna ulaşır. HAR ile ilgili Kinetics (Kay vd., 2017), Moments in Time (Monfort vd., 2019) ve Youtube-8M (Karthickkumar ve Kumar, 2020) gibi yüzlerce farklı davranış sınıfı ve milyonlarca video içeren veri setleri bulunmakta ve sürekli

geliştirilmektedir. Bu sayede başarılı algoritmalar geliştirilmektedir (Koohzadi ve Chakari, 2017; Abu-El-Haija vd., 2020).

Video görüntülerinde yer alan anormal davranış içeren çerçevelerin tespiti sürecine anormal olay algılama (Abnormal Event Detection, AED) denir. AED bir sınıflandırma problemi değildir, tespit (detection) problemidir. Anomali tanımı gereği önceden belirlenen kategoriler dışında kalan durumlardan oluşur, dolayısıyla anomali önceden öğrenilerek daha sonra tanınmaz. Önceden normal olaylar öğrenilir, sonrasında bunların dışında kalan bir durum olursa anomali denilerek tespit edilmiş olur (Kiran vd., 2018). Yapay zekanın dolayısıyla derin öğrenmenin en büyük zorluklarından biri, birden çok problemi çözen genel bir algoritma geliştirilemeyeceğidir (Wolpert ve Macready, 1997). Bu yüzden her problem için ayrı bir veri seti ve o veriye özel model geliştirilmelidir. Özellikle anomali tanımının konudan konuya, ortamdaki ortama değişmesi onu diğer video problemlerine göre daha zorlayıcı yapmaktadır. Literatürde çeşitli AED konuları bulunmaktadır. Bunlardan bazıları şunlardır:

- Kırmızı ışık ihlali tespiti
- Trafik kazası
- Düşme
- Koşma
- Cisim fırlatma, vurma
- Panik
- Hastalık
- Park ihlali
- Yangın, duman
- Metroda bilet basmama
- Raftan nesne alma
- Sahipsiz nesne tespiti

3. LİTERATÜR TARAMASI

Video görüntülerinde anormal davranış içeren çerçevelerin (frame) tespitine anormal insan davranışı algılama (AİDA) denir. AİDA hem insan davranışı hem de anomali içerdiğinden HAR ve AED alanlarının kesişimi olarak düşünülebilir. Dolayısıyla AİDA problemi, sadece insan davranışı içeren veri setleri ile anomali tespiti diye düşünülebilir. Derin öğrenmede model veri ile eğitilir. Suç algılama AİDA'nın özel bir uygulaması olarak değerlendirilebilir. Literatürde suç tespiti ile ilgili az sayıda ve kısıtlı çalışma bulunmaktadır. Bu yüzden AİDA ile ilgili literatür taraması yapılmıştır. Bunun sebebi suç ile ilgili veri seti oluşturulduğunda, genel olarak AİDA ile ilgili modellerin suç veri seti ile eğitildiğinde başarılı sonuç vermelerinin beklenmesidir.

Anomali tespiti probleminde verinin normal ya da anormal olduğu kararı verilir. Girdinin iki kategori arasından hangisine ait olduğuna bakılır. Sınıflandırma probleminde girdinin önceden belirlenen kategorilerden hangisine ait olduğu anlaşılmaya çalışılır. Bu açıdan anomali tespiti ikili sınıflandırma problemi gibi görünebilir. Ancak, anomali tespitinin sınıflandırma problemlerinden çok önemli bir farkı vardır: Girdinin normal olma sıklığı, anormal olma sıklığından kat kat fazladır (Dhiman ve Vishwakarma, 2019; Zhiguo vd., 2020). Modelin doğru eğitilmesi ve performansının doğru değerlendirilmesi için veri seti de bu dağılımı yansıtmalıdır. Bu durum model geliştirmek için sınıflandırma problemini yani, gözetimli öğrenmeyi genel olarak iyi bir yöntem olmaktan çıkarmaktadır.

Anomali tespiti için model normal verilerle gözetimsiz öğrenme ile eğitilir. Test aşamasında beklenenin dışında bir çıktı elde edilirse, girdinin anormal olduğu sonucuna karar verilir (Dhiman ve Vishwakarma, 2019; Zhiguo vd., 2020). Suç tespiti modeli tasarlanırken faydalanılabilecek, insan davranışı içeren AED veri setleri ve bu veri setinde kullanılan yöntemler detaylı şekilde araştırılmış ve açıklanmıştır. Ayrıca bu konuların dışında kalmasına rağmen literatürde suç tespiti ile ilgili bir araştırma konusu bulunmaktadır: Kavga tespiti (Fight, Violence Detection). Aslında kavga tespiti de bir anomali tespiti problemidir ve literatürde bu şekilde geçmektedir ama literatürdeki tüm veri setlerinde normal ve kavga veri sayıları eşittir. Bu durum problemin doğasını yansıtmamaktadır. Bu yüzden kavga tespiti yapan tüm çalışmalarda veri setinden en iyi başarıyı elde etmek için sınıflandırma ile gözetimli eğitim yapılmıştır. Fakat, geliştirilen yöntemler gerçek koşulları yansıtan veri seti üzerinde analiz edilmediğinden performans analizi de pratikteki

performansını yansıtmamaktadır. Bu yüzden kavga tespiti veri setleri ve bu konuda kullanılan modeller ayrı bir başlık altında incelenmiştir.

3.1. Veri Setleri

3.1.1. Anormal insan davranışı ile ilgili veri setleri

Video görüntülerinden anomali tespiti ile ilgili çeşitli veri setleri bulunmaktadır. Bu veri setleri bir veya birden çok anormal durum içermektedir. Kırmızı ışık ihlali tespiti, trafik kazası tespiti, park ihlali tespiti gibi konularla ilgili veri setleri insan davranışı içermediği için; düşme tespiti gibi konularla ilgili anomali tespiti veri setleri konu ile ilgili olmadığından araştırmaya dahil edilmemiştir. AED çalışmalarının en sınırlayıcı özellikleri, çalışmalarda kullanılan veri setlerinin oldukça az veriden oluşmasıdır (Al-Dhamari vd., 2017). AED modellerinin pratikte kullanılabilmesi için HAR’da olduğu gibi büyük veri setiyle başarılı şekilde eğitilmiş DNN’lere ihtiyaç duyulmaktadır. Başlıca AIDA veri setleri şöyledir:

UMN (Mehran vd., 2009) veri seti, 3 farklı iç ve dış mekân sahnesinde bir kaçış olayının 11 farklı senaryosunun, oyuncular tarafından canlandırılmış videolarını içerir. Kliplerin her birinde anomali aniden kaçan ve bir tahliye senaryosuna işaret eden insanlardan oluşur. Her video normal davranışın başlangıç kısmından oluşur ve anormal davranış dizileriyle sona erer. Veri kümesindeki tüm videolar kaçış paniği davranışı sergiler.

Web (Mehran vd., 2009) veri seti, insanların normal yürümelerini içeren 8 video ve kaçma, kavga, çatışma gibi anormal davranışlarını içeren 12 video görüntülerinden oluşur. Tüm çerçeveler (frame) 480 çerçeve genişliğe göre yeniden boyutlandırılmıştır.

CUHK Avenue (Lu vd., 2013) veri seti, önünde bir yaya geçidi olan bir binanın kenarına bakan tek bir açık hava gözetleme kamerasından alınan kısa video kliplerden oluşur. İnsanların normal yürümelerini veya koşma, cisim fırlatma gibi anormal davranışlarını içeren kalabalık görüntülerinden oluşan 37 video içerir. Veri kümesi nesneleri atma, dolaşma ve koşma dahil toplam 14 farklı anormal olay içeren 16 eğitim videosu ve 21 test videosu içerir. Kameranın konumu ve açısı nedeniyle insanların boyutu değişebilir.

UCSD (Chan vd., 2013) veri seti, yaya yürüyüş yollarına bakan bir yüksekliğin monte edilmiş sabit bir kamera ile elde edilmiştir. Anormal olaylar şunlardan kaynaklanır: Yaya olmayan varlıkların yaya geçitlerinde dolaşımı ve anormal yaya hareket kalıpları. Yaygın olarak ortaya çıkan anomaliler bisikletçileri, patencileri, küçük arabaları ve bir yürüyüş

yolunda veya onu çevreleyen çimenlerde yürüyen insanları içerir. Tekerlekli sandalyedeki insanların birkaç örneği de kaydedilmiştir. Tüm anormallikler doğal olarak meydana gelir, yani veri kümesinin montajı amacıyla sahnelenmemişlerdir. Veriler, her biri farklı bir sahneye karşılık gelen iki alt kümeye bölünmüştür:

- **Ped1** veri seti, kameraya doğru gelen ve kameradan uzaklaşan insan gruplarının görüntülerinden oluşur. 34 eğitim videosu örneği ve 36 test video örneği içerir.
- **Ped2** veri seti, kamera düzlemine paralel yaya hareketine sahip sahnelerden oluşur. 16 eğitim videosu örneği ve 12 test video örneği içerir.

Subway (Adam vd., 2008) veri seti, metro istasyonundan geçen insan görüntülerinden oluşan 2 saat uzunluğunda bir videodur. İki kategori vardır: Giriş ve Çıkış. Anormal olaylar, yolun yanlış tarafında yürüme, etrafta dolaşma, bilet basmadan geçme gibi 19 farklı anomaliyi içerir. Bu videolardan veri kümesinin hangi çerçeve hızında çıkarılması gerektiği ve tam olarak hangi çerçevelerin anormal olarak etiketlendiği ve eğitim ve test için hangi çerçevelerin kullanılacağı açık değildir. Ayrıca, uzamsal zemin gerçekliği sağlanamamıştır. 66 anormal olay geçici olarak etiketlenmiştir. Bu veri kümeleri artık oldukça eskidir ve mekansal açıklamaların belirsizlikleri ve eksikliği ile birlikte, bunları herhangi bir resmi kapasitede bir anomali tespit yöntemini değerlendirmek için kullanılması önerilmemektedir.

ShanghaiTech (Luo vd., 2017) veri setinde, bir üniversitenin kaldırımı boyunca yürüyen insanlar bulunur. Normal olan 330 video ve anormal durum içeren 107 kalabalık görüntüleri videolarından oluşur. Veri kümesinin karmaşık ışık koşulları ve kamera açılarıyla 13 sahnesi vardır. 130 anormal olay ve 270.000'den fazla eğitim çerçevesi içerir. Ayrıca, veri setinde anormal olayların piksel düzeyinde yer gerçeği de açıklanmaktadır.

BOSS (Velasin ve Gómez-Lira, 2017) veri seti, oyuncular tarafından canlandırılmış yürüme, bayılma, kavga ve panik gibi anormal durumlar içeren 720x576 çözünürlükte 16 video görüntüsünden oluşur. "BOSS: Yerleşik Kablosuz Güvenli Video Gözetimi BOSS" projesinin bir parçası olarak ele alınmıştır. Hareket eden bir trenin içindeki eylemlerin video ve ses kayıtlarından oluşur. Birden fazla kalibre edilmiş kamera kullanılmıştır.

Veri Seti	Gerçek Rol	Başlıca Anomaliler	Farklı Anomali Sayısı	Anomali Sayısı	Video Sayısı	Yılı
UMN	Rol	Koşarak kaçış	1	33	33	2009
Web	Gerçek	Panik Kavga Çatışma	3	8	20	2009
CUHK Avenue	Gerçek	Koşma Cisim Fırlatma	14	47	37	2013
UCSD Ped1	Gerçek	Anormal Yaya Hareketi Taşıt Geçmesi	2	40	70	2013
UCSD Ped2	Gerçek	Anormal Yaya Hareketi Taşıt Geçmesi	2	12	28	2013
Subway Entrance	Gerçek	Bilet Basmadan Geçme Yanlış Yöne Gitme	19	66	1	2008
Subway Exit	Gerçek	Bilet Basmadan Geçme Yanlış Yöne Gitme	19	19	1	2008
ShanghaiTech	Gerçek	Normal Yürüme Dışındaki Durumlar	—	130	437	2017
BOSS	Rol	Yürüme Bayılma Kavga Panik	6	12	16	2017

Tablo 2: Başlıca Anormal İnsan Davranışı Algılama Veri Setleri

Tablo 2’de AİDA veri setleri ile ilgili bilgi verilmektedir. UMN, Web, CUHK, UCSD, Subway ve ShanghaiTech veri setlerinde normal olay insanların yürümesidir. BOSS veri setinde ise insanların tren içinde oturmaları normal davranışlarıdır. Veri setlerindeki anormal durumlar ise kavga, koşma, taşıt geçmesi gibi durumlardır. Normal insan davranışlarının her videoda aynı olması problemi kolaylaştırmaktadır. Örneğin nesne uzatma, raftan ürün alma gibi günlük faaliyetler bulunmamaktadır. Veri setlerinin tamamında anomaliler bir başlık altında toplanmıştır. Tabloda belirtilen farklı anomali sayıları aynı anomalinin farklı gerçekleşme senaryosu sayısını belirtmektedir. Örneğin yaya yoluna bisiklet ve araba

girmesi farklı iki anomali olarak gösterilmektedir. Kapalı ortamda şiddet içeren suç tespiti için oluşturulacak veri setinde yürüme, durma, uzanma, nesne uzatma gibi birbirinden tamamen farklı çok sayıda normal davranış; koşma, kavga, silah doğrultma, yerde yatma gibi birbirinden tamamen farklı çok sayıda anomali olması gerekmektedir. Literatürdeki veri setleriyle kıyaslandığında problemin literatürdeki AİDA problemlerine kıyasla ne kadar zor olduğu daha net ortaya çıkmaktadır.

3.1.2. Kavga tespiti ile ilgili veri setleri

Literatürde kavga tespiti ile ilgili çeşitli veri setleri bulunmaktadır. Son yıllarda gerçek olaylardan alınmış çok sayıda videodan oluşan veri setleri yayınlanmıştır. Bu veri setleri eşit sayıda normal ve anomali videosu içermektedir. Bu yüzden geliştirilen modeller kavga tespitini gözetimli eğitime ile ele almaktadır. İkili sınıflandırma yapan ve veri seti üzerinde iyi performans gösteren modeller geliştirilmektedir. Lakin gerçekte normal veri sayısı oranı çok daha fazla olduğundan modelin pratikteki muhtemel performansı daha azdır. Literatürde bu durumu test eden bir çalışma bulunmamaktadır. Başlıca kavga algılama veri setleri şöyledir:

Action Movies (Nievas vd., 2011) veri seti, normal olan 100 video ve kavga içeren 100 videoluk film görüntülerinden oluşur. Videoların ortalama süresi 2 saniyedir.

Hockey Fight (Nievas vd., 2011) veri seti, Ulusal Hokey Ligi'nin (NHL) hokey oyunlarından 1000 aksiyon klip toplanmıştır. Normal davranış içeren 500 video ve kavga içeren 500 videoluk 1000 hokey oyunu görüntülerinden oluşur. Her klip, 720×576 çözünürlükte 50 çerçeveden oluşur. Videoların ortalama süresi 1 saniyedir.

Violent Flows (Hassner vd., 2012) veri seti, normal olan 123 video ve kavga içeren kalabalık görüntülerinden oluşan 123 video içerir. Tüm videolar YouTube'dan indirilmiştir. En kısa klip süresi 1.04 saniyedir, en uzun klip 6.52 saniyedir ve bir video klabin ortalama uzunluğu 3.60 saniyedir.

UCF-CRIME (Sultani vd., 2018) veri seti, 128 saatlik video verisi ile türünün ilk büyük ölçekli veri kümesidir. Kötüye kullanım, tutuklama, kundakçılık, saldırı, trafik kazası, hırsızlık, patlama, dövüş, soygun, atış, hırsızlık ve vandalizm dahil olmak üzere 13 gerçekçi suç içeren 1900 gerçek dünya videosundan oluşur. Bu veri kümesi iki görev için kullanılabilir. İlk olarak, bir gruptaki tüm anomalileri ve başka bir gruptaki tüm normal aktiviteleri dikkate alarak genel anomali tespiti için; ikincisi ise, 13 anormal aktivitenin her birini tanımak için.

Bu veri setindeki videolar hem suç içeren hem suç içermeyen çerçevelerden oluşabilmektedir. Video uzunlukları genelde 1 dakikadan uzundur. Videoların çerçeveleri değil, görüntünün tamamı etiketlenmiştir. Dolayısıyla bu veri seti analizi daha çok videodan konu tanıma veri seti olarak değerlendirilebilir. Veri setindeki çerçeveler etiketlenmediğinden, veri seti anormal çerçeve tespiti için kullanılamaz. Ayrıca, bir video birden çok davranış içerdiği için, suç anı tespiti için uygun bir veri seti değildir.

The LV Veri Seti (Leyva vd., 2017), toplam 3.93 saat uzunluğunda olan kavga, çatışma, yankesicilik, hırsızlık, araba kazaları, kapkaç, yangın, panik, vandalizm, adam kaçırmaya, cinayet, yanlış yöndeki arabalar, düşen insanlar, yasaklanmış U dönüşleri olmak üzere 14 farklı anormal olay içeren 30 saniyeden oluşur. Video çözünürlükleri 176x144 ile 1280x720 arasında değişiklik gösterir.

Real Life Violence Situations (Soliman vd., 2019) veri seti, Youtube videolarından toplanan 1000 şiddet içeren ve 1000 şiddet içermeyen video içerir. Veri setindeki şiddet videoları çeşitli ortam ve koşullarda birçok gerçek sokak dövüşü durumu içerir, şiddet içermeyen videolar ise spor, yemek yeme, yürüyüş gibi eylemlerden oluşur.

Surveillance Camera Fight (Aktı vd., 2019) veri seti, farklı türlerde kavgalar içeren CCTV videoları ve CCTV olmayan diğer kaynaklardan (mobil kameralar, araç ön kameraları, drone ve helikopter) toplanan gerçek kavga videoları içerir. Veri setinde 150 normal, 150 kavga içeren ortalama süresi 2 saniye olan 300 video bulunmaktadır.

RWF-2000 (Cheng vd., 2019) veri seti, Gerçek Dünya Dövüşü (Real-World Fighting-RWF) veri setinde ve YouTube'da yer alan ve gerçek dünyada güvenlik kameraları tarafından çekilen 2.000 videodan oluşur. Her video 5 saniye sürmektedir. Videoların yarısı şiddet içeren davranışlar içerirken, diğerleri şiddet içermeyen eylemlere aittir. Veri setindeki tüm videolar gerçek dünyadaki güvenlik kameraları tarafından yakalanmıştır.

Tablo 3'te literatürdeki başlıca suç tespiti veri setleri sıralanmaktadır. Veri setlerinin anomalilerinin tamamı veya çoğunu kavga oluşturmaktadır. Özellikle Real Life Violence Situations ve RWF-2000 veri setleri örneğinde olduğu gibi CCTV görüntülerinden oluşan büyük veri seti bulunmaktadır. Fakat bu veri setleri, içeriğinde yer alan sınıflardaki veri sayılarının eşit olması nedeniyle, problemin doğası gereği anomali tespiti değil sınıflandırma için kullanılmaktadır.

Veri Seti	Gerçek Rol	Başlıca Anomaliler	Farklı Anomali Sayısı	Anomali Sayısı	Video Sayısı	Yılı
Action Movies	Gerçek	Kavga	1	100	200	2011
Hockey Fight	Gerçek	Kavga	1	500	1000	2011
Violent Flows	Gerçek	Kavga	1	123	246	2012
UCF-CRIME	Gerçek	Hırsızlık Vandalizm Patlama Eve Hırsız Girmesi	13	—	1900	2018
The LV	Gerçek	Kapkaç Yangın Panik Trafik Kazası Cinayet	14	34	30	2017
Real Life Violence Situations	Gerçek	Şiddet	1	1000	2000	2019
Surveillance Camera Fight	Gerçek	Kavga	1	150	300	2019
RWF-2000	Gerçek	Kavga	1	1000	2000	2019

Tablo 3: Başlıca Kavga Algılama Veri Setleri

3.2. Derin Öğrenme Modelleri

3.2.1. Anomali tespiti

Gerçek hayatta bir süreçte anormal durumlar nadiren gerçekleşir. Bu yüzden bir konuda veri toplandığında normal durumla ilgili veri, anormal durum verilerine göre kat kat fazladır (Fernandes vd., 2019). Bu konuya bilgisayara virüs bulaşması örnek verilebilir. Bir insan, gün içinde kişisel bilgisayarıyla internette gezinirken gigabaytlarca veri alışverişi gerçekleşir ama bilgisayara nadiren virüs bulaşır. Virüsler çok çeşitlidir ve hepsinin etkisi

farklıdır. Ayrıca eskileri tespit edildikçe korsanlar tarafından sürekli yenisi geliştirilip kullanılır. Bu çeşitlilikten dolayı anormal durumları öğrenmek zordur. Dolayısıyla anormal durum tespiti ederken ne olduğu belirlenmez. Normal süreç öğrenilir ve bir anda öğrenilene aykırı bir durum belirlenirse anormal bir durum oldu diye karar verilir.

Gelen bir verinin normal veya anormal olduğunu tespit etme, iki durum arasında doğruyu anlama problemi olarak düşünülebilir. O zaman problem iki sınıf (normal, anormal) arasında sınıflandırma problemine dönüşür (Chalapathy ve Chawla, 2019). Yani gözetimli derin öğrenme yöntemini kullanmak akla gelebilir. Sınıflandırma problemlerinde sınıflar arasındaki sıklık oranı, derin öğrenme modelini eğitirken kullanılan sınıf verileri oranına da yansımalıdır ki model verilerden doğru çıkarım yapabilsin. Bir sınıfın olasılığı diğer sınıfın olasılığının 2 katı ise eğitimde kullanılan veri de 2 kat olduğunda iyi sınıflandırma yapılabilir. Anormal durum problemlerinde, anomali nadiren gerçekleşir. Örnek olarak, çok iyimser yaklaşarak normal durum sıklığı anormal durum sıklığının 999 katı diyelim. 9990 normal, 10 anormal durum verisi ile sınıflandırma yaptık. Derin öğrenme sırasında modelin amacı minimum hata elde etmektir. Model her veriye normal diyerek hata fonksiyonunu neredeyse sıfıra indirebilir. O zaman yüzde 99.9 başarı ile sınıflandırma yapılır ama tüm anormal durumlar yanlış sınıflanır. Bu yüzden, sınıflandırma modeli anormal durum tespiti için kullanılamaz. Bu durum, hata fonksiyonu tanımlanırken anormal durumu yanlış tespit etmeyi, normal durumdan çok daha fazla (örneğin 100, 1000 kat) cezalandırarak düzeltiler. Bu durumda anormal durumlar tespit edilir ama doğruluk yüzdesi düşer. Dolayısıyla yanlış pozitif sıklığı çok olur. Bu da anormal durum tespiti modelinden istenmeyen bir durumdur çünkü o durumda genel olarak normal işleyen sürecin sürekli kontrol edilmesi gerekir. Bu sebeplerden ötürü, anormal durum tespitinde gözetimli öğrenme modelleri çok tercih edilmez (Chalapathy ve Chawla, 2019).

Anormal durum tespitinde genellikle gözetimsiz öğrenme yöntemleri tercih edilir. Gözetimsiz öğrenme yöntemlerinde etiketleme yapılmaz. Derin öğrenme modelinin veriden istenilen çıkarımı kendiliğinden yapması beklenir. Gözetimsiz öğrenme ile anomali tespitinde genel yaklaşım şu şekilde açıklanabilir:

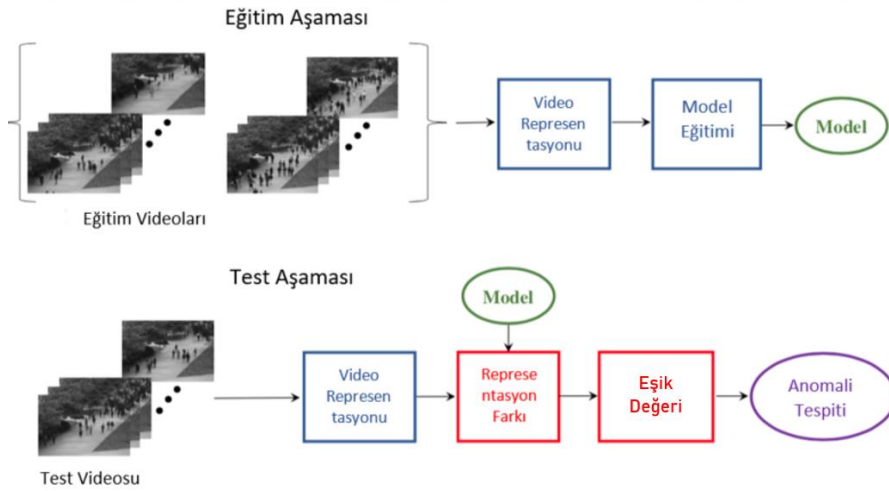
1. Eğitime için sadece normal veri kullanılır. Normal verinin bir kısmı (örneğin yüzde 20'si) ve anormal verinin tamamı test için ayrılır.
2. Derin öğrenme modeline normal verilerin ayrı ayrı veya kolektif temsili öğretilir.
3. Test sırasında bir veri modele girdi olarak verilir ve temsili elde edilir.
4. Sonuç eğitime sırasında normal verilerden elde edilen temsillere (reprezentasyonlara) benzer ise veya beklenen temsile yakın ise normal olarak çıkarım yapılır. Aksi

takdirde anormal durum çıkarımı yapılır.

Derin öğrenme yöntemleri gürültüye ve verideki küçük değişikliklere karşı dayanıklıdır. Dolayısıyla gözetimsiz öğrenme modeline normal veri verildiğinde büyük ihtimal beklenen sonuç çıkar. Lakin, modelin öğrenmediği bir veri girildiğinde, girdinin temsiline beklenenden farkının çok olması beklenir. Bu yüzden, konu ile ilgili normal durumların veri setinde yeterli çeşitlilikte ifade edilmesi ve modelin doğru eğitilmesi sağlandığında bu yöntem genel olarak iyi sonuçlar verir (Chalapathy ve Chawla, 2019; Fernandes vd., 2019). Bu yüzden anomali tespitinde gözetimsiz öğrenme yaygındır.

3.2.2. Video görüntülerinden anormal insan davranışı tespiti

Videolardan görüntü işleme yöntemleri ile öznitelik çıkarıp, çıkarılan öznitelikleri makine öğrenmesi yöntemi ile analiz ederek anormal davranış tespiti yapan çalışmalar bulunmaktadır (Kalaivani vd., 2017). Bu klasik makine öğrenmesi modelleri yerine derin öğrenme yöntemlerini kullanmanın daha iyi sonuçlar verdiği görülmektedir (Mabrouk ve Zagrouba, 2018). Bunun sebebi derin öğrenme modelinin işlenmemiş görüntüden çok katmanlı yapısı sayesinde öznitelik ve gerekli çıkarımları daha iyi yapabilmesidir.



Şekil 36: Video Görüntülerinden Gözetimsiz Öğrenme ile Anomali Tespiti Yöntemi

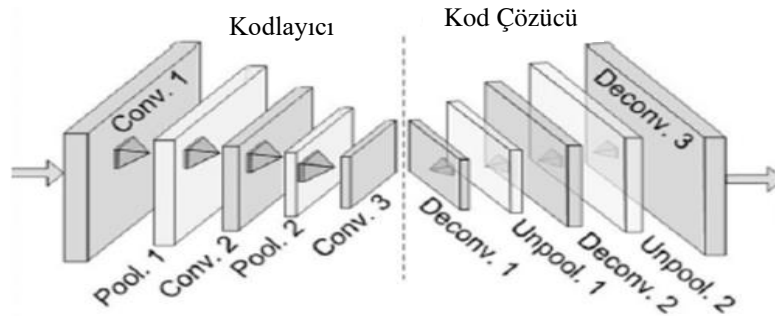
(Ramachandra, B. vd, 2020)

Bir önceki aşamada anlatıldığı gibi, gözetimsiz öğrenme ile anormal durum tespiti yapılabilir. Aynı yöntem video verileri için de uygulanabilir. Bu sürecin detaylı hali Şekil 36'da gösterilmektedir. Eğitim aşamasında, normal video görüntülerinden normal durum temsili çıkarılır. Bu temsil, 1 veya 2 boyutlu matris olarak düşünülebilir. Test aşamasında,

videoların temsili çıktı olarak alınır. Beklenen temsile uzaklığı, önceden belirlenen eşik değerinden yakın ise normal, uzak ise anormal durum kararı verilir. Video anomali tespiti için kullanılan başlıca gözetimsiz öğrenme modelleri sonraki bölümlerde anlatılmaktadır.

3.2.2.1. Konvolüsyonel özkodlayıcı (CAE)

Anormal durum tespiti için özkodlayıcılar (Autoencoders, AE) başarılı şekilde kullanılmaktadır (An ve Cho, 2015). Bu yöntemde özkodlayıcıya etiket olarak girdi verilerek gözetimsiz öğrenme yapılmaktadır. Test aşamasında bir girdi modele verilerek tekrar oluşturması sağlanmaktadır. İdeal olarak çıktının girdi ile aynı olması beklenir. Normal bir durumda çıktı ile girdi arasındaki fark azdır. Anormal bir durumda ise özkodlayıcı eğitilirken benzer bir durumu öğrenmediğinden, verilen girdinin model çıkışında doğru üretilmemesi beklenir. Dolayısıyla karar verirken girdi ile çıktı arasındaki fark hesaplanır. Bu fark önceden belirlenen eşik değerinin altında ise normal, üstünde ise anormal durum olduğuna karar verilir. Video, görüntü dizisi olarak tanımlanabilir. Dolayısıyla görüntü analizinde ilk akla gelen yöntem olan evrişimsel yapay sinir ağı (CNN) içeren modeller, video analizinde de sıkça ve başarılı olarak kullanılır. CNN'nin görüntülerdeki başarısı ile özkodlayıcıların anomali tespitindeki başarısını birleştiren bir çalışmada evrişimsel özkodlayıcı (Convolutional Autoencoder, CAE) modeli kullanılarak anormal davranış tespitinde başarılı sonuçlar elde edilmektedir (Ribeiro vd., 2018). Girdi olarak bir çerçeve görüntüsü alan model, çıktıda aynı çerçeveyi oluşturmaya çalışır. Model normal özkodlayıcılar gibi simetrik olarak önce daralıp sonra genişler, normal özkodlayıcılardan farklı olarak ise katmanlarda CNN katmanları kullanılmaktadır (Sewak vd., 2018).



