



DERİN ÖĞRENME TABANLI ŞİDDET DETEKTÖRÜ

MUSTAFA KESER

ŞUBAT 2024

ÇANKAYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VERİ ANALİTİĞİ ANA BİLİM DALI

Yüksek Lisans

VERİ ANALİTİĞİ

DERİN ÖĞRENME TABANLI ŞİDDET DETEKTÖRÜ

MUSTAFA KESER

ŞUBAT 2024

ÖZET

DERİN ÖĞRENME TABANLI ŞİDDET DETEKTÖRÜ

KESER, Mustafa

VERİ ANALİTİĞİ YÜKSEK LİSANS

Danışman: Prof. Dr. Hayri SEVER

Şubat 2024, 93 Sayfa

Hareket tanıma, bilgisayarlı görü alanı içerisinde hem RGB alanında hem de derinlik haritaları üzerinde oldukça fazla çalışılmış bir konu olsa da videolardaki şiddet içeren hareketlerin tespiti nispeten daha az çalışılmış güncel bir alandır. Günümüzde insanların yaşadığı hemen her alanda kamera sistemi bulunmasına karşın bu görüntülerin analizi konusunda hâlen çok fazla çalışma bulunmamaktadır. Artan kamera sayısı ile beraber biriken veri miktarı sonucunda akıllı aktivite tespit sistemlerine oldukça gereksinim duyulmaktadır. Bu proje ile kamera sistemlerinde video etiketleme sistemleri kapsamında kullanılacak bir şiddet tanıma yaklaşımı geliştirmek hedeflenmektedir.

Gelişen teknolojiyle beraber yaygınlaşan ve bant genişliği artan internet sayesinde büyük miktardaki video verisine kolayca erişilebilmektedir. Videolarda şiddet içeren sahnelerin etiketlenmesi, güvenlik ve içerik tabanlı video arama sistemi bakımından önem arz etmektedir. Güvenlik kameraları şiddet eylemlerini tespit etmek için uygun değildir. Büyük ölçekli kamera sistemlerinde insan operatörünün bütün kameraları izlemesi neredeyse imkânsızdır. Bu nedenle videolarda şiddet tespiti önem kazanmaktadır.

Bu projeye, mevcut yöntemlerden daha verimli sonuçlar üreten yeni bir derin öğrenme tabanlı şiddet detektörü geliştirilmesi hedeflenmiştir. Bu çalışmada, video görüntüleri üzerinden Transfer Öğrenme ve Long Short Term Memory (LSTM) ağı yaklaşımlı bir yöntem kullanılmıştır. MobileNetV2 ardışık video karelerinden uzamsal

özellikleri çıkartmak için yararlanılmıştır. Ayrıca, BILSTM yerel uzamsal özellikleri koruyup video kareleri arasındaki ilişkiler zamansal olarak analiz edilmiştir. Bu çalışmada en iyi sonucu hockey fight veri seti kullanılarak %99,37 değeri elde edilmiştir. Bu değer, yapılan birçok çalışmadan daha yüksek başarı oranına sahiptir.

Anahtar Kelimeler: Yapay Zekâ, Derin Öğrenme, Şiddet Eylemi, MobileNet, BILSTM



ABSTRACT

DEEP LEARNING-BASED VIOLENCE DETECTOR

KESER, Mustafa

M.Sc. in Data Analytics

Supervisor: Prof. Dr. Hayri SEVER

February 2024, 93 Pages

Motion recognition within the realm of computer vision has garnered substantial attention, both in RGB and depth maps. However, the identification of aggressive movements within video sequences remains an area that is relatively underexplored and contemporaneous. Despite the omnipresence of camera systems across diverse spheres of human life, there exists a palpable dearth in research pertaining to the nuanced analysis of these visual data streams. The escalating ubiquity of cameras has precipitated an exponential accumulation of data, thereby engendering a pressing exigency for sophisticated systems capable of discerning intricate human activities. This project endeavors to craft an innovative violence recognition paradigm tailored for integration within video annotation systems, specifically designed for camera setups.

The proliferation of technology has facilitated unfettered access to copious volumes of video data owing to the burgeoning internet bandwidth. The meticulous annotation of video segments portraying acts of violence assumes paramount significance, especially within the domains of security and content-driven video retrieval systems. Security cameras, however, evince limitations in accurately discerning violent acts, while the task of exhaustive monitoring by human operators in expansive camera networks approaches insurmountability. Consequently, the identification of violence within video footage has emerged as a critical concern.

This project aspires to forge a novel deep learning-driven violence detection mechanism that proffers superior efficacy when juxtaposed against prevailing methodologies. The present study methodically employs Transfer Learning in conjunction with a Long Short Term Memory (LSTM) network specifically tailored for video frames. Leveraging MobileNetV2 facilitates the extraction of spatial features from successive video frames. Concurrently, BILSTM endeavors to preserve localized spatial attributes while meticulously scrutinizing temporal interrelations inherent within video frames. In this study, the best result was obtained by using the hockey fight data set, with a value of 99.37%, eclipsing the performance benchmarks set forth by antecedent studies.

Keywords: Artificial Intelligence, Deep Learning, Violent Action, MobileNet, BILSTM

TEŐEKKÜR

Bu alıőmada, bana alıőma zamanı verdikleri iin kızlarım Beril'e ve Defne'ye, yardımları iin iő arkadaşlarıma ve bana deęerli vakitlerini esirgmeden ayırıp sorunların özümünde yardımcı olan tez danışmanım Prof. Dr. Hayri SEVER hocama ok teőekkür ederim.



İÇİNDEKİLER

İNTİHAL OLMADIĞINA DAİR BEYAN SAYFASI.....	iii
ÖZET.....	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
TABLolar LİSTESİ.....	xii
ŞEKİLLER LİSTESİ.....	xiii
KISALTMALAR LİSTESİ.....	xv
BÖLÜM I.....	1
GİRİŞ VE LİTERATÜR TARAMASI	1
1.1 GİRİŞ.....	1
1.2 LİTERATÜR TARAMASI	3
BÖLÜM II	7
YAPAY ZEKÂ VE YAPAY SİNİR AĞLARI.....	7
2.1 YAPAY ZEKÂ	7
2.1.1 Makine Öğrenmesi	9
2.1.1.1 Makine Öğrenmesi Türleri.....	11
2.1.1.1.1 Gözetimli Öğrenme (Supervised Learning)	11
2.1.1.1.2 Gözetimsiz Öğrenme (Unsupervised Learning)	12
2.1.1.1.3 Pekiştirmeli Öğrenme(Reinforcement Learning).....	12
2.1.2 Derin Öğrenme	13
2.2 YAPAY SİNİR AĞLARI	14
2.2.1 Biyolojik Sinir Hücreleri	14
2.2.2 Yapay Sinir Hücreleri.....	15
2.2.2.1 Yapay Sinir Ağlarının Temel Elemanları	17
2.2.2.2 Yapay Sinir Ağı Mimarileri	18
2.2.2.3 Tek Katmanlı Algılayıcılar (TKA).....	18
2.2.2.4 Çok Katmanlı Algılayıcılar (ÇKA).....	19
2.2.2.5 Yapay Sinir Ağlarının Genel Özellikleri.....	20

2.2.2.6	Aktivasyon Fonksiyonları	21
2.2.2.7	Kayıp Fonksiyonları.....	26
2.2.2.8	Geri Yayılım (Backpropagation).....	26
2.2.2.9	Gradyan İnişi (Gradient Descent)	27
2.2.3	Evrişimsel Sinir Ağları	28
2.2.3.1	Evrişimsel Sinir Ağları Modelleri.....	30
2.2.4	Transfer Öğrenme.....	36
2.2.5	Tekrarlayan Sinir Ağları (RNN).....	36
2.2.5.1	Uzun Kısa Vadeli Bellek (LSTM)	38
2.3	BİLGİSAYARLI GÖRÜ	38
2.3.1	Görüntü İşleme	39
BÖLÜM III.....		40
VERİ SETİ VE ÖNERİLEN MODEL.....		40
3.1	VERİ SETİ.....	40
3.1.1	Hockey Fights Dataset.....	40
3.1.2	Movies Dataset	41
3.1.3	Real-Life Violence Situations	41
3.2	ÖNERİLEN MODEL	42
3.2.1	Geliştirme Ortamı ve Kullanılan Kütüphaneler	42
3.2.1.1	Google Colab	42
3.2.1.2	Python	42
3.2.1.3	OpenCV	43
3.2.1.4	TensorFlow	43
3.2.1.5	NumPy ve Matplotlib.....	44
3.2.2	Modelin Blok Diyagramı.....	44
3.2.3	Verinin Hazırlanması.....	46
3.2.4	Modelin Oluşturulması.....	47
3.2.5	Eğitim Ayarları.....	49
3.2.6	Aktivasyon Fonksiyonları	49
3.2.7	Dropout.....	49
3.2.8	Başarının Değerlendirilmesi.....	49
3.2.8.1	Doğruluk (Accuracy):	50
3.2.8.2	Duyarlılık (Precision):.....	51
3.2.8.3	Kesinlik (Recall):	51

3.2.8.4	F-Ölçütü	51
3.2.9	Deneyler ve Sonuçları	52
3.2.9.1	Deney 1 ve Sonuçları	52
3.2.9.2	Deney 2 ve Sonuçları	54
3.2.9.3	Deney 3 ve Sonuçları	55
3.2.9.4	Deney 4 ve Sonuçları	57
3.2.9.5	Deney 5 ve Sonuçları	58
3.2.9.6	Deney 6 ve Sonuçları	59
3.2.9.7	Deney 7 ve Sonuçları	61
3.2.9.8	Deney 8 ve Sonuçları	62
3.3	BULGULAR.....	66
3.4	SONUÇ.....	70
KAYNAKÇA	72

TABLolar LİSTESİ

Tablo 1: Yapay Sinir Ağı ile Biyolojik Sinir Ağı Karşılaştırılması.....	18
Tablo 2: Aktivasyon Fonksiyonlarının Matematiksel İfadeleri	25
Tablo 3: MobileNet Genel Mimarisi.....	45
Tablo 4: Karmaşıklık Matrisi	50
Tablo 5: Deney 1'den Elde Edilen Sonuçlar	54
Tablo 6: Deney 2'den Elde Edilen Sonuçlar	55
Tablo 7: Deney 3'ten Elde Edilen Sonuçlar.....	56
Tablo 8: Deney 4'ten Elde Edilen Sonuçlar.....	58
Tablo 9: Deney 5'ten Elde Edilen Sonuçlar.....	59
Tablo 10: Deney 6'dan Elde Edilen Sonuçlar.....	60
Tablo 11: Deney 7'den Elde Edilen Sonuçlar.....	62
Tablo 12: Deney 8'den Elde Edilen Sonuçlar.....	63
Tablo 13: Real-Life Violence Situations Veri Seti İçin Deneylerde Elde Edilen Sonuçlar	64
Tablo 14: Movies Veri Seti İçin Deneylerde Elde Edilen Sonuçlar	65
Tablo 15: Hockey Fight Veri Seti İçin Deneylerde Elde Edilen Sonuçlar.....	65
Tablo 16: Önerilen Modelin Deney Sonucu	67
Tablo 17: Önerilen Model ile Diğer Çalışmaların Başarı Değerlerinin Karşılaştırılması	70

ŞEKİLLER LİSTESİ

Şekil 1: Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme	8
Şekil 2: Yeni Programlama Yaklaşımı	10
Şekil 3: Yapay Zekâ, Makine Öğrenimi ve Derin Öğrenmenin Tarihî Gelişimi.....	13
Şekil 4: Biyolojik Sinir Hücresi.....	15
Şekil 5: Bir Nöronun Doğrusal Olmayan Modeli.....	16
Şekil 6: Çok Katmanlı Yapay Sinir Ağının Genel Yapısı	17
Şekil 7: Doğrusal Aktivasyon Fonksiyonu	22
Şekil 8: Adım Aktivasyon Fonksiyonu	23
Şekil 9: Sigmoid Aktivasyon Fonksiyonu	23
Şekil 10: Tanh Aktivasyon Fonksiyonu	24
Şekil 11: ReLU Aktivasyon Fonksiyonu.....	24
Şekil 12: Softmax Aktivasyon Fonksiyonu	25
Şekil 13: Görüntü Sınıflandırma İçin Temel Bir Evrişimli Sinir Ağı Yapısı	28
Şekil 14: Lenet-5 Mimarisi.....	31
Şekil 15: AlexNet Mimarisi.....	32
Şekil 16: GoogleNet Mimari	33
Şekil 17: VGG16 Modeli.....	33
Şekil 18: Artık Blok Örneği.....	34
Şekil 19: ResNet Ağ Mimarisi.....	35
Şekil 20: Tekrarlayan Sinir Ağı İçerisindeki Döngü	37
Şekil 21: RNN Algoritmasının Çalışma Şekli.....	37
Şekil 22: LSTM Yapısı.....	38
Şekil 23: Modelin Akış Diyagramı.....	45
Şekil 24: Model Diyagramında Dense Layer	46
Şekil 25: Önerilen Model.....	48
Şekil 26: Deney 1'in Model Accuracy, Model Loss, Confusion Matrix Grafikleri ..	53
Şekil 27: Deney 2'nin Model Accuracy, Model Loss, Confusion Matrix grafikleri.	54

Şekil 28: Deneý 3'ün Model Accuracy, Model Loss, Confusion Matrix Grafikleri .	56
Şekil 29: Deneý 4'ün Model Accuracy, Model Loss, Confusion Matrix Grafikleri .	57
Şekil 30: Deneý 5'in Model Accuracy, Model Loss, Confusion Matrix Grafikleri ..	58
Şekil 31: Deneý 6'nın Model Accuracy, Model Loss, Confusion Matrix Grafikleri	60
Şekil 32: Deneý 7'nin Model Accuracy, Model Loss, Confusion Matrix Grafikleri	61
Şekil 33: Deneý 8'in Model Accuracy, Model Loss, Confusion Matrix Grafikleri ..	63
Şekil 34: Şiddet Detektörü Blok Diyagramı	66
Şekil 35: Önerilen Modelin Accuracy Grafiđi.....	67
Şekil 36: Önerilen Modelin Loss Grafiđi	68
Şekil 37: Önerilen Modelin Confusion Matrix Grafiđi	68
Şekil 38: Önerilen Modelin Sonuđları	69

KISALTMALAR LİSTESİ

AI	:Artificial Intelligence
YZ	:Yapay Zekâ
AMV	:İvme Ölçüsü Vektörü
HOG	:Histogram of Oriented Gradients
ViF	:Violent Flow (Şiddet Akışı)
SVM	:Support Vector Machine
LSTM	:Long Short-Term Memory
BILSTM	:Bidirectional Long Short-Term Memory
CNN	:Convolutional Neural Network
YSA	:Yapay Sinir Ağları
TKA	:Tek Katmanlı Algılayıcılar
ÇKA	:Çok Katmanlı Algılayıcılar
ESA	:Evrşimsel Sinir Ağı
RNN	:Recurrent Neural Network
DP	:Doğru Pozitif
DN	:Doğru Negatif
YP	:Yanlış Pozitif
YN	:Yanlış Negatif
TP	:True Positive
TN	:True Negative
FP	:False Positive
FN	:False Negative

BÖLÜM I

GİRİŞ VE LİTERATÜR TARAMASI

1.1 GİRİŞ

Tek başına anlam ifade etmeyen işlenmemiş gerçekler, olaylar, varlıklar olarak adlandırılan veriler, farklı türlerde ve farklı kaynaklardan gün geçtikçe artarak birikmekte ve kaydedilmektedir. Bu verilerin kaynağı için akla öncelikle internet gelse de asıl telefonlar, uçaklar, otomobiller, otomasyon sistemleri ve kamera sistemleri sayılmalıdır. Biriken bu verileri anlık olarak veya belli bir anda bilgiye dönüştürmek, günümüzde hem sosyoekonomik hem de güvenlik bakımından bir gereksinim hâline gelmiştir. Günümüzde çok önemli bir yeri olan bilgi çeşitliliği ve miktarı, çok arttığından mevcut yöntemler ile yönetilemez hâle gelmiştir. Veri yığınları, makine öğrenmesi ve onun alt kümesi olan derin öğrenme yöntemleri ile anlamlı hâle getirilerek sınıflandırılabilir. Böylece, bilgi yönetimi kolaylaşmaktadır.

Biriken ve yönetilmesi zorlaşan verilerden bir tanesi de video verileridir. Havaalanları, hastaneler, stadyumlar, şehir merkezli ve diğer özel ve kamu kurumları gibi insanların kalabalık olduğu alanlarda kamera sistemleri bulunmakta ve bu kameralar her geçen gün hayatımızda daha fazla yer almaktadır. Bundan dolayı, video yönetim sistemleri oluşmakta ve yaygınlaşmaktadır. Video yönetim sistemlerinin bu kadar yaygın kullanılması güvenlik endişelerini azaltırken artan görüntü miktarları nedeniyle insanlar tarafından izlenemeyecek duruma gelmiştir. Bununla birlikte, gelişen teknoloji ve bilgisayar iletişimi bu verilere erişimi kolaylaştırmıştır. Bu videolardaki şiddet içeren hareketlerin tespit edilmesi ve tanınması şehir güvenlik sistemleri bağlamında giderek önem kazanmıştır. Şiddet içeren videoların etiketlenmesi, şiddet içeren eylemlerin tanınmasının bir parçası olarak kabul edilen bilgisayarlı görü alanındaki hareket tanıma alanında incelenen bir problem durumunu almıştır (Keçeli ve Kaya 2018: 204).