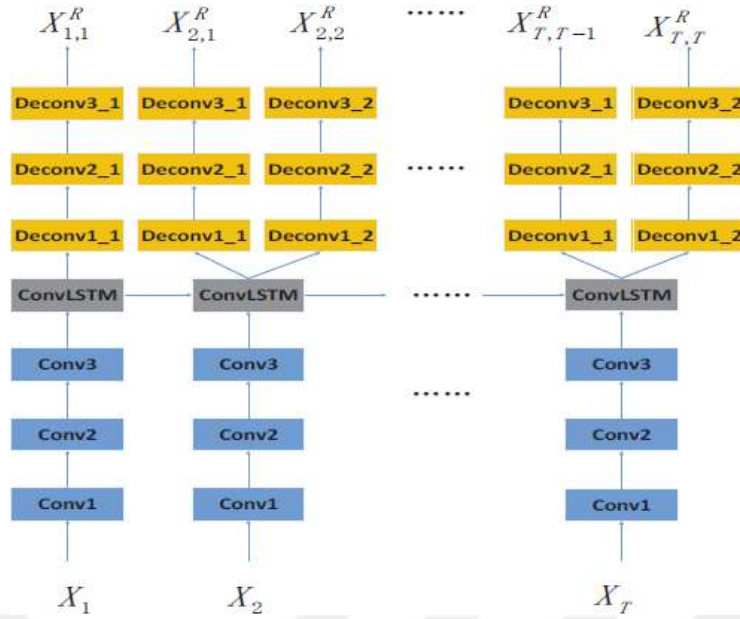
Şekil 37: CAE Modeli (Ribeiro vd., 2018)

3.2.2.2. RCNN-AE

Yinelemeli yapay sinir ağıları serilerde zaman ilişkisini çıkararak genelde diğer modellere göre daha başarılı sonuçlar verirler. Dolayısıyla, AE’de hücre olarak LSTM kullanıldığında (LSTM-AE) serilerde anomali tespiti başarılı şekilde yapılabilir. Örnek olarak ortamdaki normal sesleri öğrenen LSTM-AE modeli ile ortamdaki anormal sesler başarılı şekilde tespit edilmektedir (Marchi vd., 2015). Çünkü belirli bir zaman dilimindeki ses aslında bir vektördür. Bir ortamdaki ses, zaman dilimlerine bölündüğünde zamanla değişen bir vektör serisi elde edilmiş olur. Video ise 2 boyutlu matris serisidir. Zamansal ilişkiyi analiz etmek için RNN hibriti modeller başarılı şekilde kullanılmaktadır. LSTM-AE ile videodan anormal durum tespit etmede yaklaşık olarak CAE ile aynı başarı yakalanmıştır (Luo vd., 2017).

CNN’nin özelliği görüntü analizini, RNN’nin özelliği ise zamansal analizi iyi yapabilmektir. Bu yüzden CNN ve LSTM içeren hibrit modeller video görüntülerinden anomali tespitinde çok sık ve başarılı şekilde kullanılmaktadır (Liu vd., 2019). En basit CNN-LSTM hibriti, klasik CNN’nin sonundaki FCNN yapısında LSTM hücreleri kullanılarak elde edilebilir. Bu yapıya kısaca RCNN (Recurrent CNN) denilmektedir. Bir çalışmada bu şekilde her çerçeveden 6 elemanlı bir vektör elde edilmektedir. Belirli bir zaman aralığındaki çerçeveden elde edilen vektörler birleştirilerek o videonun temsili elde edilmiş olur. Sonrasında uzaklık karşılaştırması ile anomali kararı verilmektedir (Ko ve Sim., 2018).

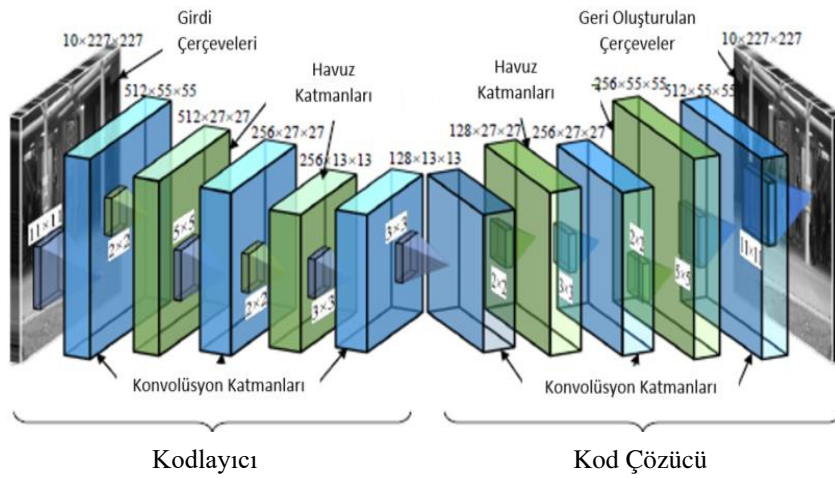
Video görüntülerinden anormal durum tespitinde, RCNN’nin videodaki başarısı ile AE’nin anormal durum tespiti başarısını birleştirerek (RCNN-AE) şu ana kadar bahsedilen modellerden daha başarılı sonuç elde edilmiştir (Luo vd., 2017). Model çıktıda etiket olarak girdi alan klasik simetrik AE yapısındadır. Dış katmanlarda konvolüsyonel katmanları bulunurken en içte LSTM katmanı bulunmaktadır. Bu şekilde gözetimsiz öğrenme ile anormal durum tespit edilmektedir.



Şekil 38: RCNN-AE Modelinin Zamansal Gösterimi (Luo vd., 2017)

3.2.2.3. ConvLSTM-AE

Video görüntülerinin çerçevelerini modele sırayla girdi olarak vermek yerine, bir zaman aralığındaki tüm çerçeveler bir adımda girdi olarak verilebilir. Zaman aralığında k tane çerçeve olsun diyelim. O halde k tane iki boyutlu matris birleştirilerek üç boyutlu (üçüncü boyut uzunluğu k olan) tek bir matris elde edilebilir. Bu şekilde elde edilen veriye de konvolüsyon uygulanabilir. Bu CNN modelinin katmanlarını iki boyut yerine üç boyutlu tasarlayarak sağlanır. Bu modele 3D CNN (C3D) denir. C3D modeli ile AE birleştirilerek (C3D-AE) videoda anormal durum tespiti yapılabilir (Hasan vd., 2016).

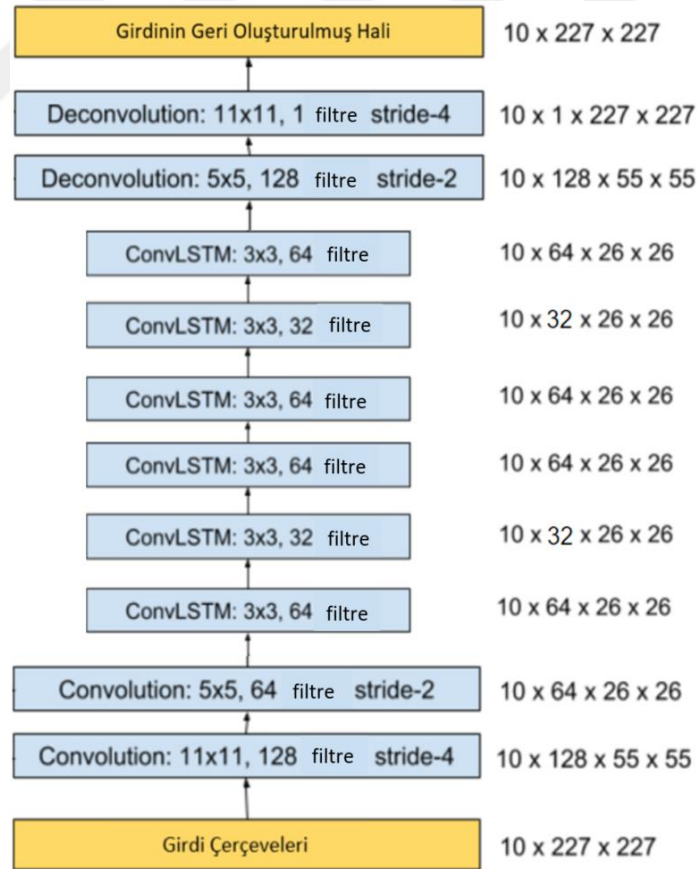


Şekil 39: C3D-AE Modeli (Hasan vd., 2016)

C3D-AE modeli RCNN'ye göre bir miktar daha iyi sonuçlar vermektedir. Bir modelde

çerçeveler girdi olarak sırayla verilirken diğerinde bir adımda verilmektedir. Ancak C3D zamansal ilişkiyi öğrenememektedir. 2. bölümde bahsedildiği üzere ConvLSTM modeli matrislerdeki zamansal ilişkiyi yakalamak için geliştirilmiştir. Bu yüzden video verilerinde büyük başarı ile kullanılmaktadır (Medel ve Savakis, 2016). Anormal video tespiti için ise ConvLSTM-AE başarı ile kullanılmaktadır.

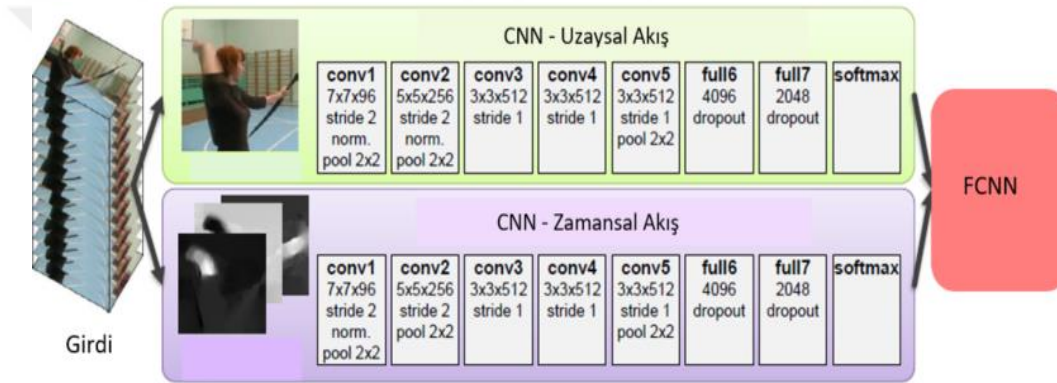
Şekil 40'ta gösterilen model ile şu ana kadar bahsedilen modellerden genel olarak daha başarılı anormal görüntü tespiti yapılmıştır (Chong ve Tay, 2017). Video analizinde kullanılan derin öğrenme modellerine girdi olarak çerçeve verilmek zorunda değildir. Videoda çerçeveler sürekli değişmektedir. Bazı uygulamalarda görüntünün içeriğinden çok görüntüdeki değişim merak edilebilir. Bu gibi durumlarda girdi olarak art arda gelen iki çerçeve arasındaki farkın verilmesi performansı artırabilir. Videoda anomali tespitinde de ConvLSTM-AE modeline girdi ve etiket olarak ardışık çerçeveler arasındaki fark verilerek modelin performansı artırılabilmiştir (Duman ve Erdem, 2019). Ancak bu yöntemin başarısı uygulamadan uygulamaya değişmektedir ve standart girdi verme şekli haline gelmemiştir.



Şekil 40: ConvLSTM-AE Modeli (Chong ve Tay, 2017)

3.2.2.4. Two-Stream CNN-AE

Videolar tek bir çerçeveye göre çok daha karmaşık verilerdir. Bir aralıktaki çerçeve serisinde hem nesneler bulunmaktadır hem de akış (flow) bulunmaktadır. Tek bir modelle tüm çıkarımları yapmak ne kadar derin olursa olsun mümkün olmamaktadır. Bu yüzden iki ayrı model eğitme fikri ortaya çıkmıştır. Bir model çerçevelerdeki uzaysal ilişkileri öğrenmek için eğitilirken, diğer modelin zamansal ilişkileri öğrenmesi için eğitilmesi amaçlanmıştır. Sonrasında bu iki koldan gelen bilgiler birleşiminden sonuç çıkaran üçüncü bir model kullanılmaktadır. Bu hibrit modele Two-Stream CNN denilmektedir (Simonyan ve Zisserman, 2014). İlk olarak videolardan davranış tanıma için geliştirilen model Şekil 41’de detaylı olarak gösterilmektedir.



Şekil 41: Two-Stream CNN Modeli (Simonyan ve Zisserman, 2014)

Two-Stream CNN ve AE hibrit modeli (Two-Stream CNN-AE) ile videodan başarılı olarak anomali tespiti yapılmaktadır (Yan vd., 2018). Yalnız, beklenenin aksine performansı ConvLSTM-AE'nin performansı ile yaklaşık aynıdır. Bu nedenle daha basit yapısı ile ConvLSTM-AE anormal durum tespiti için daha çok tercih edilmektedir.

Tablo 4'te şu ana kadarki algoritmaların performanslarının AUC değerinde göre karşılaştırmaları verilmektedir. En iyi performans gösteren yöntem girdi olarak çerçeve farkını alıyor. Bu yöntem dışındaki yöntemler arasında öne çıkan yoktur. Anormal durum tespitinde modellerin veri setlerindeki performansının çok değişken olabildiği görülüyor. ConvLSTM-AE ve Two-Stream CNN-AE modelleri arasında net şekilde görüldüğü gibi, modelin biri bir veri setinde daha iyi çalışırken, diğeri başka veri setinde daha iyi sonuç verebiliyor. Genel olarak ConvLSTM-AE türevi modeller iyi sonuç vermektedir. Ancak henüz belirli bir standarda ulaşılmadığı görülmektedir.

Makale	Yöntem	UCSD Ped1	UCSD Ped2	CUHK Avenue	Subway Entrance	Subway Exit
Luo vd. 2017	LSTM-AE	—	0.8362	0.7533	—	—
Ribeiro vd. 2018	CAE	0.895	0.847	0.754	—	—
Luo 2017	RCNN	0.755	0.881	0.770	0.933	0.877
Hasan vd. 2016	C3N-AE	0.81	0.900	0.745	0.943	0.807
Chong ve Tay 2017	ConvLSTM-AE	0.899	0.874	0.80	0.847	0.940
Yan vd. 2018	Two-Stream AE	0.750	0.910	0.796	0.851	0.917
Duman ve Erdem 2019	ConvLSTM-AE * Akışlı	0.924	0.929	0.895	—	—

Tablo 4: AE Tabanlı Başlıca Anomali Tespiti Modellerinin Çeşitli Veri Setlerindeki Performansları (AUC)

*Akışlı (with flow) diye bahsedilen modellere, girdi olarak görüntü çerçeveleri yerine ardışık gelen çerçeveler arasındaki fark verilmektedir.

3.2.3. Video görüntülerinden kavga tespiti

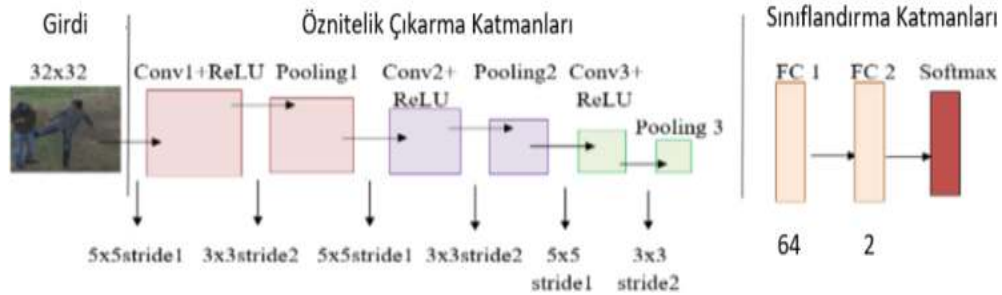
Video görüntülerinden tespit edilebilen durumlardan biri kavga veya birine vurmaktır. Kavga'nın günlük hayatta nadiren gerçekleşmesi onu anormal durum kategorisine koyar. Önceden detaylı olarak açıklandığı gibi anormal olması, onu gözetimsiz eğitime daha yatkın kılmaktadır. Ancak literatürdeki veri setlerinin tamamında normal video sayısı, kavga içeren video sayısına eşit veya çok yakındır (Cheng vd., 2019; Soliman vd., 2019). Bu da veri setlerindeki veri oranının gerçek hayattaki oranları yansıtmamasına neden olmaktadır. Bilindiği kadarıyla da bu veri setlerinden biri üzerinde geliştirilmiş ve pratikte kullanılan bir model bulunmamaktadır. Dolayısıyla veri setlerindeki çalışmaların daha çok, veri setindeki verilerle test edildiğinde en iyi performansı almayı amaçladığı görülebilir.

Literatürde de veri setlerindeki normal ve anormal veri sayısı eşit olduğu için gözetimli öğrenme modelleri kullanılmaktadır. Neredeyse tüm yöntemler normal ve kavga olmak

üzere ikili sınıflandırma problemini çözecek şekilde geliştirilmiştir (Cheng vd., 2019; Soliman vd., 2019). Önceki bölümde videodan anormal durum tespiti modellerinden bahsedilmektedir. Aslında bahsedilen bölümde uygulanan teknik video sınıflandırma problemi için geliştirilmiş olan modellerin amacı, AE ile hibritleri oluşturularak videodan başarıyla anormal durum tespiti yapabilen bir model geliştirmektir. Dolayısıyla bahsedilen modellerin AE'siz yalın halleri kullanılarak videodan kavga tespiti başarıyla yapılmaktadır. Kavga tespiti gerçek hayatta anomali problemi olsa bile, bu çözüm yöntemlerinde sınıflandırma problemi olarak düşünülerek tespit edilmektedir.

3.2.3.1. CNN ve LSTM hibriti modeller

CNN modeli ile kavga tespiti yapılmıştır (Tay vd., 2019) ama çerçeveler (frame) arası zaman ilişkisini öğrenmediğinden iyi sonuç vermemektedir. Şekil 42'de gösterilen modelin son katmanında iki tane perceptron hücresi ve softmax aktivasyon fonksiyonu bulunmaktadır. Problem ikili sınıflandırma problemi olduğu için bu bölümde bahsedilecek tüm modellerde son katman aynıdır. Bir başka çalışmada dört farklı CNN modeli eğitilmiş ve çıktılarından sınıflandırma yapan yeni bir model eğitilerek kavga tespiti yapılmıştır (Carneiro vd., 2019). Bu şekilde bir CNN modeli geliştirilmiştir.

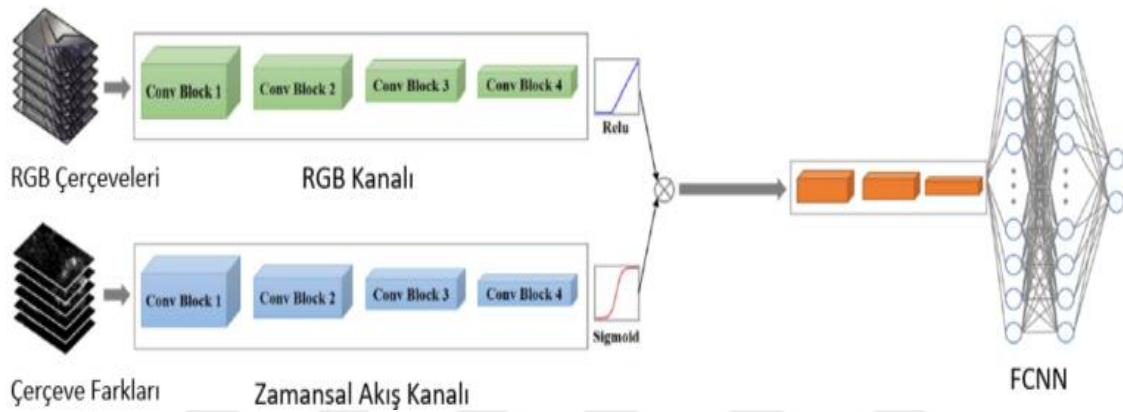


Şekil 42: CNN Modeli (Tay vd., 2019)

Çerçeveler arası zaman ilişkisini öğrenebilen LSTM ve CNN hibriti (RCNN) model ile de kavga tespiti yapılmaktadır (Abdali vd., 2019; Tay vd., 2019). CNN ve LSTM katmanlarını art arda uygulamak yerine matrisler için tasarlanan ConvLSTM katmanını kullanmak video verilerinde daha iyi sonuçlar vermektedir. Dolayısıyla ConvLSTM modeli ile daha başarılı kavga tespiti yapılmaktadır (Sudhakaran vd Lanz, 2017).

Şekil 43'de gösterilen Two-Stream CNN modeli ile başarılı şekilde kavga tespiti yapılmaktadır (Cheng vd., 2020). Videodaki uzaysal bilgiler yeşil kanal tarafından, zamansal

bilgiler mavi kanal tarafından çıkarılarak birleştirilmekte, sonrasında FCNN ile sınıflandırma yapılmaktadır. Anomali tespitinde ConvLSTM ile Two-Stream CNN birbirine yakın performans gösteriyordu. Kavga tespiti sınıflandırma problemi olduğundan Two-Stream CNN, ConvLSTM'ye göre önemli derecede (Hockey Fight veri setinde %3.9 daha fazla doğruluk oranı) daha iyi sonuç vermektedir. Farklı CNN modellerinin çıktılarından FCNN veya LSTM modeli ile sınıflandırma yapan farklı hibrit modeller bulunmakta ama Two-Stream CNN ile yaklaşık aynı performansı vermektelerdir (Aktı vd., 2019; Zhou vd., 2017).



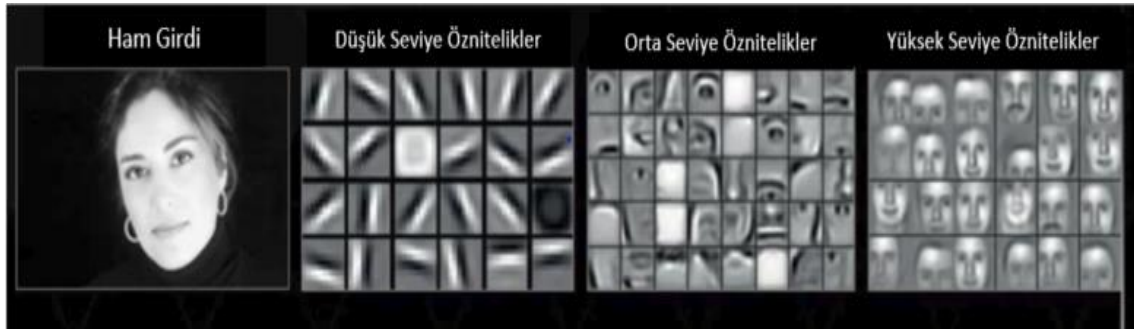
Şekil 43: Two-Stream CNN Modeli (Cheng vd., 2020)

Anormal durum tespitinde Two-Stream CNN-AE, C3D-AE'den daha başarılı idi. Kavga tespitinde ise C3D modeli şu ana kadar bahsedilen tüm modellerden daha iyi sonuç vermektedir (Jiang vd., 2019; Song vd., 2017). Ayrıca bu başarının test edilen tüm veri setlerinde geçerli olması, sınıflandırma problemleri için C3D modelinin tercih edilebilir olduğunu göstermektedir.

3.2.3.2. Öğrenme aktarımı ile geliştirilen modeller

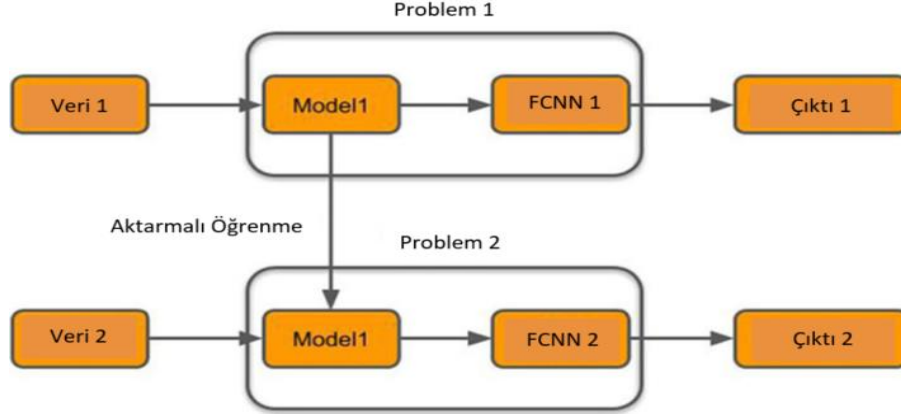
Derin öğrenme çok katmanlı yapay sinir ağı olarak tanımlanabilir. İlk katmanlar düşük seviyeli çıkarımlar yapar. Bu duruma örnek olarak görüntü verilerinden doğruları, doğrulardan eğriler, eğrilerden şekilleri tanıma verilebilir. Sonraki katmanlar önceki katmanların çıkarımlarından orta ölçekli çıkarımlar yaparlar. Örneğin; şekillerden nesne tanıma, sonraki katmanlarda doku tanıma gibi. Son katmanlar ise davranış tanıma, video konusu tanımı gibi yüksek seviyeli çıkarımlar yapılabilir. Şekil 44'de yüz tanıma yapan derin öğrenme modelinin katmanlarının öğrendiği öznelilikler gösterilmektedir. Yüz

görüntüsünden insan tanıyan modelin ilk katmanları eğrileri, orta katmanları şekilleri ve son katmanları önceki katmanların çıkarımlarından yüz tanıyabilmektedir.



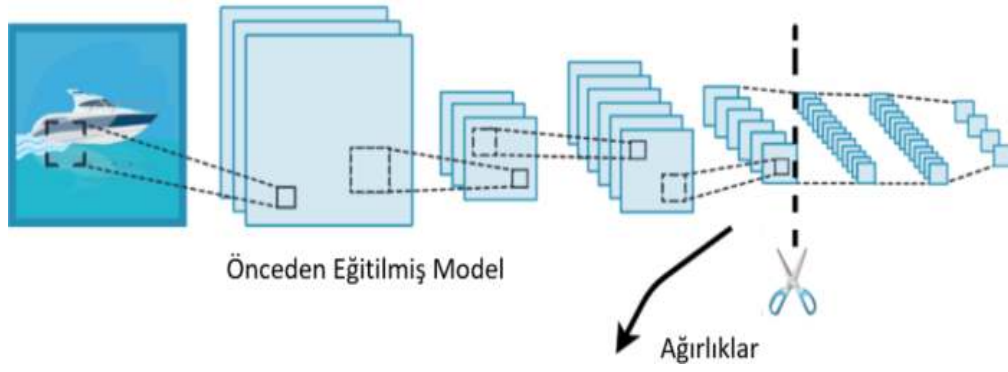
Şekil 44: Yüz Tanıma Problemi Öznitelikleri

Derin öğrenmenin davranış tanıma gibi karmaşık çıkarımları yapabilmesi için çok sayıda öznitelik analiz etmesi gerekir. Bu aşamaları öğrenebilmesi için modelin katman sayısının çok olması gerekmektedir. Problemin karmaşıklığı ile modelin derinliği dolayısıyla modeldeki değerinin öğrenilmesi gereken parametre sayısı artmaktadır. Bununla birlikte modelin doğru eğitilmesi için gerekli veri sayısı da artmaktadır. Bu durum derin öğrenmenin en önemli zorluklarından biridir. Her probleme özel model ve veri seti geliştirmek gerekmektedir. Yalnız bazı problemler benzer olduğundan öğrenilmesi gereken özniteliklerin bazıları ortak olabilir. Örneğin yüz tanıma ve nesne tanıma modellerinin ikisi de doğru ve eğrileri ayırt edebilmelidir. Normalde bir modelin parametreleri rastgele ilklendirilerek model eğitimeye başlanır. Bu durumda yeterli veri olmayınca model problemin çözümünü iyi öğrenemez. Yapılan çalışmalarda bu duruma pratik bir çözüm yer almaktadır. Benzer problem için önceden eğitilmiş (pre-trained) bir modelin parametrelerinin değeri ile modelin parametreleri ilklendirilir. Böylece model ortak öznitelikleri anlamayı verileri kullanmaya başlamadan öğrenmiş olur (Sewak vd., 2018). Sonrasında veri seti ile modeli eğiterek modelin problemin öğrenilmeyen bölümlerini anlaması sağlanır. Böylece model aynı veri seti ile daha iyi performans verecek şekilde eğitilmiş olur. Bu şekilde önceden eğitilmiş modelin ağırlıklarıyla yeni model eğitmeye aktarmalı öğrenme (transfer learning) denir (Zhuang vd., 2019).



Şekil 45: Aktarmalı Öğrenme Süreci Gösterimi

Aktarmalı öğrenmeyi pratikte uygulama yöntemi Şekil 45’te gösterilmektedir. Benzer bir problem için, büyük bir veri seti ile eğitilmiş bir modelin ağırlıkları alınır. O modelin son birkaç katmanı alınmaz. Tasarlanacak modelin ilk katmanları alınan modelin katmanlarıyla aynı şekilde tasarlanır ve ağırlıklar aynen aktarılır (Sewak vd., 2018). Modelin sonuna çözülmesi istenilen probleme uygun birkaç katman eklenir ama bu katmanların ağırlıkları rastgele başlatılır. Modelin ağırlıkları alınan kısmının sonucu öznitelik gibi düşünülebilir. Çünkü ham veriden bir matris veya dizi şeklinde genellikle boyutu girdiden daha küçük bir sonuç elde edilmiş ve bu elde edilen sonuç veriden gerekli bilgileri çıkarmıştır. Sadece nihai sonuç için bu veri incelenmelidir. Problemler farklı olduğundan önceden eğitilmiş problemle çözülmesi istenen problemin son katmanları birbiriyle uyuşmaz ama ikisi de aynı ham veriyi analiz ettiğinden ilk katmandaki çıkarımlar az çok aynı olmalıdır. Yalnız önceden eğitilmiş model daha önceden ve daha çok veri ile eğitildiğinden onun ağırlıkları kullanıldığında modele, eğitime için günler haftalar harcamadan ve çok daha büyük bir veri setinden faydalanarak, öznitelik çıkarma öğretilmiş olur. Sonrasında ağırlıkları önceden eğitilmiş katmanların parametreleri değiştirilmeden sadece son katmanlardaki parametreler problemin veri seti ile eğitilir (Sewak vd., 2018). Bu şekilde elde edilen veri setinden normal yollarla geliştirilebilecek modellerden daha iyi performans veren bir model geliştirilmiş olur.