Bilgisayarlı görü alanında hareket tanıma, hem RGB videolarında hem de derinlik haritaları üzerinde fazlasıyla çalışılmış bir konu olsa da şiddet içeren hareketlerin tespiti bu alanda fazla çalışılmamış bir konudur (Keçeli ve Kaya 2019: 933).

Akıllı şehirler ve askerî amaçlı kullanılan yöntemlerde kamera sistemi bulunmakta ve bu kameralardan elde edilen görüntülerin incelenmesinde akıllı algoritmaların kullanımı ve görüntülerin otomatik işlenmesi önemlidir. Diğer taraftan, artan nüfusa bağlı olarak ortak kullanılan alanların yönetilmesi, bu alanların kapasitelerinin verimli kullanılması gibi kalabalık analizine ihtiyaç duyulan alanlar bulunmaktadır. Toplumun yaşam kalitesini artırmayı hedefleyen bu çalışmalarda video ve fotoğraf görüntülerinden yola çıkarak problemlere çözümler getirilmesi oldukça önemli bir konu hâline gelmiştir (Kızrak ve Bolat 2018: 263).

Veri miktarı ve çeşitliliğinin artması ile beraber bilgisayar sistemlerindeki donanımların gelişmesi, veriden anlam çıkartma ihtiyacının doğması, toplumun yaşam kalitesinin artırılması ve güvenlik sistemlerine destek sağlamak amacıyla video verilerini otomatik olarak analiz edecek ve verilerdeki şiddet içeren görüntüleri tespit edecek derin öğrenme tabanlı şiddet tespiti detektörü oluşturulması hedeflenmiştir. Bu çalışma, kamera sistemlerinden kaydedilen video görüntülerinin ve verilerinin analiz edilmesini kapsamaktadır.

Bu çalışma, videolardaki şiddet içeriğini tespit etmeye yönelik yapay zekâ ve derin öğrenme yöntemini kullanarak bir sistem geliştirmeyi amaçlamaktadır. Şiddet içeren videoların saptanması, dijital platformların güvenliğini artırmak, içerik üreticileri ve kullanıcıları korumak adına önemli bir adım olacaktır. Ayrıca, bu tür bir sistem, sosyal medya, video paylaşım platformları ve diğer dijital medya kanallarındaki içerikleri daha etkili bir şekilde filtreleme ve moderasyon süreçlerini güçlendirme potansiyeli taşımaktadır.

Video görüntüden, görüntü ise üç boyutlu diziden oluşur. Video verilerini işlemek için iyi hesaplama gücü ve performansı iyi olan algoritmalara ihtiyaç duyulmaktadır. Derin öğrenme öncesi dönemde bu işlemleri gerçekleştirmek mümkün değildir. Günümüzde, derin öğrenme, GPU'nun icadından sonra çok popüler bir araç hâline gelmiş, problemlere bu yöntemle çözüm üretilmeye başlanmıştır. Videolarda şiddet içeren görüntüleri tespit etmek ikili bir sınıflandırma problemidir, önceden toplanmış video görüntülerinde şiddet içeren ve içermeyen veriler üzerinde eğitilecek bir derin öğrenme mimarisi oluşturulacaktır.

Video görüntüleri üzerinde transfer öğrenme ve Long Short Term Memory (LSTM) ağı yaklaşımli bir yöntem çalışılmıştır. Videolardaki şiddeti tahmin etmek için çift yönlü LSTM modeli (CNN-BILSTM) mimarisi kullanılmıştır. Videolar önce karelere ayrılmıştır. MobilNetV2 kullanılarak derin öznitelikler elde edilmiştir. Öznitelikleri, zamansal olarak çıkartmak için BILSTM ağı kullanılmıştır. Bu nedenle zamansal özellikler, bu mimari mekânsal özelliklerin yanı sıra her iki yönde de tahmin analizi için kullanılmıştır. Kullanılan veri setleri literatürde bu tarz çalışmaların test edilmesinde yaygın olarak kullanılan Real-Life Violence Situations veri seti ile test edilmiştir. Deneysel sonuçları literatürdeki çalışmaların çoğunun üzerindedir. Derin öğrenme yardımıyla bir videoda bulunan şiddet içeren hareketler, insan yardımı olmadan otomatik olarak belirlenebilmektedir.

Bu çalışmanın giriş bölümü, videolarda şiddetin tespiti konusundaki mevcut sorunları ve bu alandaki literatürü ele alarak bu çalışmanın önemini vurgulamaktadır. Ayrıca, şiddet içeren içeriklere karşı duyarlı bir tespit sistemi geliştirmenin, dijital medya güvenliği ve toplumsal refah açısından ne kadar kritik olduğunu ortaya koymaktadır. Tezin 2. Bölümü'nde yapay sinir ağları ile ilgili genel bilgiler verilmekte, 3. Bölüm'de ise kullanılan yöntemler, bulgular ve sonuçlar ile birlikte, bu amaç doğrultusunda gerçekleştirilen çalışmanın detaylı bir analizi sunulmaktadır.

1.2 LİTERATÜR TARAMASI

Derin öğrenme yöntemleri ile videolardaki şiddet içeren aktivitelerin tespit edilmesi alanında birçok yaklaşım önerilmiştir.

Datta vd.nin (2002) yaptıkları çalışmada, insan şiddetini tespit etme sorununu; yumruk dövüşü, tekme atma, nesnelere vurma vb. eylemler şeklinde ele alıyorlar. İnsan şiddetini ele alan çalışmada bir İvme Ölçüsü Vektörü (AMV) tanımlanıyor.

Şiddet tespiti, kişinin uzuvlarının yönelim bilgisine ve hareket yörüngesi bilgileri güvenine dayanmaktadır. Hareketin yönü, büyüklüğü ve sarsıntısı AMV'nin zamansal türevi olarak tanımlanıyor. Bu çalışmada, dik silüetler ile ilgili olumlu sonuçlar elde edilse de çete veya grup şiddetinde başarılı sonuçlar elde edilememiştir.

Clarín vd.nin (2005) yaptıkları çalışmada, film videoları analiz edilerek şiddet olaylarını otomatik olarak tespit etmeye çalışmışlardır. Kan, cilt, hareket tespiti ile bıçaklama, yumruklama ve benzeri hareketleri sınıflandırmaya çalışmışlardır. Bu çalışma; filmi sahnelere ayırarak, sahneleri izole ederek, filmin içinde aktif olan cildi

belirleyerek bu sahnelerdeki kan rengi bölgeleri ve bu bölgelerin tüm sahne boyunca yoğunlaşp yoğunlaşmadığını kontrol etmektedir.

Mohtavipour vd.nin (2022) yaptıkları çalışmada el yapımı yöntemlerden türetilen belirli özelliklere dayalı algılama yöntemini kullanmışlardır. CNN'ler, ilki ham karelerden uzamsal akış, ikincisi zamansal akış, üçüncüsü uzamsal-zamansal akış kullanılarak eğitilmiştir. Uzamsal akış, çevre modellerini öğrenmek için videodaki her bir kareyle ağı eğitmiştir. Hareketli nesnelere için hareket hızını artıran gelen kareleri optik akış ile hesaba katan zamansal akıştır. Uzamsal-zamansal akışta, eylemlerin şeklini anlamak için ardışık kareler arasındaki farkları kullanarak hareket enerjisi görüntüsü oluşturulur. Daha sonra bu işlem için kullanılan üç farklı CNN'nin çıktıları, şiddet davranışlarını tespit etmek amacıyla bir sınıflandırıcıyla beslenir. Bu çalışmada Hokey ve Violent Flow veri setleri kullanılmıştır. Mohtavipour vd. bu veri setlerine göre sırasıyla %100 ve %99.35 başarı oranı elde ettiklerini iddia etmişlerdir.

Giannakopoulos vd. (2010) yaptıkları çalışmada, filmlerden gelen video görüntü akışlarındaki şiddeti belirleyebilmek için bir yöntem sunmuşlardır. Buna göre öğrenmeye dayalı işitsel ve görsel birleştirme yöntemini kullanarak filmlerdeki şiddet içeren görüntülerin tespitine yönelik bir araştırma yapmışlardır. Bu amacı yerine getirebilmek için çok aşamalı bir yaklaşım izlemişlerdir: İlk olarak belirli işitsel ve görsel bilgilerle ilgili sınıflara ilişkin olasılıksal ölçümleri tahmin etmek amacıyla otomatik işitsel ve görsel işleme ve analiz yöntemini gerçekleştirmişlerdir. İkinci aşamada, orta vadeli video segmentlerini “şiddet içerir” ya da “şiddet içermez” şeklinde sınıflandırmak için işitsel ve görsel bilgileri birleştiren bir meta-sınıflandırma mimarisi kullanmışlardır. 10 filmde oluşan gerçek bir veri kümesi üzerinde önerilen bir şemayı değerlendirmişlerdir. Deneyler, ses verilerini analiz ederek sınıflandırma ve algılamanın görsel tabanlı yöntemle kıyasla sırasıyla %1,6 ve %6,1 değerlerinde daha başarılı olduğunu ortaya koymuştur. Bununla birlikte, birleştirilmiş meta-sınıflandırma, en iyi bireysel yöntemle (yani ses tabanlı yöntemle) nispeten genel başarı oranında yükselme sağlamıştır. Son olarak, genel olay tespit performansı, şiddet içeren olayların sadece %17'sinin tespit edilemediğini, tespit edilen 2 olaydan neredeyse 1'inin gerçekten şiddet içeren olaylar olduğunu göstermiştir.

Dinesh Jackson vd. (2019) yaptıkları çalışmada, devasa veri tabanını işleyen gerçek zamanlı bir şiddet tespit sistemi önermektedirler. Bu öneri, girdi için farklı kaynaklardan gelen ve işlenen büyük miktarda gerçek zamanlı video akışıdır. Bu sistemde, akış verileri girilir ve insan zekâsı simülasyonu ile şiddet tespit etmeye

çalışılır. Spark çerçevesinde, fotoğraf kareleri ayrılır ve her bir karenin özellikleri HOG (Histogram of Oriented Gradients) fonksiyonu sayesinde çerçeveler şeklinde tespit edilir. Daha sonra bu fotoğraf kareleri, şiddet modeli, insan parçası modeli ve negatif model gibi niteliklere göre işaretlenir. Bu kareler şiddet görüntülerinin bulunabilmesi için Çift Yönlü Uzun Kısa Süreli Bellek (BILSTM) ağında eğitilir. Çift yönlü LSTM bilgiye, hem ileri hem de geri yönde erişilebilir. Böylece elde edilen çıktı hem geçmiş hem de gelecek bilgiler sayesinde üretilir. Şiddet içermeyen sahneler ve şiddet sahneleri içeren 409 video klip futbol stadyumundan elde edilmiştir. Bu modelin performansı doğrulanmış ve şiddet eylemini tanıma oranı yüzde 94,5 olarak hesaplanmıştır.

Dursun ve Türkmen'in (2021) yaptıkları çalışmada süper piksel tabanlı bir yöntem önerilerek kanlı bölge tespit edilmeye çalışılmıştır. Öncelikle, süper piksel yardımıyla çıkartılmış renk ve doku nitelikleri elde edilerek görüntülerde kan içeren alanları saptamak amacıyla kan görüntüsü olup ve olmayan şeklinde bir veri seti oluşturulmuştur. Bu veri setindeki görüntülere süper piksel bölütleme yöntemi uygulanmıştır. Daha sonra renk ve doku nitelikleri kullanılarak benzer olan anlamlı parçaların elde edilmesi hedeflenmiştir. Sistemin başarısını test etmek için bölütleme algoritmasının üreteceği süper piksel miktarı, üç farklı üst sınır ile test edilmiştir. Bölütleme yöntemi ile meydana gelen parçacıklardan renk ve doku nitelikleri çıkartılmış ve destek vektör makinesi (DVM) sayesinde kanlı bölgeleri belirleyebilecek modeller elde edilmiştir. Farklı çekirdek fonksiyonları oluşturularak modellerin başarıları karşılaştırılarak test edilmiştir. Önerilen bu sistemde yaklaşık %97 doğruluk elde edilmiştir.

Bu alanda birçok araştırmacının CNN farklı modellerini kullanmalarının bir nedeni de videolardaki insan davranışlarının sınıflandırmada zamansal özelliklerinin önemidir. Araştırmacılar, uzamsal özellikleri analiz etmek için yalnızca 2B CNN'leri kullanırken (Abdali ve Al-Tuma 2019) 2B CNN'yi LSTM (Uzun Kısa Süreli Bellek) adı verilen bir ağla birleştirerek derin öğrenmeye dayalı gerçek zamanlı bir şiddet dedektörü yöntemi önermişlerdir. LSTM, çoğunlukla önceden eğitilmiş niteliklerin bir bölümü (ardışık kareler) yeniden değerlendirmek için kullanılmıştır. Bu durum tüm ağın hem uzamsal (spatial) hem de zamansal (temporal) niteliklerin kullanılmasına izin vermiştir. Bellek yönetimi bakımından kaynaklar sınırlı olduğundan bu modelin araştırmacılar tarafından faydalı olduğu bildirilmiştir. Önerilen model, Hokey veri kümesi ile test performansı, 131 kare/sn. hızında %98'doğruluk değerini vermiştir.

Arceda vd.nin (2016) yaptıkları çalışmada, Şiddet Akışı (ViF) varyasyonu önerilmektedir. Yinelemeli Yeniden Ağırlıklandırılmış En Küçük Kareler yerine Horn-Schunck optik akış algoritması kullanılmıştır. Sınıflandırıcı olarak Destek Vektör Makinesi (SVM) kullanılarak farklı veri kümeleri üzerinde performans değerlendirmesi yapılmıştır. Hokey veri kümesinde Horn-Schunck daha iyi bir performansa sahipken filmler ve kalabalık veri kümelerinde Yinelemeli Yeniden Ağırlıklandırılmış En Küçük Kareler'in (IRLS) daha iyi olduğu tespit edilmiştir.

Duman ve Erdem'in çalışmasında, derin öğrenme modelleri ile beraber ön tarafta objelerin hız ve yön bilgilerini elde etmek için çerçeveden yoğun optik akışa dayalı öz nitelik çıkarma yöntemi kullanılmıştır. Videolarda anormal durumu belirlemek amaçlanmış, Evrişimli Otomatik Kodlayıcı ve Evrişimli Uzun Kısa Süreli Belleği denetimsiz bir şekilde kullanarak bu amacı gerçekleştirme çalışmaları yapılmıştır. Bunun için Avenue, UCSD Ped1 ve UCSD Peds2'den oluşan üç popüler genel veri seti kullanılmıştır (Duman ve Erdem 2019).

Videolardaki şiddet içeren görüntüleri tespit etmek için yapılan görüntü analizi çalışmalarından biri de Ullah ve arkadaşları tarafından yapılmıştır. Bu çalışmada diğerlerinden farklı olarak kaynak kullanımını azaltmak için işe yaramayan görüntüleri ortadan kaldırma yoluna gidilmiştir. Bunun için insan içeren görüntü çerçevelerini tespit edip bunlara Evrişimsel Sinir Ağları kullanılarak şiddet içeren ya da içermeyen olarak ikili sınıflama yapılmıştır. Yani önce karelerdeki kişileri algılamak için 2B CNN'den yararlanılmış, ardından ise şiddet belirleme amacı ile sadece insanları içeren kareler 3B CNN beslenerek bir model tasarlanmıştır. Önerilen modelde veri kümesi olarak Hockey Fight ve Crowd Violence kullanılmıştır ve test edilmiştir. Doğruluk değerleri sırasıyla %96 ve %98 şeklinde elde edilmiştir (Ullah vd. 2019).

BÖLÜM II

YAPAY ZEKÂ VE YAPAY SİNİR AĞLARI

2.1 YAPAY ZEKÂ

Yapay zekâ (YZ) terimi ilk duyulduğunda anlaması ve anlatması bakımından birçok karışıklığa ve yanlış anlamalara neden olacaktır çünkü zekâ kavramının kendisi göreceli, kişiden kişiye değişen, soyut bir kavramdır. Bundan dolayı yapay zekâyı bilimsel yönden tanımlamak kolay olmayacaktır.

Yapay zekâ; bir bilgisayara ya da bilgisayar kontrollü bir makineye insan zekâsına özgü nitelikler olduğu varsayılan algılama (görme, işitme), anlam çıkartma, akıl yürütme ve geçmiş deneyimlerden tecrübe edinme gibi zihinsel işlevleri yerine getirme yeteneği kazandırmak olarak tanımlanmaktadır (Nabiyev 2016: 25).

Yapay zekâ fikri yüzlerce yıl öncesine uzanmaktadır. İnsanlar mekanik ve otomatik olarak ihtiyaçlarını karşılamaya yönelik makineler yapmak için uğraşmışlardır. İlk bilgisayarlar 1940'larda ortaya çıkınca bu ihtiyaçlar ve istekler daha da artmış, böylece işin içerisine elektronik ve yazılım da karışmıştır. Yapay zekâ kavramı 1956'ya kadar resmî olarak oluşturulamamıştır.

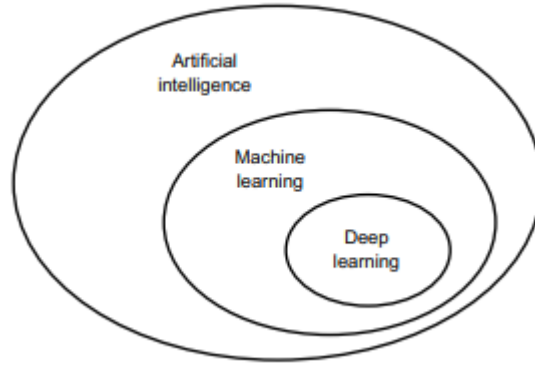
Yapay zekâ kavramı ilk kez 1956 yılında Dortmund Konferansı'nda John McCarthy, Marvin L. Minsky, Nathaniel Rochester ve Claude E. Shannon tarafından verilen bir öneri mektubunda ortaya atılmıştır. Ancak genellikle yapay zekâ kavramının mucidi olarak John McCarthy kabul edilmektedir. Bu alanda yapılan araştırmalar dijital bilgisayarlar kadar eskidir (Alpaydın 2014: 309).

1956 yazında Dartmouth Colledge'de düzenlenen konferansta yapay zekâ kavramı tartışılmıştır. Bu konferansa katılan John McCarthy yapay zekâ kelimesini kullanım için önermiştir. Jon McCarthy ile birlikte konferansa katılım sağlayanlar arasında Marvin Minsky, Allen Newell ve Herbert Simon da yapay zekânın öncülerinden kabul edilmektedir.

Yapay zekâ 1950'li yıllarda gelişmeye başlamış ve "Bilgisayarlar düşünebilir mi?" sorusuyla karşılaşan bilgisayar bilimcileri hâlen bu sorunun cevabını

aramaktadırlar. Yapay zekâ kısaca, insanlar tarafından düşünülerek yerine getirilen faaliyetleri bilgisayarlar aracılığıyla otonom hâle getirmektir (Chollet 2021: 2).

Makine öğrenmesi ve derin öğrenmeyi kapsayan yapay zekâ genel bir alandır.



Şekil 1: Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme
(Chollet 2021)

1997 yılında Deep Blue adlı bilgisayarın dünya satranç şampiyonu Garry Kasparov’u yenmesinin yapay zekâ alanında bir dönüm noktası olması, 2014 yılında Google’ın sürücüsüz otomobil projesi için çalışan ilk gerçek araç prototipini test sürüşüne çıkarması, 2011 yılında Apple firması tarafından üretilen yapay zekâ kullanan ilk kişisel asistanı olan Siri’nin IOS işletim sistemli akıllı telefonlarda kullanıma sunulması, 2016 yılında Google’ın yapay zekâ ile uğraşan firmalarından olan DeepMind tarafından geliştirilen AlphaGo adlı yapay zekânın satrançtan daha zor olan “Go” adlı oyunda dünya şampiyonunu yenmesi, 2017 yılından bu yana uzay görevlerinde keşif yapmak için kullanılması, yapay zekânın kendisinden daha da söz ettirmesine neden oldu (Tan 2019: 4).

Yapay yaratıcılık faaliyetleri, akıl yürütme faaliyetleri, yüz tanıma sistemleri, konuşma tanıma, bilgisayarlı görü, görüntü işleme, zeki sözcük tanıma, nesne tanıma, el yazısı tanıma, yüz tanıma, hastalığa tanı koyma, uzman sistemler, yazı madenciliği ve soru yanıtlama sistemleri alanları yapay zekânın kullanıldığı alanlardan bazılarıdır.

Yapay zekânın alt alanlarını oluşturan pek çok konu vardır:

- Makine Öğrenmesi
- Yapay Sinir Ağları ve Derin Öğrenme
- Bulanık Sistemler (Fuzzy Logic)
- Evrimsel Yöntemler (Genetic Algoritmalar, Differential Evolution)

- Üst Sezgisel (Meta Heuristic) Yöntemler (Karınca Kolonisi, Particle Swarm Optimization)
- Olasılıksal (Probabilistic) Yöntemler (Bayesian Network, Hidden Markov Model, Kalman Filter)

Bu alanda birçok uzman yapay zekânın kodlarla bilgiyi işleme ve değiştirilmesinin mümkün olabileceği düşüncesi, 1950'lerin ortalarından 1980'lerin sonuna kadar yapay zekâ araştırmalarının baskın paradigmasıydı. Bu yaklaşıma, sembolik yapay zekâ denmekteydi. Sembolik yapay zekânın iyi tanımlanmış problemlerde başarı elde ettiği görülse de ses tanıma, diller arası çeviri yapma, görüntü sınıflandırma gibi daha karmaşık alanlarda istenen sonuçları veremediği görüldü. İlerleyen yıllarda karmaşık problemlere çözüm üretebilen makine öğrenmesi ortaya çıktı (Chollet 2021: 3).

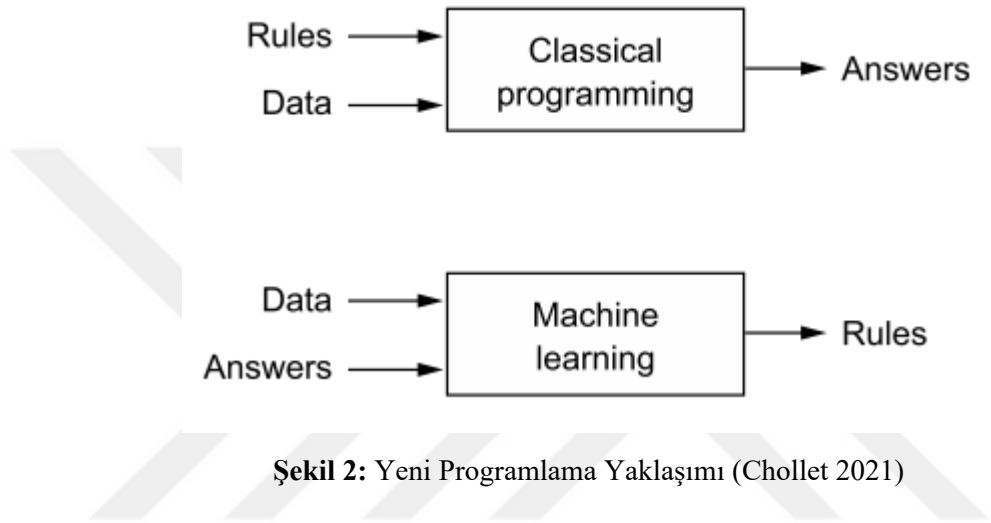
2.1.1 Makine Öğrenmesi

Öğrenme bir süreçtir ve bu süreç içerisinde davranışlarda değişim olması beklenmektedir (Bilgin 2018: 13). Pek çok süreci içerisinde barındırdığından öğrenmeyi de zekâ gibi tanımlamak zordur. Öğrenme, sözlükte “Çalışma, eğitim veya deneyimle bilgi, anlayış ya da yetenek kazanmak.” ve “Deneyimle davranışsal eğilimdeki değişiklik.” olarak tanımlanmaktadır (Nilsson 1996: 1).

Makine öğrenmesi, bilgisayarların verilerden öğrenip edinmiş olduğu deneyimlerle yapıyı iyileştirdiği yapay zekânın (AI) bir alt kümesidir. Makine öğrenmesi bilgisayar, istatistik ve veri tabanı bilimi disiplinlerinin kesişim noktasında ortaya çıkmıştır.

Yapay zekânın bir alt dalı olan makine öğrenmesi, derin öğrenmeyi de içermektedir. Makine öğrenmesi bir problemi geleneksel programlama mantığı ile çözmeyen bunun yerine bu problemin çözümlerini kendi kendine öğrenebilen bilgisayar sistemidir. Makine öğrenmesinin ana problemi, kendi kendine öğrenebilen bilgisayar sistemleri yöntemlerini incelemek, bu sistemleri geliştirmek ve bunları uygulamaktır. Bunun için matematiksel ve istatistiksel yöntemlerden yararlanarak veriler üzerinden çıkarımlar yapmak ve tahminlerde bulunan sistemler oluşturmaktır. Bu sistemde öğrenme süreci, içerisinde başarı ölçülerini artıracak uygun problemi en iyi sonuçla çözen modeller oluşturmaktır. Yöntem, daha önceden belirlenen kurallar

sınırlarında giriş verisi ile bir çıktının üretildiği geleneksel programlamanın aksine, giriş ve çıktılar arasındaki bağlantıya göre kuralların oluşturulmasına dayanmaktadır. Klasik yapay zekâ algoritmalarında sembolik yaklaşımlar kullanılmaktadır. Girdi olarak veriler ve kurallar sunulmakta, veriler ve bu kuralların adımları izlenerek bir çıktı elde edilmektedir. Makine öğrenmesinde farklı bir yol izlenir: Girdi olarak veri ile beraber veriden beklenen çıktı sunulur. Çıktı olarak da kurallar elde edilir. Bu kurallar yeni verilere uygulanarak yeni bir çıktı üretilmesi beklenir (Chollet 2021: 4).



Şekil 2: Yeni Programlama Yaklaşımı (Chollet 2021)

Bir makine öğrenmesi yöntemi, girişlere karşılık tahmin etmek için çeşitli türde çıktı üretebilir. Bu çıktıya kategorik ise sınıflandırma (classification) problemi, nümerik ise regresyon (regression) problemi adı verilir. Açıklayıcı bir modelleme olan kümeleme (Clustering) ise benzer özellikleri en çok olanları aynı kümelere atama işlemidir. Birliktelik kuralları (Association Rules) ile gözlemler arasındaki basit ve karmaşık bağlantılar bulunabilir. Sınıflandırma ve regresyon problemleri eğitim verisine gereksinim duyduğu için gözetimli (supervised) öğrenme olarak tanımlanırken kümeleme ve birliktelik kuralları, eğitim verisine gerek duymadığı için gözetimsiz (unsupervised) öğrenme olarak tanımlanır.

Makine öğrenmesi yöntemleri; veri madenciliği, doğal dil işleme, görüntü işleme ve robotik gibi birçok alanda kullanılarak bu alanlarda çözümler üretmiştir. Ayrıca dolandırıcıları tespit etmek için bankacılık sektöründe, kanserli hücre tespitiyle de sağlık alanında kullanılmaktadır.

2.1.1.1 Makine Öğrenmesi Türleri

Makine öğrenmesi gözetimli, gözetimsiz ve pekiştirmeli olarak temelde üç kategoriye ayrılabilir.

2.1.1.1.1 Gözetimli Öğrenme (Supervised Learning)

Makine öğrenmesi, elimizdeki veriyi kullanarak tahminde bulunmaktır. Veri setinin içerisinde sonuçların olduğu varsayıldığında bu verilere göre oluşturulan model, yeni gelen veriyi tahmin ediyor ise bu öğrenme biçimine “denetimli öğrenme” denilmektedir. Kullanılan algoritma, girdi-çıkı verilerinin etiketli şekilde kullanılmasıyla eğitilir. Daha sonra eğitim verilerindeki desenler tanır ve yeni verilerle karşılaştırılarak doğru çıkarımlar yapılır.

Gözetimli öğrenme problemleri, temelde Regresyon (Regression), Sınıflandırma (Classification) diye ikiye ayrılabilir:

Regresyon (Regression): Nümerik değerler verilir ve bu değerler arasındaki bağlantılar yardımıyla bir matematiksel fonksiyon elde edilir, gelen verilere göre bu fonksiyon yardımıyla sonuç tahmini yapılır. Regresyon, eldeki veriler görsel olarak ifade edildiğinde bunların arasındaki bağlantı eğrisini çizmek olarak da düşünülebilir.

Bazı regresyon algoritmaları şunlardır:

- Doğrusal Regresyon (Linear Regression)
- Çoklu Doğrusal Regresyon (Multilinear Regression)
- Polinomsal Regresyon (Polynomial Regression)
- Lojistik Regresyon (Logistic Regression)
- Karar Ağacı Regresyonu (Decision Tree Regression)
- Rastgele Ormanlar Regresyonu (Random Forest Regression)
- Destek Vektör Regresyonu (Support Vector Regression)

Sınıflandırma (Classification): Veri kümesindeki elemanların parametrelerine bakılarak onları sınıflara ayırmaya çalışan bir algoritmadır. Çıkış değeri kategorik olup iki veya daha fazla gruba sahiptir. Örneğin bir resmin köpek içerip içermediği tahmini vb.

Bazı sınıflandırma algoritmaları şunlardır:

- Naif Bayes Sınıflandırılması (Naive Bayes Classifier)
- Karar Ağacı Sınıflandırılması (Decision Tree Classifier)
- K-En Yakın Komşu (K-Nearest Neighborhood)

- Destekçi Vektör Makinesi (Support Vector Machine)
- Rastgele Orman Sınıflandırması (Random Forest Classification)

2.1.1.1.2 Gözetimsiz Öğrenme (Unsupervised Learning)

Gözetimsiz makine öğrenmesi, verileri etiketlemeden analiz etmek ve kümelemek amacıyla makine öğrenimi yöntemlerini kullanır. Bu yöntem, verilerdeki desenleri ve ilişkileri bularak öğrenme sürecini gerçekleştirir. Etiketlenmemiş verilerin kullanılması, algoritmanın veri setindeki yapıları ve nitelikleri öğrenmesine olanak tanır. Bu algoritmalar, insan müdahalesine gereksinim duymaksızın gizli kalıpları veya veri gruplamalarını ortaya çıkartır. Algoritma, veriler arasındaki benzerlikleri ve farklılıkları bulup bu verileri gruplandırır. Gözetimsiz öğrenme çoğunlukla kümeleme problemleri çözümünde kullanılmaktadır.

Kümeleme, veri kümesindeki verileri benzerliklerine ve farklılıklarına göre otomatik olarak gruplara ayırır. Anormali tespiti, veri setindeki olağan dışı veri noktalarını bulabilir. Bu yöntemden bankacılık alanında hileli işlemleri bulmak için yararlanır, müşteri segmentasyonu ve görüntü tanıma için çözüm üretmektedir.

Çok fazla veri, genellikle daha doğru sonuçlar verirken diğer taraftan makine öğrenimi algoritmalarının performansını düşürebilir, veri kümelerinin görselleştirilmesini zorlaştırabilir. Veri setindeki niteliklerin veya boyutlarının miktarı çok fazla olduğunda boyut küçültme yöntemi kullanılabilir. Bu teknik, veri setinin bütünlüğünü mümkün olduğunca korurken veri girişlerinin miktarını yönetilebilir bir boyuta indirir.

2.1.1.1.3 Pekiştirmeli Öğrenme(Reinforcement Learning)

Bu tür makine öğrenmesi, öğrenmenin sonucu karşılığında alınan ödüllere dayanır. Çevreyi gözetleyen ve ajan adı verilen öğrenme sistemi, işlemleri seçip gerçekleştirebilir buna karşın ödüller alabilir (Aurélien Géron 2023: 39). Algoritma, amaçlanan görev için en doğru sonucu üreten stratejileri bulmak için deneme-yanılma yöntemini kullanır. Bunun için bir karar verme ağacı oluşturarak ödül kazandıran eylemlerin sıklığını artırır.

Makine öğrenmesi sürecinde ham veriler bulunur ve algoritmaya verilir. Bu veriler genel olarak sayısal değerler, puanlamalar ve metin gibi bilgilerdir. Ardından sırasıyla; özellik çıkarımı, model oluşturma, değerlendirme, tahmin adımları izlenir. Başarılı sonuçlar elde edilebilmesi için değerlendirme aşaması önem arz etmektedir.

Bu aşamada doğruluk (accuracy), hassaslık (sensitivity), belirginlik (specificity) gibi testler yapılır ve en iyi çıkarımın yapıldığı gözetlenir. Eğer istenilen sonuç elde edilmez ise model yeniden değerlendirilerek gerekirse yeni model oluşturulup en iyi sonuç bulunana kadar sürdürülür.

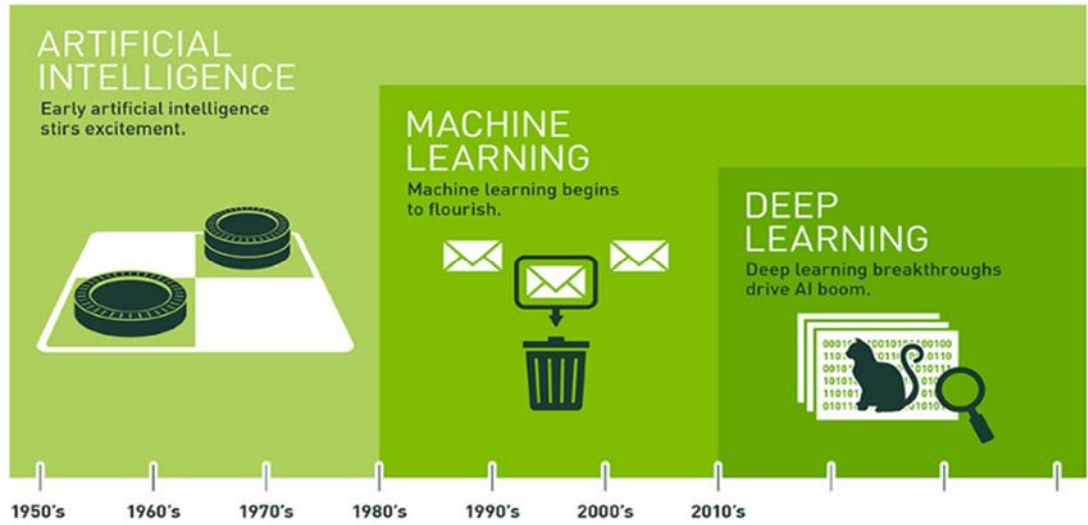
Makine öğrenmesi, insanlar veri analizi yaparken ve karar verirken onların yerini almamalıdır ancak bu öğrenme teknolojileri insan kontrolü altında kullanılarak iş süreçlerinde verimliliği artırma ve karar verme aşamalarında yardım alınabilir.

2.1.2 Derin Öğrenme

Yapay sinir ağlarını temel alan ve popülerliği giderek artan derin öğrenme, yapay zekâ ve makine öğrenmesinin bir alt kümesidir. Günümüzde yapay zekânın bu denli önem kazanmasına en çok katkıda bulunan da derin öğrenme yöntemleridir.