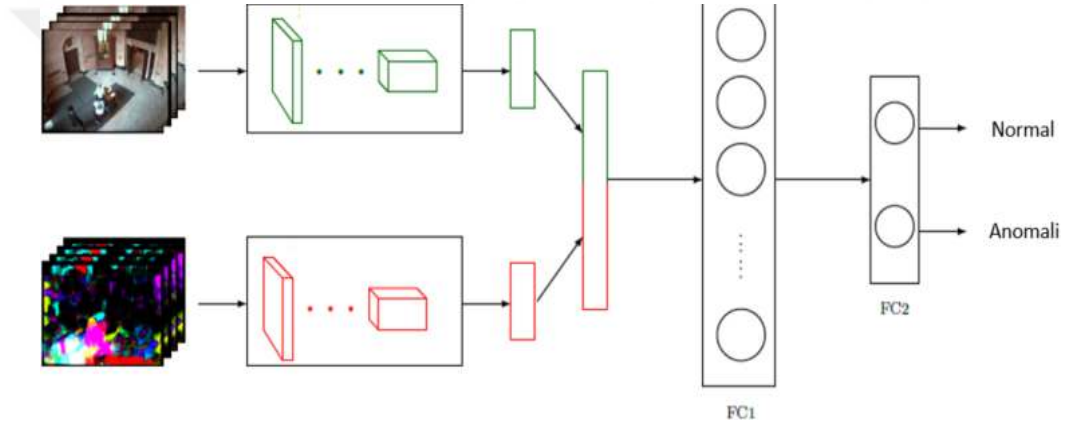


Şekil 46: Aktarmalı Öğrenme ile Nesne Tanıma Yapan CNN Modeli Örneği

Aktarmalı öğrenmeye şu şekilde bir örnek verilebilir. Şekil 46’da gösterildiği gibi görüntüde köpek, kedi, bot ve kuş tanıma yapmak isteyelim. Her sınıf için 10.000’er resim elimizde olsun. Resimlerden çıkarılması gereken piksel, doğru, eğri, kenar, şekil, doku, motif, parça, nesne gibi uzun bir bilgi dizisi var. Bunun için çok katmanlı bir derin öğrenme modeli tasarlanmalı ancak modelin doğru eğitilmesi için yeterli veri yok. Mecburen veri sayısına en uygun olan az katmanlı bir CNN modeli tasarlanması gerekecek ve eğitilecek. Model dört nesneyi sınıflandırmayı öğrenir ama bu durum ideal bir senaryo değildir. Literatürde nesne tanıma için geliştirilmiş Imagenet veri seti bulunmaktadır (Berg vd., 2011). Oluşturulmasına 2006’da başlanmış ve tüm resimleri insanlar tarafından etiketlenmiş ve etiketlenmeye devam etmektedir. Imagenet’te 14 milyondan fazla resim binlerce farklı nesne sınıfı ve onların da çok sayıda alt sınıfları bulunmaktadır. Literatürde Imagenet ile eğitilmiş VGG (Simonyan ve Zisserman, 2015), ResNet (Kaiming vd., 2016), Inception (Szegedy vd., 2015), AlexNet (Krizhevski vd., 2012) gibi CNN modelleri bulunmaktadır. Dolayısıyla bu modellerden biri seçilip onun konvolüsyonel katmanlarının modeli ve ağırlıkları alınır. Sonrasında problem büyüklüğüne göre FCNN kısmı tasarlanır ve eldeki 40.000 veri ile model eğitilince önceki yöntemlerden çok daha iyi performans sağlayan model geliştirilmiş olur.

Görüntüler ile derin öğrenme yaparken konvolüsyonel katmanlar kullanmak standart bir yöntemdir. Dolayısıyla bu katmanları eldeki verilerle eğitmek yerine Imagenet veya hazır bir video veri seti ile önceden eğitilmiş bir modelin konvolüsyonel katmanları aynen alınıp modelin kalanının tasarlanarak uygun veri seti ile eğitilmesi ile kavga tespiti yapma, aktarmalı öğrenmesiz yöntemlere göre daha iyi sonuç verir (Xia vd., 2018; Bansod ve Nandedkar, 2019; Dubey vd., 2019). İki boyutlu görüntüyü girdi olarak alan RCNN gibi modellerle kavga tespiti yaparken aktarmalı öğrenme kullanılmaktadır (Abdali vd., 2019; Sumon vd., 2019). Aktarmalı eğitme modeli özellikle Two-stream CNN modellerinde başarı

ile kullanılmaktadır (Majhi vd., 2019). Two-Stream CNN birinin uzaysal diğerinin zamansal öznitelikleri çıkarması amaçlanan iki bağımsız CNN'den oluşur. Şekil 47'de ikili sınıflandırma yapan Two-Stream modeli detaylı olarak gösterilmektedir. Uzaysal tanıma yapması planlanan CNN görüntülerdeki durağan bilgileri çıkarmalıdır. Bu yüzden burada Imagenet ile önceden eğitilmiş hazır bir model kullanılır ve girdi olarak çerçeve görüntüleri verilir. İkinci CNN'nin zamansal bilgileri yani hareketleri anlaması amaçlanır. Bu yüzden, bu modele girdi olarak ardışık çerçevelerin farkı verilir. İki CNN'nin çıktıları bir dizi haline getirilerek birleştirilir ve FCNN'ye verilir. Model kavga verilerinden eğitilirken sadece zamansal tanıma yapan CNN ve FCNN'nin katmanları eğitilir. Genel olarak bu şekilde geliştirilen modeller daha başarılı sonuçlar vermektedirler.



Şekil 47: Kavga Tanıma Yapan Two-Stream CNN Modeli (Majhi vd., 2019)

En güncel çalışmaların birinde C3D modeli, önceden Sport-1M veri seti (Karpathy vd., 2014) üzerinde önceden eğitilen modeller ilklendirilerek eğitilerek önceki yöntemlerden daha başarılı kavga tanıma yapan bir model geliştirilmiştir (Accattoli vd., 2020). Sport-1M veri seti spor videolarından oluşmaktadır. Bu veri seti ve kavga hareketi benzer hareketler içermektedir. Literatürde aktarmalı öğrenme için kullanılabilecek üç boyutlu görüntü videoları ile eğitilmiş 3D ResNet gibi çeşitli hazır modeller bulunmaktadır (Hara vd., 2017).

Tablo 5'te şu ana kadarki algoritmaların performanslarının doğruluk yüzdesine göre karşılaştırmaları verilmektedir. Tablodan C3D modelinin diğer yöntemlere göre daha iyi performans gösterdiği ve aktarmalı öğrenme ile modellerin performansında ciddi artış sağlanabildiği sonucu çıkarılmaktadır.

Makale	Yöntem	Action Movies	Hockey Fight	Violent Flows
Tay vd. 2019	RCNN	100.00	85.81	—
Sudhakaran vd 2017	ConvLSTM	—	94.10	—
Sudhakaran vd. 2017	ConvLSTM Aktarmalı Öğrenmeli	—	96.00	—
Sudhakaran vd. 2017	ConvLSTM Aktarmalı Öğrenmeli *Akışlı	100.00	97.10	94.57
Cheng vd. 2020	Two-Stream	100.00	98.00	88.87
Song vd. 2017	C3D	99.97	99.62	94.30
Karpathy vd. 2014	C3D Aktarmalı Öğrenmeli	—	98.51	99.29

Tablo 5: Başlıca Kavga Tespiti Modellerinin Çeşitli Veri Setlerindeki Performansları (Doğruluk)

* Akışlı (with flow) diye bahsedilen modellere, girdi olarak görüntü çerçeveleri yerine ardışık gelen çerçeveler arasındaki fark verilmektedir.

4. YÖNTEM

Araştırmanın temel amacı, kapalı alanlarda güvenlik kameralarından elde edilen video görüntülerini kullanarak suç davranışlarını belirlemeye yönelik akıllı bir çözüm üretmektir. Bu kapsamda çalışma, Tasarım Bilimi Araştırma Yöntemi (Design Science Research) (TBAY) (Hevner vd., 2004; Vaishnavi ve Kuechler 2008) çerçevesinde, derin öğrenme yöntem, teknik ve araçları kullanılarak bütünleşik bir yapıda yürütülmüştür. Araştırmanın kuramsal temellerini suç ve suç tespit yöntemleri, yapay zeka ve yapay sinir ağları, derin öğrenme yöntem ve teknikleri ile görüntü işleme bilgi alanları oluşturmaktadır. Cevaplanmaya çalışılan araştırma soruları şöyledir:

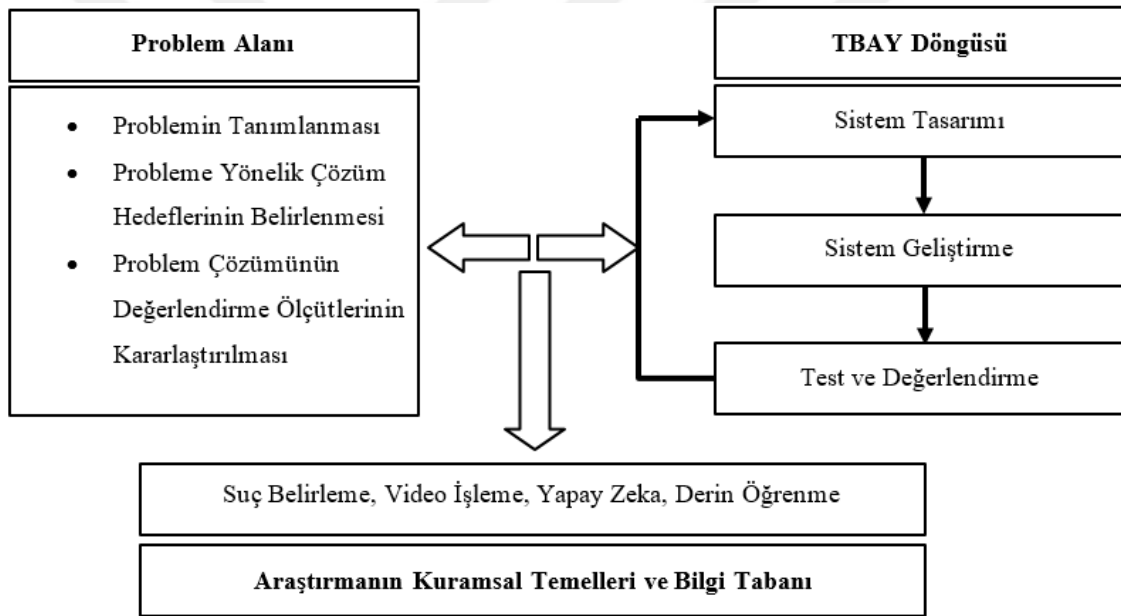
- **Araştırma Sorusu-1:** Suça yönelik davranışların belirlenmesine yönelik kullanılan derin öğrenme yöntem, teknik ve araçları nelerdir?
- **Araştırma Sorusu-2:** Derin öğrenme yöntem, teknik ve araçları kullanılarak kapalı alanlarda suç davranışlarının belirlenmesine yönelik bir uygulama çerçevesi nasıl geliştirilebilir?

Birinci Araştırma Sorusuyla ilgili elde edilen bulgular bu raporun İkinci ve Üçüncü Bölümlerinde sunularak yorumlanmaktadır. Bu bulgulardan hareket edilerek İkinci Araştırma Sorusu aşağıdaki gibi alt sorulara ayrılmıştır:

- Suçla ilgili veri seti oluşturulurken nelere dikkat edilmelidir?
- Hangi davranışlar suçu işaret etmektedir? Bir başka ifade ile suç tespiti kapsamında hangi davranışlar anomali olarak değerlendirilebilir?
- Anomalilerin tespitindeki zorluk dereceleri ve bunların nedenleri neler olabilir?
- Suç tespitinde en iyi performansı elde etmek için modele girdi çerçeveleri nasıl verilmelidir?
- Suç tespiti problemi, anomali tespiti yerine ikili sınıflandırma problemi biçiminde ele alındığında, geliştirilecek modelin başarı düzeyi ne olacaktır?
- Suç tespitinde elde edilebilecek genel başarı düzeyleri nelerdir?

4.1. Araştırma Modeli

Bilişim sistemleri araştırmalarında TBAY'nin son yıllarda artan sıklıkla kullanıldığı söylenebilir. TBAY ile endüstride çeşitli uygulama alanlarındaki problemlere istenilen özellik ve işlevlere sahip sistem, teknik, yöntem, model ve araçların tasarlanması veya geliştirilmesi amaçlanmaktadır. Bu çalışmada benimsenen TBAY'a ait araştırma modeli Şekil 48'de gösterilmektedir. TBAY problem alanı, TBAY döngüsü, kuramsal temeller ve bilimsel bilgi tabanı olmak üzere üç bileşenden oluşmaktadır. Araştırmanın çıktısı olan derin öğrenme çözümü ve içerdiği özgün algoritma, TBAY döngüsü doğrultusunda yinelemeli olarak dört aşamada gerçekleştirilmiştir. Her bir yinelemenin çıktısı geliştirilecek çözüme veya modele ait bir aşamayı içermektedir. Bu aşamalarda ilgili bileşen değerlendirilerek, problem alanına ve bilimsel bilgi alanına olan katkısı ortaya konulmaktadır.



Şekil 48: Tasarım Bilimi Araştırma Yöntem ve Modeli (Uysal, 2017)

Araştırma kademeli olarak gerçekleştirilmiştir. Öncelikle problem ve gerçekleştirilmesi düşünülen çözüm ve bileşenleri belirlenmiştir. Problem çözümünde kullanılacak yöntemin tasarlanıp geliştirilmesi için gerekli kuramsal kavramlar araştırılmış ve elde edilen çıkarımlar sonucu problem çözümü için gerekli hedef ve ölçütler belirlenmiştir. Son aşamada derin öğrenme modeli tasarlanmış, eğitilmiş ve test edilip değerlendirilmiştir. Bu aşamadaki etkinlikler önceden belirlenen hedefleri gerçekleştirecek model elde edilene kadar tekrar edilmiştir. Bu sürecin detayları sonraki bölümlerde yer almaktadır.

4.2. Tasarım, Geliştirme ve Uygulama

Birinci araştırma sorusuyla ilgili detaylı literatür taraması yapılmıştır. Video görüntülerinden akıllı karar veren otonom bir yazılım geliştirmek için en iyi yöntemin derin öğrenme olduğu sonucuna varılmıştır. Bu bağlamda problem derin öğrenme problemi olarak tekrar tanımlanmış, problemin çözüm hedefi ve ölçütleri belirlenmiştir. Tasarlanacak modelleri eğitmek için gerekli veriler toplanıp işlenerek Güvenlik Kamerası Suç Veri Seti oluşturulmuştur. Son olarak, çeşitli derin öğrenme modelleri tasarlanarak eğitilmiştir. Bu süreç dört ana aşamada gerçekleştirilmiştir. Aşamalarda yer alan etkinlikler Tablo 6’da gösterilmektedir.



	Aşamanın Amacı	Uygulama Adımları
Aşama 1	Problem Analizi ve İhtiyaç Tanımlama	• Problemin Tanımlaması • İhtiyaç Analizi ve Tanımlanması • Derin Öğrenme Kapsamında Problemi Tekrar Tanımlama • Probleme Uygun Derin Öğrenme Yönteminin Tasarlanması • Derin Öğrenme Probleminin Değişkenlerini Belirleme • Derin Öğrenme Yöntemi için Veri Özelliklerini Belirleme
Aşama 2	Derin Öğrenme Geliştirme Platformunun Hazırlanması	• Probleminin Çözümüne Yönelik Veri Kaynaklarının Belirlenmesi • Verilerin Toplanması ve Depolanması • Verilerin İşlenmesi ve Etiketlenmesi için Platform ve Yazılımların Belirlenmesi ve Hazırlanması • Derin Öğrenme Yazılımlarının ve Kütüphanelerinin Belirlenmesi • Derin Öğrenme Platform ve Yazılımlarının Hazırlanması • Diğer Bileşenlerin Kurulumu ve Entegrasyonu • Probleme Yönelik Çözüm Hedeflerinin Belirlenmesi • Problem Çözümünün Değerlendirme Ölçütlerinin Belirlenmesi
Aşama 3	Veri Seti Oluşturma	• Veri Hazırlama • Veri Görselleştirme • Veri Temizleme ve İyileştirme • Veri Etiketleme • Veriyi ve Etiketlerini Uygun Formata Dönüştürme
Aşama 4	Derin Öğrenme Model ve Yöntemlerinin Geliştirilmesi	• Derin Öğrenme Modellerinin Tasarlanması • Test ve Karşılaştırma Yöntemlerinin Belirlenmesi • Derin Öğrenme Modellerinin ve Yöntemlerinin Geliştirilmesi • Derin Öğrenme Modellerinin Eğitimi • Derin Öğrenme Modellerinin Geçerlemesi • Derin Öğrenme Modellerinin ve Yöntemlerinin Test Edilmesi ve Değerlendirilmesi • Veri Setinin Değerlendirilmesi • Veri Seti ve Modellerin Güçlü ve Zayıf Yönlerinin Tespiti • Çözümü İyileştirme Yöntemlerinin ve Gelecek Araştırma Sorularının Çıkarımı

Tablo 6: Tasarım, Geliştirme ve Uygulama Adımları

4.2.1. Aşama 1: Problem analizi ve ihtiyaç tanımlama

Araştırma amacını gerçekleştirmeye yönelik hedeflerden birisi de çözümün gerçek hayatta uygulanabilir bileşenlerini ortaya koymaktır. Geliştirilen algoritmanın girdisi olarak kapalı alanlara ait CCTV video görüntüleri kullanılmıştır. Belirli bir zaman aralığında suç analizi için o zaman aralığındaki veri çerçevelerinin algoritmaya girdi olarak verilmesinin yeterli olduğu söylenebilir. Algoritmanın yüksek başarı ile suç tespiti yapabilmesi, her suç gerçekleştiğinde ilgilileri gerçek zamanlı ve doğru bilgilendirme adına gerekli bir özelliktir. Dolayısıyla, çözüm ve bileşenlerinin suç gerçekleştiğinde yüksek başarı oranıyla tespit edebilmesi onun güvenilir bir şekilde kullanımı için gerekmektedir. Ayrıca, sistemin suç olmadığı zamanlarda sıkça yanlış uyarı vermemesi diğer bir önemli gereksinimdir. Aksi takdirde yazılımla sürekli bir insanın ilgilenmesi gerekeceğinden kullanılabilirliği istenilen düzeyde olamayacaktır.

Resim ve video görüntülerinden akıllı çıkarımlar yapan yöntem ve algoritmaların bütünü bilgisayarla görü bilgi alanını oluşturmaktadır. 2000’li yıllarda bu alanda geliştirilen uygulama ve çözümlerin çoğu makine öğrenmesi yöntem ve teknikleriyle mümkün olmuştur. 2010’lu yıllarda ise yapay sinir ağları ve derin öğrenme ile önemli gelişmeler sağlanmıştır. Bunun en büyük nedeni CNN’nin görüntü işleme ve sonuç çıkarımını birlikte ve yüksek başarıyla gerçekleştirebiliyor olmasıdır. Bu bilgiler doğrultusunda araştırma probleminin çözümünde iki derin öğrenme yaklaşımının kullanılmasına karar verilmiştir.

1. **Gözetimsiz Öğrenme:** Sürekli olarak incelenen görüntüde hangi zaman (veya çerçeve) aralıklarında ‘Anomali’ (suç) olduğunun tespiti
2. **Gözetimli Öğrenme:** Belirli bir zaman (veya çerçeve) aralığındaki görüntünün ‘Normal’ ve ‘Suç’ olmak üzere iki sınıftan hangisine ait olduğunun belirlenmesi

Problemin çözümünde derin öğrenmenin en iyi yöntem olduğuna karar verildikten sonra, belirlenen yöntemlerin uygulanabilmesi için gerekli makine öğrenmesi, yapay sinir ağları ve derin öğrenme konuları, formül, terminoloji, yöntem ve modelleri araştırılmış ve öğrenilmiştir. Bu bilgiler 3. bölümde açıklanmaktadır. Sonrasında problem çözümünde kullanılacak model, yöntem ve bunların parametreleri belirlenmiştir. Derin öğrenme modellerinin başarılı eğitilmesi için çok sayıda veriye ihtiyaç duyulmaktadır. Genel olarak, eğitimde kullanılan veri sayısı arttıkça modelin performansı da artmaktadır. Modelin başarısında veri sayısı kadar verinin hazırlanma, etiketlenme ve girdi olarak verilme formatı da önemlidir. Bu yüzden veri ve verilerin gerekli özellikleri tespit edilmiştir. Sonraki bölümlerde sırasıyla model geliştirmek için gerekli platformun hazırlanması, veri setinin oluşturulması ve modellerin geliştirilmesi süreçleri detaylı olarak anlatılmaktadır.

4.2.2. Aşama 2: Derin öğrenme platformunun hazırlanması

Araştırmanın ikinci aşaması dört başlık altında ve her süreçte önceki süreçlerde yapılan çıkarımlar doğrultusunda gerçekleştirilmiştir.

4.2.2.1. Veri seti için video toplanması

Gözetimli öğrenmede her sınıf için eğitim ve test verisi gerekmektedir. Gözetimsiz öğrenmede, eğitim sadece normal verilerle yapılırken; test etme normal veri ve anomali verileriyle birlikte yapılır. Gerekli normal veri sayısı, anomali verisi sayısından daha büyüktür. Problem karmaşıklığı arttıkça başarılı model geliştirmek için yapay sinir ağlarının derinliği ve karmaşıklığını arttırmak gerekmektedir. Bir modelin derinliği ve karmaşıklığı arttıkça, modeli genellenebilir ve yüksek başarılı eğitime için veri sayısını da artırmak gerekmektedir. Bu yüzden suç içeren ve içermeyen çok sayıda video görüntüsüne ihtiyaç duyulmaktadır.

Veri setinde hem normal hem de kavga, silahlı soygun gibi suç görüntülerinin bulunması gerekmektedir. Veri setini oluşturmak için Youtube web sitesinde halkın kullanımına açık, kapalı ortamlarda CCTV kameralarıyla kaydedilmiş çok sayıda normal ve suç görüntüleri araştırılıp seçilmiştir. Veri toplama sürecinde ilk olarak YouTube web sitesinde suça ilişkin çeşitli anahtar kelimeler (ör. kavga, soygun, silahlı saldırı vb.) aranmış ve bir URL listesi elde edilmiştir. Ardından, elde edilen bağlantılardan videoları otomatik olarak indirmek için bir yazılım kullanılmış ve ilgisiz olanların ayıklanması için her video kontrol edilmiştir. Veri setinin çeşitliliğini kapsamlı bir şekilde genişletmek için, anahtar kelimelerin dilini Türkçe, İngilizce, Hintçe dahil olmak üzere çeşitli dillere çevirerek yukarıdaki süreçler tekrarlanmıştır.

4.2.2.2. Derin öğrenme platformunun hazırlanması ve kütüphanelerin belirlenmesi

Derin öğrenme modellerinin oluşturulması, eğitilmesi ve test edilmesi, karar metodu gibi genel algoritmalar kullanımı, değerlendirme metriklerinin hesaplanması ve verinin okunması, işlenmesi ve etiketlenmesi için Python 3 Programlama Dili kullanılmıştır. Python açık kaynaklı olması ve kapsamlı makine öğrenmesi, yapay sinir ağları ve matematik

kütüphaneleri içermesi, kolay erişim ve kullanım imkanlarından dolayı tercih edilmiştir (Van Rossum ve Drake, 2009). Python'a uyumlu, rahat ve pratik kullanımı, kolaylıkla kütüphane yükleme ve kullanımı ve akademisyenler için ücretsiz lisans imkanlarından dolayı bilgisayar programı olarak Pycharm kullanılmıştır (PyCharm, 2010). Videoları inceleyip düzenleyerek girdi videoları elde etmek için açık kaynaklı Kdenlive video düzenleyicisi kullanılmıştır (Kdenlive, 2015). Verilerde uygulanacak diğer işlemler için Python kütüphaneleri yeterli olmuştur. Videoların okunması için OpenCV kütüphanesi (Bradski, 2000), işlenmesi için OpenCV ve Numpy kütüphanesi (Oliphant, 2006), videoların uygun girdi formatına dönüştürülmesi ve girdilerin etiketlerinin çıkarılması Numpy kütüphanesi kullanılarak gerçekleştirilmiştir.

Derin öğrenme kütüphaneleri olarak Tensorflow (Abadi vd., 2016) ve Keras (Chollet vd., 2015) kullanılmıştır. Bu kütüphanelerde yapay sinir ağı tasarlamak için gerekli hücreler, katmanlar, eniyileştiriciler, yapay sinir ağını eğitme, test etme, erken durdurma gibi süreçler için metotlar bulunmaktadır. Çıktıların etiketlerle karşılaştırılması ve metrik değerlerinin hesaplanması Numpy ile yapılmıştır. Metriklerle tablolar oluşturularak sonuçlar incelenmiştir. Ayrıca metrikler ve test etme süreleri ile MATLAB'da çizimler (plot) üretilerek performans analizinin detaylandırılması sağlanmıştır (MATLAB, 2010).

Makine öğrenmesi ve özellikle yapay sinir ağları modellerinin başarılı şekilde eğitilmesi için çok sayıda veri gerekmektedir. Derin öğrenmede ise bu sayı daha da artmaktadır. Bu da bir eğitme süresini günlerden haftalara çıkarmaktadır. Model geçerleme için defalarca eğitme gerektiğinden bu süreci hızlandırmak için GPU'ların paralel hesaplama gücünden faydalanılmaktadır. Video verileri çerçeve serilerinden oluşur. Örneğin 3 saniyelik bir videoda yaklaşık 90 görüntü çerçevesi bulunmaktadır. Bu yüzden modelleri bütün veri seti ile eğitmeyi iyi bir GPU'ya sahip bile olsa kişisel bilgisayarlar üzerinde araştırma süresi içinde gerçekleştirmek mümkün olmamıştır. Derin öğrenme için özel tasarlanan bilgisayarları almak veya bulut hizmetinden faydalanmak için yeterli araştırma bütçesi olmadığı için Google'ın akademisyenler, araştırmacılar ve öğrenciler için ücretsiz sunduğu Google Colab bulut hizmeti kullanılmıştır (Google Colab, 2017). Bu hizmette kullanımına izin verilen GPU ve rastgele erişimli bellekteki (Random-Access Memory, RAM) kapasitesi sınırlıdır. Dolayısıyla endüstride kullanılabilir bir çözüm geliştirmek için yeterli seviyede olmasa da araştırma probleminin çözümü için yeterli düzeyde hizmet sunulabilmiştir. Google Colab verileri buluta yükledikten sonra Jupyter Notebook yazılımının çevrim içi kullanımına da imkân tanımaktadır (Project Jupyter, 2015). Jupyter Notebook programının dili Python'dur ve bu nedenle araştırma sürecinde öncelikle

büyüklüğü az olan veri için PyCharm ortamı kullanılmıştır. Ardından Google Colab'da Jupyter Notebook ile öğrenme modelleri geliştirilerek sonuçlar gözlenmiştir.

4.2.2.3. Verilerin işlenmesi ve etiketlenmesi için platformun hazırlanması, kütüphanelerin belirlenmesi

Anomali, bir süreçteki ani veya kısa zamanlı sıra dışı durumları ifade etmektedir. Bir videoda da benzer şekilde suç uzun sürse bile suça işaret eden durumlar çok kısa süreler alırlar. Örneğin birine vurma, cisim fırlatma gibi olaylar en fazla 1-2 saniye sürmektedir. Koşma, yerde yatma, el kaldırma gibi davranışlar uzun sürse bile davranış tanıma için 2-3 saniyelik görüntü yeterlidir. Bu yüzden literatürdeki davranış tanıma ve anomali tespiti veri setleri genellikle 2-3 saniyelik video verilerinden oluşmaktadır. Bu çalışmada da indirilen videoların normal davranış veya suç işaret eden kısımları çıkarılarak verilerin 2-3 saniyelik videolardan oluşturulmasına karar verilmiştir. Bu şekilde geliştirilen bir model ile kapalı bir mekânda otomatik olarak şu şekilde suç analizi yapılabilmektedir: Kameradan gelen görüntü her üç saniye sonunda girdiye dönüştürülerek modele verilmektedir. Modele videoyu girdi olarak vermek için çerçeve serisine dönüştürmek yeterli olmaktadır. Daha sonra geliştirilen algoritmanın görüntü işleme ve girdi oluşturma kısımları yapılacaktır. Model çıktısı ile normal veya suç kararı verilmektedir. Dolayısıyla algoritma üç saniyelik girdide suç tespit ederse, hangi zaman anında anomali olduğunu da bulmuş olur. Bu noktada yazılım zamanı kaydedebilir, görüntüyü bir ekranda gösterebilir, görüntüyü kaydedebilir ve ilgililere bildirebilir. Bu yüzden başarılı şekilde veri seti ve suç tespiti yapan model geliştirmek problem çözümü için yeterlidir.

Video ve resimler çok sayıda pikselden oluşan büyük verilerdir. Bunlar çeşitli sıkıştırma (compression) yöntemleri kullanılarak sabit diskte depolanırlar. Program görüntüleri sabit diskten alıp açınca RAM'deki boyutu önemli ölçüde artmaktadır. Bilgisayar işlemcisi veya GPU işlem yaparken, üzerinde işlem yapılmayan veriler hızlı erişim için RAM'de tutulmaktadır. Bu yüzden hızlı eğitim için RAM büyüklüğü de GPU performansı kadar olmasa da önemlidir. Suç veri setinin sabit diskteki toplam boyutu yaklaşık 300 megabayt iken Python'da görüntüleri açınca veri, 32 gigabaytlık RAM'e sığmamaktadır. Sabit diskten veri okumak RAM'e göre yaklaşık 50 kat, sabit diske veri yazmak ise yaklaşık 200 kat daha yavaştır (Storagereview, 2020). Bu yüzden videolar Google Colab'da OpenCV kullanılarak resim çerçeveleri şeklinde depolanmıştır. Gerekğinde Keras kütüphanesinin ImageLoader sınıfı ile bir dizindeki belirlenen sayıdaki

ardışık resim çerçeveleri bir bütün olarak alınıp, girdiye dönüştürülerek modele girdi olarak verilmiştir. ImageLoader sınıfı GPU verinin belirli bir kısmı üzerinde hesaplama yaparken sonraki aşamada gerekli verileri hazırlamaktadır. Bu sayede RAM'e sığan veri ile model eğitime kadar hızlı olmasa da eğitime sürecini pratikte gerçekleştirecek kadar hızlandırmaktadır. Veri seti büyüklüğü arttıkça eğitilen modelin test ve genellenebilirlik performansı artmaktadır. Lakin yukarıdaki nedenden dolayı veri seti büyüklüğü 617 video ile sınırlı tutulmuştur. Literatürdeki anomali ve kavga veri setleri büyüklükleri düşünüldüğünde, 617 videoluk veri setinin literatüre katkı sağlama ve bu çalışmada kullanabilme adına yeterli olduğu söylenebilir. Ancak, ticari kullanıma uygun bir yazılım geliştirmek için veri sayısının artırılıp yüksek GPU veya bulut hizmetinin alınması gerektiği vurgulanmalıdır.

4.2.2.4. Problemin değerlendirme ölçütlerinin ve çözüm hedeflerinin belirlenmesi

Önceki bölümlerde açıklandığı gibi geliştirilen yazılımın uygulanabilir olması için suçu yüksek başarı ile tespit etmesi ve düşük FP değerine sahip olması gerekmektedir. Bu yüzden modelin performans değerlendirmesi için doğruluk değeri önemli bir ölçüttür ve modellerin suç tespiti başarısı da bu ölçüte göre değerlendirilecektir. Ayrıca gözetimsiz öğrenme modellerinin başarısı literatürde sıklıkla AUC değeri ile ölçülmektedir. Düşük FP performansı ise FPR ile değerlendirilecektir.

Pratikte performansı mümkün olduğunca artırmak tercih edilmekle birlikte gerçek hayatta kullanılan yazılımın doğruluk oranı gereksinimi kullanıldığı mekâna göre değişebilir. Doğruluk oranı 0.99 olan bir model ortalama 100 suçtan 99'unu tespit eder. Bir AVM veya kuyumcu ortamında yazılım kullanılırsa 0.99 doğruluk değerinin sağlanması beklenirken, bir markette 0.95 doğruluk yeterli görünebilir. Hatta kişi sayısının az olduğu sokaklarda veya gece saatlerinde kullanılan mekânlarda 0.90 doğruluk bile yeterli görülebilir. FPR değeri suç olmadığı süre boyunca modelin ortalama kaç karardan birinde yanlışlıkla suç kararı verdiğini gösterir. Örneğin FPR değeri 0.01 ise model 100 karardan 1'inde yanlışlıkla suç çıkarımı yapar. 3 saniyede bir karar verildiğinden bu durumda ortalama 300 saniyede (5 dakika) bir yanlış uyarı verilir. Bu durumda, gerçek hayatta, geliştirilen bu yazılımı kullanmak mümkün değildir. FPR 0.001 iken ortalama 50 dakikada bir yanlış alarm gerçekleşir. Gerçek hayatta kullanılmak için üretilen model için bu performans yeterli olur. Çünkü yazılım bir ortamda kullanılmadan önce ortamda

kullanılacak kamera görüntüleri bir hafta kadar toplanabilir. Sonrasında eğitilen model buradan çıkarılan normal verilerle tekrar eğitilerek modelin ortamı tanınması sağlanabilir. Böylece FPR ciddi oranda azaltılarak yanlış alarm sıklığı haftada veya ayda bire indirilebilir. Gerçek hayatta kullanılan mekâna göre değişmekle birlikte, genel olarak FPR'nin düşürülmesi doğruluğu yükseltmekten daha öncelikli olabilir çünkü ortalama iki günde bir alarm veren bir yazılım akademik olarak çok başarılı olabilir ama yazılımın kullanıldığı mekânlara iki günde bir gereksiz yere yetkiliyi olay yerine sevk etmek önemli bir kaynak israfıdır.

Uygulanabilir bir başlangıç hedefi olarak, doğruluğu $[0.9, 0.95]$ aralığında, FPR'si $[0.001, 0.01]$ aralığında olan bir model geliştirmek belirlenebilir. Modelin doğruluğu ile FPR değeri arasında bir ödünleşme (trade off) vardır. Yani bir modelde biri iyileştirilirse diğeri kötüleşir. Bu sebeple 0.001 FPR elde etmek kaynak gerektiren zor bir hedeftir. Bu araştırmada suç tespiti mümkün olduğu, üretilen model ve veri setlerinin geliştirildiği takdirde bunun mümkün olduğunun gösterilmesi amaçlanmaktadır. Bu sebeple her ne kadar FPR değerinin düşük olması amaçlansa bile doğruluk değerinin yüksek olması öncelikli hedef olarak belirlenmiştir. Yukarıda bahsedilen araştırma kısıtları ve sınırlılıklarının yanı sıra literatürde bulunan anomali ve özellikle kavga veri setlerine oranla daha çeşitli davranış içeren, daha zorlayıcı ve problemin zorluğunu yansıtan bir veri seti oluşturulmuştur. Bu durum bir sonraki bölümde veri seti anlatılırken detaylı olarak açıklanacaktır. Dolayısıyla benzer karmaşıklık ve zorlukta bir çalışma örneği olmadığından, referans olarak alıp ulaşılan performansın üstünde bir model geliştirmeye çalışılacak bir algoritma veya model tespit edilememiştir. Bahsedilen koşullar altında araştırma sonunda doğruluk değeri 0.70 olan model geliştirilmesinin uygun olabileceği değerlendirilmiştir.

Doğruluk ile FPR arasında ödünleşme var. Örnek olarak doğruluk hedefi 0.95 iken 0.99 doğruluğa ulaşılmış olsa algoritmanın doğruluk değerinin azaltılması ile başarılan FPR değeri azaltılabilir. 0.70 doğruluk için düşük FPR elde etmek veya ödünleşme gerçekleştirmek mümkün değildir. Dolayısıyla ikinci hedef olarak 0.25 FPR belirlenmiştir.

4.2.3. Aşama 3: Veri seti oluşturma

Model eğitilirken geri yayılım ile veri setindeki her veriden çıkarımlar yapmayı öğrenir. Dolayısıyla her videodaki davranış, nesne, ortamdaki ışık ve renk miktarı gibi unsurlar öğrenilir. Bu yüzden kaliteli-kalitesiz kameradan alınan; az ışıklı-çok ışıklı ortamda çekilen; çok insanlı-az insanlı ortamda çekilen; dar açıdan bakan-geniş açıdan bakan

kameradan alınan birbirinden farklı videolardan veri seti oluşturmak önemlidir. Bu sebeple farklı ortamlarda kullanılabilir ve genellenebilir başarılı bir model geliştirme veri seti büyüklüğü ve veri çeşitliliği ile doğrudan ilgilidir. Bu sebeple olabildiğince çeşitli ortam ve davranış videoları toplanmaya çalışılmıştır. Ayrıca aynı ortamdan alınan normal görüntüler ve suç görüntüleri çok değerlidir çünkü model birbirine yakın açı, ışık miktarı gibi özelliklere sahip iki videodan (biri normal, diğeri suç) normal ve suç unsurları farkını daha rahat öğrenebilir. Bu yüzden her video detaylı olarak incelenerek normal durum veya anomali işaret eden davranışlar içeren bütün kısımları belirlenmiştir.

Toplanan videoların ilgisiz veya kullanılmayacak kısımları Ek-4'te bulunan Şekil 54'teki gibi Kdenlive ile kesilerek çıkarılmıştır. Videoların davranış içeren bölümleri 2-3 saniyelik videolar halinde ayrılmıştır. Seçilen videoların çözünürlükleri OpenCV ile 300x300 olacak şekilde ölçeklendirilmiştir. Videoların saniyelik çerçeve sayısı (frame per second) 30'dur. Suç tespitini derin öğrenme yöntemleriyle gerçekleştirebilmek için, bu şekilde çeşitli normal ve suç içeren videolar elde edilmiştir. Bu çalışmada YouTube'da yer alan halka açık görüntülerden elde edilen 617 video incelenmiştir. Normal videolar bir klasöre, suç videoları ayrı bir klasöre konulmuştur. Son olarak önceden detaylı olarak açıklandığı üzere verilerin ImageLoader sınıfı ile okunabilmesi için her klasördeki videolar Ek-4'te bulunan Şekil 60'da gösterildiği gibi OpenCV ile resim çerçevelerine ayrılarak çerçeve sırası korunarak kaydedilmiştir. Bir modele girdi verilmek istenildiği zaman resim dizisi OpenCV ile alınır. OpenCV görüntüyü gri tonlamalı görüntüye daha sonra ise Numpy dizinine (array) dönüştürür. Her dizinin elemanları 255'e bölünerek alınan değerler [0, 255] aralığından [0,1] aralığına taşınır. Bu şekilde elemanların oranları korunarak her çerçevenin normalizasyonu yapılmış olur. Böylece girdi hazırlanmış olur çünkü Keras'ta geliştirilen bir model değişken olarak Numpy dizini almaktadır. Daha sonra aynı klasördeki veriler aynı sınıf olarak etiketlenerek verilerin etiketlenmesi tamamlanır. Sınıflandırma problemlerinde etiketler one-hot vektörlerden oluşur ve model çıktıda en büyük değere sahip hücrenin karşılığı olan sınıf kararı vermiş olur. Gözetimli öğrenme modelinde ise etiket olarak girdi verilirken, modelin uzaklığa göre verdiği kararlar verinin ait olduğu klasörün etiketi kontrol edilerek test edilirler.

Her video isimlendirilirken mekân numarası da belirtilmiştir. Belirli bir mekân görüntülerinden elde edilen videolar aynı mekân numarası ile etiketlenmiştir. Bir video "mekanNumarası_videoNumarası" adıyla kaydedilmiştir. Derin öğrenmede eğitme veri setinde kullanılan veriler test sırasında kullanılmamalıdır aksi takdirde modelin performansı doğru tespit edilmemiş olur. Çünkü gerçek hayatta kullanılırken farklı bir mekânda farklı

koşullardan gelen veri analiz edilmek istenmektedir. Bu yüzden eğitime veri setinden elde edilen analiz ve değerler test edilirken kullanılmamalıdır. Aynı mekânda eğitilen bir video ile model o arka planı öğrenir. Dolayısıyla test aşamasında aynı mekândan gelen farklı bir video bile test edilse eğitim ve veri setleri tamamen ayrı olmamış olur. Bu yüzden veri seti ile modeller eğitilirken aynı mekâna ait bütün normal videolar ve suç videoları ya eğitim ya da test veri setinde kullanılmıştır.

Güvenlik Kamerası Suç Veri Setinde 288 normal, 329 suç videosu bulunmaktadır. Tüm suç videoları aynı klasör altında toplandı. Bu videolar farklı davranışlar içerebilmektedir, dolayısıyla suç videoları 8 kategoriye ve her suç videosu da kendi içerdiği anormal duruma göre Ek-4'te bulunan Şekil 55, 56 ve 57'de gösterildiği gibi alt kategoriye ayrıldı. Veri seti ile ilgili detaylı bilgi Tablo 7'de verilmektedir. Suç alt kategorisi olarak belirlenen anormal durumlar şöyledir:

1. **Kavga:** Kavga, vurma gibi şiddet içeren suç görüntüleri
2. **Koşma:** Suç ortamında kişi veya gruplar halinde aynı veya farklı yönlerde koşma görüntüleri
3. **Silah Doğrultma:** En az bir kişinin silah, kesici ve delici alet gibi tehlikeli cisimleri diğer insanlara doğrulttuğu video görüntüleri
4. **Yerde Uzanma:** Suç ortamında en az bir kişinin yan, sırtüstü veya yüzüstü şekilde yerde uzandığı video görüntüleri
5. **Atlama:** Suç ortamında en az bir kişinin zıpladığı veya bir cismin üstünden atlayarak arka tarafına geçtiği video görüntüleri
6. **El Kaldırma:** Suç ortamında en az bir kişinin iki veya tek elinin havada, iki elin başta veya iki elin de yanlarda açılmış olduğu video görüntüleri
7. **Eylemsizlik:** Suç belirten davranış içermeyen ama suç olduğunu işaret eden silah, kesici alet, biber gazı, kask, maske gibi suç olduğunu işaret eden nesneler içeren suç görüntüleri. Maske gibi nesneler içerip suç içermeyen videolar normal video içerisinde değerlendirilmiştir.
8. **Diğer:** Üstte belirtilen durumları içermeyen suç görüntüleri; problemi zorlaştıracı olarak, göz yaşartıcı sprey sıkma, camları veya kapıları kırma, ürün bulunan dolapları kırma görüntülerinden oluşmaktadır.

Silah doğrultma, koşma, atlama, el kaldırma gibi anomaliler belirli davranış öğeleri ve örüntüsü içerir. Bir dükkânda yerde uzanma hastalık, rahatsızlık veya suç ortamından kaynaklanan nedenlerle de gerçekleşebileceği için bir anomali olarak değerlendirilmiştir. Kavga ise çok çeşitli unsurlar barındırabilir: İttirme, vurma, cisim fırlatma gibi unsurlar akış

süreci içinde değerlendirilerek suç olup olmadığı sonucuna varılır. Suçun ve bireylerin doğası gereği çok farklı davranışlar da suça konu olmaktadır, bu durumlar diğer kategorisi altında toplanmıştır. Bu nedenle olabildiğince çeşitli davranış videoları veri setine dahil edilmiştir. Bu sayede problemin doğasını yansıtacak bir veri seti oluşturulmuştur, ancak bu durum veri setinde yüksek başarı gösteren model geliştirmeyi zorlaştırmıştır. Bu açıdan bakıldığında, 0.70 doğruluk hedefine ulaşmış olmak araştırma hedefleri ve sınırlılıkları açısından yeterli olarak değerlendirilmektedir. Bunun yanı sıra özellikle literatürdeki kavga veri setleri göz önünde bulundurulduğunda geliştirilen suç veri seti ve modellerin literatüre katkısı açısından araştırmanın önemini daha da artırdığı düşünülmektedir.

Video Grubu Türü	Mekân Sayısı	Bir Mekândaki Azami Video Sayısı	Video Sayısı	Ortalama Video Süresi (Saniye)
Kavga	37	1	51	2.22
Koşma	30	4	41	2.18
Silah Doğrultma	53	4	70	2.30
Yerde Uzanma	17	2	23	2.30
Atlama	15	2	21	2.28
El Kaldırma	16	2	20	2.29
Eylemsizlik	15	3	18	2.28
Diğer	33	5	44	2.22
Suç	131	9	288	2.26
Normal	244	8	329	2.43
Veri Seti	332	12	617	2.35

Tablo 7: Güvenlik Kamerası Suç Verilerinin Gruplara Göre Dağılımı

Birinin elini kaldırarak raftan bir şey alması normal kategorideki videolar arasına konulmuştur. Ama silah taşıyan bir kolluk kuvvetinin görüntüsü normal kategoriye eklenmemiştir, çünkü bu durumda akıllı bir algoritmanın normal ve anormal arasındaki farkı anlaması oldukça güçtür. Bu durumlar için eylemsizlik kategorisi oluşturulmuştur. Bu kategoride bir veya birden çok kişinin elinde silahla ayakta durduğu video görüntüleri yer almaktadır. Bu açıdan eylemsizlik kategorisi özel bir kategoridir. Suçu işaret eden net bir davranış olmamakla birlikte normal bir insanın suç ve suçluyu tespit edebileceği görüntülerden oluşmaktadır. Dolayısıyla standart yöntemlerle geliştirilen modellerin, ilgili davranışa bakarak normal davranışlardan ayırt etmesi pek mümkün olmayan suç

görüntülerinden oluşmaktadır. Bu tür davranışlarda da suç tespitinin öneminden dolayı, bu videoların veri setine dahil edilince modelin performansını düşüreceği öngörülmesine rağmen, veri setine dahil edilmiştir. Yukarıda bahsedilen anomaliler düşünüldüğünde, modelin veri setinde ortalama performans sergilemesi hedeflenmiştir. Kavga ve koşmada yüksek başarılı, eylemsizlik gibi anomalilerde ise düşük başarılı olması öngörülmüştür. İleriki çalışmalarda nesne tanıma algoritması ile geliştirilen modelin sonuçlarından yeni bir sınıflandırıcı eğitilerek hibrit model geliştirmesi ile daha başarılı çözüm yolu elde edilebilir.

Güvenlik Kamerası Suç Veri Setindeki sekiz farklı anormal durum videolarının hepsi suç kategorisi altında toplanmaktadır. Toplam suç sayısı, normal sayısına yakındır. Veri setinin dağılımı Ek-4'te bulunan Şekil 58 ve 59'da detaylı olarak gösterilmektedir. Dolayısıyla anormal durum yerine sınıflandırma problemi olarak değerlendirilebilir ve bir ikili sınıflandırıcı modeli başarılı şekilde eğitilebilir. Ayrıca gözetimsiz eğitime ile de anormal tespiti yapmak da tercih edilebilir. Veri setinden modeller eğitilirken suç klasöründeki videolar tek bir sınıf gibi etiketlenerek eğitilmektedir. Dolayısıyla test ederken, model videolar için sadece normal veya suç tahmini yapabilmektedir. Ama sekiz durumun da suçun alt kategorisi olduğu düşünülürse bu durum bir sorun değildir. Ayrıca sekiz durum her seferinde sadece bir anormal durum kategorisi ve normal kategori altındaki videolar ile test edilerek ayrı ayrı değerlendirilebilir.

Veri setinde sekiz farklı anormal durum belirlenmiştir. Dolayısıyla performans değerlendirilirken sadece o anormal sınıf ve normal videolardaki performans değerlendirilmesi yani sekiz ayrı değerlendirilme yapılması daha sağlıklı olur. Çünkü bir anormal durumu iyi tespit eden model diğerini kötü tespit edebilir. Burada bir anormal sınıf ve normal durum videoları ele alındığında normal video sayısı önemli ölçüde fazladır. Gerçek hayatta anormal durumun nadir olduğu düşünülürse veri seti bu şekilde değerlendirildiğinde modelin gerçekte kullanılabilirliği daha doğru analiz edilebilir.

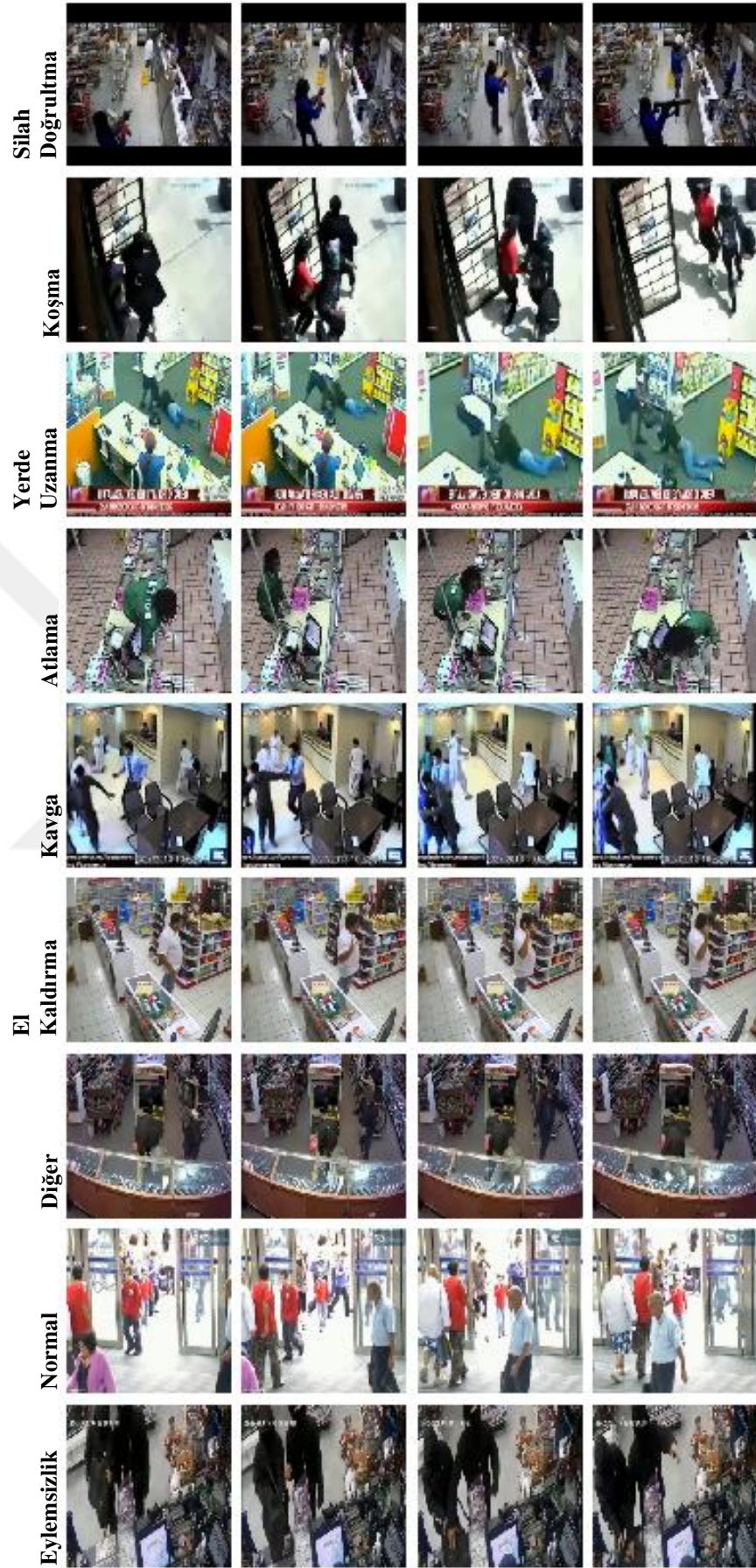
Örnek olarak test veri setinde normal sayısı 150, suç sayısı 100, koşma sayısı 20 olsun. Eğitilen model normal videoların 10'una suç desin ($FP=10$) ve suçların hepsini başarıyla tespit etsin. Burada suç ve koşma değerlendirilirken FP sayısı aynı olmasına rağmen ehemmiyeti aynı değildir. Bu durum hassasiyet hesaplaması yapılırken şu şekilde ortaya çıkmaktadır:

$$Suç Hassasiyeti = \frac{100}{100 + 10} = 0.909$$

$$Koşma Hassasiyeti = \frac{20}{20 + 10} = 0.667$$

Koşma anomalisinin hassasiyeti suç tespiti hassasiyetine göre ciddi derecede düşüktür. Burada önemli olan suç tespit etmedeki performansı diye düşünülerek bunun kritik olmadığı yorumlanabilir ama yine veri setini analiz etmede, sonraki süreçlerde veri seti veya suç tespitine özel model geliştirmede önemli fayda sağlayabileceği söylenebilir.





Şekil 49: Güvenlik Kamerası Suç Veri Seti Anomalileri Örneği

Normal veri sayısı, değerlendirilen anormal durum verisi sayısından en az birkaç kat fazla olmalıdır. Bu sayede özellikle FP gibi değerlerden etkilenen AUC ve benzeri metrikler, modelin pratik hayattaki performansını daha iyi yansıtır. Veri setinde suç tespiti performansı yapılmalıdır ama ayrı ayrı yapılan analizler de önemli çıkarımlar yapmayı sağlar. Dolayısıyla problem, suç ve normal veri sayısına bakılınca yanıltıcı şekilde sınıflandırma problemi gibi görülebilir. Hazırladığımız veri setinde 1 normal ve 8 anormal durum videoları bulunmaktadır. Ayrıca normal video oranı normal durumun gerçekleşme sayısında olduğu gibi daha fazladır. Burada 8 ayrı anormal durumun hepsi aynı sonuca, yani suçun varlığına işaret etmektedir. Gelecekteki araştırmacılar tercih ederlerse normal ve suç olarak ikili sınıflandırma yerine, 9 kategorili sınıflandırma yapan modeller geliştirebilirler. Suç verileri alt kategorilere bölündüğünden veri seti farklı senaryolar için esneklik sağlamaktadır.

Literatürde benzer anormal davranışlardan bir anomali sınıfı oluşturarak geliştirilen anomali veri setleri bulunmaktadır. Farklı anomali veri setleri olmasına rağmen hepsinin kendi normal verileri bulunmaktadır. Güvenlik Kamerası Suç Veri Setinde koşma, kavga gibi farklı anomaliler ve ortak normal veriler bulunmaktadır. Bu sayede literatürdeki veri setlerinin sağlamadığı, aynı normal veri ile eğitilen bir modelin farklı anormal durumu tespit etmedeki başarısı karşılaştırılabilecektir. Böylece farklı anomali problemlerini karşılaştırmalı analiz etme imkânını sağlaması, bu çalışmanın literatüre olan bir başka katkısı olduğu söylenebilir.

Literatürdeki gerçek olaylardan alınmış çok sayıda videodan oluşan kavga/suç tespiti yapan veri setlerinin normal ve anormal sınıf veri sayılarının eşit olması, geliştirilen modellerin analizinin yanıltıcı olmasına neden olmaktadır. Güvenlik Kamerası Suç Veri Seti sadece CCTV görüntülerinden oluşmasının ve veri sayısının çokluğunun yanı sıra sınıflar arası orantısız veri sayısı ile endüstride kullanmak için daha iyi model geliştirme imkânı sunmaktadır.

Güvenlik Kamerası Suç Veri Seti kapsamı bakımından birçok yenilik sağlamaktadır. Literatürdeki diğer anomali veri setleriyle kıyasla veri sayısı bakımından çok daha büyük ve birden çok anomali problemini birlikte çözen bir model geliştirme imkânı sağlayan bir veri setidir. Suç videosu sınıflandırma problemi dışında, suç anı tespit etme problemi için de sadece şiddet değil, suç işaret eden farklı anomaliler bulunduran bir veri seti de olduğu söylenebilir.

AİDA ve suç veri setleri ile Güvenlik Kamerası Suç Veri Seti Tablo 8’de karşılaştırılmaktadır. Literatürde kavga, koşma, düşme veri setleri bulunmaktadır. Bu gibi anomalilerin bir veri setinde toplanması ile gerçek hayatta kullanılabilecek suç tespiti yapan

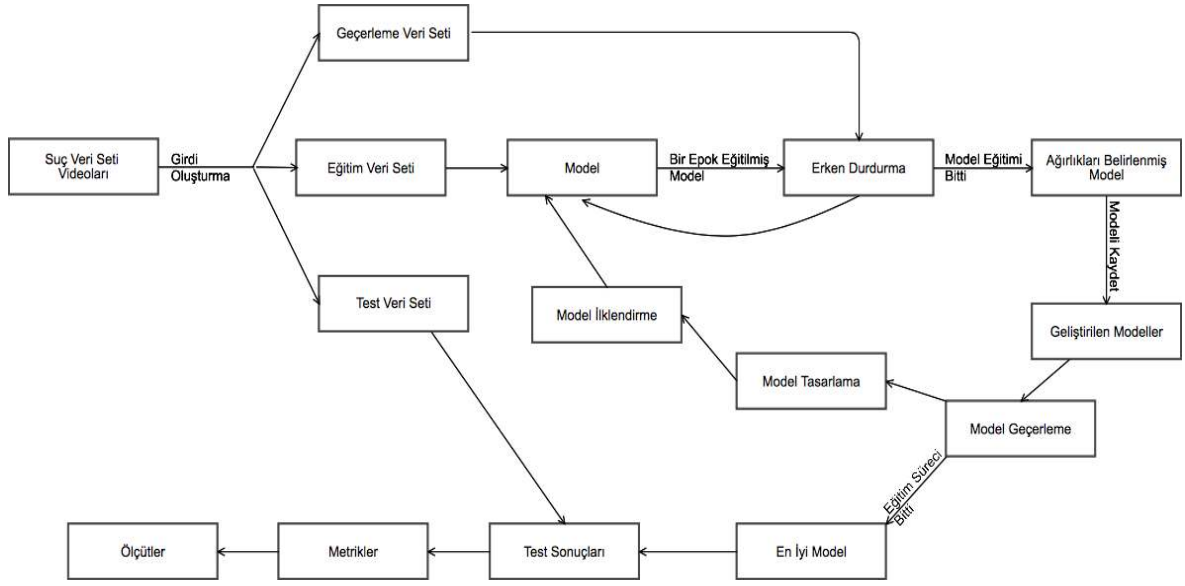
model geliřtirmede kullanılabilecek veri setinin literatüre kazandırılması amaçlanmıřtır.

Veri Seti	Gerçek Rol	Başlıca Anomaliler	Farklı Anomali Sayısı	Anomali Sayısı	Video Sayısı	Yılı
CUHK Avenue	Gerçek	Kořma Cisim Fırlatma	14	47	37	2013
Subway	Gerçek	Bilet Basmadan Geçme Yanlış Yöne Gitme	19	85	2	2008
ShanghaiTech	Gerçek	Normal Yürüme Dıřındaki Durumlar	—	130	437	2017
UCF-CRIME	Gerçek	Hırsızlık Vandalizm Patlama Eve Hırsız Girmesi	13	—	1900	2018
Real Life Violence Situations	Gerçek	řiddet	1	1000	2000	2019
RWF-2000	Gerçek	Kavga	1	1000	2000	2019
Güvenlik Kamerası Suç	Gerçek	Kavga Silah Doğrultma El Kaldırma Yerde Yatma	8	288	617	2020

Tablo 8: Güvenlik Kamerası Suç Veri Seti ile Başlıca AİDA ve Kavga Veri Setlerinin Karşılaştırılması

4.2.4. Ařama 4: Derin öğrenme model ve yöntemlerinin geliřtirilmesi

Bu bölümde geliřtirilen modeller, modellere girdi çerçevelerinin verilme yöntemi ve çıktılarından suç tespiti kararı verme algoritmaları anlatılmaktadır. Suç tespiti modellerinin geliřtirilme ve test edilme süreci řekil 50’de gösterilmektedir. Modellerin eğitim kodları Ek 5’te bulunan řekil 61, 62, 63, 64, 65 ve 66’da; test kodları Ek 6’da bulunan řekil 67, 68 ve 69’da detaylı olarak gösterilmektedir.