Derin öğrenme; doğal dili işleme, yüz tanıma, konuşma ve tanıma gibi alanlarda birden çok katmanda çalışabilen yapay sinir ağlarını kullanan yapay zekâ yöntemi olup makine öğrenmesinin alt dalıdır (Yılmaz ve Kaya 2019: 1).

Veri ne kadar çok olursa yapay zekâ özellikleri de o kadar iyi açığa çıkacaktır, yapı ise karmaşık hâle gelecektir. Yapı karmaşık hâle geldikçe yapay zekâdan, makine öğrenimine doğru yönelim meydana gelecektir. Yapı daha da karmaşıklaşınca makine öğreniminden derin öğrenime doğru geçişler başlayacaktır. Veri büyüdükçe sistem çok iyi çalışacaktır. Makine öğrenimi tek katmanda çalışırken derin öğrenme çok katmanda ve aynı anda çalışacaktır. Basitçe söylemek gerekirse derin öğrenme, çok katmanlı yapay sinir ağına verilen addır.



Şekil 3: Yapay Zekâ, Makine Öğrenimi ve Derin Öğrenmenin Tarihi Gelişimi (Nvidia)

Derin öğrenme kavramı, 2006 yılında Geoffrey Hinton tarafından çok katmanlı yapay sinir ağlarının daha verimli eğitilebileceği fikrini öne sürmesi ile ilk kez duyulmuştur (Yılmaz ve Kaya 2019: 1).

Derin öğrenmede, verilerin birden çok özellik seviyesinin veya temsillerinin öğrenilmesine göre oluşturulan bir yapı vardır (Şeker vd. 2017: 48).

Derin öğrenme yöntemleri, ham veriden başlayarak her biri bir seviyedeki bir temsili; daha yüksek, daha soyut seviyedeki bir temsile dönüştüren, basit fakat doğrusal olmayan modüllerin oluşturulmasıyla elde edilen, çoklu temsil seviyelerine sahip öğrenme yöntemleridir. Karmaşık işlemler, bu tür dönüşümlerin yeterli sayıda bir araya getirilmesiyle öğrenilir (Lecun vd. 2015: 436).

2.2 YAPAY SİNİR AĞLARI

Yapay Sinir Ağları (YSA), biyolojik sinir ağlarından ilham alarak bilgi işleyen paradigmadır. Yapay sinir ağları, insan beyninin niteliklerinden olan öğrenme yolu ile yeni bilgiler oluşturabilme ve keşfedebilme gibi insana özgü yetenekleri herhangi bir yardım almadan, otonom olarak gerçekleştirmek amacı ile geliştirilen bilgisayarlardır (Öztemel 2020: 29). Yapay sinir ağı, biyolojik olarak insan beyninden ilham alınmış ve bilgiyi işlemek üzere dizayn edilmiş bir öğrenme algoritmasıdır. Aynı insan beyninde olduğu gibi yapay sinir ağları yöntemleriyle makinelerin eğitilmesi, öğrenmesi ve karar vermesi amaçlanmaktadır. Yapay sinir ağı; makine öğrenmesi modellerinden biridir ve derin öğrenmenin merkezinde bulunmaktadır. Güçlü, çok yönlü ve ölçeklenebilir olmaları karmaşık ve zor makine öğrenme problemlerinin çözümlemesinde sıklıkla kullanılmaktadır.

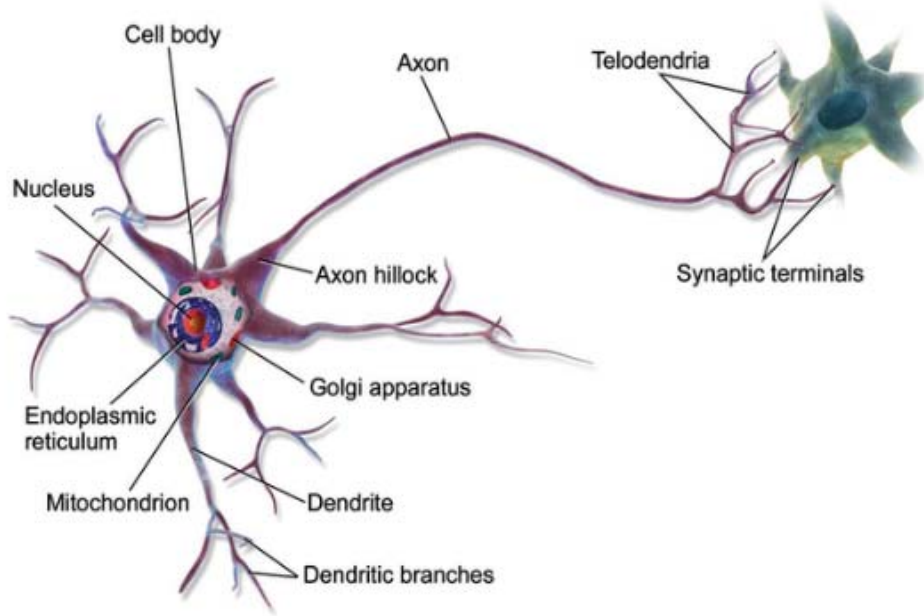
Biyolojik sinir sistemi oldukça karmaşık bir yapıya sahiptir. Yapay sinir ağı modellenmesi, bu karmaşıklığı soyutlamaya ve bilgi işleme açısından teorik olarak en önemli olabilecek şeylere odaklanmaya çalışmaktadır.

Yapay sinir ağlarının temelleri, 1940'ların başlarında atılmıştır. 1943 yılında, matematikçi Warren McCulloch ve Walter Pitts, insan beyninin işleyişine benzemeye çalışan basit algoritmaları içeren bir devre sistemi oluşturarak bu alanda ilk adımları atmışlardır.

2.2.1 Biyolojik Sinir Hücreleri

Yapay sinir hücreleri, biyolojik sinir hücrelerinden ilham alınarak geliştirilmiştir. İnsanlarda bulunan biyolojik bir sinir hücresi; bir gövde (Cell body),

bir çekirdek (Nucleus), çok sayıda sinir ucu (dendrite) ve aksondan (Axon) oluşur. Komşu iki sinir hücresi arasında kalan ince uzantılar (sinaps) aracılığıyla iletişim kurulur ve bilgi iletimi yapılır.

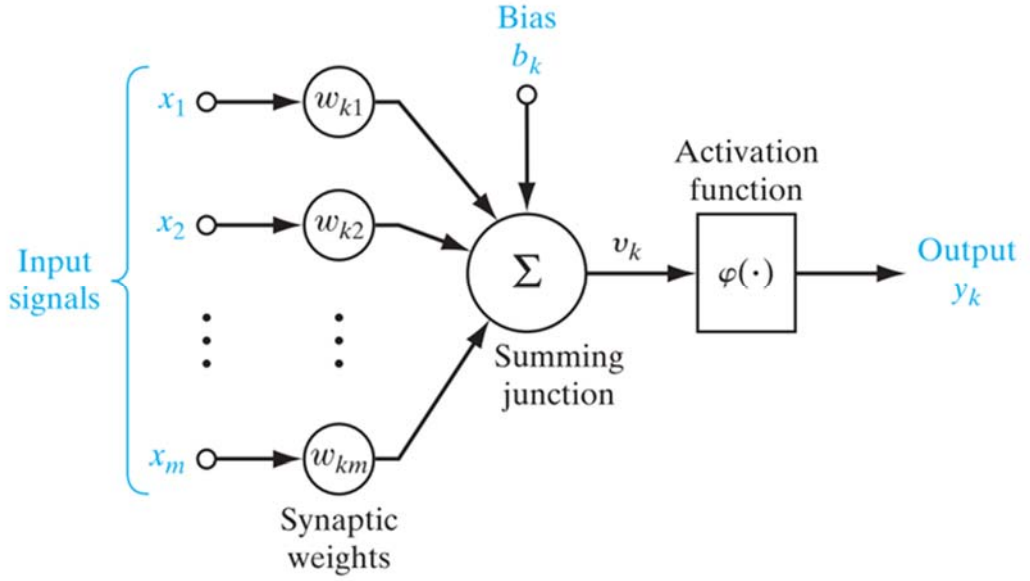


Şekil 4: Biyolojik Sinir Hücresi (Aurélien Géron 2023)

Biyolojik sinir ağı, nöron isimli bilgi işleme kabiliyeti olan sinir hücrelerinin bir araya gelmesinden oluşmaktadır. Nöronlar sinir hücreleri olarak tanımlanır ve birbirleriyle bağlanarak fonksiyonlarını yerine getirir. Sinir sistemi, her biri bir gövde ve bir sinir lifi (akson) içeren sinir hücresinden oluşan bir ağdır. İki sinir hücresi arasındaki bağlantılar ya da sinapslar her zaman bir sinir hücresinin aksonu ile diğerinin gövdesi arasındadır. Herhangi bir anda bir nöronun, bir atak başlatabilmesi için uyarımın aşması gereken bir eşik değeri vardır ve bu değer aşılmamalıdır. Bu, uyarım tarafından değil nöron tarafından belirlenir (Pitts 1943: 115).

2.2.2 Yapay Sinir Hücreleri

Yapay sinir ağları, biyolojik sinir ağlarının modelini oluşturarak bilgisayarların insanlar gibi öğrenmesini sağlamaya çalışır. Yapay sinir ağının temelini yapay nöronlar oluşturur. Sinir sisteminin esas bölümünü oluşturan sinir hücreleri olan nöronların taklit edilmesi neticesinde yapay nöron kavramı ortaya çıkarılmıştır (Yılmaz ve Kaya 2019: 30).



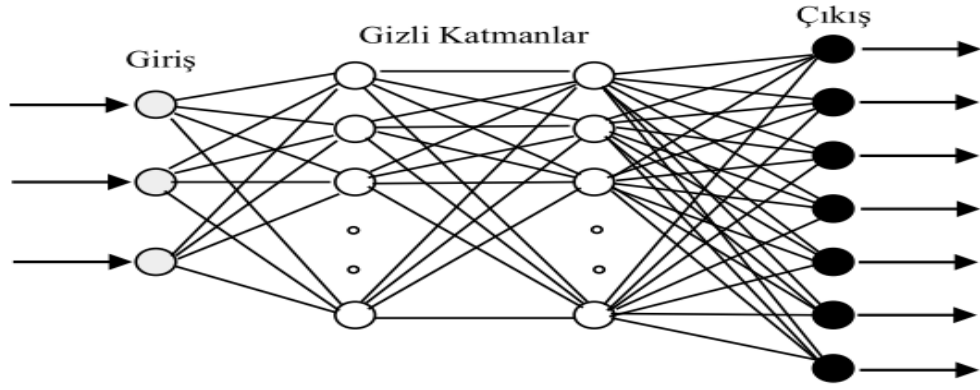
Şekil 5: Bir Nöronun Doğrusal Olmayan Modeli (Haykin 2018)

Girdiler (x_1, x_2, \dots, x_m), diğer yapay sinir hücrelerinden ya da dış ortamdan hücreye giren verilerdir.

Bu girdiler, ağırlık öğrenmesi istenen veriler tarafından belirlenir. Ağırlıklar (w_1, w_2, \dots, w_m), girdi kümesi veya kendinden önceki bir tabakada bulunan başka bir işlem elemanının bu işlem elemanı üzerindeki etkisini gösteren değerlerdir. Her bir girdi, o girdiyi işlem elemanına bağlayan ağırlık parametre değeriyle çarpılarak toplam fonksiyon vasıtasıyla birleştirilir.

$$y_k = \sum_{j=1}^m (w_{kj} x_j) + b_k \quad (2.1)$$

Yapay sinir hücresi, ağırlıklı girdilerin toplamını hesaplar; aktivasyon fonksiyonuna uygulayarak elde ettiği çıktıyı bir sonraki yapay sinir hücresine aktarır. Algılayıcı, her bir girdinin bir nörona bağlanmasıyla meydana gelir. Girdilerin en az bir nörona ve en az bir çıktıya bağlandığı yapıya “yoğun katman” veya “tam bağlanmış katman” denir. Bu şekilde oluşan modele ise “çok katmanlı algılayıcı (MLP)” denir. MLP girdi katmanı, gizli katman ve çıktı katmanından oluşmaktadır. Gizli katman en az bir algılayıcı içeren katman ya da katmanlar olarak ta bilinmektedir (Çetinkaya 2022: 12).



Şekil 6: Çok Katmanlı Yapay Sinir Ağının Genel Yapısı (Atalay ve Çelik 2017)

Yapay sinir hücreleri, mühendislikte “işlemci eleman” olarak adlandırılmakta ve bu hücrelerin beş temel işlevi vardır (Öztemel 2020: 48)

2.2.2.1 Yapay Sinir Ağlarının Temel Elemanları

Yapay sinir ağları, çok sayıda birbirine bağlı yapay sinir hücrelerinden oluşmuş, paralel olarak çalışan yapılardır. Yapay sinir ağlarının temel yapı taşları yapay bir sinir hücresidir. Bu hücre, biyolojik sinir hücresine göre daha basitleştirilmiştir ve biyolojik sinir hücresinin 4 temel işlevini taklit etmektedir (Elmas 2021). Biyolojik sinir ağları yapay sinir hücrelerinden oluştuğu gibi yapay sinir ağları da yapay sinir hücrelerinden oluşmaktadır ve her yapay sinir hücresi 5 temel elemandan oluşmaktadır (Öztemel 2020: 49). Bunlar girişler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkışlardır.

Girişler, yapay nörona çevreden ya da başka nöronlardan gelen bilgileri sinir hücresine getirir. Bir sinir hücresi gelişigüzel birçok girişe sahiptir.

Her yapay sinir hücresinin her girişi için bir ağırlık değeri tanımlanmıştır. Girişlerin yapay sinir hücresi üzerindeki önemini ve etkisini belirleyen ise katsayılarıdır. Ağırlık parametrelerinin değerleri pozitif de olabilir, negatif de olabilir (Caner ve Akarşlan 2009: 223). Yapay sinir hücreleri arasındaki bağlantının gücünü ağırlık değeri belirler.

Toplama işlevi, birçok fonksiyon kullanılarak hesaplanır. En yaygın olanı yapay sinir hücrelerinin her bir girişi (x_i) ile o girişlere ait ağırlıkların (w_i) çarpımlarının toplamıdır. Bu toplam, aktivasyon fonksiyonuna gönderilmektedir. Birçok uygulamada eşik değeri (b_k), toplama dâhil edilmektedir. b_k eşik değeri

sıfırdan farklı bir değer seçilerek giriş değerlerinin hepsi sıfır olması durumunda çıkış değerinin sıfır olmasını engellemektedir.

Aktivasyon fonksiyonu, işaretini çıkış işaretine dönüştürmek için kullanılan bir fonksiyondur ve bu fonksiyonun çıkışı yapay sinir hücresinin çıkışıdır. Bu da yapay sinir ağında bir sonraki katmandaki giriştir. Bir ağdaki bütün yapay sinir hücrelerinin aynı aktivasyon fonksiyonunu kullanması gerekmez. Aktivasyon fonksiyonu olarak çıkış değerini hesaplamak amacıyla farklı fonksiyonlar kullanılabilir. En fazla kullanılan aktivasyon fonksiyon çeşitleri “lineer, rampa, basamak, sigmoid, relu ve softmax” aktivasyon fonksiyonlarıdır.

Yapay sinir hücresinin çıkışı aktivasyon fonksiyonu çıkışında meydana gelen değerdir ve bu, çıkış değeridir. Giriş değeri birden çok olmasına karşın çıkış değeri 1 tane dir.

Tablo 1: Yapay Sinir Ağı ile Biyolojik Sinir Ağı Karşılaştırılması (Elmas 2021)

Biyolojik Sinir Ağı	Yapay Sinir Ağı
Sinir Sistemi	Sinirsel Hesaplama Sistemi
Sinir	Düğüm (Sinir, İşlem Elemanı)
Sinaps	Sinirler arası Bağlantı Ağırlıkları
Dentrit	Toplama İşlevi
Hücre Gövdesi	Etkinlik İşlevi
Akson	Sinir Çıkışı

2.2.2.2 Yapay Sinir Ağı Mimarileri

YSA, ağ mimarisinde temel olarak üç değişik yapıyı gösterir. Bunlar tek katmanlı ileri beslemeli, çok katmanlı ileri beslemeli ve geri beslemeli ağ mimarileridir (Haykin 2018: 21). YSA katman sayılarına göre tek ve çok katmanlı olarak ikiye ayrılır:

2.2.2.3 Tek Katmanlı Algılayıcılar (TKA)

Tek katmanlı yapay sinir ağları, giriş ve çıkış katmanlarından meydana gelir; doğrusal olmayan problemlerde tercih edilmez. Ayrıca, birden çok giriş değerini içerebilirler. Şekil 5’te tek katmanlı algılayıcı modeli gösterilmiştir.

Tek katmanlı yapay sinir ağının çıkışının sıfır olmasını önleyen bir eşik değeri vardır. Bu değer daima birdir. Bu ağın çıktısının matematiksel denklemi (2.2)’deki gibi

ateşleme değeri ağırlık parametrelerinin girdi değerlerinin toplanması sonucu elde edilir.

$$y_k = f\left(\sum_{i=1}^n w_i x_i + b_k\right) \quad (2.2)$$

Çıktı fonksiyonu tek katmanlı algılayıcıda doğrusaldır. Bu durumda, ağa verilen örnekler eşik değeri fonksiyonu ile iki sınıfı birbirinden ayıran doğru bulunmaya çalışılır. Ağın çıktısı 1 ya da -1 değeri alır.

$$f(g) = \begin{cases} 1, & y_k > 0 \\ -1, & y_k \leq 0 \end{cases} \quad (2.3)$$

İki girdili ve tek çıktılı bir tek katmanlı algılayıcıda iki sınıfı birbirinden ayıran doğruyu bulmak için

$$w_1 x_1 + w_2 x_2 + b_k = 0 \quad (2.4)$$

denklemini yazabiliriz. Bu denklemde doğru x_1 ve x_2 değerlerini bularak geometrik olarak çizilebilir.