Şekil 50: Gözetimsiz Öğrenme Süreci Akış Şeması

4.2.4.1. Girdi formatı seçimi

Derin öğrenme modelinin girdi boyutları sabittir. Tasarlanan bir model her zaman aynı boyutlarda girdi alır. Dolayısıyla bir zaman dilimindeki çerçevelerden belirli bir miktarı seçilerek modele verilmelidir. Girdideki çerçeve sayısı performansı etkiler. Girdinin çerçeve sayısı arttıkça modele verilen bilgi de arttığından modelin performansı artar, fakat belirli bir sayıdan sonra performansta ciddi bir artış olmaz çünkü yeterli miktarda çerçeveden model gerekli çıkarımı yapabilmektedir. Çerçeve sayısı ile modelin çalışma süresi orantılıdır dolayısıyla modele gereğinden fazla çerçeveli girdi vermek yanlıştır. Bu başlık altında farklı çerçeve sayılarıyla girdi alan modeller eğitilerek performansları karşılaştırılmıştır. Böylece model için ideal girdi çerçeve sayısının belirlenmesi amaçlanmıştır. Ayrıca sadece bütün ardışık çerçeveleri birlikte girdi olarak vermek dışında, videolardan çeşitli yöntemlerle girdi çerçeveleri seçilerek de modeller eğitilerek analiz edilmiştir. Böylece bir videodan alınacak çerçeve numaralarını belirleyerek en iyi girdi oluşturma yöntemi araştırılmıştır.

Eşit Aralıklı Örnekleme: Klasik video girdisi üretme yönteminde girdi olarak verilmek istenen aralıktaki tüm çerçeveler çıkarılır ve modele girdi olarak verilir. Ancak girdi olarak verilmek istenen videonun çerçeve sayısı, girdi boyutundan büyük ise bazı çerçeveleri vermek mümkün olmaz. Bu durumda genellikle şu iki yöntemden biri izlenir:

- Videonun çerçeve sayısı girdi büyüklüğünden aşırı büyük değilse videonun başındaki ilk girdi büyüklüğü kadar çerçeve alınarak girdi oluşturulur.
- Videonun çerçeve sayısı girdi büyüklüğünden aşırı büyükse eşit aralıklı örnekleme

(Uniform Sampling) ile girdi elde edilir.

Eşit aralıklı örneklemede çerçeveler sırasıyla birden başlayarak numaralandırılır. Toplam çerçeve sayısına N , girdi büyüklüğüne K diyelim. Bu durumda $[1, N]$ aralığı $(K-1)$ eşit aralığa bölünür. İlk çerçeve girdiye alınır. Sonrasında eşit aralıklarla birer çerçeve alınarak girdiye dâhil edilir. En son N . çerçeve alınarak girdi oluşturulur. Aralığı eşit aralığa bölünce değerler tam sayı çıkmazsa en yakın tam sayıya yuvarlanırlar. Örneğin çerçeve sayısı 56 olan videodan eşit aralıklı örnekleme ile 12 çerçeveli bir girdi elde etmek için alınan çerçeve numaraları şöyledir:

$$\{1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56\}$$

İnsan davranışı tanıma ve anomali tespiti problemlerinde eşit aralıklı örnekleme standart girdi oluşturma yöntemi olarak değerlendirilebilir. Bu yüzden birinci yöntem olarak eşit aralıklı örnekleme seçilmiştir. Bu yöntemde girdi boyutunun performanslarının karşılaştırılması için her videodan 10, 12, 15 ve 20 boyutlu birer girdi elde edilerek veri setleri oluşturulmuştur. Seçilen bir model veri setleri ile eğitilerek başarı ve hız karşılaştırılması yapılmıştır. Sonuçlar 5. bölümde değerlendirilmiştir. Ayrıca detaylı olarak Ek 1’de bulunan Tablo 14’te gösterilmektedir

Dinamik Örnekleme: Bir çalışmada HAR problemi için çerçeve numaralarının eşit aralıklı seçilmesi (eşit aralıklı örnekleme) yerine her girdinin ayrı ayrı incelenerek çerçevelerinin girdinin yapısına göre seçilmesinin modelin performansını arttırdığı gösterilmektedir (Seo vd., 2016). Algoritma girdi yapısının önceden belirlenen sabit yöntemle değil işlem yapılacağı zaman duruma göre belirlenmesinden dolayı dinamik olarak adlandırılır. Dinamik algoritmalarda parametre değeri girdiden hesaplanır ve girdiden girdiye değişebilir. Dolayısıyla bu yöntemle girdi oluşturmaya dinamik örnekleme (Dynamic Sampling) denilmektedir.

Yukarıda bahsedilen çalışmada bir çerçeve alındıktan sonra, sıradaki çerçevelerle arasındaki değişim miktarı hesaplanır. Belirli bir eşikten fazla değişimin gerçekleştiği ilk çerçeve seçilir. Süreç tekrarlanarak çerçeve numaralarına karar verilir. Bu çalışmada ise iki çerçeve arasındaki değişim olarak çerçeve matrislerinin elemanlarının farklarının mutlak değerlerinin ortalaması seçilmiştir. A ve B çerçeveleri arasındaki fark şu formülle ifade edilebilir:

$$\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |A_{ij} - B_{ij}|$$

Geliştirilen modelde girdi boyutu sabittir, bu yüzden çerçeveler bahsedilen çalışmadakinden farklı algoritma ile seçilmiştir. Öncelikle ilk ve son çerçeveler her zaman girdiye alınmıştır. Diğer çerçeveler videonun başından sonuna kadar gerçekleşen değişimleri eşit miktarda yakalayacak şekilde seçilmiştir. Bir video için aradaki çerçeve numaralarını belirleyen algoritma şu şekildedir:

1. Videodaki ardışık çerçeveler arasındaki değişim hesaplandı.
2. Değişimler toplamı (girdi çerçeve sayısı - 1)'e bölünerek girdideki ardışık çerçeveler arası olması gereken ortalama değişim miktarı belirlendi.
3. $i=1$ seçildi.
4. $j=0$ ve $d=0$ seçildi
5. $(i+j)$. çerçeve ile $(i+j+1)$. çerçeve arasındaki değişim d 'ye eklendi.
6. d değeri 2. adımda hesaplanan ortalamaya yakınsa veya onu geçtiyse $(i+j+1)$. çerçeve girdiye eklendi. i , $(i+j+1)$ olarak güncellenerek 4. adıma gidildi.
 - Aksi takdirde j 1 artırılarak 5. adıma gidildi.
7. Girdi için gerekli tüm çerçeve numaraları belirlenene kadar üstteki adımlar izlendi.

Bu şekilde video eşit zaman aralıkları yerine, ortalama piksel değişimi eşit olan aralıklara bölünmüş oldu. Bu yöntemde girdi boyutunun performanslarının karşılaştırılması için her videodan 10, 12, 15 ve 20 boyutlu birer girdi elde edilerek veri setleri oluşturulmuştur. Seçilen bir model veri setleri ile eğitilerek, başarı ve hız karşılaştırılması yapılmıştır. Sonuçlar 5. bölümde değerlendirilecektir. Ayrıca detaylı olarak Ek 1'de bulunan Tablo 15'te gösterilmektedir.

4.2.4.2. Çoklu girdi ile eğitime

Literatürdeki insan davranışı tespiti ve anomali tespiti çalışmalarında genellikle bir videodan bir girdi oluşturularak model geliştirilir. Lakin eşit aralıklarla girdi oluşturulurken çoğu çerçeve kullanılmaz. Bu yöntemde seçilen çerçeve numaraları sabit miktarda artarak büyümektedir. Dolayısıyla girdinin tüm çerçeve numaraları artış miktarından küçük olmak şartıyla eşit miktarda artırılırsa yeni bir girdi elde edilmiş olur. Örneğin çerçeve sayısı 60 olan videodan eşit aralıklı örnekleme ile 12 çerçeveli 5 girdi elde edilebilir. Bu girdilerin

çerçeve numaraları şöyledir:

{1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56}

{2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57}

{3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58}

{4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59}

{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60}

Bu şekilde bir videodan elde edilen girdiler arasında önemli fark olmayabilir lakin girdilerin hepsi eğitimde kullanıldığında model bir girdiden öğrenemediği detayı diğer girdiden öğrenebilir. Böylece veri setindeki girdi sayısı artırılmış olur. Bu yöntem, girdi sayısını örnek olarak 5 katına çıkarmanın video sayısını 5 katına çıkarmak kadar olmasa da modelin performansını artırması beklendiği için geliştirilmiştir.

Yöntemi test etmek için her eğitim veri seti videosundan 10, 12, 15, 20 boyutlu sırasıyla 6, 5, 4 ve 3 girdi elde edilerek eğitim veri setleri oluşturulmuştur. Modeller eğitildikten sonra farklı karar verme algoritmaları ile test edilerek, girdi ve karar verme yöntemi arasındaki ilişki de incelenmiştir. Bu yöntemle hem modeli eğitirken ve hem de eğittikten sonra performansını artırma yöntemleri aranmıştır. Sonuçlar 5. bölümde değerlendirilecektir. Ayrıca detaylı olarak Ek 2’de bulunan Tablo 16, 17, 18 ve 19’da gösterilmektedir

Çoklu girdi ile eğitilen modeller için geliştirilen test girdisi oluşturma ve suç kararı algoritmaları şöyledir:

Tek Girdi ile Test Etme: Bu yöntemde her test veri seti videosundan bir girdi çıkarılarak test veri setleri oluşturulmuştur. Test sürecinde iki girdi oluşturma yöntemi kullanılmıştır. Bu yöntemler şöyledir:

1. Eşit Aralıklı Örnekleme: Bir videodan eşit aralıklı örnekleme ile bir girdi elde etme
2. Dinamik Örnekleme: Bir videodan dinamik örnekleme ile bir girdi elde etme

Birden Çok Girdi ile Test Etme: Eğitim videolarından birden çok girdi çıkarılarak kullanıldığı gibi test videolarından da birden çok girdi çıktıları elde edilebilir. Bir videodan elde edilen her girdinin çıktısı için model farklı çıkarımlar yapabilir ama önemli olan video için doğru karar verebilmektir. Bir girdi ile model test edilirken modelin kararına göre hareket etmek gereklidir. Ancak bir videodan alınan çeşitli girdilerden modelin çıktıları değerlendirilirse, video için daha doğru ve istikrarlı çıkarımlar yapılarak suç tespiti algoritmasının videolardaki performansı artırılabilir.

Bu yöntemde test veri seti videolarından 10, 12, 15, 20 boyutlu sırasıyla 6, 5, 4 ve 3 girdi elde edilerek test veri setleri oluşturulmuştur. Test aşamasında bir videodan elde edilen girdilerin analizi ile video için suç tahmini yapılmıştır. Bunun için dört karar verme algoritması geliştirilerek test edilmiştir. Bu algoritmalar şunlardır.

1. Suç Tespiti Yapılan Çerçeve Sayısı: Videodan elde edilen girdilerden iki veya daha fazlası için model suç çıkarımı yapmışsa video için de suç kararı verilmiştir.
2. Ortalama Suç Tespiti Olasılığı: Bir girdi için anomali kararı verme şekli eğitime yöntemine göre değişmektedir. Gözetimsiz öğrenmede bir girdi için beklenen ile, elde edilen temsil arasındaki uzaklık eşik değerinden fazla ise anomali kararı verilir. Bu yüzden bir video için, çıktıların uzaklıkları ortalaması belirlenen farklı bir eşik değerinden fazla ise anomali kararı verilmiştir. Gözetimli öğrenmede bir girdi için anomali olasılığı eşik değerinden (genelde 0.5) fazla ise anomali kararı verilir. Bir video için, çıktıların olasılıkları ortalaması belirlenen farklı bir eşik değerinden (0.4) fazla ise anomali kararı verilmiştir.
3. Çerçeve Sayısı ve Ortalama Olasılık: Video için 1. ve 2. yöntemlerin ikisi birden suç sonucuna vardıysa suç kararı verilmiştir.
4. Çerçeve Sayısı veya Ortalama Olasılık: Video için 1. ve 2. yöntemlerden en az biri suç sonucuna vardıysa suç kararı verilmiştir.

Çıktı Tümlleştirme ile Test Etme: Bir problemin çözümünde matematiksel ve mantıksal kural yerine ANN veya DL modellerini kullanmak genel olarak daha başarılı sonuçlar verir. Çünkü bu yöntemler özellikle derin öğrenme tüm veri detaylı olarak incelenerek klasik yöntemlerle elde edilemeyecek çıkarımlar yapılabilir. Bu yüzden bu yöntemde geliştirilen modelin bir videonun girdileri için verdiği sonuçlarla yeni bir sınıflandırma modeli eğitilerek video için suç tespiti yapılmıştır. Yöntem gözetimli öğrenme modellerine uygulanabilmektedir. Bu yöntemde model çıktı elde etmek için kullanılmamıştır. Modelin çıktı katmanından bir önceki katmanın çıktı vektörü, bir başka ifadeyle çıktı katmanının girdi vektörü alınmıştır. Bu vektör problem için öznitelik vektörü olarak düşünülebilir, çünkü modelin eğitilirken problemde edindiği tüm çıkarımlarla girdinin değerlendirilmesi sonucu elde edilmiştir. Bu açıdan bu yöntem aktarmalı öğrenme olarak adlandırılabilir.

Yöntem uygulanırken öncelikle eğitim veri setinde birinci model eğitilmiştir. Geçerleme veri setindeki videolardan birinci model ile öznitelikler elde edilmiştir. Daha sonra her video için çıkarılan öznitelik vektörleri birleştirilerek girdi oluşturulmuş ve ikinci model geçerleme veri seti üzerinde eğitilmiştir. Test veri setindeki her videodan çıkarılan

girdilerin birinci modele verilmesi, birinci modelin çıktılarının da ikinci modele verilmesi ile performans değerlendirilmesi yapılmıştır. Birden çok model çıktısını birleştirerek hesaplamalara devam ettiği için bu yönteme tümleştirme (fusion) yöntemi denilmektedir. İkinci model olarak iki FCNN geliştirilmiştir. İlk modelden bir girdi için alınan vektör büyüklüğü N, bir videoya ait girdi sayısı ise K iken geliştirilen modeller şöyledir:

1. Tümleştirme (MLP): K vektör birleştirilerek elde edilen $K*N$ boyutlu girdi vektörleri ile iki gizli katmanlı MLP modeli eğitilip kullanılmıştır.
2. Tümleştirme (LSTM): K vektör, ikinci bir boyut doğrultusunda birleştirilerek elde edilen $K*N$ 'lik iki boyutlu girdiler (K tanelik vektör dizisi) ile LSTM modeli eğitilip kullanılmıştır.



Şekil 51: MLP (soldaki, Yeşil Katman: ReLu), ve LSTM (sağdaki, N: Girdi Boyutu) FCNN Modelleri

4.2.4.3. Gözetimsiz eğitime modeli

AİDA probleminde ConvLSTM-AE başarı ile kullanılmaktadır (Chong ve Tay, 2017). Bu yüzden gözetimsiz öğrenme modeli olarak ConvLSTM-AE geliştirilmiştir. Model gri tonlamalı videodan elde edilen ve boyutları $N \times 227 \times 227$ olan 3 boyutlu girdi almaktadır. Buradaki N çerçeve sayısıdır. Veri seti şu şekilde parçalanmıştır:

- Eğitim Seti: 200 normal video
- Geçerleme Seti: 53 normal video
- Test Seti: 76 normal video ve 288 suç videosu

Modele etiket olarak girdi verilmektedir. Test aşamasında normal videolarda çıktıda,

girdinin az hatayla geri oluşturulması beklenmektedir. Dolayısıyla çıktı ile girdi arasındaki fark (piksel farklarının mutlak değerlerinin ortalaması) eşik değerinden fazla ise anomali sonucu çıkarılmaktadır. A girdisi ve B çıktısı arasındaki fark şu şekilde ifade edilebilir:

$$\frac{1}{N \times 227 \times 227} \sum_{i=1}^N \sum_{j=1}^{227} \sum_{k=1}^{227} |A_{ijk} - B_{ijk}|$$

Geçerleme veri setindeki ortalama fark eşik değeri olarak alınmıştır. Böylece eşik değeri verilerden dinamik olarak hesaplanmıştır. Gözetimsiz öğrenme modelinin eğitim ve test süreci aşamaları şu şekildedir:

1. Bir videodan girdi oluşturma yöntemi belirlenir.
2. Videolardan belirlenen yöntemle göre çerçeveler seçilip $N \times 227 \times 227$ boyutlu gri tonlamalı girdiler ve etiketler oluşturulur. Girdiler normalize edilir.
3. Azami epok sayısı olarak 100 seçilir.
4. Tasarlanan modelin katman sayısı, katman büyüklüğü, öğrenme oranı vb. değişkenleri belirlenir.
5. Modelin ağırlıkları (beklentisi 0.0, standart sapması 0.05 olan normal dağılımdan (Gaussian Distribution) örnekleme ile) rastgele ilklendirilir.
6. Model eğitim veri seti ile bir epok eğitilir.
7. Modelin geçerleme veri setindeki doğruluk değeri ölçülür.
 - Doğruluk değeri o ana kadarki en büyük değer ise modelin o anki mimarisi ve ağırlıkları kaydedilir.
8. Modelin en yüksek geçerleme doğruluğunda üst üste 10 epok gelişme olmadıysa eğitim sonlandırılır (erken durdurma) ve sonraki adıma gidilir.
 - Aksi takdirde 6. adıma gidilir.
9. Modelin geçerleme veri setindeki performansı o ana kadar eğitilen modellerinki ile karşılaştırılarak değişkenler değerlendirilir. Son birkaç adımda öncekilere göre daha iyi performans gösteren model elde edilmediyse, modelin iyileşmesi için planlanan tüm değişken kombinasyonları test edildiyse ve modelin performansının daha fazla geliştirilemeyeceği düşünülüyorsa sonraki adıma gidilir.
 - Aksi takdirde 4. adıma gidilir.
10. Geçerleme veri setinde en yüksek doğruluk veren model seçilir ve test veri setinde metrikler ve ortalama karar verme süresi hesaplanır.

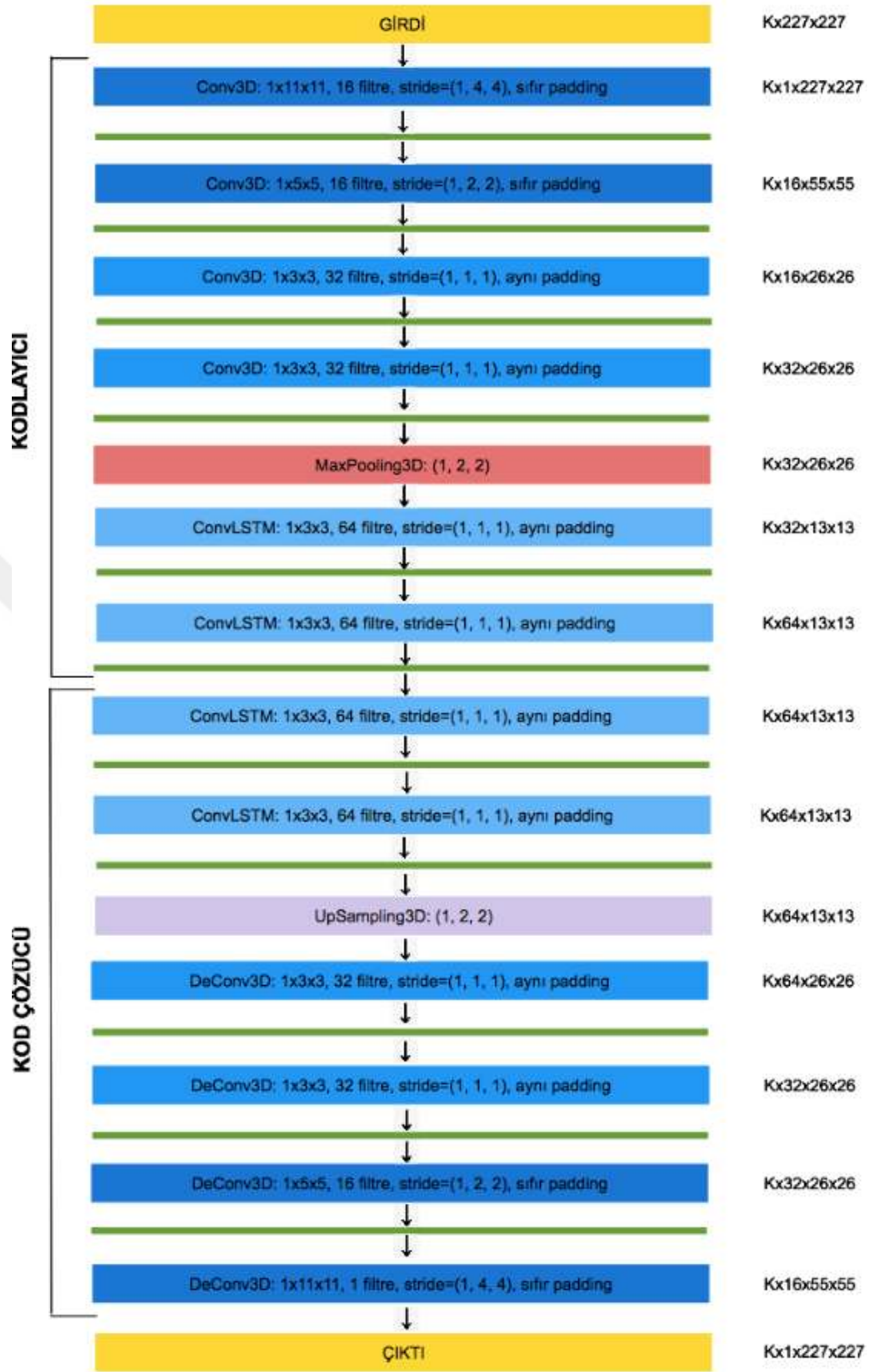
Gözetimsiz derin öğrenme modelinin sözde kodu (Pseudocode) ise şu şekildedir:

Algoritma 1: Eğitimsiz Öğrenme Modeli Eğitim ve Test Etme

```
Girdi:  $x_{train}, x_{val}, x_{test}, y_{test}$   
Çıktı:  $m_{best}, dist, metrikler$   
 $m \leftarrow model\ olustur$   
 $m.opt \leftarrow Adam$   
 $m.loss \leftarrow \text{Çapraz Entropi}$   
 $m.lr \leftarrow 0.00001$   
 $m.w \leftarrow Normal(0, 0.05)$   
 $epokSayısı \leftarrow 100$   
 $m_{best} \leftarrow m$   
 $valLoss_{best} \leftarrow 0$   
 $patience \leftarrow 0$   
for  $i$  from 1 to  $epokSayısı$  do  
   $m.fit(x_{train}, batchSize = 32)$   
   $valLoss \leftarrow m.evaluate(x_{val})$   
  if  $valLoss_{best} > valLoss$  then  
     $m_{best}.w \leftarrow m.w$   
     $valLoss_{best} \leftarrow valLoss$   
     $patience \leftarrow 0$   
  else  
     $patience \leftarrow patience + 1$   
    if  $patience == 10$  then  
      break  
    end  
  end  
end  
 $m.w \leftarrow m_{best}.w$   
 $dist \leftarrow mean(valLoss_{best})$   
 $metrikler \leftarrow m.evaluate(x_{test}, y_{test}, dist)$ 
```

Model geçерleme sonucu geliştirilen ConvLSTM-AE modeli Şekil 52’de gösterilmektedir. Kullanılan diğer değişkenler ve değerleri şunlardır:

- Hata Fonksiyonu: Çapraz entropi
- Eniyileştirici: Adam
- Öğrenme Oranı: 0.00001
- Mini-Yığın Büyüklüğü = $360 / (\text{girdi çerçeve sayısı})$



Şekil 52: Suç Tespiti için ConvLSTM-AE Modeli (Yeşil Katman: Tanh Katmanı)

4.2.4.4. Gözetimli eğitim modelleri

Kavga veri setlerinde gözetimli eğitim yöntemleri kullanıldığından, Güvenlik Suç Veri Setinde gözetimli öğrenme yöntemlerinin performanslarının değerlendirilmesine karar

verilmiştir. Bu yöntemler gri tonlamalı görüntü alıp ikili sınıflandırma yaparlar. Normal videolar 0, suç videoları 1 olarak bire bir kodlama ile etiketlenmiştir. Çıktı olarak girdinin suç olma olasılığı dönülür. Veri seti şu şekilde parçalanmıştır:

- Eğitim Seti: 200 normal video ve 150 suç videosu
- Geçerleme Seti: 53 normal video 36 suç videosu
- Test Seti: 76 normal video ve 102 suç videosu

Gözetimli öğrenme modellerinin eğitim ve test süreci aşamaları şu şekildedir:

1. Bir videodan girdi oluşturma yöntemi belirlenir.
2. Videolardan belirlenen yonteme göre çerçeveler seçilip gri tonlamalı girdiler ve etiketler oluşturulur. Girdiler normalize edilir.
3. Test sırasındaki karar verme algoritmasına karar verilir.
4. Azami epok sayısı olarak 100 seçilir.
5. Tasarlanan modelin katman sayısı, katman büyüklüğü, öğrenme oranı vb. değişkenleri belirlenir.
6. Modelin ağırlıkları (beklentisi 0.0, standart sapması 0.05 olan normal dağılımdan (Gaussian Distribution) örnekleme ile) rastgele ilklendirilir.
7. Model eğitim veri seti ile bir epok eğitilir.
8. Modelin geçerleme veri setindeki doğruluk değeri ölçülür.
 - Doğruluk değeri o ana kadarki en büyük değere ise modelin o anki mimarisi ve ağırlıkları kaydedilir.
9. Modelin en yüksek geçerleme doğruluğunda üst üste 10 epok gelişme olmadıysa eğitim sonlandırılır (erken durdurma) ve sonraki adıma gidilir.
 - Aksi takdirde 7. adıma gidilir.
10. Modelin geçerleme veri setindeki performansı o ana kadar eğitilen modellerinki ile karşılaştırılarak değişkenler değerlendirilir. Son birkaç adımda öncekilere göre daha iyi performans gösteren model elde edilmediyse, modelin iyileşmesi için planlanan tüm değişken kombinasyonları test edildiyse ve modelin performansının daha fazla geliştirilemeyeceği düşünülüyorsa sonraki adıma gidilir.
 - Aksi takdirde 5. adıma gidilir.
11. Geçerleme veri setinde en yüksek doğruluk veren model seçilir ve test veri setinde metrikler ve ortalama karar verme süresi hesaplanır.

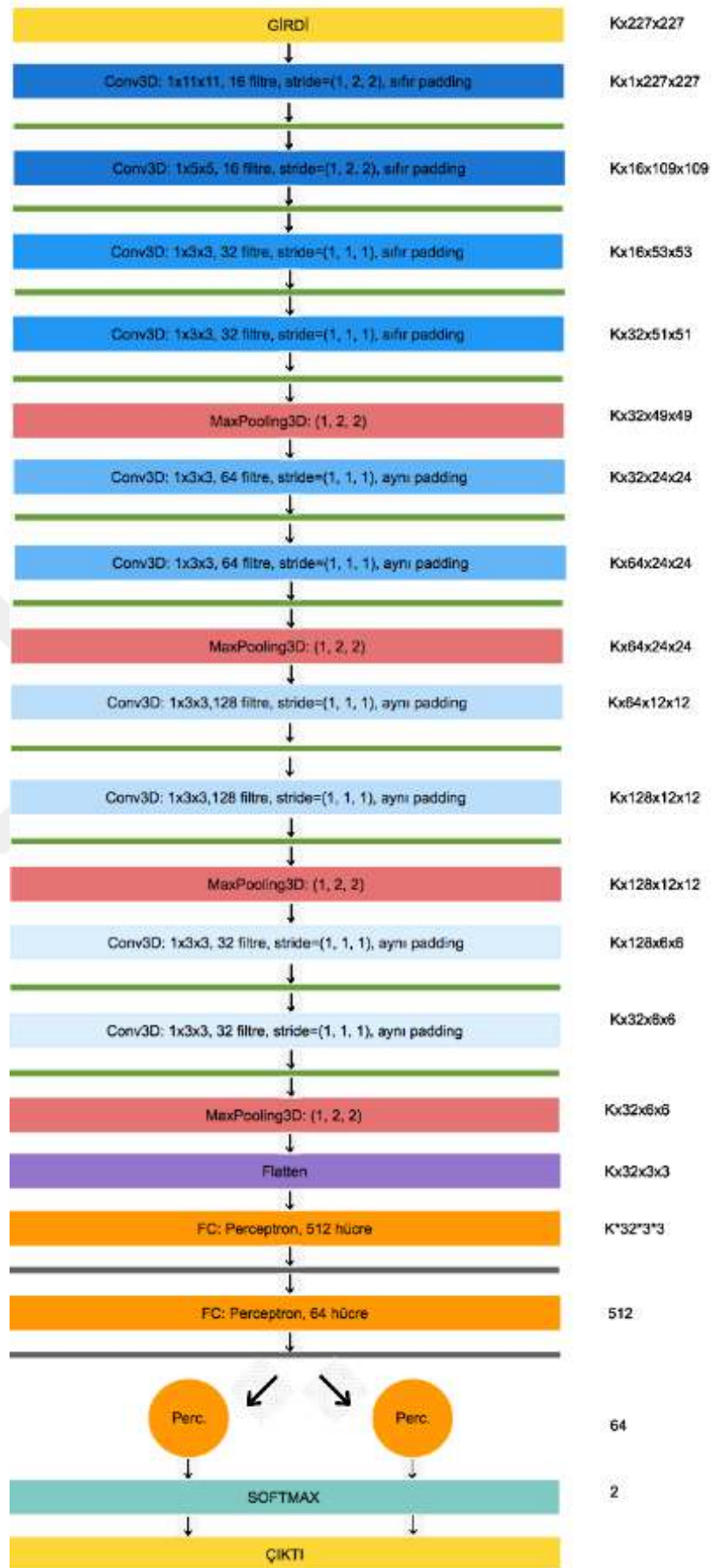
Gözetimli derin öğrenme modellerinin sözde kodu (Pseudocode) ise şu şekildedir:

Algoritma 2: Eğitmenli Öğrenme Modeli Eğitme ve Test Etme

```
Girdi:  $x_{train}, y_{train}, x_{val}, y_{val}, x_{test}, y_{test}$   
Çıktı:  $m_{best}, metrikler$   
 $m \leftarrow model\ olustur$   
 $m.opt \leftarrow Adam$   
 $m.loss \leftarrow Kategorik\ Çapraz\ Entropi$   
 $m.lr \leftarrow 0.00001$   
 $m.w \leftarrow Normal(0, 0.05)$   
 $epokSayısı \leftarrow 100$   
 $m_{best} \leftarrow m$   
 $valAcc_{best} \leftarrow 0$   
 $patience \leftarrow 0$   
for  $i$  from 1 to  $epokSayısı$  do  
   $m.fit(x_{train}, y_{train}, batchSize = 32)$   
   $valAcc \leftarrow m.evaluate(x_{val}, y_{val})$   
  if  $valAcc_{best} < valAcc$  then  
     $m_{best}.w \leftarrow m.w$   
     $valAcc_{best} \leftarrow valAcc$   
     $patience \leftarrow 0$   
  else  
     $patience \leftarrow patience + 1$   
    if  $patience == 10$  then  
      break  
    end  
  end  
end  
 $m.w \leftarrow m_{best}.w$   
 $metrikler \leftarrow m.evaluate(x_{test}, y_{test})$ 
```

C3D: Gözetimsiz öğrenme modeli olarak anomali tespitindeki başarısından dolayı ConvLSTM-AE kullanılmıştır. Gözetimli öğrenme modeli olarak ise kavga veri setindeki yüksek performansından dolayı C3D tercih edilmiştir. Model gri tonlamalı videodan elde edilen ve boyutları $N \times 227 \times 227$ olan 3 boyutlu girdi almaktadır. Buradaki N çerçeve sayısıdır. Model geçerleme sonucu geliştirilen C3D modeli Şekil 53'te gösterilmektedir. Kullanılan diğer değişkenler ve değerleri şunlardır:

- Hata Fonksiyonu: Çapraz entropy
- Eniyileştirici: Adam
- Öğrenme Oranı: 0.00001
- Mini-Yığın Büyüklüğü = $360 / (\text{girdi çerçeve sayısı})$



Şekil 53: C3D Modeli (Yeşil Katman: Tanh Katmanı, Gri Katman: ReLu Katmanı)

4.3. Geçerlik ve Güvenirlik

Çalışmanın görünüş ve kapsam geçerliliğini artırmak amacıyla alan uzmanlarının görüşlerine yer verilmiştir. Güvenirliği artırmak amacıyla sistematik haritalama aracılığıyla içerik ve veri analizinde çeşitli kodlar kullanılmış, modeller oluşturulmuştur. Veri analizine farklı kişiler dâhil edilerek yansızlığın ve iç güvenilirliğin artırılması amaçlanmıştır. Veri kaydı için Google Drive’da ve Microsoft OneDrive’da bir dizin oluşturulmuştur. Dış güvenirliği artırmak için araştırma süreci ayrıntılı planlanmış, veri toplama süreci detaylandırılarak kontrol edilmiş, kayıtlar ayrıntılı ve diğer araştırmacılara açık olacak biçimde bulut dizininde tutulmuştur.