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \quad (2.5)$$

$$b_k(t+1) = b_k(t) + \Delta b_k(t) \quad (2.6)$$

İki sınıfı birbirinden ayıran en iyi doğruyu bulurken ağırlık değerleri 2.5 formülü ile her iterasyonda değiştirilerek en uygun ağırlık değeri bulunabilir. Eşik değeri, benzer şekilde 2.6 formülüyle de her iterasyonda güncellenebilir.

2.2.2.4 Çok Katmanlı Algılayıcılar (ÇKA)

ÇKA, bir girdi katmanı, birden çok gizli katman ve çıktı katmanı içerir. Çok katmanlı algılayıcıların ortaya çıkma sebebi bazı yöntemlerin yetersiz kalmasıdır. ÇKA'nın genel yapısı Şekil 6'da verilmiştir. Giriş katmanında hesaplama için nöronlar

bulunmaz, burası giriş katmanıdır ve sadece birinci ara katmandaki nöronları beslemek içindir. Bu katmanı, diğer ara katmanlar takip eder. Son olarak da çıkış katmanı yer alır. Aynı katmandaki nöronlar arasında bir bağ bulunmamaktadır. Çok katmanlı algılayıcılarda katmanlar arası ileri yayılım ve geri yayılım olarak bilinen geçişler bulunur. İleri yayılım aşamasında, ağın çıktısı ve genellikle belirlenen bir hedefle karşılaştırılarak hata değeri hesaplanır. Geri yayılım aşamasında ise hesaplanan hata değerinin en aza indirilmesi için katmanlar arası bağlantı ağırlık parametre değerleri güncellenir.

İleri beslemeli yapılarda her nöron, bitişikteki katmanın tüm nöronları ile bağlantılıdır ancak diğer nöronlarla bağlantılı değildir. Bu özellik, bilgi ilerleyişinin bir yöne doğru yani giriş katmanından çıkış katmanına doğru ilerlediği anlamına gelir.

Bu ağ yapısı, her katmandaki sinir hücrelerinin sadece bir sonraki katmanda bulunan sinir hücreleri ile bağlantılı olduğu anlamına gelir. Katman içindeki sinir hücreleri arasında doğrudan bağlantılar bulunmaz. Bu, ağı basitleştirir ve matematiksel işlemleri daha yönetilebilir kılar. Bu tür bir yapının avantajlarından biri, parametre sayısının kontrol altında olmasıdır. Her nöronun yalnızca bir sonraki katmandaki nöronlarla bağlantısı olduğu için, modelin karmaşıklığı daha iyi yönetilebilir. Ancak, daha karmaşık problemler için genellikle daha derin ve karmaşık ağ yapıları gerekebilir.

ÇKA modeli doğrusal algılayıcıda en küçük kareler algoritmasının geliştirilmiş hâli olan geri yayılım (backpropagation) öğrenme algoritmasını kullanır.

Geri yayılım (backpropagation) yöntemi, ağın çıktı değerinin belirlendiği ileri besleme ve meydana gelen hatanın gradiyent değerini azaltmak için geri yayılarak ağırlık parametrelerinin güncellendiği geri besleme adımlarından oluşmaktadır.

2.2.2.5 Yapay Sinir Ağlarının Genel Özellikleri

Yapay sinir ağları, aktivasyon fonksiyonları sayesinde doğrusal olmayan gerçek dünya özelliklerini tanıtmak amacıyla kullanılır. Bundan dolayı bu ağların doğrusal olmama özelliği bulunmaktadır.

Yapay sinir ağlarının toplanmış ve biriktirilmiş mevcut verileri öğrenebilme; bu verilerden faydalanarak kendi tecrübesini oluşturabilme, bu bilgi ve tecrübelerine dayanarak insanlara benzer şekilde tahminde bulunabilme özellikleri bulunmaktadır.

Yapay sinir ađı, veri setindeki desenleri öğrenerek istenilen görevi yerine getirebilecek genelleştirmeleri yapar (Atalay ve Çelik 2017: 162). Yapay sinir ağlarının genelleştirme özelliđi vardır.

Bu ağlar hata toleransı ve esnekliđine sahiptir. Kullanılan deđişken ve parametre miktarına göre bu hata toleransı deđişmektedir.

2.2.2.6 Aktivasyon Fonksiyonları

Yapay sinir hücrelerinin çıkışını tespit etmek için aktivasyon fonksiyonları kullanılır ve bu fonksiyon ile ilişkili olarak 0 ila 1 veya -1 ila 1 arasında bir çıktı üretilir (Alkayal 2023: 11).Yapay sinir ağlarında aktivasyon fonksiyonlarına ihtiyaç duyulmasının en önemli sebebi, gerçek dünyadaki doğrusal olmayan özellikleri tanıyabilmektir.

Aktivasyon fonksiyonları yapay sinir ağlarında performansı önemli ölçüde etkilediđi için performansı artırmak amacıyla kullanılan aktivasyon fonksiyonlarına ait özelliklerin bilinmesi önemlidir.

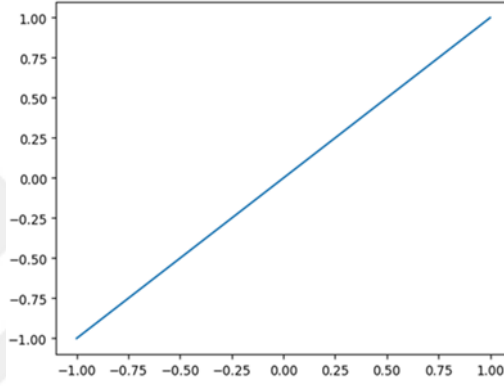
YSA'lar, deđiştirilebilir parametreleri olan basit işlem birimlerinden oluşur ve öğrenme yeteneđine sahiptir. YSA bir matematiksel modeldir. Modelin öğrenmesi, ağırlık diye adlandırılan parametrelerin ayarlanması ile gerçekleştirilir. Modelin önemli hiper parametrelerinden bir tanesi de aktivasyon fonksiyonudur. Bu fonksiyon her bir sinir hücresinin çıkışında bulunur ve doğrusal olmama özelliđini de sağlamaktadır. YSA başarısını, aktivasyon fonksiyonu seçimi büyük oranda deđiştirebilmektedir (Kaytan vd. 2022: 30).

Bir yapay sinir ağında aktivasyon fonksiyonları kullanılmazsa çıkış deđeri birinci dereceden bir fonksiyon olacaktır. Bu da ancak basit verilere sahip çözümlerde kullanılabilir. Yapay sinir ađı, doğrusal regresyon modeli gibi davranır. YSA videoların; ses, metin gibi karmaşık veri türlerini modellemek gibi çok daha karmaşık işlemleri yerine getirmesi istenmektedir. Birden fazla gizli katmana sahip yüksek boyutlu, karmaşık ve doğrusal olmayan modellerin çözümü ya da gerçek dünya özelliklerini tanıtmak için aktivasyon fonksiyonları kullanılmaktadır.

Ađı dinamik hâle getirmek, veriden karmaşık bilgileri çıkarmak ve giriş ile çıkış arasında doğrusal olmayan kıvrımlı rastgele fonksiyonel eşlemeleri temsil etme özelliđi eklemek için aktivasyon fonksiyonu eklemek gerekmektedir (Sharma vd. 2020: 310). Doğrusal, basamak aktivasyon fonksiyonu olmakla beraber yaygın olarak Sigmoid, ReLU, Softmax, Tanh aktivasyon fonksiyonları kullanılmaktadır.

Doğrusal aktivasyon fonksiyonu, giriş ile doğrusal orantılı olan ve değeri sıfır olmayan bir aktivasyon fonksiyonudur. Gradyan değeri sıfır olmayıp giriş değerinden bağımsız bir değerdir. Bundan dolayı güncelleme faktörü aynı olmasına rağmen ağırlıkların geriye yayılma adımı sırasında güncelleneceği anlaşılmaktadır. Sinir ağı her iterasyonda aynı gradyan değeri nedeniyle hatayı iyileştiremeyeceğinden dolayı bu fonksiyonun kullanımının pek bir faydası yoktur. Bu fonksiyon, basit görevler için idealdir. Bu fonksiyonun matematiksel denklemi (2.7)'de gösterilmiştir.

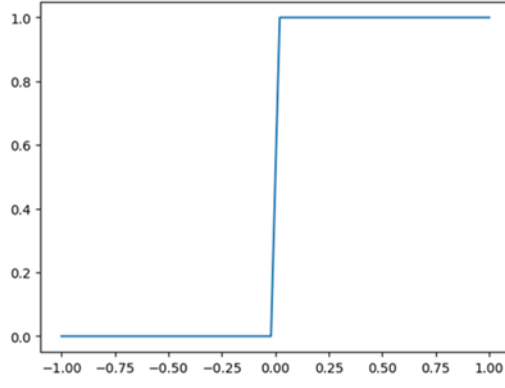
$$f(x) = ax \quad (2.7)$$



Şekil 7: Doğrusal Aktivasyon Fonksiyonu

Aktivasyon fonksiyonlarından adım fonksiyonu en basit fonksiyondur. İkili sınıflandırmada genellikle bu fonksiyon kullanılır ancak çok sınıflı sınıflandırmada kullanılmaz. Gradyanı sıfır olduğundan dolayı geri yayılım adımında bir engelleme nedeni olur. Bu fonksiyonun türevi sıfırdır. Bu fonksiyonun matematiksel denklemi (2.8)'de gösterilmiştir.

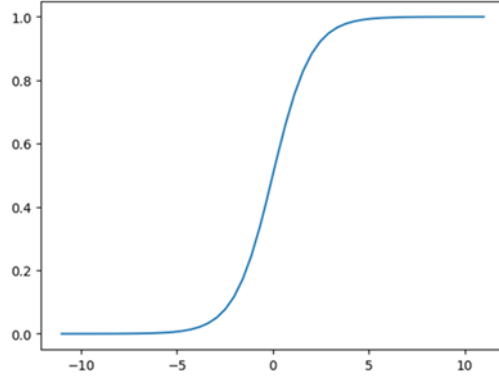
$$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ 1 & \text{ için } x \geq 0 \end{cases} \quad (2.8)$$



Şekil 8: Adım Aktivasyon Fonksiyonu

Sigmoid fonksiyonu, makine öğrenimi ve derin öğrenme alanında yaygın olarak kullanılan aktivasyon fonksiyonlarından bir tanesidir ve doğrusal değildir. Bu fonksiyon, bir önceki katmandan gelen çıktıyı alan ve girdi değerlerini 0 ile 1 arasına getiren gizli katmanlarda kullanılabilir. Bu fonksiyon sürekli türevlenebilir bir fonksiyondur. Bu fonksiyonun matematiksel denklemi (2.9)'da gösterilmiştir.

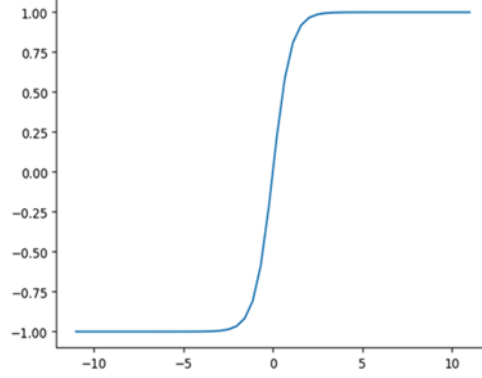
$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.9)$$



Şekil 9: Sigmoid Aktivasyon Fonksiyonu

Tanh aktivasyon fonksiyonu sigmoid fonksiyonuna benzer fakat orijin etrafında simetriktir. Bu fonksiyon hiperbolik tanjant fonksiyonudur. Değeri -1 ile 1 arasındadır ve sürekli türevlenebilir bir fonksiyondur. Belli bir yönde değişmek üzere kısıtlanmamış gradyanlara sahip ve sıfır merkezli olduğundan sigmoide göre tercih edilmektedir. Bu fonksiyonun matematiksel denklemi (2.10)'da gösterilmiştir.

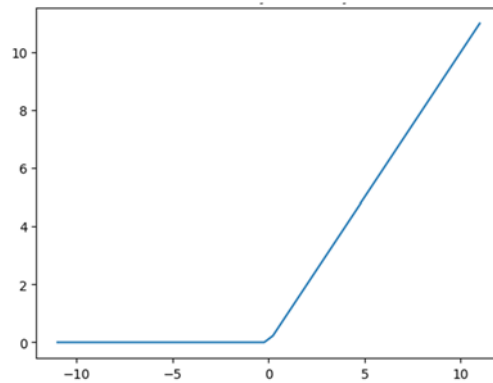
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.10)$$



Şekil 10: Tanh Aktivasyon Fonksiyonu

ReLU aktivasyon fonksiyonu herhangi bir negatif girdiyi sıfır yapar ve pozitif girdileri değiştirmeden korur. Bu fonksiyon, hesaplama verimliliği ve kaybolan gradyan probleminden kaçınma kabiliyeti nedeniyle derin öğrenme problemlerinde geniş çapta kullanılmaktadır ve popüler hâle gelmiştir. Doğrusal olmayan bir fonksiyondur. ReLU fonksiyonunda tüm nöronlar aktif edilmez bunun yerine bir seferde belirli sayıda nöron aktif edilir bundan dolayı diğer fonksiyonlara göre daha verimlidir. Bu fonksiyonun matematiksel denklemi (2.11)'de gösterilmiştir.

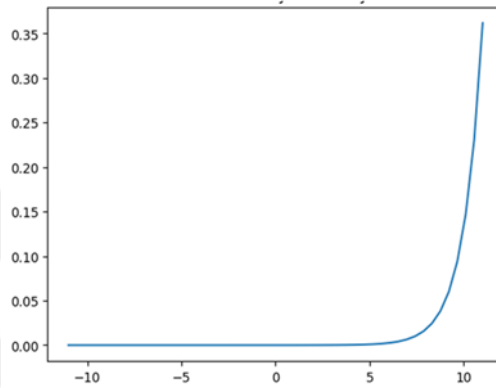
$$f(x) = \begin{cases} 0 & \text{için } x < 0 \\ x & \text{için } x \geq 0 \end{cases} \quad (2.11)$$



Şekil 11: ReLU Aktivasyon Fonksiyonu

Softmax fonksiyonu birden fazla sigmoid fonksiyonunun bir kombinasyonudur. Softmax aktivasyon fonksiyonu yaygın olarak çok sınıflı sınıflandırma problemleri çözümünde bir yapay sinir ağında son aktivasyon fonksiyonu olarak tercih edilir. Girdilerini birden fazla sınıfa ayırmak için bir olasılık dağılımına dönüştürür. Bu fonksiyonun matematiksel denklemi (2.12)'de gösterilmiştir.

$$f(x) = \frac{e^{x_i}}{\sum_j^n e^{x_j}} \quad (2.12)$$



Şekil 12: Softmax Aktivasyon Fonksiyonu

Tablo 2: Aktivasyon Fonksiyonlarının Matematiksel İfadeleri

Fonksiyon Adı	Fonksiyon Denklemi	Aralığı
Doğrusal Fonksiyon	$f(x) = x$	$(-\infty, \infty)$
Basamak Fonksiyonu	$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ 1 & \text{ için } x \geq 0 \end{cases}$	$\{0, 1\}$
Sigmoid Fonksiyonu	$f(x) = \frac{1}{1 + e^{-x}}$	$(0, 1)$
Tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$
ReLU	$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ x & \text{ için } x \geq 0 \end{cases}$	$[0, \infty)$
Softmax	$f(x) = \frac{e^{x_i}}{\sum_j^n e^{x_j}}$	

2.2.2.7 Kayıp Fonksiyonları

Bir yapay sinir ağının performansını ölçmek ve sınıflandırıcının ne kadar iyi tahmin ettiği hakkında bilgi almak için kayıp fonksiyonu kullanılır (Alkayal 2023: 15). Kayıp fonksiyonunun oluşturduğu çıktı, maliyet ya da ceza diye adlandırılan gerçek bir değerdir. Görüntü sınıflandırma problemlerinde kullanılan kayıp fonksiyonu türlerinden en yaygın olan çapraz-entropi (cross-entropy) fonksiyonudur (Akdağ 2017: 13). Çapraz -entropi denklemi (2.13)'te verilmiştir.

$$L = - \sum_i^n y_i \log p_i \quad (2.13)$$

Bu denklemde n çıkış katmanındaki beklenen sınıfların sayısıdır, y hedef vektör, p değeri ise her sınıf için ağ tarafından tahmin edilen olasılık değeridir.