4.4. Çalışmanın Sınırlılıkları

Video ve resimler çok sayıda pikselden oluşan büyük verilerdir. Bu yüzden hızlı eğitim için RAM büyüklüğü de GPU performansı kadar olmasa da önemlidir. Suç veri setinin sabit diskteki toplam boyutu yaklaşık 300 megabayt iken Python’da görüntüleri açınca, veri 32 gigabaytlık RAM’e sığmamaktadır. Sabit diskten veri okumak RAM’e göre yaklaşık 50 kat, sabit diske veri yazmak ise yaklaşık 200 kat daha yavaştır (Storagereview, 2020). Derin öğrenme için özel tasarlanan bilgisayarlar almak veya ücretli bulut hizmetinden faydalanmak için araştırma kaynakları yeterli olmadığı için videolar Google Colab’ın ücretsiz versiyonunda OpenCV kullanılarak resim çerçeveleri şeklinde depolanmıştır. Bu hizmette kullanımına izin verilen GPU ve RAM hizmeti de sınırlıdır. Veri seti büyüklüğü arttıkça eğitilen modelin test ve genellebilirlik performansı artmaktadır. Fakat yukarıdaki depolama nedenlerinden dolayı veri seti büyüklüğü 617 video ile sınırlı tutulmuştur. Literatürdeki anomali ve kavgı veri setleri büyüklükleri düşünüldüğünde 617 video verisinin literatüre katkı sağlamak ve suç tespitinin mümkün olduğunu göstermek için yeterli olacağı değerlendirilerek araştırma bu şekilde sınırlandırılmıştır.

Model eğitilirken geri yayılım ile veri setindeki her veriden çıkarımlar yapmayı öğrenir. Dolayısıyla her videodaki davranış, nesne, ortamdaki ışık ve renk miktarı gibi unsurlar öğrenilir. Bundan dolayı kaliteli, kalitesiz kameradan alınan; az ışıklı-çok ışıklı ortamda çekilen; çok insanlı-az insanlı ortamda çekilen; dar açıdan-geniş açıdan bakan kameradan alınan birbirinden farklı videolardan veri seti oluşturmak önemlidir. Araştırma sürecinde olabildiğince çeşitli ortam ve davranış videoları toplanmaya çalışılmıştır. Araştırmanın iç geçerliliğini etkileyebilecek değişkenler için uzman görüşlerine başvurulmuş ve ilgili çalışmalar incelenerek eksiklerin giderilmesi hedeflenmiştir.

Arařtırmadaki bir dięer sınırlılık ise geliřtirilen modelin kapalı alanlara yönelik olmasıdır.



5. TARTIŞMA

İkinci araştırma sorusuna cevap bulmak için geliştirilen derin öğrenme modelleri test edilmiş, karşılaştırılmış ve değerlendirilmiştir. Ayrıca farklı nitelikteki modele girdi verme yöntemleri ve eğitme yöntemleri, karar verme algoritmaları, performans değerlendirme yöntemleri ile detaylı analizler yapılarak daha doğru kararlara ulaşılması amaçlanmıştır. Araştırma için gerçekleştirilen uygulama sonuçları ve bunların değerlendirilmesi ile elde edilen ikinci araştırma sorusuna ilişkin bulgular sonraki bölümlerde sıralanmaktadır.

5.1. Tek Girdi ve Çoklu Girdi Yöntemlerinin Performanslarının Karşılaştırılması

Bu deneyde 4.2.4.1 ve 4.2.4.2. bölümlerdeki yöntemler kullanılarak girdi oluşturma yöntemleri, karar verme yöntemleri ve bunların kombinasyonlarının suç tespiti modelindeki performansları analiz edilmiştir. Her yöntem ile C3D modeli eğitilmiş, test veri seti ile suç tespiti yapılmıştır. Tablo 9’da görüldüğü gibi çerçeve sayısından bağımsız olarak model tek girdi ile eğitildiğinde dinamik örnekleme ile elde edilen girdiler beklenildiği gibi çözüm başarısını artırmaktadır. Tablo 9 ve 10 birlikte incelendiğinde ise, çoklu girdi ile model eğitmenin performansı artırdığı görülmektedir.

Çerçeve Sayısı	Doğruluk (EAÖ)	Doğruluk (DÖ)	AUC (EAÖ)	AUC (DÖ)
10	0.623	0.647	0.683	0.675
12	0.628	0.657	0.678	0.713
15	0.638	0.657	0.678	0.715
20	0.657	0.662	0.693	0.716

Tablo 9: Eşit Aralıklı Örnekleme (EAÖ) ve Dinamik Örnekleme (DÖ) ile Girdi Oluşturma Yöntemlerinin Performans Karşılaştırması

Tablo 10 incelendiğinde karar verme algoritmaları hakkında önemli çıkarımlar yapılmaktadır. Tek girdi ile karar vermenin, eğitmede olduğu gibi iyi bir seçenek olmadığı gözlenmektedir. İncelenen veri arttıkça elde edilen çıkarımlar artacağından daha başarılı tespitler yapmak mümkün olabilmektedir. Bu nedenle çoklu girdilerle kullanan dört algoritmanın daha başarılı sonuçlar verdiği gözlenmektedir.

Girdi Çerçeve Sayısı	10	12	15	20
Eşit Aralıklı Örnekleme	0.633	0.643	0.647	0.671
Dinamik Örnekleme	0.647	0.651	0.657	0.676
Suç Tespiti Yapılan Çerçeve Sayısı	0.671	0.682	0.676	0.681
Ortalama Suç Tespiti Olasılığı	0.666	0.686	0.681	0.700
Çerçeve Sayısı ve Ortalama Olasılık	0.666	0.686	0.700	0.681
Çerçeve Sayısı veya Ortalama Olasılık	0.671	0.682	0.705	0.700
Tümleştirme (MLP)	0.676	0.686	0.710	0.705
Tümleştirme (LSTM)	0.682	0.705	0.696	0.710

Tablo 10: Çoklu Girdi ile Eğitilen Modellerin Girdi Çerçeve Sayısına ve Karar Verme Algoritmasına Göre Performansları (Doğruluk)

Tablo 10’da tümleştirme yönteminin suçu belirlemede genel olarak en iyi karar verme yöntemi olduğu gözlenmektedir. Bu yöntemde suç tespiti için ikinci bir model geliştirildiğinde diğer yöntemlerin sınırlılıklarından dolayı öğrenemediği çıkarımını yapmak mümkündür. Bu sonuçlar doğrultusunda sonraki deneylerde girdi büyüklüğü olarak 20 kullanılmıştır. Gözetimsiz öğrenme yönteminde ise *çerçeve sayısı veya ortalama olasılık* algoritması ile suç tespiti yapılmış ve sonuçlar Ek 3’te bulunan Tablo 22’de detaylı olarak gösterilmektedir. Gözetimli öğrenme yöntemlerinde ise tümleştirme yönteminin MLP ve LSTM’li versiyonları için ayrı modeller geliştirilmiş ve test edilmiştir. Bu modellerin sonuçları ise detaylı olarak Ek 3’te bulunan Tablo 20 ve 21’de sunulmaktadır.

5.2. Gözetimsiz Öğrenme Modeli Performansını Geliştirme

Literatürde özkodlayıcı ile geliştirilen anomali tespit model eğitim yöntemlerinin standart olduğu söylenebilir. Ancak uzaklık alma yöntemi, aralarındaki uzaklıkları alınan matrisler, eşik değeri belirleme stratejisi ve karar verme stratejisi değiştirilerek suç tespiti performansı artırılabilir. Normal veri seti ile eğitilen ConvLSTM-AE modelini performansı aşağıda anlatılan yöntemlerle iyileştirilmiştir.

5.2.1. Uzaklık hesaplamadan önce havuz katmanı kullanma

Girdi ve çıktı aslında normalize edilmiş gri tonlamalı çerçeve dizisidir. Dolayısıyla yüzbinlerce piksel vardır. Bu durum girdi ile çıktı arasındaki uzaklığı gürültüye karşı hassaslaştırmaktadır. Bu yüzden girdi ve çıktıyı havuz katmanından geçirerek boyutları küçültüldükten sonra fark hesaplayarak suç tespiti yapmanın daha başarılı sonuçlar verdiği araştırma süresince test edilip görülmüştür. Geliştirilen gözetimsiz öğrenme modellerinde fark hesaplamadan önce girdi ve çıktı 1x2x2 boyutlu maksimum havuz katmanından iki defa geçirilmektedir.

5.2.2. Eğitimden suç verisini kullanma

Anomali tespitinde genel olarak normal veri sayısı çok iken anormal veri sayısı azdır. Özkodlayıcı içeren modele normal durum öğretilir. Yeni bir veri geldiğinde model ile normal durum ve anomali tespiti yapılır. Bu problem için anormal durum suç demektir. Bu geliştirilen 1. ConvLSTM-AE modelidir. Suç veri setinde suç verisi miktarı normal anomali tespiti problemlerine kıyasla normal veri sayısına yakındır. Dolayısıyla bu deneyde suç verileri ile 2. Bir ConvLSTM-AE modeli eğitilmiştir. 2. Model için normal durum suç, anomali ise ortamda suç olmaması anlamına gelmiş oldu. Test sırasında iki model birlikte kullanılarak suç analizi yapılmıştır. Bu süreç şu şekilde açıklanabilir:

- 1. Model eğitildi (normal=suç yok, anomali=suç var).
- 2. Model eğitildi (normal=suç var, anomali=suç yok).
- Bir video iki modele ayrı ayrı verildi ve girdi ile çıktı arasındaki fark hesaplandı.
- 1. modelden elde edilen fark 2. modelden elde edilen farktan küçük ise, videodaki davranış 1. modelin normal durumuna daha yakın demektir yani suç yok kararı verildi.
- 2. modelden elde edilen fark 1. modelden elde edilen farktan küçük ise, videodaki davranış 2. modelin normal durumuna daha yakın demektir yani suç var kararı verildi.

Yukarıda anlatılan yöntemle iki öğrenmesiz öğrenme modeli kullanarak suç tespiti yapmanın başarısı Ek 3'te bulunan Tablo 23'te detaylı olarak görülebilir. Tablo 12'den görülebileceği gibi geliştirilen suç tespiti algoritmasının performansı artırılabilmiştir.

5.3. Gözetimli ve Gözetimsiz Öğrenme Modellerin Performanslarının

Karşılaştırılması

Suç tespiti problemi anomali tespiti problemi olduğundan gözetimsiz öğrenme ile çözüm modeli geliştirmenin doğru bir strateji olduğu sonucuna varılmıştır. Ancak literatürde kavga tespiti, gözetimli öğrenme ile çözülmektedir. Bu sebeple suç tespitinde gözetimli ve gözetimsiz öğrenme yöntemlerinin performanslarının karşılaştırılmasına karar verilmiştir. C3D ve ConvLSTM-AE modellerinin suç tespitinde en başarılı versiyonlarının anomali tespitindeki başarıları Tablo 11’de karşılaştırılmıştır. Veri setindeki 8 farklı anomali suç verisi ve normal veri sayısı birbirine yakındır. Ancak anomali sınıflarındaki veri sayısı ile normal veri sayısı arasındaki fark gerçeğe yakındır. Dolayısıyla önceki deneylerde doğruluk ölçütüne göre çıkarımlar yapılmıştır. Suç verisini alt kategorilere ayırınca her anomali sınıfının veri sayısı ile normal veri sayısı arasındaki fark önemli derecede artmaktadır. Bu açıdan her alt kategori grubuna ait verilerin suç tespiti başarısı ölçülmek istenildiğinde literatürde anomali modellerinin performans ölçütü olarak sıkça kullanılan AUC değerleri incelenmiştir.

Anomali Türü	Tümleştirmeli C3D (MLP)	Tümleştirmeli C3D (LSTM)	ConvLSTM-AE	ConvLSTM-AE (2 Modelli)
Eylemsizlik	0.735	0.714	0.744	0.744
Yerde Uzanma	0.697	0.678	0.695	0.695
El Kaldırma	0.750	0.738	0.769	0.744
Silah Doğrultma	0.728	0.734	0.747	0.735
Diğer	0.782	0.795	0.794	0.797
Atlama	0.639	0.617	0.648	0.631
Kavga	0.622	0.588	0.618	0.614
Koşma	0.627	0.635	0.634	0.643
Suç	0.717	0.715	0.728	0.724

Tablo 11: Gözetimli ve Gözetimsiz Öğrenme Modellerin Anomali Tespiti Performansı (AUC)

Tablo 11’deki sonuçlar çıkarılmadan önce modeller normal ve suç olmak üzere iki grup veri ile yukarıda anlatılan standart yöntemlerle eğitilmiştir. Test edilirken bir anomali grubu verileri ve normal verilerin tamamı alınmıştır. Modelin bu verilere karşın verdiği

normal ve suç kararları alınmıştır. Sonra sadece normal verilerin ve belirlenen anomali verilerinin olduğu bir ortamda modelin anomali gurubuna ait suç tespiti başarısı ölçülmüştür. Tablo 11’de anomali gruplarının tespitinde gözetimsiz öğrenme modelleri daha yüksek performans göstermektedir. Bunun sebebinin gözetimli öğrenme modellerinin yüksek performans göstermesi için kategorilerdeki veri sayılarının birbirine yakın olmasının gerektiği düşünülmektedir. Gözetimsiz öğrenme için böyle bir zorunluluk yoktur. Sadece çok sayıda normal veri gerekmektedir. Tablo 12’den görülebileceği gibi suç tespitinde gözetimsiz öğrenme yöntemleri daha başarılı sonuçlar vermektedir.

Metrik	Tümleştirmeli C3D (MLP)	Tümleştirmeli C3D (LSTM)	ConvLSTM-AE	ConvLSTM-AE (2 Modelli)
AUC	0.717	0.715	0.728	0.724
Doğruluk	0.705	0.710	0.720	0.729

Tablo 12: Gözetimli ve Gözetimsiz Öğrenme Modellerin Suç Tespiti Performansı

6. SONUÇ VE ÖNERİLER

Bu çalışmanın temel amacı, kapalı alanlarda güvenlik kameralarından (CCTV) elde edilen video görüntülerini kullanarak suç davranışlarını gerçek zamanlı belirlemeye yönelik derin öğrenme yöntem ve tekniklerine dayalı akıllı bir çözüm üretmektir. Bunlara ek olarak suç tespitine yönelik 617 videodan oluşan bir veri seti literatüre kazandırılmıştır. Veri seti üzerinde yapılan çalışma ve değerlendirmelerde doğruluk (accuracy) ölçütü göz önünde bulundurulmuştur. Araştırmanın hedefleri doğrultusunda en iyi performans gösteren model doğruluk, AUC ve FPR ölçütleriyle yorumlanarak, derin öğrenme yönteminin problem çözümündeki başarısı değerlendirilmiştir.

Tablo 13'te görüldüğü gibi araştırma sonunda araştırma hedef ölçütlerini gerçekleştiren derin öğrenme modeli geliştirilmiştir. FPR değerine göre ortalama yaklaşık beş normal videodan biri suç olarak tanımlanmaktadır. Yöntem bölümünde belirtildiği gibi modelin geliştirilmesi ve değerlendirilmesinde iki veya üç saniyeden oluşan video görüntüleri kullanılmış, yaklaşık on beş saniye aralıklarda çeşitli uyarılar alınmıştır. Geliştirilen algoritma ve model yardımıyla sekiz kategoride sınıflanan suç davranışları %73 düzeyinde doğru olarak tespit edilebilmektedir. Ancak bu durumun, geliştirilen akıllı çözümün gerçek hayatta kullanılabilirliğine ve ticari ürüne dönüştürülmesine kısıtlamalar getirdiğini de belirtmek gerekmektedir.

Model	AUC	FPR	Doğruluk
ConvLSTM-AE	0.728	0.211	0.720
ConvLSTM-AE (2 Modelli)	0.724	0.184	0.729

Tablo 13: Gözetimsiz Öğrenme Modellerin Suç Tespiti Performansı

Elde edilen bir diğer önemli bulgu ise geliştirilen modelin farklı anomali sınıflarında farklı performanslar sergilemesidir. Örneğin koşma, kavga ve silah doğrultmada yüksek başarıyla performans gösterirken; eylemsizlik, yerde uzanma, atlama kategorilerinde ise düşük performans göstermiştir. Bu durumun, literatürdeki çalışmalardan farklı olarak araştırmada kullanılan veri setindeki anomali kategorilerinin karmaşıklığı, çok farklı davranış senaryolarının bulunması, veri setinin büyüklüğü vb. faktörlerden kaynaklanabileceğini söylemek mümkündür.

Bu çalışmada elde edilen bulgular ve gözlemler doğrultusunda, derin öğrenme kapsamında geliştirilen çözümün araştırma amaç ve hedeflerini karşıladığı ve etkili olduğunu söylemek mümkündür. Ancak, araştırma sınırlılıkları dikkate alındığında iyileştirilmesi gereken yönleri de bulunmaktadır. Bu doğrultuda gelecekte yapılması planlanan çalışmalar aşağıdaki gibidir:

- Güvenlik Kamerası Suç Veri Setinin veri büyüklüğünün artırılmaya devam edilmesi,
- Suç tespiti modelinin performansının kamera uzaklığı, insan sayısı gibi ortam koşullarına göre tekrar değerlendirilmesi ve bu doğrultuda çözümün iyileştirilmesi,
- Anomalileri tespit eden ayrı modeller tasarlanıp sonuçlarını birlikte değerlendirerek daha detaylı ve geniş yelpazede suç tespitinin yapılması,
- Anomali türlerine özgü yeni modellerin geliştirilmesi,
- Eylemsizlik anomali türü başta olmak üzere ortam analizini kuvvetlendirmek için geliştirilen model ve nesne tanıma modelini birlikte barındıran yeni bir hibrit suç tespiti modelinin geliştirilmesi,
- Two-stream CNN başta olmak üzere görüntülerdeki zamansal ve uzaysal çıkarımları ayrı kollarla gerçekleştiren tümleştirme (fusion) modellerinin geliştirilmesidir.

Araştırma raporu, bu çalışmada geliştirilen modeller, elde edilen bulgular ve gözlemler, araştırma sınırlılıkları ve önerilerin ışığı altında yeni çalışmaları yapılması çağrısıyla son bulmaktadır.

EKLER

EK-1: Bir Girdi ile Eğitim Performansları

Çerçeve Sayısı	Hassasiyet	Kesinlik	Doğruluk	AUC	Ortalama Test Süresi (ms)
10	0.589	0.730	0.623	0.683	19.2
12	0.605	0.728	0.628	0.678	21.2
15	0.605	0.743	0.638	0.678	26.5
20	0.831	0.673	0.657	0.693	29.6

Tablo 14: Eşit Aralıklı Örnekleme ile Bir Videodan Bir Girdi Oluşturularak Eğitilen ve Test Edilen C3D Modelinin Girdi Çerçeve Sayısına Göre Suç Tespiti Performansı

Çerçeve Sayısı	Hassasiyet	Kesinlik	Doğruluk	AUC	Ortalama Test Süresi (ms)
10	0.719	0.644	0.647	0.675	20.5
12	0.839	0.671	0.657	0.713	21.8
15	0.831	0.673	0.657	0.715	25.7
20	0.661	0.745	0.662	0.716	33.1

Tablo 15: Dinamik Örnekleme ile Bir Videodan Bir Girdi Oluşturularak Eğitilen ve Test Edilen C3D Modelinin Girdi Çerçeve Sayısına Göre Suç Tespiti Performansı

EK-2: Çoklu Girdi ile Eğitim Performansları

Suç Kararı Yöntemi	Hassasiyet	Kesinlik	Doğruluk	AUC	Ortalama Test Süresi (ms)
Eşit Aralıklı Örnekleme	0.621	0.726	0.633	0.699	21.3
Dinamik Örnekleme	0.645	0.734	0.647	0.699	21.9
Suç Tespiti Yapılan Çerçeve Sayısı	0.887	0.671	0.671	0.618	52.2
Ortalama Suç Tespiti Olasılığı	0.710	0.765	0.666	0.692	52.2
Çerçeve Sayısı ve Ortalama Olasılık	0.710	0.765	0.666	0.692	52.2
Çerçeve Sayısı veya Ortalama Olasılık	0.887	0.671	0.671	0.618	52.2
Tümleştirme (MLP)	0.790	0.705	0.676	0.709	63.6
Tümleştirme (LSTM)	0.708	0.748	0.682	0.728	65.5

Tablo 16: Girdi Çerçeve Sayısı 10 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı

Suç Kararı Yöntemi	Hassasiyet	Kesinlik	Doğruluk	AUC	Ortalama Test Süresi (ms)
Eşit Aralıklı Örnekleme	0.729	0.736	0.643	0.722	22.2
Dinamik Örnekleme	0.718	0.742	0.651	0.714	22.6
Suç Tespiti Yapılan Çerçeve Sayısı	0.708	0.748	0.682	0.728	51.8
Ortalama Suç Tespiti Olasılığı	0.766	0.736	0.686	0.678	51.8
Çerçeve Sayısı ve Ortalama Olasılık	0.766	0.736	0.686	0.678	51.8
Çerçeve Sayısı veya Ortalama Olasılık	0.708	0.748	0.682	0.728	51.8
Tümleştirme (MLP)	0.742	0.736	0.686	0.745	62.2
Tümleştirme (LSTM)	0.823	0.723	0.705	0.698	63.7

Tablo 17: Girdi Çerçeve Sayısı 12 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı

Suç Kararı Yöntemi	Hassasiyet	Kesinlik	Doğruluk	AUC	Ortalama Test Süresi (ms)
Eşit Aralıklı Örnekleme	0.737	0.738	0.647	0.727	24.3
Dinamik Örnekleme	0.831	0.673	0.657	0.715	25.4
Suç Tespiti Yapılan Çerçeve Sayısı	0.847	0.686	0.676	0.634	55.5
Ortalama Suç Tespiti Olasılığı	0.855	0.688	0.681	0.638	55.5
Çerçeve Sayısı ve Ortalama Olasılık	0.718	0.767	0.700	0.696	55.5
Çerçeve Sayısı veya Ortalama Olasılık	0.726	0.769	0.705	0.700	55.5
Tümleştirme (MLP)	0.847	0.719	0.710	0.754	62.8
Tümleştirme (LSTM)	0.694	0.775	0.696	0.786	64.5

Tablo 18: Girdi Çerçeve Sayısı 15 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı

Suç Kararı Yöntemi	Hassasiyet	Kesinlik	Doğruluk	AUC	Ortalama Test Süresi (ms)
Eşit Aralıklı Örnekleme	0.731	0.696	0.671	0.644	33.0
Dinamik Örnekleme	0.766	0.714	0.676	0.715	33.5
Suç Tespiti Yapılan Çerçeve Sayısı	0.831	0.696	0.681	0.644	69.8
Ortalama Suç Tespiti	0.847	0.719	0.700	0.676	69.8
Çerçeve Sayısı ve Ortalama Olasılık	0.831	0.696	0.681	0.644	69.8
Çerçeve Sayısı veya Ortalama Olasılık	0.847	0.719	0.700	0.676	69.8
Tümleştirme (MLP)	0.831	0.720	0.705	0.717	72.9
Tümleştirme (LSTM)	0.863	0.713	0.710	0.715	74.5

Tablo 19: Girdi Çerçeve Sayısı 20 ve Çoklu Girdi ile Eğitilen C3D Modelinin Karar Verme Yöntemine Göre Suç Tespiti Performansı

EK-3: Modellerin Girdi Büyüklüğü 20 iken Anomali Tespiti Performansları

Anomali Türü	Hassasiyet	Kesinlik	Doğruluk	AUC
Eylemsizlik	0.857	0.130	0.544	0.735
Yerde Uzanma	0.857	0.130	0.544	0.697
El Kaldırma	0.857	0.130	0.544	0.750
Silah Doğrultma	0.879	0.420	0.621	0.728
Diğer	0.919	0.459	0.642	0.782
Atlama	0.600	0.070	0.523	0.639
Kavga	0.733	0.216	0.551	0.622
Koşma	0.615	0.167	0.531	0.627
Suç	0.831	0.720	0.705	0.717

Tablo 20: C3D MLP Tümleştirme Modelinin Anomali Tespiti Performansı

Anomali Türü	Hassasiyet	Kesinlik	Doğruluk	AUC
Eylemsizlik	1.000	0.140	0.522	0.714
Yerde Uzanma	0.857	0.122	0.511	0.678
El Kaldırma	0.857	0.122	0.511	0.738
Silah Doğrultma	0.909	0.411	0.603	0.734
Diğer	0.946	0.449	0.625	0.795
Atlama	0.600	0.065	0.489	0.617
Kavga	0.733	0.204	0.520	0.588
Koşma	0.692	0.173	0.510	0.635
Suç	0.863	0.713	0.710	0.715

Tablo 21: C3D LSTM Tümleştirme Modelinin Anomali Tespiti Performansı

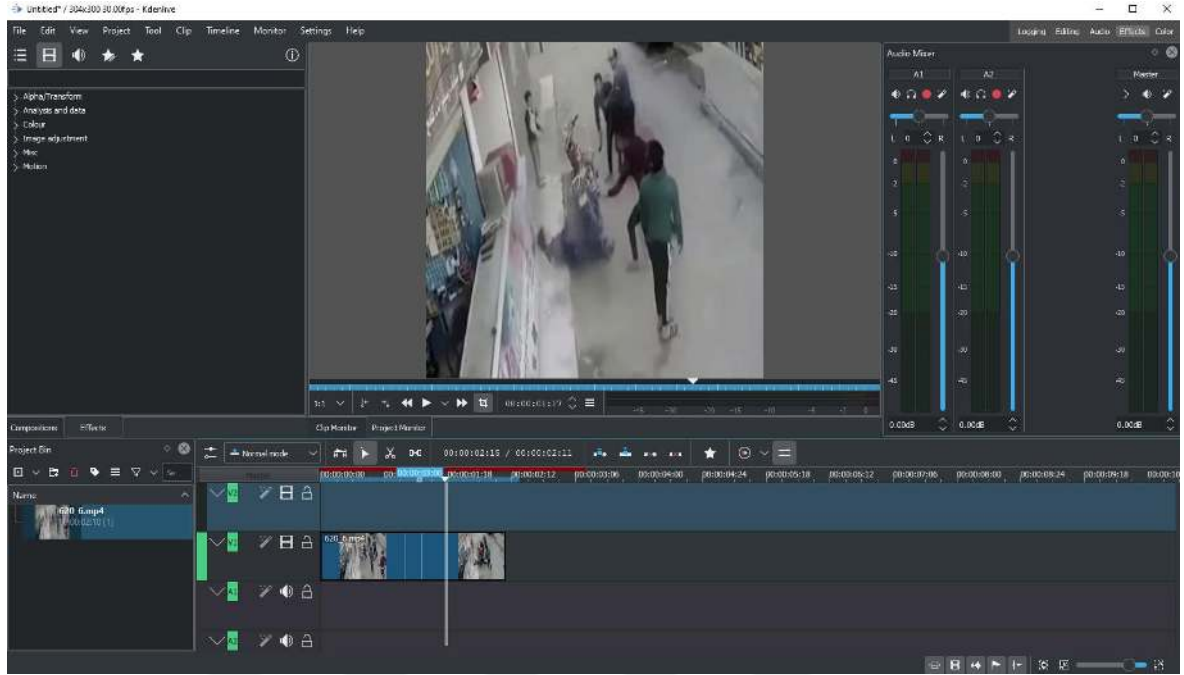
Anomali Türü	Hassasiyet	Kesinlik	Doğruluk	AUC
Eylemsizlik	0.857	0.133	0.556	0.744
Yerde Uzanma	0.857	0.133	0.556	0.695
El Kaldırma	0.857	0.133	0.556	0.769
Silah Doğrultma	0.909	0.435	0.638	0.747
Diğer	0.919	0.466	0.650	0.794
Atlama	0.600	0.071	0.534	0.648
Kavga	0.733	0.220	0.561	0.618
Koşma	0.692	0.188	0.552	0.634
Suç	0.847	0.729	0.720	0.728