2.2.2.8 Geri Yayılım (Backpropagation)

Geri yayılım yönteminde, her iterasyon ileri yayılım ve geri yayılım olmak üzere iki temel evreden geçer. İleri yayılım aşamasında YSA'nın o andaki durumunda YSA'ya uygulanan giriş işaretlerine karşı YSA'nın çıkışlarında oluşan değerler bulunur. Geri yayılım sürecinde, çıkışlarda oluşan hatalardan yola çıkılarak devredeki ağırlıklar baştan düzenlenmektedir ve bu hata (e) olarak tanımlanmaktadır. Burda x girdiyi ve (f) aktivasyon fonksiyonu temsil eder, verilen her katmandaki ağırlık değerlerine (w_{ki}) göre kayıp fonksiyonunun (L) bir gradyanı olarak hesaplanabilir.

$$y_k = f \left(\sum_{i=1}^n w_{ik} x_k \right) \quad (2.14)$$

$$e = \frac{\partial L}{\partial w_{ik}} = \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial w_{ik}} \quad (2.15)$$

Gradyan hesaplaması, w_{ik} nin L'ye göre kısmi türevinin hesaplanması ve zincir kuralının uygulanmasıyla yapılır. Ağırlıklar, hata değeri yardımıyla gradyan inişi benzeri bir optimizasyon algoritması tarafından yeniden hesaplanır.

2.2.2.9 Gradyan İnişi (Gradient Descent)

Yaygın kullanılan gradyan iniş algoritmasının amacı kayıp fonksiyonlarımızın eğitim değerlerini kullanarak fonksiyonumuzun minimum noktasına inmektir. Gradyan inişi, tanımlanmış bir fonksiyon üzerinde herhangi bir noktadan başlayarak o noktanın koordinatlarını tek bir değişken için kısmi türevlerin tersi yönünde değiştirerek küçük adımlarla minimum noktaya yaklaşmayı amaçlayan bir optimizasyon yöntemidir. Atılacak adımların büyüklüğünü belirleyen çarpana "adım boyutu" veya "öğrenme oranı" denir. Öğrenme oranı, ağırlıkların değişim değerini kontrol eden, genellikle (0,1) aralığında bir değerdir. Parametrelerde (ağırlıklarda) güncelleme yapmak için, eğitim setinin tamamı kullanılmalıdır. Ancak büyük eğitim kümelerinde bu yöntem, hesaplama bakımından maliyeti artırabilir.

Stokastik Gradyan İnişi (SGD) optimizasyon yöntemi diğer optimizasyon yöntemlerine göre daha verimlidir. SGD'de, model ağırlık parametrelerini güncellemek için eğitim veri setinin alt kümesi kullanılır. Her parametrenin güncellenmesinde herhangi bir küme seçilir. Buna ilaveten SGD, normal gradyan inişinde görülen yerel minimuma takılma durumunda değildir. Ancak SGD'nin bir dezavantajı vardır o da normal gradyan inişine göre daha yavaş bir yakınsama değerine sahip olmasıdır.

$$w^{(t+1)} = w^t - \alpha \nabla_w L(w^t) \quad (2.16)$$

Ağırlıklar, kayıp fonksiyonunun ağırlıklara göre gradyanının negatif yönde alınmasıyla güncellenir. Bu değişiklik, öğrenme oranı (α) ile sınırlıdır.

Uyarlanabilir Moment Tahmini (Adam), her parametre için uyarlamalı öğrenme oranlarını hesaplayan bir optimizasyon yöntemidir. Her iterasyonda öğrenme katsayısını günceller ve parametrelerin güncellenmesinde önceki gradyan değerlerini ve karelerini dikkate alarak adapte edilmiş moment tahminleri kullanır (Ünlü 2019). Adam optimizasyon algoritması, öğrenme oranlarını parametrelere göre otomatik olarak adapte eden ve bu şekilde optimize edilmekte olan kayıp fonksiyonunu iyileştirmeye çalışan bir optimizasyon algoritmasıdır. Diğer taraftan, bu fonksiyonun en hızlı azaldığı yolda optimum değere doğru salınımı düşürmenin bir yolu, ağırlıkların güncellemesine bir momentum terimi eklemektir. Her ağırlık için birbirine

benzemeyen bir öğrenme sabiti sağlamak amacıyla bu algoritmanın öğrenme değerinin ölçekleme parametresi aşağıda gösterildiği gibidir:

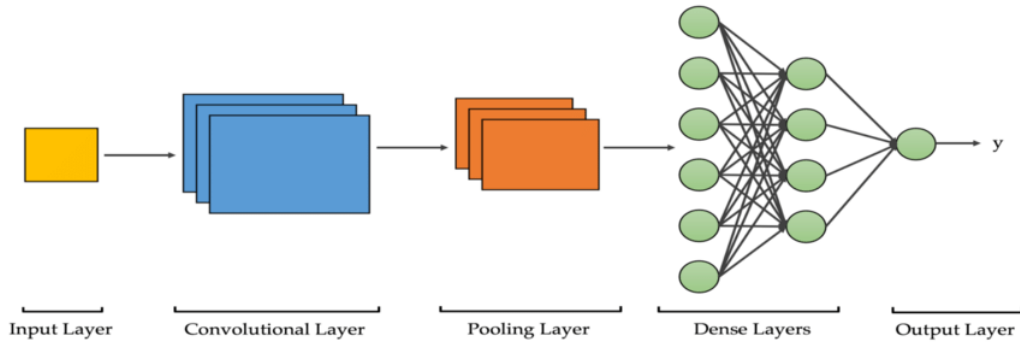
$$v_1 = \beta_2 v_t + (1 - \beta_2) [\nabla E(w_t)]^2 \quad (2.17)$$

$$m_1 = \beta_1 m_{t-1} + (1 - \beta_1) \nabla E(w_1) \quad (2.18)$$

$$w_{t+1} = w_t - \frac{\alpha m_1}{\sqrt{v_1} + e} \quad (2.19)$$

2.2.3 Evrişimsel Sinir Ağları

Evrişimsel Sinir Ağı (ESA) görüntü analizi yapmak için özellikle tasarlanmış derin bir sinir ağıdır (Çalışkan ve Demir 2022: 31). ESA bilgisayarla görme alanında; görüntü tanıma, yüz tanıma, nesne algılama, şiddet tespiti, sınıflandırma gibi alanlarda titizlikle ve yaygın olarak kullanılır. Farklı türdeki görüntülerden çeşitli özellikleri çıkararak sınıflandırmak için tasarlanmıştır. Başta iki boyutlu görüntü verileriyle çalışmak için geliştirilmiş olsa da tek boyutlu ve üç boyutlu verilerle de kullanılabilir.



Şekil 13: Görüntü Sınıflandırma İçin Temel Bir Evrişimli Sinir Ağı Yapısı (Sit vd. 2020)

Bir Evrişimli Sinir Ağı (ESA), giriş katmanı, çıkış katmanı ve birden çok gizli katmandan oluşur. Bu gizli katmanlar yaygın olarak filtre uygulanan evrişimli katmanlar, havuzlama katmanları, tek boyuta dönüştüren tamamen bağlı katmanlar ve sınıflandırma için normalleştirme katmanları (ReLU) içerir (Tokmak ve Kırac 2021: 74). Bu temel evrişimli sinir ağı; giriş katmanı, bir konvolüsyon katmanı, havuzlama katmanı, bir tam bağlantılı katman ve çıkış katmanı olmak üzere toplamda beş farklı

katmandan oluşmaktadır. Bu katmanlar nitelik çıkartma ve sınıflandırma şeklinde iki grup içinde yer alır. Giriş, konvolüsyon ve havuzlama katmanı özellik çıkarma bölümünü oluştururken tam bağlantılı katman ve çıkış katmanı ise sınıflandırma bölümünü oluşturur.

Konvolüsyon katmanı, giriş görüntülerinden farklı nitelikleri çıkarmak için kullanılan giriş katmanından sonra ilk katmandır. Bu katmanda, giriş görüntü verisi ile belirli bir $M \times M$ boyutundaki bir filtre arasında evrişim (konvolüsyon) matematiksel işlemi uygulanır. Bir çekirdek şablonu (kernel veya matris), resim üzerindeki piksellerle "kaydırma ve çarpma" işlemi olarak tanımlanabilir. Bu işlem, özellikle Evrişimli Sinir Ağları (CNN) gibi derin öğrenme modellerinin temelini oluşturur.

Çekirdek şablonu, resim üzerinde belirli bir boyutta (örneğin, $m \times m$ veya $n \times n$ gibi) kaydırılır ve çekirdeğin değerleri ile aynı boyutta bir bölge üzerindeki pikseller arasında bulunan eleman bazlı çarpma işlemi yapılır. Sonuç olarak, bu işlem resimde belirli özellikleri (örneğin, kenarlar, köşeler veya belirli desenler) vurgulayan yeni bir matris veya görüntü oluşturur. Bu işlem, özellik çıkarma veya evrişim adı verilen bir işlem olarak bilinir.

Bu işlem, özellikle görüntü işleme, nesne tanıma ve daha birçok görsel veri analizi uygulamasında kullanılır.

Konvolüsyon katmanlarını takiben genel olarak "Aktivasyon Katmanı (Activation Layer)" olarak adlandırılan doğrusal olmayan (Non-Linearity) katman gelir. Bu katmanda doğrusallığı bozan aktivasyon fonksiyonlarından biri kullanılır. Geçmişte, sigmoid ve Tanh gibi doğrusal olmayan fonksiyonlar sıkça kullanılmaktadır ancak Rectifier (ReLU) fonksiyonu, sinir ağlarının daha hızlı eğitilmesini sağladığı ve bazı durumlarda daha iyi sonuçlar verdiği için artık çok yaygın bir şekilde kullanılmaktadır. ReLU, özellikle derin sinir ağlarında aşırı öğrenmeyi önlemeye yardımcı olabilir ve gradient vanishing sorununu azaltabilir.

Pooling katmanı konvolüsyon katmanı içinde sıklıkla kullanılan bir katmandır ve onun temel görevi, özellik haritasındaki boyutları küçültmek ve hesaplama sayısını azaltmak için veri küçültme veya özetlemenin bir türünü uygulamaktır. Bu sayede ağır hesaplama karmaşıklığı azalırken özelliklerin öğrenilmesi sırasında ağdaki uyumsuzluk (overfitting) sorunu kontrol altına alınabilir.

En yaygın kullanılan Pooling işlemi "max pooling"dir. Bu işlem bir özellik haritasındaki her bölge için en büyük değeri seçer ve bunu yeni bir özellik haritasına ekler. Bu, özelliklerin belirli yönlerini veya desenlerini vurgulamak için kullanışlıdır.

Ayrıca, "average pooling" ve "L2-norm pooling" gibi farklı pooling algoritmaları da mevcuttur. Average pooling, her bölgenin ortalamasını alırken L2-norm pooling, bölge içindeki değerlerin L2-normunu alır. Bu farklı pooling yöntemleri, belirli görsel problemlere uyum sağlamak için kullanılabilir.

Pooling katmanı, özellik haritalarını küçültmek ve önemli bilgileri korumak için kullanıldığı için CNN'lerin performansını artırmak ve hesaplama maliyetini azaltmak için önemli bir bileşendir.

Düzleştirici katmanın (Flattening Layer) görevi basitçe, son ve en önemli katman olan tam bağlantılı katmanın (Fully Connected Layer) girişindeki verileri hazır hâle getirmektir. Konvolüsyon ve pooling katmanından gelen matrisleri tek boyutlu diziye çevirerek Fully Connected Layer'in girişine verir.

Hesaplanan değer en son tamamen bağlı (fully connected) katmanda işlenir ve bu katman, görüntünün hangi sınıfa ait olduğu bilgisini içeren tahmin değerini üretir (Aktaş 2020: 6). Tamamen bağlı katmanlar, önceki gizli katmanlardan gelen özellikleri bir araya getirir ve bu özellikleri temsil eden bir vektör oluşturur.

Öğrenme süreci boyunca ağ, eğitim verilerindeki girdi görüntüleriyle ilişkilendirilmiş doğru sınıf etiketlerini öğrenir. Tamamen bağlı katman, bu öğrenilen özellik vektörlerini kullanarak girdiyi sınıflandırır. Sınıflandırma işlemi genellikle bir aktivasyon fonksiyonu, genellikle softmax, kullanılarak gerçekleştirilir.

Sonuç olarak, tamamen bağlı katman, önceki katmanlardan gelen özelliklerle beslenen ve nihai sınıf tahminini üreten kritik bir bileşen hâline gelir. Bu sayede, sinir ağı eğitildikten sonra görmediği veriler üzerinde sınıflandırma yapabilir ve görüntünün hangi sınıfa ait olduğunu tahmin edebilir.

Günümüzde en çok bilinen bazı CNN mimarileri şunlardır:

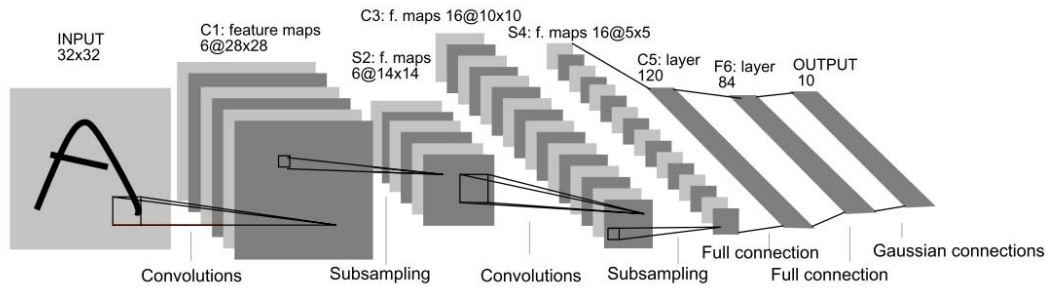
- R-CNN
- GoogLeNet
- AlexNet
- ZFNet
- Microsoft ResNet

2.2.3.1 Evrimsel Sinir Ağları Modelleri

Derin öğrenme, 2012 yılında yapılan ve dünya çapında bir yarışma olan ImageNet Yarışması'yla birlikte ilgiyi üzerine çekti. Derin öğrenme, yarışmacıların

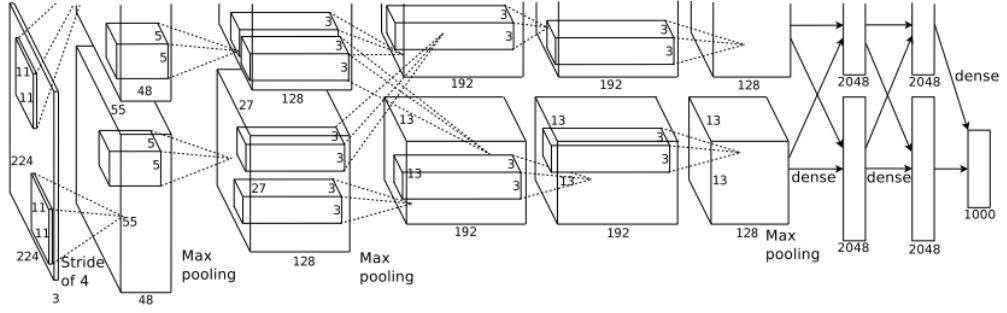
çoğu için bu mimari kullanmalardan dolayı daha da önemli hâle geldi. Derin öğrenme mimarileri, her geçen yıl değişen katman sayıları ve başarımlarıyla gittikçe daha da artmaktadır. Bu mimariler derin öğrenmenin temel yapıları olarak kabul edilmektedir (Tan 2019: 15). AlexNet, VGGNet, ResNet, GoogLeNet, MobileNet ve DenseNet en çok bilinen evrimsel sinir ağı mimarilerinden bazılarıdır (Minaee VD. 2022: 2). Evrimsel sinir ağlarının ilk başarılı modeli Lenet-5'tir.

Lenet-5, rakamları sınıflandırmak için 1998 yılında Yann LeCun tarafından belirlenen prensiplerle geliştirilmiştir (Lecun vd. 1998). İki konvolüsyon katmanı, iki tam bağlı katmandan meydana gelmiştir. Bu modeli diğerlerinden ayıran önemli özelliklerden biri, boyut azalma aşamalarında ortalama havuzlama işlemine ve aktivasyon fonksiyonu olarak sigmoid ve hiperbolik tanjant kullanımına dayanmasıdır. Şekil 14'te Lenet-5 mimarisi gösterilmektedir.



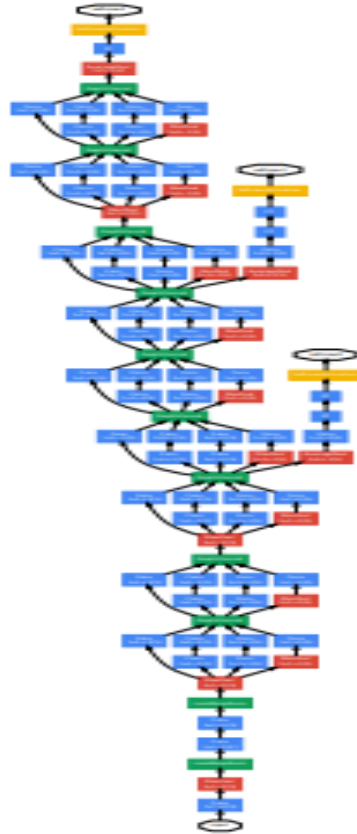
Şekil 14: Lenet-5 Mimarisi (Lecun vd. 1998)

AlexNet, Alex Krizhevsky tarafından geliştirilen ve 30 Eylül 2012 tarihinde yapılan ImageNet Yarışması'nı kazanarak adını duyuran bir konvolüsyonel sinir ağıdır (Xiao 2022: 15). LeNet mimarisine, ardışık evrişim ve ortaklama katmanları bulunduğu için benzetilmektedir. AlexNet'te yaklaşık 60 milyon parametre kullanılmış ve ILSVRC Yarışması'nda sınıflandırma doğruluk oranı %74,3'ten %83,6'ya çıkarılmıştır (Krizhevsky vd. 2012). AlexNet ReLu aktivasyon fonksiyonunu, ortalama katmanlarında ise maksimum ortalama kullanmaktadır



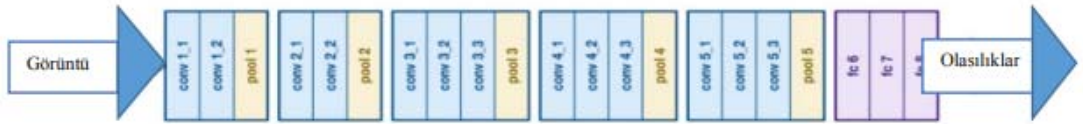
Şekil 15: AlexNet Mimarisi

2014 ImageNet Yarışması'nı kazanan GoogleNet modeli, Inception modüllerinin birleşiminden oluşur ve oldukça karmaşık bir yapıya sahiptir. %5.7 hata oranına sahip olması, o dönemde oldukça etkileyici olmuş ve veri kümelerinde başarılı sonuçlar elde edilmesine olanak tanımıştır. Inception modülleri, farklı ölçeklerdeki konvolüsyonları ve pooling katmanlarını kullanarak bilgiyi daha etkili bir şekilde çıkarabilen özgün bir yaklaşım sunar. GoogleNet, bu modüllerin birleşimiyle büyük ve karmaşık bir sinir ağı oluşturarak birçok görevde başarılı sonuçlar vermiştir.