Tablo 22: ConvLSTM-AE Modelinin Anomali Tespiti Performansı

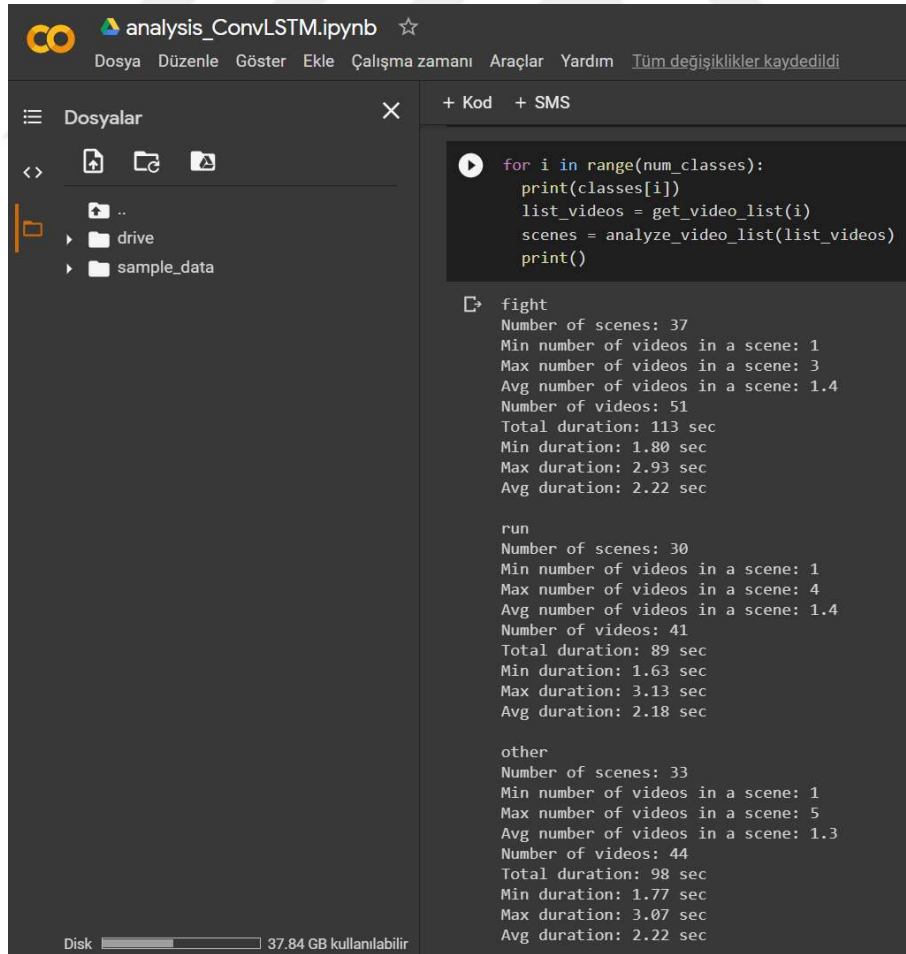
Anomali Türü	Hassasiyet	Kesinlik	Doğruluk	AUC
Eylemsizlik	0.857	0.133	0.556	0.744
Yerde Uzanma	0.857	0.133	0.556	0.695
El Kaldırma	0.857	0.133	0.556	0.744
Silah Doğrultma	0.909	0.435	0.638	0.735
Diğer	0.946	0.473	0.658	0.797
Atlama	0.600	0.071	0.534	0.631
Kavga	0.561	0.561	0.561	0.614
Koşma	0.769	0.204	0.562	0.643
Suç	0.863	0.733	0.729	0.724

Tablo 23: Suç Verisi ile de Eğitilen ConvLSTM-AE Modelinin Anomali Tespiti Performansı

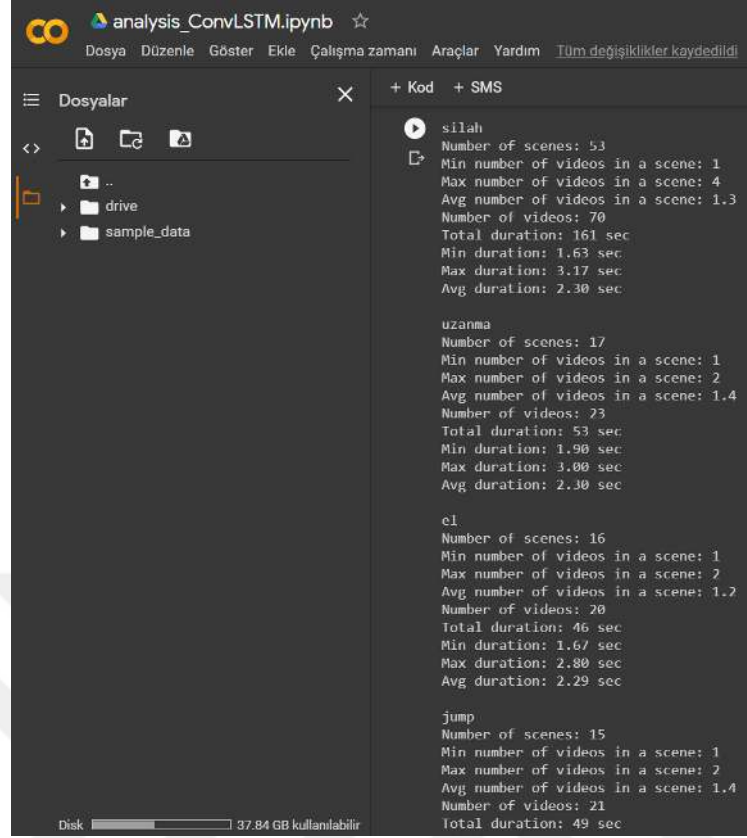
EK-4: Veri Hazırlama ve Etiketleme Ekran Görüntüleri



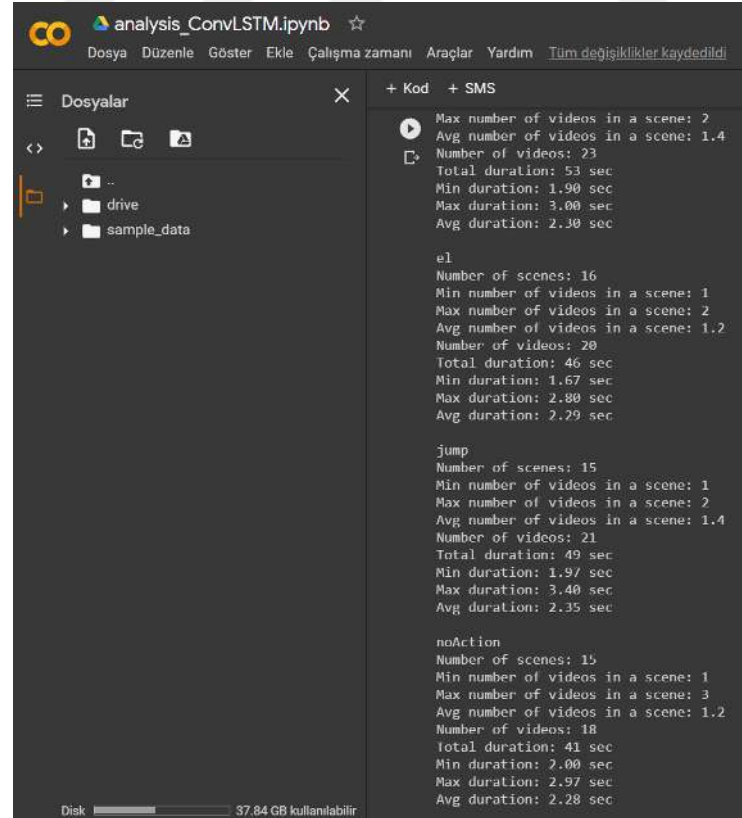
Şekil 54: Kdenlive Ekran Görüntüsü



Şekil 55: Güvenlik Kamera Suç Veri Seti Anomali Sınıfları Analizi - 1. Kısım



Şekil 56: Güvenlik Kamera Suç Veri Seti Anomali Sınıfları Analizi - 2. Kısım



Şekil 57: Güvenlik Kamera Suç Veri Seti Anomali Sınıfları Analizi - 3. Kısım

The screenshot shows a Jupyter Notebook titled 'analysis_ConvLSTM.ipynb'. The left sidebar displays a file explorer with folders 'drive' and 'sample_data'. The main area shows a code cell with the following Python code:

```
Avg duration: 2.28 sec

list_videos = get_video_list(1)
for i in range(2,num_classes):
    list_videos = get_video_list(i, list_videos)
    print('Crime')
    scenes = analyze_video_list(list_videos)
    print()

list_videos = get_video_list(0, list_videos)
print('All')
scenes = analyze_video_list(list_videos)
print()
```

The output of the code is as follows:

```
Crime
Number of scenes: 131
Min number of videos in a scene: 1
Max number of videos in a scene: 9
Avg number of videos in a scene: 2.2
Number of videos: 288
Total duration: 650 sec
Min duration: 1.63 sec
Max duration: 3.40 sec
Avg duration: 2.26 sec

All
Number of scenes: 332
Min number of videos in a scene: 1
Max number of videos in a scene: 12
Avg number of videos in a scene: 1.9
Number of videos: 617
Total duration: 1450 sec
Min duration: 1.63 sec
Max duration: 3.40 sec
Avg duration: 2.35 sec
```

The bottom status bar indicates 'Disk 37.84 GB kullanılabilir'.

Şekil 58: Güvenlik Kamera Suç Veri Seti Suç Verileri Analizi

The screenshot shows a Jupyter Notebook titled 'analysis_ConvLSTM.ipynb'. The left sidebar displays a file explorer with folders 'drive' and 'sample_data'. The main area shows a code cell with the following Python code:

```
total duration: 1450 sec
[8] Min duration: 1.63 sec
Max duration: 3.40 sec
Avg duration: 2.35 sec

print('Normal')
list_videos_normal = get_video_list(0)
scenes_normal = analyze_video_list(list_videos_normal)
print()

print('Crime')
list_videos_crime = get_video_list(1)
for i in range(2,num_classes):
    list_videos_crime = get_video_list(i, list_videos_crime)
    scenes_crime = analyze_video_list(list_videos_crime)
```

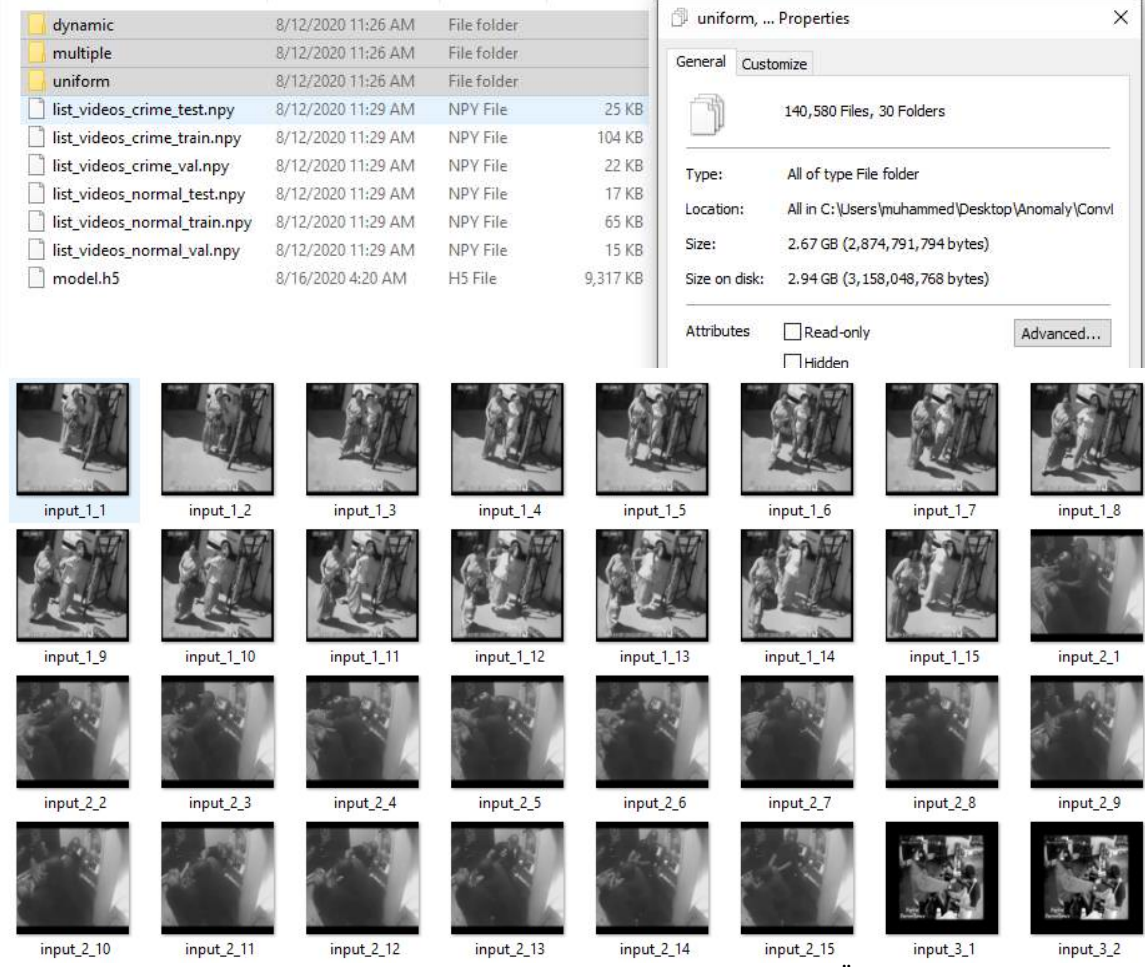
The output of the code is as follows:

```
Normal
Number of scenes: 244
Min number of videos in a scene: 1
Max number of videos in a scene: 8
Avg number of videos in a scene: 1.3
Number of videos: 329
Total duration: 800 sec
Min duration: 1.80 sec
Max duration: 3.40 sec
Avg duration: 2.43 sec

Crime
Number of scenes: 131
Min number of videos in a scene: 1
Max number of videos in a scene: 9
Avg number of videos in a scene: 2.2
Number of videos: 288
Total duration: 650 sec
Min duration: 1.63 sec
Max duration: 3.40 sec
Avg duration: 2.26 sec
```

The bottom status bar indicates 'Disk 37.84 GB kullanılabilir'.

Şekil 59: Güvenlik Kamera Suç Veri Seti Normal ve Suç Sınıfları Analizi



Şekil 60: Güvenlik Kamera Suç Veri Seti Çerçevesi Örneği

EK-5: Derin Öğrenme Modelleri Eğitim Kodları Ekran Görüntüleri

```
from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

import tensorflow.keras as ker
from tensorflow.keras.layers import Input,Dense,Flatten,Dropout,Reshape,Conv2D,MaxPooling2D,UpSampling2D,Conv2DTranspose,concatenate,add
from tensorflow.keras.layers import BatchNormalization,Conv3D,ConvLSTM2D,Conv3DTranspose,Permute,MaxPooling3D
from tensorflow.keras.models import Model,Sequential
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import cv2
import os
import time
import random
import sys
sys.path.insert(0, '/content/drive/My Drive/Colab Notebooks/Surveillance Camera Crime/')
from image_loader import *

Using TensorFlow backend.
```

Şekil 61: Derin Öğrenme Modeli Eğitimde Kullanılan Kütüphaneler

```
train_ConvLSTM.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

Dosyalar
<>
-
drive
sample_data

[7] def load_model():
    model=Sequential()
    # N 227 227
    model.add(Conv3D(filters=16, kernel_size=(1,1,1), strides=(1,2,2), padding='valid', activation='tanh',
        input_shape=(sequence_length,image_size,image_size,1)))
    # N 16 109 109
    model.add(Conv3D(filters=16, kernel_size=(1,5,5), strides=(1,2,2), padding='valid', activation='tanh'))
    # N 16 53 53
    model.add(Conv3D(filters=32, kernel_size=(1,3,3), strides=(1,1,1), padding='valid', activation='tanh'))
    model.add(Conv3D(filters=32, kernel_size=(1,3,3), strides=(1,1,1), padding='valid', activation='tanh'))
    model.add(MaxPooling3D(pool_size=(1,2,2)))
    # N 32 24 24
    model.add(Conv3D(filters=64, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(Conv3D(filters=64, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(MaxPooling3D(pool_size=(1,2,2)))
    # N 64 12 12
    model.add(Conv3D(filters=128, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(Conv3D(filters=128, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(MaxPooling3D(pool_size=(1,2,2)))
    # N 128 6 6
    model.add(Conv3D(filters=128, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(Conv3D(filters=32, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(MaxPooling3D(pool_size=(1,2,2)))
    # N 32 3 3
    model.add(Flatten())
    model.add(Dense(512, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(2, activation='softmax'))
    return model

[8] model = load_model()
adam = Adam(learning_rate=0.001)
model.compile(optimizer=adam, loss='categorical_crossentropy', metrics=['accuracy'])
```

Şekil 62: Gözetimli Derin Öğrenme Modeli Keras Kodu

train_ConvLSTM.ipynb
☆

[Dosya](#)
[Düzenle](#)
[Göster](#)
[Ekle](#)
[Çalışma zamanı](#)
[Araçlar](#)
[Yardım](#)
[Tüm değişiklikler kaydedildi](#)

Dosyalar

..

- drive
- sample_data

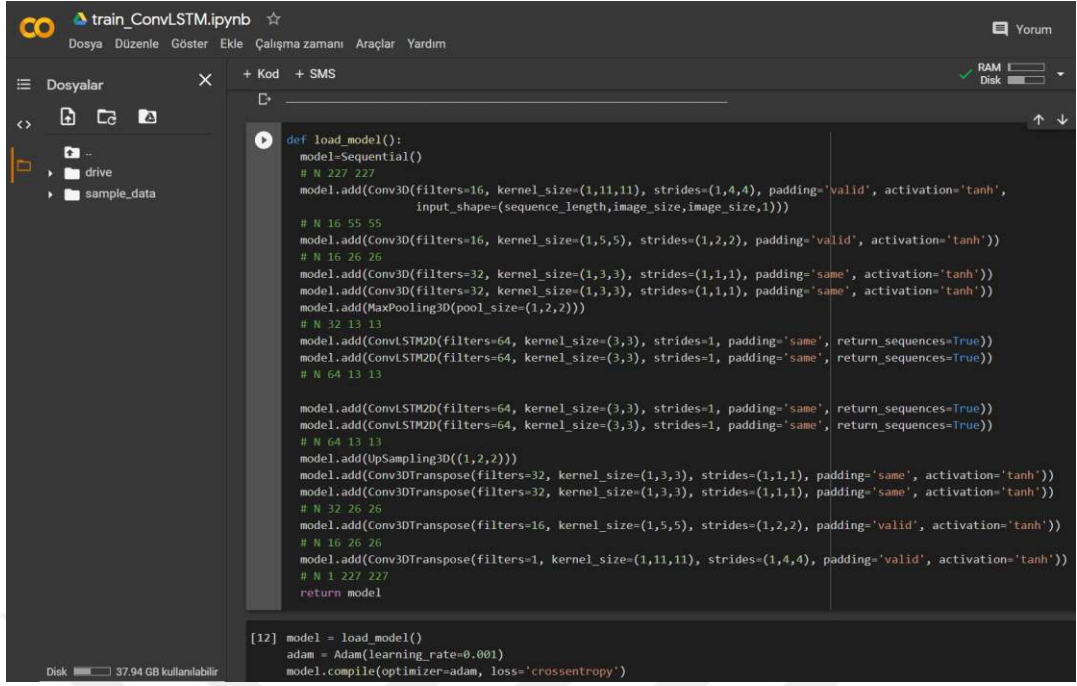
+ Kod
+ SMS

Model: "sequential"

Layer (type)	Output Shape	Param #
conv3d (Conv3D)	(None, 15, 109, 109, 16)	1952
conv3d_1 (Conv3D)	(None, 15, 53, 53, 16)	6416
conv3d_2 (Conv3D)	(None, 15, 51, 51, 32)	4640
conv3d_3 (Conv3D)	(None, 15, 49, 49, 32)	9248
max_pooling3d (MaxPooling3D)	(None, 15, 24, 24, 32)	0
conv3d_4 (Conv3D)	(None, 15, 24, 24, 64)	18496
conv3d_5 (Conv3D)	(None, 15, 24, 24, 64)	36928
max_pooling3d_1 (MaxPooling3D)	(None, 15, 12, 12, 64)	0
conv3d_6 (Conv3D)	(None, 15, 12, 12, 128)	73856
conv3d_7 (Conv3D)	(None, 15, 12, 12, 128)	147584
max_pooling3d_2 (MaxPooling3D)	(None, 15, 6, 6, 128)	0
conv3d_8 (Conv3D)	(None, 15, 6, 6, 128)	147584
conv3d_9 (Conv3D)	(None, 15, 6, 6, 32)	36896
max_pooling3d_3 (MaxPooling3D)	(None, 15, 3, 3, 32)	0
flatten (Flatten)	(None, 4320)	0
dense (Dense)	(None, 512)	2212352
dense_1 (Dense)	(None, 64)	32832
dense_2 (Dense)	(None, 2)	130
Total params: 2,728,914		
Trainable params: 2,728,914		
Non-trainable params: 0		

Disk 37.94 GB kullanılabilir

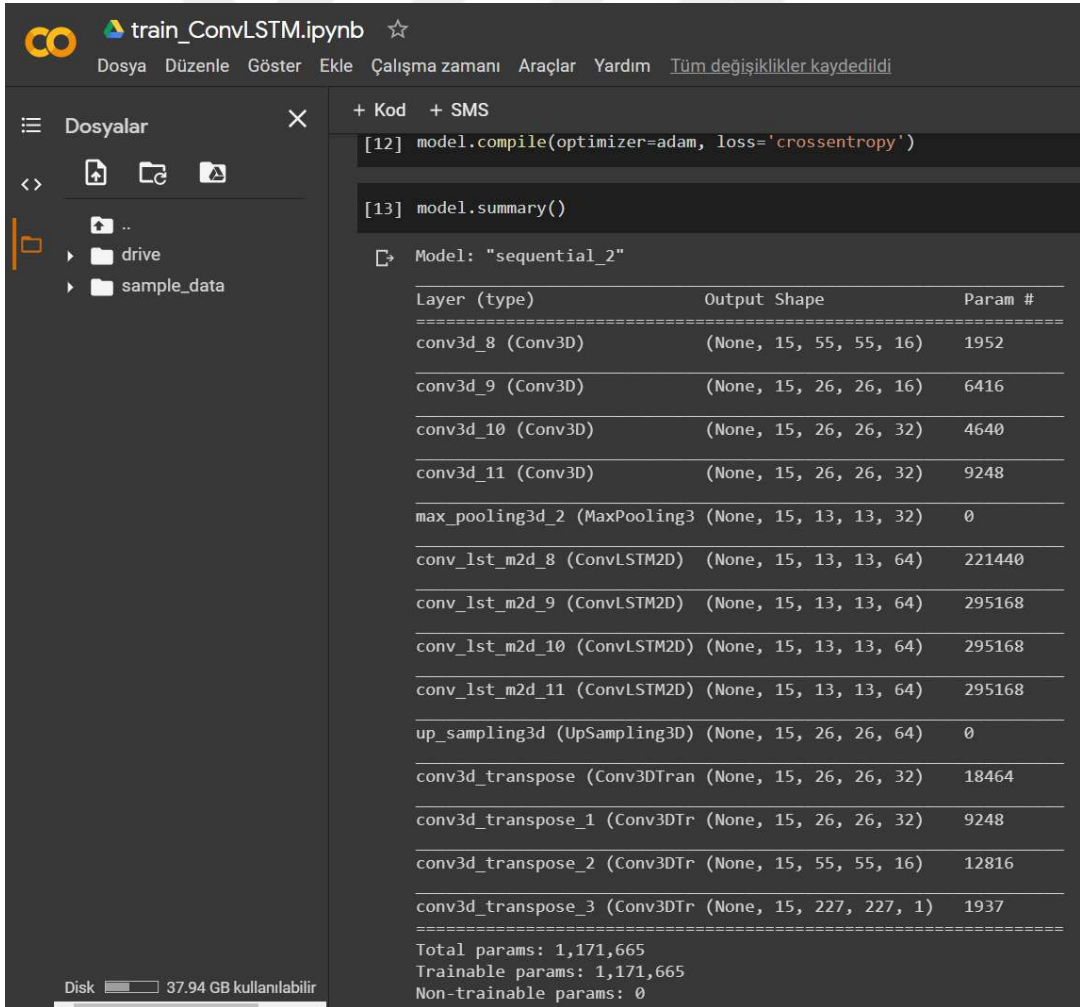
Şekil 63: Gözetimli Derin Öğrenme Modeli



```
def load_model():
    model=Sequential()
    # N 227 227
    model.add(Conv3D(filters=16, kernel_size=(1,11,11), strides=(1,4,4), padding='valid', activation='tanh',
        input_shape=(sequence_length,image_size,image_size,1)))
    # N 16 55 55
    model.add(Conv3D(filters=16, kernel_size=(1,5,5), strides=(1,2,2), padding='valid', activation='tanh'))
    # N 16 26 26
    model.add(Conv3D(filters=32, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(Conv3D(filters=32, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(MaxPooling3D(pool_size=(1,2,2)))
    # N 32 13 13
    model.add(ConvLSTM2D(filters=64, kernel_size=(3,3), strides=1, padding='same', return_sequences=True))
    model.add(ConvLSTM2D(filters=64, kernel_size=(3,3), strides=1, padding='same', return_sequences=True))
    # N 64 13 13
    model.add(ConvLSTM2D(filters=64, kernel_size=(3,3), strides=1, padding='same', return_sequences=True))
    model.add(ConvLSTM2D(filters=64, kernel_size=(3,3), strides=1, padding='same', return_sequences=True))
    # N 64 13 13
    model.add(UpSampling3D((1,2,2)))
    model.add(Conv3DTranspose(filters=32, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    model.add(Conv3DTranspose(filters=32, kernel_size=(1,3,3), strides=(1,1,1), padding='same', activation='tanh'))
    # N 32 26 26
    model.add(Conv3DTranspose(filters=16, kernel_size=(1,5,5), strides=(1,2,2), padding='valid', activation='tanh'))
    # N 16 26 26
    model.add(Conv3DTranspose(filters=1, kernel_size=(1,11,11), strides=(1,4,4), padding='valid', activation='tanh'))
    # N 1 227 227
    return model

[12] model = load_model()
adam = Adam(learning_rate=0.001)
model.compile(optimizer=adam, loss='crossentropy')
```