Şekil 16: GoogleNet Mimari (Szegedy vd. 2015)

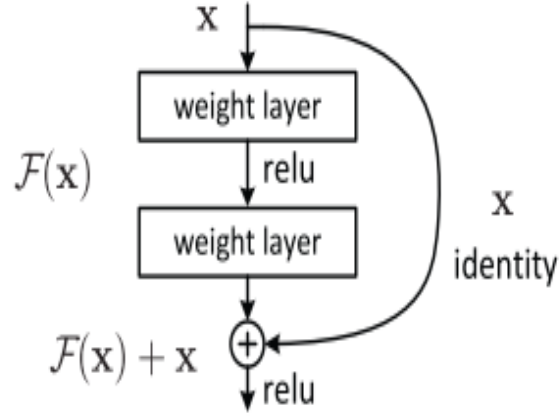
VGG VGGnet olarak da bilinen klasik bir evrişimsel sinir ağı (CNN) mimarisidir. VGGNet, AlexNet ağına benzer temel prensiplerle tasarlanmış bir yapay sinir ağıdır. Model performansını artırmak amacı ile bu tür CNN'lerin derinliği artırılarak geliştirilmiştir. 16 katmandan oluşan VGGNet mimarisi, Simonyan ve Zisserman tarafından geliştirilmiştir (Yılmaz ve Kaya 2019: 81). VGG16 evrişimsel sinir ağı mimarisi, Oxford Üniversitesi Görsel Geometri Grubu tarafından ILSVRC-2014 Yarışması'nda daha iyi sonuçlar elde etmek amacıyla geliştirilmiştir. VGG16 evrişimsel sinir ağı mimarisi 13 evrişim (konvolüsyon) ve 3 tam bağlı (Fully Connected) katmandan meydana gelen bir ağıdır. VGG16 ağı, Maxpool, Fully Connected Layer, ReLU Layer, Dropout Layer ve Softmax Layer katmanları dâhil olmak üzere toplamda 41 katmandan oluşur (Ferdî ve Türkoğlu 2018: 422). VGG16 modelinin girdi katmanı 224x224 piksel boyutunda ve 3 kanallı (RGB) bir görüntüyü kabul eder. Son katman ise sınıflandırma katmanıdır, bu katman sınıflandırma işlemini gerçekleştirir ve sonuçları sınıf etiketlerine dönüştürür. VGG 19 modeli ise 19 katmandan oluşmaktadır.



Şekil 17: VGG16 Modeli

ResNet, ImageNet Yarışması'nda 2015 yılı kazananı olmuştur. 34 katmanlı ilk ağ olan yapı, residual bloklardan oluşmaktadır. Oluşturulan tüm mimarilerden daha derin olarak tasarlanan bir modeldir. ResNet (Residual Network), özellikle derin sinir ağlarındaki performansı artırmak ve eğitimi kolaylaştırmak için residual block olarak adlandırılan özel bir mimari özellik eklenerek diğer klasik sinir ağı modellerinden farklı şekilde geliştirilmiştir (Kızrak ve Bolat 2018: 267).

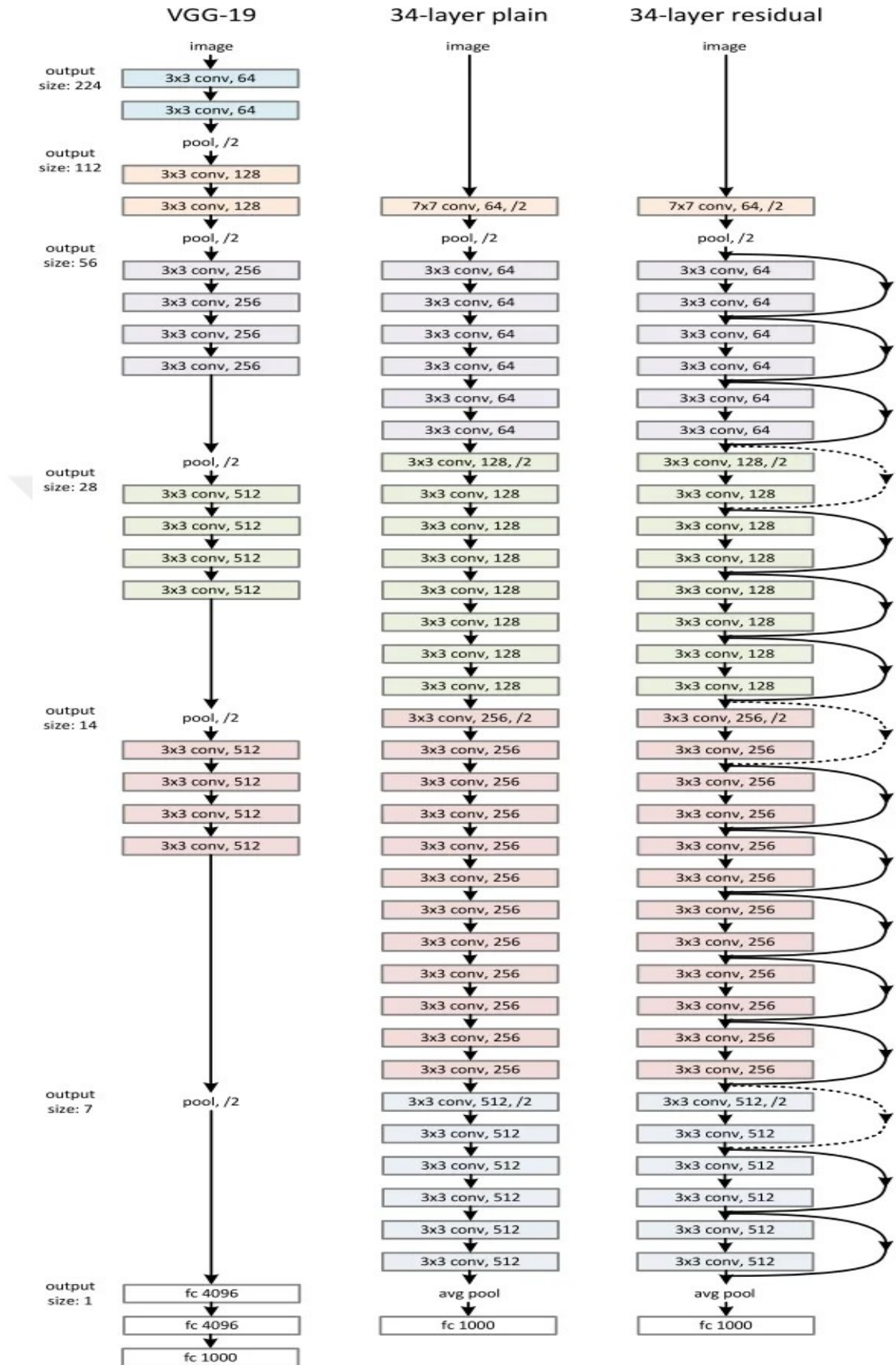
Teorikte derinliği artırılan bir ağda eğitim hatasının azalması beklenirken gerçekte doğruluk doğar ve ardından hızlı bir şekilde düşerek eğitim hatası artar. Bu da aşırı uyumdan (overfitting) değil daha derin modellerin optimize edilmesinin kolay olmadığını gösteren bir bozulma (degradasyon)/optimizasyon probleminden kaynaklıdır.



Şekil 18: Artık Blok Örneği (He vd. 2016)

X katman girdisini toplama işlemine taşıyan düz çizgiye "kısayol bağlantısı" veya "bağlantı" denir. Bu tür bağlantılar, bir veya daha fazla katmanı atlayarak girdiyi doğrudan ilgili çıkış katmanına taşır. Kısayol bağlantıları, özellikle derin sinir ağlarında kullanıldığında ağın eğitim sürecini daha verimli hâle getirebilir ve gradiyentlerin daha iyi iletilmesine katkıda bulunabilir.

ResNet ağında, VGG-19 ağ mimari yapısından yararlanılarak VGG ağlarından daha az filtreye ve daha az karmaşıklığa sahip 34 katmandan oluşan düz bir ağ mimarisi kullanır. Daha sonra ResNet ağ mimarisi, bu düz ağa, atlama bağlantıları ya da artık bloklar eklenerek artık ağ meydana getirilir.



Şekil 19: ResNet Ağ Mimarisi (He vd. 2016)

MobileNet, derinlemesine ayrılabilir filtreler üzerine inşa edilmiştir (Howard vd. 2017). MobileNet, derin evrişimsel sinir ağı modellerinden biridir ve özellikle mobil cihazlarda hafif ve hızlı bir performans sağlamak üzere tasarlanmıştır. Bu model, diğer birçok popüler modelden daha küçük boyutta olup düşük hesaplama gücüne sahip cihazlarda bile etkili şekilde kullanılabilir. Boyut ve gecikmeyi azaltmak için kabul edilebilir miktarda doğruluktan ödün vererek genişlik çarpanı ve çözünürlük çarpanı kullanarak geliştirilen daha küçük ve daha hızlı bir modeldir (Howard vd. 2017: 2). Çıktıyı sınıflandırma yapmak için softmax katmanını besleyen tam bağlantılı katman haricinde, tüm katmanlardan sonra bir batchnorm ve ReLU doğrusal olmayan aktivasyon katmanı gelir. MobileNet, derinlikli ve noktasal evrişimler dışında 28 katmana sahiptir (Howard vd. 2017: 3).

MobileNet, mimaride işlem hacmini ve model karmaşıklığını düşürmüş olduğundan bir donanımın yetersiz olduğu yerlerde de kullanılabilir.

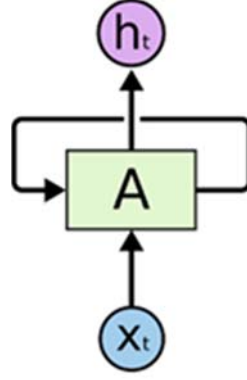
2.2.4 Transfer Öğrenme

Transfer öğrenme, bir yapay öğrenme sisteminin bir görevi öğrendikten sonra bu öğrenilen bilgiyi farklı ancak genellikle benzer bir görevde kullanma yeteneğine dayanan bir öğrenme yaklaşımıdır. Makine öğrenmesi yöntemlerinin ana çalışma prensibi, eğitim verisi ile eğitilen bir sistemin aynı dağılıma sahip ve aynı sayıda giriş ve çıkış parametresi içeren hiç bilinmeyen veriler üzerine sınıflandırma tahminleri yapması kurallarına dayanmaktadır. Eğitim verisini bulmak ya da oluşturmak çok zaman gerektiren, masraflı ve zahmetli bir iştir. Bunun dışında yeteri kadar veri bulunamayabilir. Modeli eğitmek çok uzun zaman alabilir. Dolayısıyla, bu durumlarda transfer öğrenme yöntemini kullanmak yarar sağlayacaktır.

CNN'ler milyonlarca ağırlık barındırabilen oldukça karmaşık yapılara sahip, binlerce sınıf etiketi ve görüntü verisi içeren çok büyük veri setleri ile eğitilmiş birçok model içermektedir. AlexNet, LeNet-5, VGGnet, GoogLeNet, ResNet ve MobileNet transfer öğrenme yapılarından bazılarıdır.

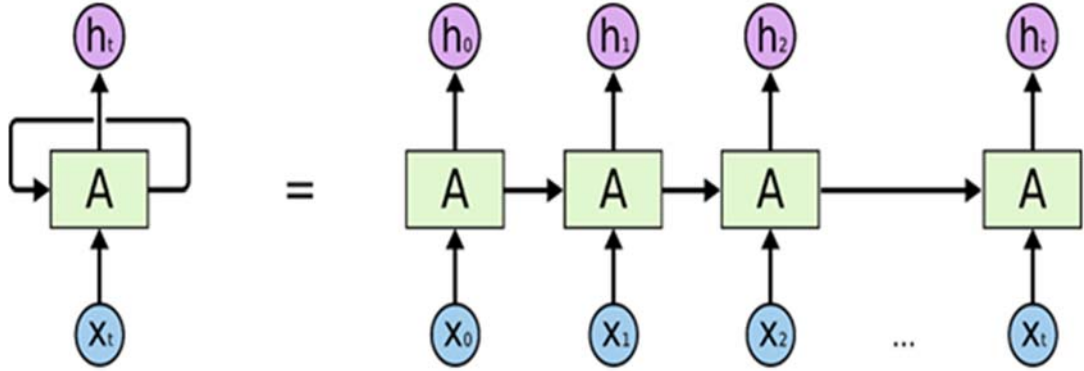
2.2.5 Tekrarlayan Sinir Ağları (RNN)

Tekrarlayan Sinir Ağları (RNN), döngü oluşturarak bilgilerin ağ içinde depolanmasına izin veren bir tür yapay sinir ağıdır (Yağın 2022).



Şekil 20: Tekrarlayan Sinir Ağı İçerisindeki Döngü (Olah 2015)

İnsanların düşüncelerinin kalıcılığı vardır ve bir konu hakkında düşünürken hep sıfırdan başlamazlar. Tekrarlayan sinir ağları geleneksel sinir ağlarının yapamadığı bilginin devama etmesi sorununu ele alır. Örneğin, geleneksel sinir ağlarında bir videodaki tüm karelere bakarak hareketleri sınıflandırmak istendiğinde, filim kareleri arasında insanlar gibi anlam çıkartamayacağından, sınıflandırma yapamayacaktır. Tekrarlayan Yapay Sinir Ağları (Recurrent Neural Network) ise bir döngü oluşturarak, geçmiş bilgilerin kullanılmasını sağlayacak ve böylece video kareleri arasında anlam çıkartabilecek ve sınıflandırma yapabilecektir.

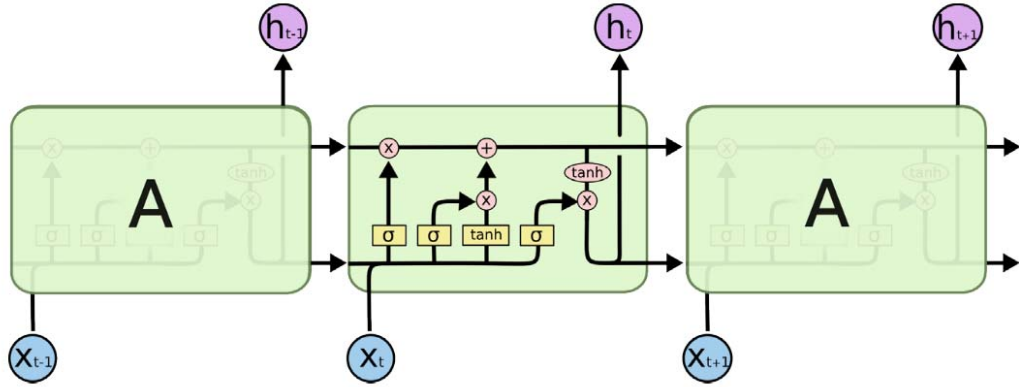


Şekil 21: RNN Algoritmasının Çalışma Şekli (Olah 2015)

Tekrarlayan sinir ağları (RNN), ağa giren giriş örnekleri ile beraber daha önce zaman serisi içerisindeki ağa giriş örneklerini de alır ve bundan dolayı yaygın olarak dil çevrimleri için kullanılmakla beraber sıralı şekilde meydana gelen bir dizi yapıdan bir sonraki noktayı tahmin etmek amacıyla da kullanılır. (Doğan ve Türkoğlu 2019: 414).

2.2.5.1 Uzun Kısa Vadeli Bellek (LSTM)

Bazı problem tiplerinde RNN kısa vadeli bir hafızası olmasından dolayı iyi çalışamaz durumdaydı. Yeterince uzun bir cümle, RNN'nin girişine verildiğinde geçmiş bilgileri hatırlamakta başarısız olunuyordu ve bu nedenle tahminde bulunmakta zorluk çekiliyordu. Bunun dışında geriye yayılım (backpropagation) esnasında gradyanın yok olması problemi de yaşanmaktaydı. Gradyan değerleri iyice küçüldüğünde RNN öğrenme problemi yaşanmaya başlanıyordu. Bu tip problemlere, 1997 yılında Hochreiter tarafından uzun-kısa vadeli bellek ağları (Long Short Term Memory-LSTM) çözüm önerisi olarak getirilmiştir. Giriş katmanını, unutma katmanını ve çıkış katmanını olmak üzere üç katmandan meydana gelmektedir. Klasik özyinelemeli sinir ağı modelinde unutma bölümü bulunmamaktadır (Hochreiter ve Schmidhuber 1997).