Şekil 64: Gözetimsiz Derin Öğrenme Modeli Keras Kodu

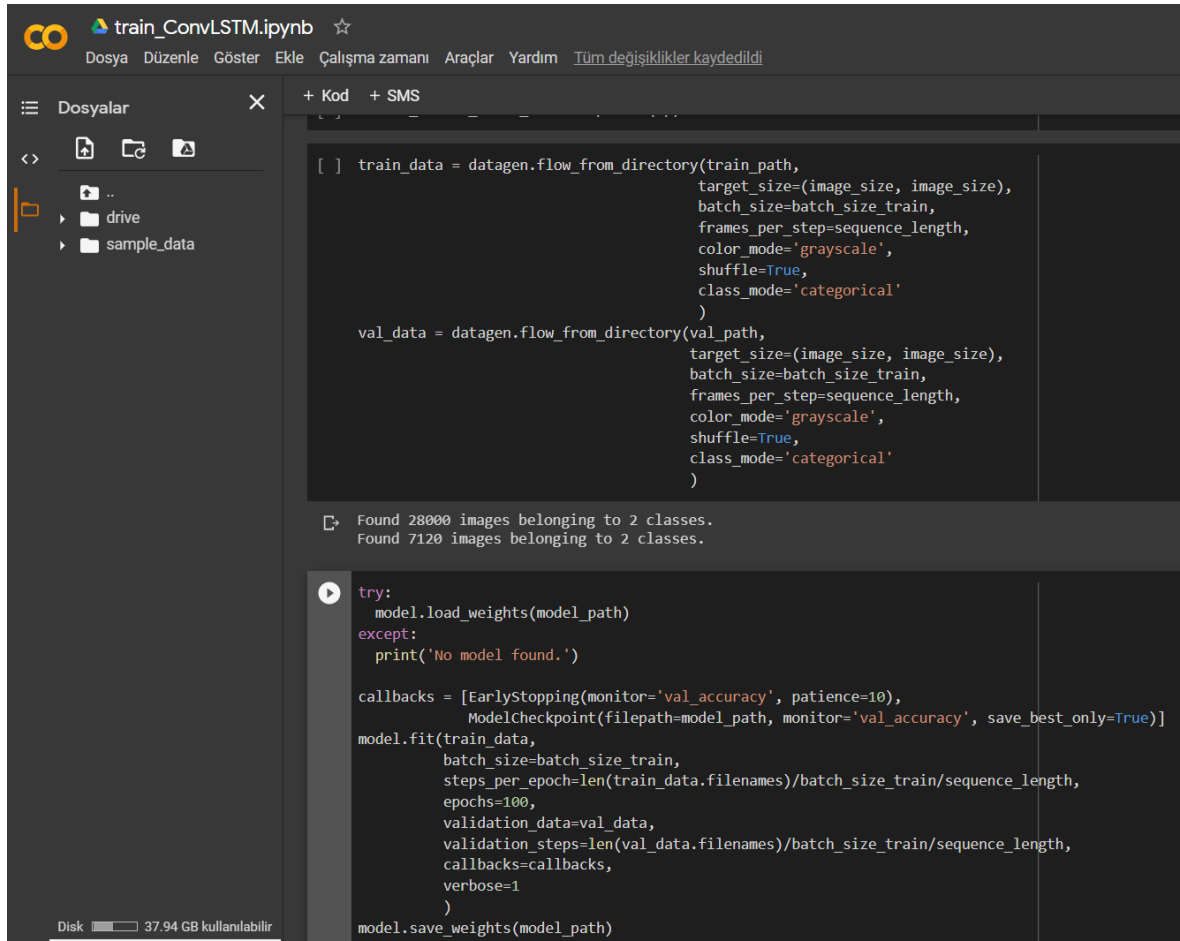


```
[12] model.compile(optimizer=adam, loss='crossentropy')

[13] model.summary()

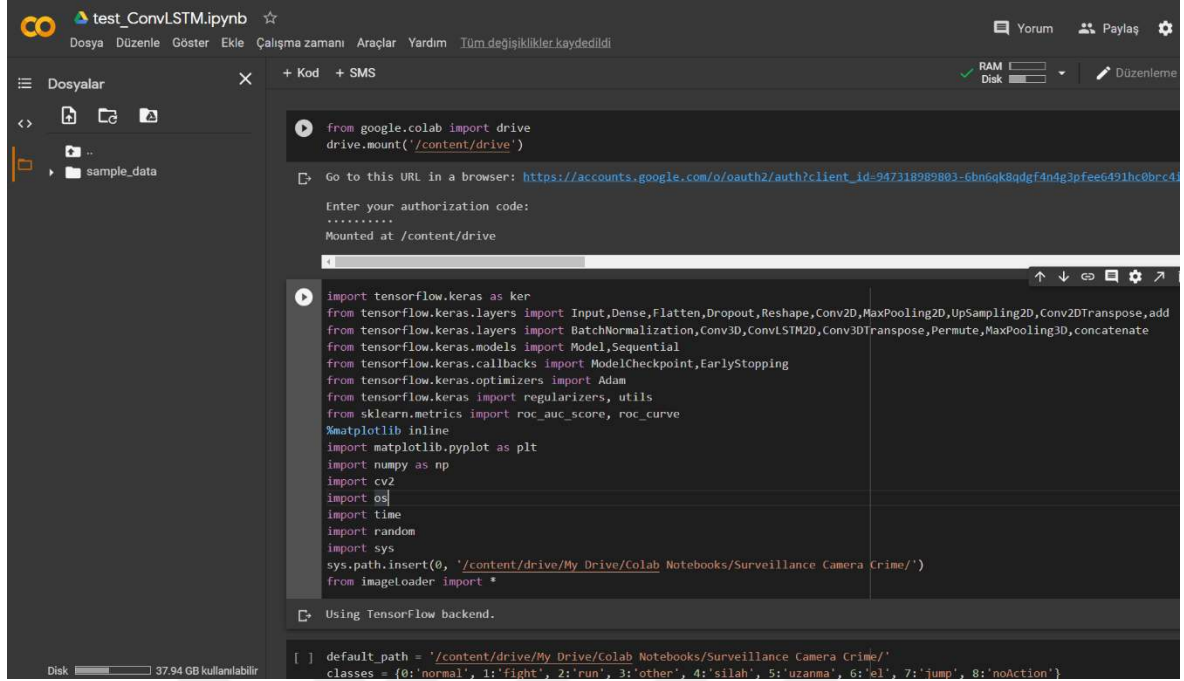
Model: "sequential_2"
Layer (type) Output Shape Param #
=====
conv3d_8 (Conv3D) (None, 15, 55, 55, 16) 1952
conv3d_9 (Conv3D) (None, 15, 26, 26, 16) 6416
conv3d_10 (Conv3D) (None, 15, 26, 26, 32) 4640
conv3d_11 (Conv3D) (None, 15, 26, 26, 32) 9248
max_pooling3d_2 (MaxPooling3 (None, 15, 13, 13, 32) 0
conv_lst_m2d_8 (ConvLSTM2D) (None, 15, 13, 13, 64) 221440
conv_lst_m2d_9 (ConvLSTM2D) (None, 15, 13, 13, 64) 295168
conv_lst_m2d_10 (ConvLSTM2D) (None, 15, 13, 13, 64) 295168
conv_lst_m2d_11 (ConvLSTM2D) (None, 15, 13, 13, 64) 295168
up_sampling3d (UpSampling3D) (None, 15, 26, 26, 64) 0
conv3d_transpose (Conv3DTran (None, 15, 26, 26, 32) 18464
conv3d_transpose_1 (Conv3DTr (None, 15, 26, 26, 32) 9248
conv3d_transpose_2 (Conv3DTr (None, 15, 55, 55, 16) 12816
conv3d_transpose_3 (Conv3DTr (None, 15, 227, 227, 1) 1937
=====
Total params: 1,171,665
Trainable params: 1,171,665
Non-trainable params: 0
```

Şekil 65: Gözetimsiz Derin Öğrenme Modeli



Şekil 66: Derin Öğrenme Modeli Eğitim Kodu

EK-6: Derin Öğrenme Modelleri Test Kodları Ekran Görüntüleri

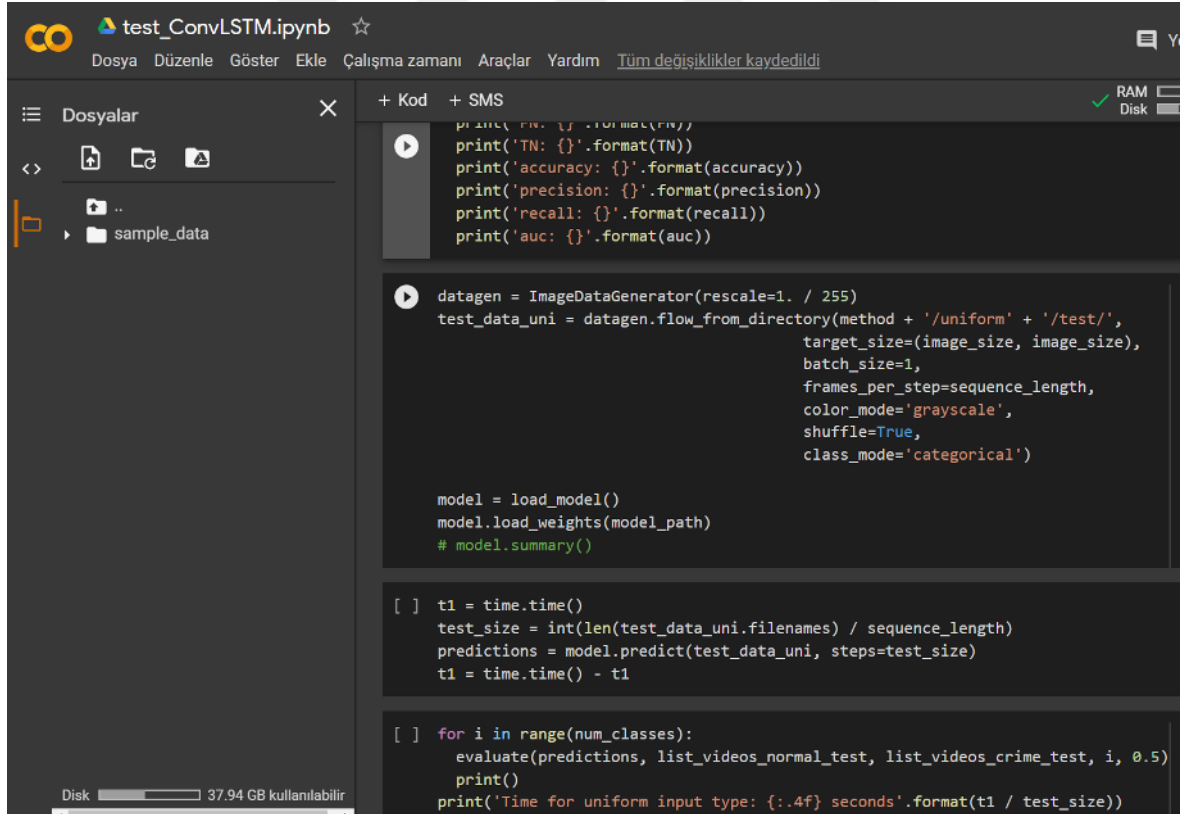


```
from google.colab import drive
drive.mount('/content/drive')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4ndg3pfee6491hc8brc41...
Enter your authorization code:
.....
Mounted at /content/drive

import tensorflow.keras as ker
from tensorflow.keras.layers import Input,Dense,Flatten,Dropout,Reshape,Conv2D,MaxPooling2D,UpSampling2D,Conv2DTranspose,add
from tensorflow.keras.layers import BatchNormalization,Conv3D,ConvLSTM2D,Conv3DTranspose,Permute,MaxPooling3D,concatenate
from tensorflow.keras.models import Model,Sequential
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import regularizers,utils
from sklearn.metrics import roc_auc_score,roc_curve
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import cv2
import os
import time
import random
import sys
sys.path.insert(0, '/content/drive/My Drive/Colab Notebooks/Surveillance Camera Crime/')
from imager import *
```

Şekil 67: Derin Öğrenme Modeli Testte Kullanılan Kütüphaneler



```
print('FN: {}'.format(FN))
print('TN: {}'.format(TN))
print('accuracy: {}'.format(accuracy))
print('precision: {}'.format(precision))
print('recall: {}'.format(recall))
print('auc: {}'.format(auc))

datagen = ImageDataGenerator(rescale=1. / 255)
test_data_uni = datagen.flow_from_directory(method + '/uniform' + '/test/',
target_size=(image_size, image_size),
batch_size=1,
frames_per_step=sequence_length,
color_mode='grayscale',
shuffle=True,
class_mode='categorical')

model = load_model()
model.load_weights(model_path)
# model.summary()

t1 = time.time()
test_size = int(len(test_data_uni.filesnames) / sequence_length)
predictions = model.predict(test_data_uni, steps=test_size)
t1 = time.time() - t1

for i in range(num_classes):
    evaluate(predictions, list_videos_normal_test, list_videos_crime_test, i, 0.5)
    print()
    print('Time for uniform input type: {:.4f} seconds'.format(t1 / test_size))
```

Şekil 68: Derin Öğrenme Modeli Test Kodu

```
def evaluate(predictions, list_videos_normal,
             list_videos_crime, class_number=0, threshold=0.5):
    test_normal_size = len(list_videos_normal)
    if class_number==0:
        abnormal_name = 'Crime'
        test_abnormal_size = len(list_videos_crime)
        pred = predictions
    else:
        class_number = class_number
        abnormal_name = classes[class_number]
        test_abnormal_size = 0
        pred = predictions[:test_normal_size]
        for i, item in enumerate(list_videos_crime):
            if int(item[3])==class_number:
                test_abnormal_size += 1
                pred.append(predictions[test_normal_size+i])

    labels = np.zeros((test_normal_size+test_abnormal_size, 2))
    labels[:test_normal_size, 0] = 1
    labels[test_normal_size:, 1] = 1

    FP = 0
    for i in range(test_normal_size):
        if threshold <= pred[i][1]:
            FP += 1
    TN = test_normal_size - FP

    TP = 0
    for i in range(test_abnormal_size):
        if threshold <= pred[test_normal_size+i][1]:
            TP += 1
    FN = test_abnormal_size - TP

    accuracy = (TP+TN) / (test_normal_size+test_abnormal_size)
    precision = TP / (TP+FP)
    recall = TP / (test_abnormal_size)
    auc = roc_auc_score(labels, pred)

    print('normal: {}'.format(test_normal_size))
    print('{}: {}'.format(abnormal_name, test_abnormal_size))
    print('TP: {}'.format(TP))
    print('FP: {}'.format(FP))
    print('FN: {}'.format(FN))
    print('TN: {}'.format(TN))
    print('accuracy: {}'.format(accuracy))
    print('precision: {}'.format(precision))
    print('recall: {}'.format(recall))
    print('auc: {}'.format(auc))
```

Şekil 69: Derin Öğrenme Modeli Test Metodu

KAYNAKLAR

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283).
- Abdali, A. M. R., & Al-Tuma, R. F. (2019, March). Robust Real-Time Violence Detection in Video Using CNN And LSTM. In *2019 2nd Scientific Conference of Computer Sciences (SCCS)* (pp. 104-108). IEEE.
- Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., & Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*.
- Accattoli, S., Sernani, P., Falcionelli, N., Mekuria, D. N., & Dragoni, A. F. (2020). Violence Detection in Videos by Combining 3D Convolutional Neural Networks and Support Vector Machines. *Applied Artificial Intelligence*, 34(4), 329-344.
- Adam, A., Rivlin, E., Shimshoni, I., & Reinitz, D. (2008). Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE transactions on pattern analysis and machine intelligence*, 30(3), 555-560.
- Afouras, T., Chung, J. S., & Zisserman, A. (2020, May). ASR is all you need: Cross-modal distillation for lip reading. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2143-2147). IEEE.
- Aggarwal, C. C. (2018). *Neural networks and deep learning*. Springer.
- Aktı, Ş., Tataroğlu, G. A., & Ekenel, H. K. (2019, November). Vision-based Fight Detection from Surveillance Cameras. In *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)* (pp. 1-6). IEEE.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." NIPS 2012

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Al-Dhamari, A.H.L.A.M., Sudirman, R. and Mahmood, N.H., 2017. Abnormal Behavior Detection in Automated Surveillance Videos: A Review. *Journal of Theoretical & Applied Information Technology*, 95(19).

An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1).

Backhaus, W. G., Kliegl, R., & Werner, J. S. (Eds.). (2011). *Color vision: Perspectives from different disciplines*. Walter de Gruyter.

Bansod, S., & Nandedkar, A. (2019). Transfer learning for video anomaly detection. *Journal of Intelligent & Fuzzy Systems*, 36(3), 1967-1975.

Berg, A., Deng, J., & Fei-Fei, L. (2011). ImageNet large scale visual recognition challenge 2010. *ILSVRC2011*.

Bhardwaj, A., Di, W., & Wei, J. (2018). *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing Ltd.

Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.

Boss Research Project, <http://www.multitel.be/image/researchdevelopment/research-projects/boss.php>

Bradski, G. (2000). The OpenCV Library. *Dr. Dobbs's Journal of Software Tools*.

Brown, D.E. 1998. "The Regional Crime Analysis Program (RECAP): A framework for Mining Data to Catch Criminals", IEEE, pp. 2848-2853.

- Browy, E., Woods, M. J., & Rabinovich, A. (2020). *U.S. Patent No. 10,580,213*. Washington, DC: U.S. Patent and Trademark Office.
- Buduma, N., & Locascio, N. (2017). *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. " O'Reilly Media, Inc."
- Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10023-10033).
- Carneiro, S. A., da Silva, G. P., Guimaraes, S. J. F., & Pedrini, H. (2019, October). Fight Detection in Video Sequences Based on Multi-Stream Convolutional Neural Networks. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (pp. 8-15). IEEE.
- Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.
- Chan, A., & Vasconcelos, N. (2008). Ucsd pedestrian dataset. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(5), 909-926.
- Chen, H., Li, J., & Hu, X. (2020). Delving Deeper into the Decoder for Video Captioning. *arXiv preprint arXiv:2001.05614*.
- Cheng, M., Cai, K., & Li, M. (2019). RWF-2000: An Open Large Scale Video Database for Violence Detection. *arXiv preprint arXiv:1911.05913*.
- Chollet, F., & others. (2015). Keras. GitHub. Retrieved from <https://github.com/fchollet/keras>
- Chong, Y.S. and Tay, Y.H. 2017, June. Abnormal event detection in videos using spatiotemporal autoencoder. In *International Symposium on Neural Networks* (pp. 189-196). Springer, Cham.
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Andrew Rabinovich. "Going Deeper with Convolutions." CVPR 2015.

- Clarke, R. V., & Eck, J. E. (2005). *Crime analysis for problem solvers in 60 small steps*. Washington, DC: US Department of Justice, Office of Community Oriented Policing Services.
- Clemente, F. and Kleiman, M.B. 1977. Fear of crime in the United States: A multivariate analysis. *Social Forces*, 56(2), 519-531.
- Çolak, H. (1998). Kolluk Teşkilatı ve Adli Kolluk. *Sayıştay Dergisi Ocak-Mart*, 85-91.
- Dağ, H. 2003. “Suçla Mücadele”, *Polis Dergisi*, 34, 106-109.
- Dai, C., Liu, X., & Lai, J. (2020). Human action recognition using two-stream attention based LSTM networks. *Applied Soft Computing*, 86, 105820.
- Daniel, G. (2013). *Principles of artificial neural networks* (Vol. 7). World Scientific.
- Dhiman, C., & Vishwakarma, D. K. (2019). A review of state-of-the-art techniques for abnormal human activity recognition. *Engineering Applications of Artificial Intelligence*, 77, 21-45.
- Dubey, S., Boragule, A., & Jeon, M. (2019, October). 3D ResNet with Ranking Loss Function for Abnormal Activity Detection in Videos. In *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)* (pp. 1-6). IEEE.
- Duman, E., & Erdem, O. A. (2019). Anomaly Detection in Videos Using Optical Flow and Convolutional Autoencoder. *IEEE Access*, 7, 183914-183923.
- Fernandes, G., Rodrigues, J. J., Carvalho, L. F., Al-Muhtadi, J. F., & Proença, M. L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems*, 70(3), 447-489.
- Forsyth, D. A., & Ponce, J. (2002). *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference.
- Gibbons, S. (2004). The costs of urban property crime. *The Economic Journal*, 114(499), F441-

F463.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Google Colab. (2017) Retrieved July 30, 2020, from
<https://colab.research.google.com/notebooks/intro.ipynb>.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. and Lew, M.S., 2016. Deep learning for visual understanding: A review. *Neurocomputing*, 187, pp.27-48.

Hara, K., Kataoka, H., & Satoh, Y. (2017). Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 3154-3160).

Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., & Davis, L. S. (2016). Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 733-742).

Hassner, T., Itcher, Y., & Kliper-Gross, O. (2012, June). Violent flows: Real-time detection of violent crowd behavior. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1-6). IEEE.

Haykin, S. S. (2009). *Neural networks and learning machines*/Simon Haykin.

Hevner A, March S, Park J, Ram S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75e105.

Introduction to RAM Disks. (2020). Retrieved 30 July 2020, from
<https://www.storagereview.com/introduction-to-ram-disks#:~:text=The%20main%20advantage%20of%20a,for%20small%204KB%2Dsize%20transfers.>

Jensen, J. R. (1996). *Introductory digital image processing: a remote sensing perspective* (No. Ed. 2). Prentice-Hall Inc..

- Jiang, B., Xu, F., Tu, W., & Yang, C. Channel-wise Attention in 3D Convolutional Networks for Violence Detection. In *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)* (pp. 59-64). IEEE.
- Jiang, Y., Bosch, N., Baker, R. S., Paquette, L., Ocumpaugh, J., Andres, J. M. A. L., ... & Biswas, G. (2018, June). Expert feature-engineering vs. deep neural networks: Which is better for sensor-free affect detection?. In *International Conference on Artificial Intelligence in Education* (pp. 198-211). Springer, Cham.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." CVPR 2016.
- Kalaivani, P., Mohamed Mansoor Roomi, S. and Annalakshmi, M. (2017). COMPREHENSIVE FRAMEWORK FOR LEARNING EVENTS. *The Online Journal of Distance Education and e-Learning*, 5(3).
- Karakehya, H. (2009). "GÖZETİM VE SUÇLA MÜCADELE": Gözetimin Tarihsel Gelişimi ile Yakın Dönemde Gerçekleştirilen Hukuki Düzenleme ve Uygulamalar Bağlamında Bir Değerlendirme.
- Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." ICLR 2015
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1725-1732).
- Karthickkumar, S., & Kumar, K. (2020, January). A survey on Deep learning techniques for human action recognition. In *2020 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-6). IEEE.
- Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P. and Suleyman, M., 2017. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950.

Kdenlive | Libre Video Editor. (2015). Retrieved 30 July 2020, from <https://kdenlive.org/en/>

Kiran, B. R., Thomas, D. M., & Parakkal, R. (2018). An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2), 36.

Ko, K.E. and Sim, K.B., 2018. Deep convolutional framework for abnormal behavior detection in a smart surveillance system. *Engineering Applications of Artificial Intelligence*, 67, pp.226-234.

Koohzadi, M. and Charkari, N.M., 2017. Survey on deep learning methods in human action recognition. *IET Computer Vision*, 11(8), pp.623-632.

Kubat, M. (2017). *An introduction to machine learning*. Springer International Publishing AG.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature*, 521(7553), p.436.

Leyva, R., Sanchez, V., & Li, C. T. (2017, April). The LV dataset: A realistic surveillance video dataset for abnormal event detection. In *2017 5th International Workshop on Biometrics and Forensics (IWBF)* (pp. 1-6). IEEE.

Li, Y., Min, M. R., Shen, D., Carlson, D., & Carin, L. (2018, April). Video generation from text. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. and Alsaadi, F.E., 2017. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, pp.11-26.

Liu, W., Luo, W., Li, Z., Zhao, P., & Gao, S. (2019, August). Margin learning embedded prediction for video anomaly detection with a few anomalies. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (pp. 3023-3030). AAAI Press.

Lu, C., Shi, J., & Jia, J. (2013). Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision* (pp. 2720-2727).

- Lu, Y., & Yan, J. (2020). Automatic Lip Reading Using Convolution Neural Network and Bidirectional Long Short-term Memory. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(01), 2054003.
- Luo, W., Liu, W. and Gao, S., 2017. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 341-349).
- Luo, W., Liu, W., & Gao, S. (2017, July). Remembering history with convolutional lstm for anomaly detection. In *2017 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 439-444). IEEE.
- Mabrouk, A. B., & Zagrouba, E. (2018). Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications*, 91, 480-491.
- Majhi, S., Dash, R., & Sa, P. K. (2019). Two-Stream CNN Architecture for Anomalous Event Detection in Real World Scenarios.
- Mandic, D., & Chambers, J. (2001). *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Wiley.
- Marchi, E., Vesperini, F., Eyben, F., Squartini, S., & Schuller, B. (2015, April). A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1996-2000). IEEE.
- MATLAB. (2010). *version 9.7.0 (R2019a)*. Natick, Massachusetts: The MathWorks Inc.
- Medel, J. R., & Savakis, A. (2016). Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*.
- Mehran, R., Oyama, A., & Shah, M. (2009, June). Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 935-942). IEEE.

- Michie, E. D. (1994). Machine Learning, Neural and Statistical Classification.(D. Michie, DJ Spiegelhalter, & CC Taylor, Eds.). *Technometrics*, 37(4), 459.
- Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S.A., Yan, Y., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C. and Oliva, A., 2019. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*.
- Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., & Venkatesh, S. (2019). Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 11996-12004).
- Nawaratne, R., Alahakoon, D., De Silva, D., & Yu, X. (2019). Spatiotemporal Anomaly Detection Using Deep Learning for Real-Time Video Surveillance. *IEEE Transactions on Industrial Informatics*, 16(1), 393-402.
- Nievas, E. B., Suarez, O. D., García, G. B., & Sukthankar, R. (2011, August). Violence detection in video using computer vision techniques. In *International conference on Computer analysis of images and patterns* (pp. 332-339). Springer, Berlin, Heidelberg.
- Nilsson, N. J. (2014). *Principles of artificial intelligence*. Morgan Kaufmann.
- Oliphant, T. E. (2006). A guide to NumPy (Vol. 1). *Trelgol Publishing USA*.
- O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., ... & Walsh, J. (2019, April). Deep learning vs. traditional computer vision. In *Science and Information Conference* (pp. 128-144). Springer, Cham.
- Piza, E. L., Welsh, B. C., Farrington, D. P., & Thomas, A. L. (2019). CCTV surveillance for crime prevention: A 40-year systematic review with meta-analysis. *Criminology & Public Policy*, 18(1), 135-159.
- Poppe, R., 2010. A survey on vision-based human action recognition. *Image and vision computing*,

28(6), pp.976-990.

Problem Çözücüler için 60 ‘Küçük’ Adımda Suç Analizi,
https://popcenter.asu.edu/sites/default/files/library/reading/PDFs/60steps_turkish.pdf

Project Jupyter. (2015). Retrieved July 30, 2020, from <https://jupyter.org/>

PyCharm: the Python IDE for Professional Developers by JetBrains. (2010). Retrieved 30 July 2020, from <https://www.jetbrains.com/pycharm/>

Ramachandra, B., Jones, M. J., & Vatsavai, R. R. (2020). A Survey of Single-Scene Video Anomaly Detection. arXiv preprint arXiv:2004.05993.

Ribeiro, M., Lazzaretti, A. E., & Lopes, H. S. (2018). A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters*, 105, 13-22.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks*, 61, pp.85-117.

Seo, J. J., Kim, H. I., De Neve, W., & Ro, Y. M. (2017). Effective and efficient human action recognition using dynamic frame skipping and trajectory rejection. *Image and Vision Computing*, 58, 76-85.

Sewak, M., Karim, M. R., & Pujari, P. (2018). *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python*. Packt Publishing Ltd.

Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568-576).

Skansi, S. (2018). *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer.

- Soliman, M. M., Kamal, M. H., Nashed, M. A. E. M., Mostafa, Y. M., Chawky, B. S., & Khattab, D. (2019, December). Violence Recognition from Videos using Deep Learning Techniques. In *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)* (pp. 80-85). IEEE.
- Song, W., Zhang, D., Zhao, X., Yu, J., Zheng, R., & Wang, A. (2019). A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks. *IEEE Access*, 7, 39172-39179.
- Soyaslan, D., Ceza Hukuku Genel Hükümler, Yetkin Kitabevi, Ankara, 1998.
- Sudhakaran, S., & Lanz, O. (2017, August). Learning to detect violent videos using convolutional long short-term memory. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.
- Sultani, W., Chen, C. and Shah, M., 2018. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6479-6488).
- Sumon, S. A., Shahria, M. T., Goni, M. R., Hasan, N., Almarufuzzaman, A. M., & Rahman, R. M. (2019, April). Violent Crowd Flow Detection Using Deep Learning. In *Asian Conference on Intelligent Information and Database Systems* (pp. 613-625). Springer, Cham.
- Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
- Tay, N. C., Connie, T., Ong, T. S., Goh, K. O. M., & Teh, P. S. (2019). A robust abnormal behavior detection method using convolutional neural network. In *Computational Science and Technology* (pp. 37-47). Springer, Singapore.
- Tay, N. C., Tee, C., Ong, T. S., & Teh, P. S. (2019, November). Abnormal Behavior Recognition using CNN-LSTM with Attention Mechanism. In *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)* (pp. 1-5). IEEE.

Trial Procedure, <https://www.britannica.com/topic/crime-law/Trial-procedure>

Uysal, M. P., & Giray, G. (2017). Yazılım Yeniden Yapılanmada Öz Çerçeve Yaklaşımı.

U.S. Jewelry Crime Statistics, <https://jewelerssecurity.org/crime-statistics/>

Ünver, M. (2008). Suçun Ekonomik Analizi: Genel Bir Değerlendirme. Dumlupınar Üniversitesi Sosyal Bilimler Dergisi, (20), 35–56.

Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

Vaishnavi VK, Kuechler WJ. (2008). Design science research methods and patterns: innovating information and communication technology. first ed. NW, USA: Auerbach Publications, Taylor & Francis Group

Velastin, S. A., & Gómez-Lira, D. A. (2017, November). People Detection and Pose Classification Inside a Moving Train Using Computer Vision. In *International Visual Informatics Conference* (pp. 319-330). Springer, Cham.

Welsh, B. C., & Farrington, D. P. (2009). Public area CCTV and crime prevention: an updated systematic review and meta-analysis. *Justice Quarterly*, 26(4), 716-745.

Wolfgang, E. (2011). Introduction to artificial intelligence. *Translated by Nathanael Black With illustrations by Florian Mast, Springer-Verlag London Limited.*

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67-82.

Wu, D., Sharma, N. and Blumenstein, M., 2017, May. Recent advances in video-based human action recognition using deep learning: A review. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 2865-2872). IEEE.

Wu, W., He, D., Tan, X., Chen, S., Yang, Y., & Wen, S. (2020). Dynamic Inference: A New

Approach Toward Efficient Video Action Recognition. *arXiv preprint arXiv:2002.03342*.

- Xia, Q., Zhang, P., Wang, J., Tian, M., & Fei, C. (2018, August). Real time violence detection based on deep spatio-temporal features. In *Chinese Conference on Biometric Recognition* (pp. 157-165). Springer, Cham.
- Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems* (pp. 802-810).
- Yan, S., Smith, J. S., Lu, W., & Zhang, B. (2018). Abnormal event detection from videos using a two-stream recurrent variational autoencoder. *IEEE Transactions on Cognitive and Developmental Systems*.
- Yao, Y., Wang, X., Xu, M., Pu, Z., Atkins, E., & Crandall, D. (2020). When, Where, and What? A New Dataset for Anomaly Detection in Driving Videos. *arXiv preprint arXiv:2004.03044*.
- Yu, C., Ward, M.W., Morabito, M., & Ding, W. 2011. Crime Forecasting Using Data Mining Techniques. *2011 IEEE 11th International Conference on Data Mining Workshops*, 779-786.
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2019). Dive into deep learning. *Unpublished Draft*. Retrieved, 19, 2019.
- Zhang, H. B., Zhang, Y. X., Zhong, B., Lei, Q., Yang, L., Du, J. X., & Chen, D. S. (2019). A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19(5), 1005.
- Zheng, A. (2015). Evaluating machine learning models: a beginner's guide to key concepts and pitfalls.
- Zhiguo, W. A. N. G., & Yujin, Z. H. A. N. G. (2020). Anomaly detection in surveillance videos: A survey. *Journal of Tsinghua University (Science and Technology)*, 60(6), 518-529.

Zhou, P., Ding, Q., Luo, H., & Hou, X. (2017, June). Violent interaction detection in video based on deep learning. In *Journal of physics: conference series* (Vol. 844, No. 1, p. 012044). IOP Publishing.

Zhu, Y., Nayak, N. M., & Roy-Chowdhury, A. K. (2012). Context-aware activity recognition and anomaly detection in video. *IEEE Journal of Selected Topics in Signal Processing*, 7(1), 91-101.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2019). A Comprehensive Survey on Transfer Learning. *arXiv preprint arXiv:1911.02685*.