Şekil 22: LSTM Yapısı (Olah 2015)

Çift Yönlü Uzun Kısa Süreli Bellek (BILSTM), uzun kısa süreli belleğin (LSTM) bir çeşididir. Zıt yönlerde iki LSTM ağından oluşturulur ve çift yönlü aktarım yoluyla önceki ve sonraki derinlik dizisi verilerinin etkileri birleştirilerek bir derinlik dizisinin içerdiği bilgiler öğrenilebilir.

2.3 BİLGİSAYARLI GÖRÜ

Bilgisayarlı görü, insanlar gibi bilgisayarların da görebilmesini, anlayabilmesini ve yorumlayabilmesini konu alır. Yapay zekâ ve görüntü işleme alanlarında kullanılan tekniklerin birlikte kullanılmasıyla oluşturulan bir teknolojidir (Aksoylu 2021).

2.3.1 Görüntü İşleme

Görüntü işleme, dijital ortamlar üzerinde kaydedilmiş görüntüleri bazı bilgisayar algoritmaları ve görsel teknikler kullanılarak amaca uygun hâle getirme yöntemidir. İşlenmiş görüntü elde etmek için matematiksel işlemler uygulanarak istenen bilgileri elde etmek amacıyla kullanılan matematiksel yöntemlerdir (Aksoylu 2021: 448).

Görüntü işleme, daha çok önceden kamera gibi cihazlarla kaydedilmiş olan görüntüleri işlemek yani mevcut resim ve grafiklerin değiştirilerek düzenlenmesi veya iyileştirilmesi için kullanılmaktadır (Peker 2009: 8).

Her görüntü, kullanım amacına yönelik kullanılabilirlik açısından elverişli olmayabilir. Bu görüntüler, artık görüntü işleme teknolojisi kullanılarak amaca uygun hâle getirilebilmektedir. Kaydedilen görüntü, görüntü işleme algoritmaları kullanılarak bir fonksiyona dönüştürülür. Görüntü iki boyut olarak bir $f(x, y)$ fonksiyonu olarak tanımlanabilir (Gonzalez 2008: 2). Her renk, her nokta fonksiyonunun parçaları olarak algılanır ve algoritmalar bu fonksiyon üzerinde işlem yapabilir.

Görüntü işleme; güvenlik, tıp, savunma, trafik gibi alanlarda şekil tanıma, nesne tanıma, gürültü giderme, çözünürlüğü artırma amaçlarıyla kullanılmaktadır.

Görüntü işleme beş temel aşamada ele alınabilir. Kamera ile görüntü sayısal olarak elde edilir, görüntü elde edildikten sonra görüntü iyileştirme, görüntü onarma, görüntü sıkıştırma gibi işlemlerle görüntüde ön işleme yapılır. Ön işlemde geçen görüntüler bölütleme, parçalara ayırma, segmentasyon olarak adlandırılan bölütleme işlemine tabi tutularak görüntü içerisinde bulunan nesne ve alanların farklı özelliklerinin tespit edilip birbirinden ayrılması sağlanır. İstenilen farklı özellikler, ayrıntılar ön plana çıkarılarak sonuç elde edildikten sonra, yapay zekâ algoritmaları kullanılarak işlenen bu görüntü içerisindeki istenen nesnelerin ve alanların sınıflara ayrılması ve etiketlenmesi yapılır.

BÖLÜM III

VERİ SETİ VE ÖNERİLEN MODEL

3.1 VERİ SETİ

Giderek artan video verilerini analiz etme ve bu verileri yorumlama, bilgisayarla görme ve uygulamalar alanında giderek büyüyen bir konu hâline gelmiştir. Video verileri, çevredeki değişiklikler hakkında bilgi sağladığından gözetim, video indeksleme ve navigasyon gibi birçok görsel etkinlik için çok önemlidir. Büyük veri setlerinin ve hesaplama kaynaklarının oluşması sayesinde birçok derin öğrenme tekniklerinin geliştirilmesi, bilgisayarla görme alanında tarihî bir değişime yol açmıştır.

Videolarda şiddet tanıma alanında en çok kullanılan, kamuya açık veri kümelerinin geneli el sallama, yürüme gibi basit bir eylemi gerçekleştiren tek bir aktöre odaklanmaktadır. Arka planda düzensiz ise bu tür veri setlerinden şiddet tespit edilmesi oldukça zordur. Bu nedenle belirtilen veri setlerini kullanmak uygun değildir.

Derin öğrenme yöntemiyle videolarda şiddet tespiti edebilmek için özel veri setleri oluşturulmuştur. Bu veri setleri ile çalışmak daha uygun olacaktır.

Bu çalışmada modeli eğitmek ve test etmek için yoğun araştırma ve analizler yapıldıktan sonra 3 adet veri kümesi seçilmesine karar verilmiştir. Her bir veri seti çeşitli kaynaklardan ve farklı ortamlarda oluşturulmuş olup kendine özgü avantajları ve dezavantajları vardır.

3.1.1 Hockey Fights Dataset

Hockey fight veri seti (Bermejo Nieves vd. 2011: 3), Ulusal Hokey Ligi'nden 500'ü şiddet içeren ve 500'ü şiddet içermeyen 1000 adet aksiyon içeren klipten oluşmaktadır. Her klip 50 karelik çerçeve içermektedir. Çözünürlük 720x576 pikseldir. Bu veri seti dövüş kalıplarını öğrenmek için faydalı bilgiler içerir. Videolar nispeten ünifordur.

Hockey fight veri seti (Shilaskar vd. 2023: 1); eşit sayıda görüntüden oluşan veri kümesi, profesyonel hokey sırasında şiddet içeren ve şiddet içermeyen eylemleri kapsayan maçlardan oluşur. Bu maçlarda genellikle, iki oyuncu yakın vücut etkileşiminde bulunur.

3.1.2 Movies Dataset

Movies Dataset (Shilaskar vd. 2023: 1) veri kümesi, filmlerden toplanan dövüş sahnelerinden oluşmaktadır. Veri kümesi eşit sayıda şiddet içeren ve şiddet içermeyen film kliplerini içermektedir. Hokey fight veri kümesinin aksine, bu veri kümesi değişiklik göstermektedir ve örnekler arasında derin farklılıklar bulunmaktadır.

Bu veri seti çeşitli aksiyon filmlerinden seçilip alınan 200 klipten oluşmaktadır. Bu kliplerden 100 tanesi bir dövüş sahnesi içerirken diğer 100 tanesi dövüş sahnesi içermeyen görüntülerden oluşmaktadır. Çözünürlük 360 x 250 pikseldir. Klipler farklı filmlerden alınırken sahneler seçilmiştir. Videolardan bazıları sahne, bağlam ve arka planı aynı olanlardan seçilmiştir. Diğer veri setlerine göre çözünürlüğün daha düşük olduğu ve eğitim için video sayısının yeterli olmadığı görülmüştür.

3.1.3 Real-Life Violence Situations

Real-Life Violence Situations (Soliman vd. 2019: 82) veri seti, araştırma veya mühendislik amacıyla kullanılması durumunda izin verilen bir veri setidir. YouTube kanalından toplanan bu veri seti 2000 adet video görüntüsünden oluşmaktadır. Çözünürlüğü 480X720 p olan klipler 1000 tane şiddet içeren ve 1000 tane de şiddet içermeyen video görüntüleri olarak iki gruba ayrılmıştır. Bu veri seti, kaggle'da bir video veri kümesi olarak bulunmaktadır. Şiddet videolarında ve koşullarında çeşitli ortamlardaki birçok gerçek sokak çatışması durumu tasvir edilirken şiddetsizlik videoları spor, yemek yeme, gezinme gibi çeşitli insan faaliyetlerinden derlenmiştir. Veri setindeki şiddet içeren videolar farklı yerlerde ve koşullarda birçok gerçek sokak kavgası görüntülerini içermektedir. Tüm videolar RGB formatında olup renklidir.

Toplamda 3200 adet klipten oluşan veri setleri hazırlanırken her video 16 kare içerecek şekilde düzenlenmiş olup 16 x 64 x 64 x 3 şekline getirilmiştir.

3.2 ÖNERİLEN MODEL

Günümüzde videolardaki şiddet tespitinde hem doğruluk oranının yüksek olması hem de hızlı olunması beklenmektedir. Bu çalışmada en uygun, en iyi çözümü üretebilmek için önce konsept belirlenmiş daha sonra pratik uygulaması yapılmıştır.

3.2.1 Geliştirme Ortamı ve Kullanılan Kütüphaneler

3.2.1.1 Google Colab

Kodların geliştirilmesi ve çalıştırılması Google tarafından sunulan ücretsiz bir bulut tabanlı Python kodlama ortamı olan Google Colab (Google Colaboratory) ortamı kullanılmıştır. Bu sayede GPU ve yüksek ram kullanma imkânı bulunmuştur.

Google Colab, yapay zekâ ve derin öğrenme projeleri için oluşturulmuş etkileşimli, tamamen bulut tabanlı, kullanımı kolay, önceden hazırlanmış ve ortak çalışmaya dayalı bir programlama ortamıdır. Ücretsiz olan ve kurulum ihtiyacı olmayan Google Colab, 2017 senesinde Google firması tarafından piyasaya sürülmüştür. Jupyter Notebook'un Google'ın bulut sunucularında barındırılan ve birden fazla kullanışlı özelliğe sahip, güçlendirilmiş bir sürümü olarak düşünülebilir.

3.2.1.2 Python

Python basit söz dizimine sahip, nesne yönelimli (verilere dayalı), yorumlamalı, modüler, etkileşimli, öğrenmesi kolay yüksek seviyeli bir programlama dilidir. Bu programlama dili 90'lı yılların başında Hollandalı bir programcı olan Guido Van Rossum tarafından geliştirilmeye başlanmıştır. Bu programlama dilinin ismi The Monty Python adlı altı kişilik bir İngiliz komedi grubunun, Monty Python's Flying Circus adlı gösterisinden etkilenilerek adlandırılmıştır. Python'un dili diğer programlama dilleri ile karşılaştırıldığında bu dilin daha az kod ile işlemleri yapmasının mümkün olduğu görülmüştür. Çalışmada tüm analizler ve kodlamalar Python dili ile yapılmıştır.

Veri biliminde makine öğrenmesinde, güçlü kütüphanelere sahip olan python kullanımı, kolay ve açık kaynak kod mantığına dayanan ve geliştirme araçları ile diğer birçok kütüphanenin, lisansa gerek duyulmaksızın açık kaynak kod olarak ücretsiz şekilde indirilebilir olmasından dolayı tercih edilmiştir. Bu projede python 3 sürümü kullanılmıştır.

3.2.1.3 OpenCV

Görüntü işlemek için açık kaynak bilgisayar görü kütüphanesi olan OpenCV Kütüphanesi kullanılmıştır. OpenCV (Açık Kaynak Bilgisayarla Görü), bilgisayarla görme ve görüntü işleme uygulamaları geliştirmek için kullanılan intel tarafından üretilen açık kaynaklı bir kütüphanedir. İlk olarak 1999 yılında geliştirilmeye başlanmış ve zamanla geniş bir topluluk tarafından desteklenmiştir. Windows, Linux, MacOS X gibi farklı işletim sistemlerinde kullanılabilir olması, bu kütüphaneyi diğer görüntü işleme araçlarından avantajlı duruma getirmiştir.

OpenCV, resim veya video içindeki anlamlı bilgileri belirleyip işleyebilmek C ve C++ dilleri kullanılarak geliştirilmiş, birçok programlama dilini destekleyen, hazır algoritmalar, işlevler ve araçlar içeren açık kaynak kodlu bir kütüphanedir. Bu kütüphane, görüntü ve video işleme, nesne tanıma, yüz tanıma, hareket tespiti gibi görevler için kullanıla bilinir.

OpenCV'nin açık kaynak kodlarının genel topluluk tarafından erişilebilir ve geliştirilebilir olması; görüntü işleme, bilgisayarla görme ve makine öğrenimi alanlarında kullanılan birçok algoritmaya sahip olması; Python, C++, Java gibi popüler programlama dilleriyle kullanılması, bu kütüphaneyi kullanmak için avantaj sağlamaktadır.

Bilginin işlenmesi ve anlamlı sonuçlar çıkartılmasının yanı sıra yapay sinir ağlarını kolaylaştıran, içerisinde birçok fonksiyon barındıran TensorFlow da kullanılmıştır.

3.2.1.4 TensorFlow

TensorFlow, Google tarafından 2015 yılında geliştirilen açık kaynaklı ve kodlu bir derin öğrenme kütüphanesidir. Makine öğrenmesi ve derin öğrenme projelerini gerçekleştirmek için ortam sunmaktadır. Bu kütüphane çok farklı ortamlarda çalışabileceği gibi değişik programlama dilleri ve değişik donanımlarda da çalışabilmektedir.

TensorFlow; makine öğrenimi, derin öğrenme algoritmaları ve modellerini entegre edebilmektedir. Makine öğrenimi ve derin öğrenme için Python'un kullanılmasına imkân tanır ve uygulamalar oluşturabilmek için bir front end API sunar.

Bu kütüphane sayesinde el yazısıyla yazılmış rakamlar sınıflandırılabilir; tekrarlayan sinir ağları, görüntü tanıma, doğal dil işleme ve kısmi diferansiyel denklem

gibi birçok makine öğrenmesi uygulamaları için derin sinir ağları kolayca eğitilebilir ve çalıştırılabilir.

TensorFlow, ücretsiz ve kullanımının kolay olmasının ötesinde her düzeyde makine öğrenimi geçmişine sahip tüm geliştiricilerin makine öğrenmesi ve derin öğrenme modellerini sıfırdan oluşturmak yerine onların güçlü bir kütüphaneye erişmesini sağladığı için oldukça popülerdir. Diğer yandan hem mobil uygulamalar hem de back end hizmetleri için bu alanda yazılım geliştirmeyi kolaylaştırmaktadır. Bu nedenlerden dolayı kullanımı tercih edilmiştir.

3.2.1.5 NumPy ve Matplotlib

Bunların dışında çalışmaya Numpy ve Matplotlib kütüphaneleri de dâhil edilmiştir.

NumPy, bilimsel hesaplamalarda ve işlemlerde kullanılan bir python kütüphanesidir. Bu kütüphane matematiksel işlemler yaparak veri işlemleri gerçekleştirir. Bu kütüphane, pythonun içerisinde bulunan bir kütüphane olmayıp ayrıca kurularak pythona dâhil edilmesi gereken bir kütüphanedir.

Bu paket çok boyutlu dizi nesnelere manipüle edebildiğiniz ve hızlıca yaratabildiğiniz; sıralama, seçme, lineer cebirden, Fourier Transformasyona kadar tamamen matematiğin gücünü kullanabileceğiniz eşsiz bir kütüphanedir.

Homojen dizi desteği, matris işlemleri, sayısal işlemler ve çok boyutlu diziler özellikleriyle veri işleme, istatistik, sayısal hesaplamalar, makine öğrenmesi, derin öğrenme alanlarında kullanılmaktadır.

Matematiksel işlemler ve matris işlemleri python dizileri yerine, NumPy ile daha verimli ve hızlı bir şekilde kullanıldığından bu kütüphane tercih edilmiştir.

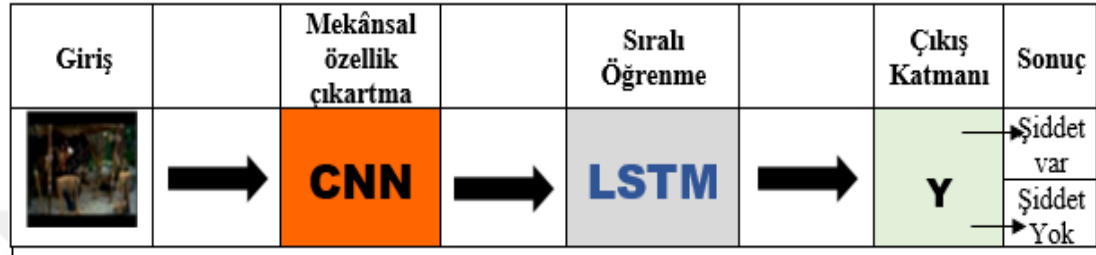
Matplotlib; karmaşık ve dağınık verileri düzenleyerek kolay anlaşılır, yorum yapılacak hâle getirmek, bir başka deyişle veri görselleştirmesi için kullanılan temel python kütüphanesidir. 2 ve 3 boyutlu çizimler yapabilmek için kullanılır. Bu projede 2 boyutlu çizim yapmak için kullanılmıştır.

3.2.2 Modelin Blok Diyagramı

Görüntü verilerini işlemek için Evrişimsel Sinir Ağları çok iyi bir çözümdür. LSTM ağları ile çok harika çalışmaktadır. Bilgisayarla görme problemlerinden zor olanların ikisi birleştirilerek çözüm üretmek mümkündür. Şiddet tahmin etmek için LSTM (CNN-BI-LSTM) modeli uygundur. Bu modelde, ilk olarak videolar karelere

bölünür, her kareyi bir konvolüsyonel sinir ağından geçirerek o anki karede bulunan bilgiler çıkartılır, ardışık olay akışını tanımlamak için çift yönlü LSTM katmanı kullanılır.

Aşağıda açıklanan deneylerin sonucuna göre seçilen model MobileNetV2 ve BILSTM modelidir. Çıkışta Dense Layer kullanılarak bu katmanın çıkışında videoda şiddet olup olmadığı tahmin edilmiştir.



Şekil 23: Modelin Akış Diyagramı

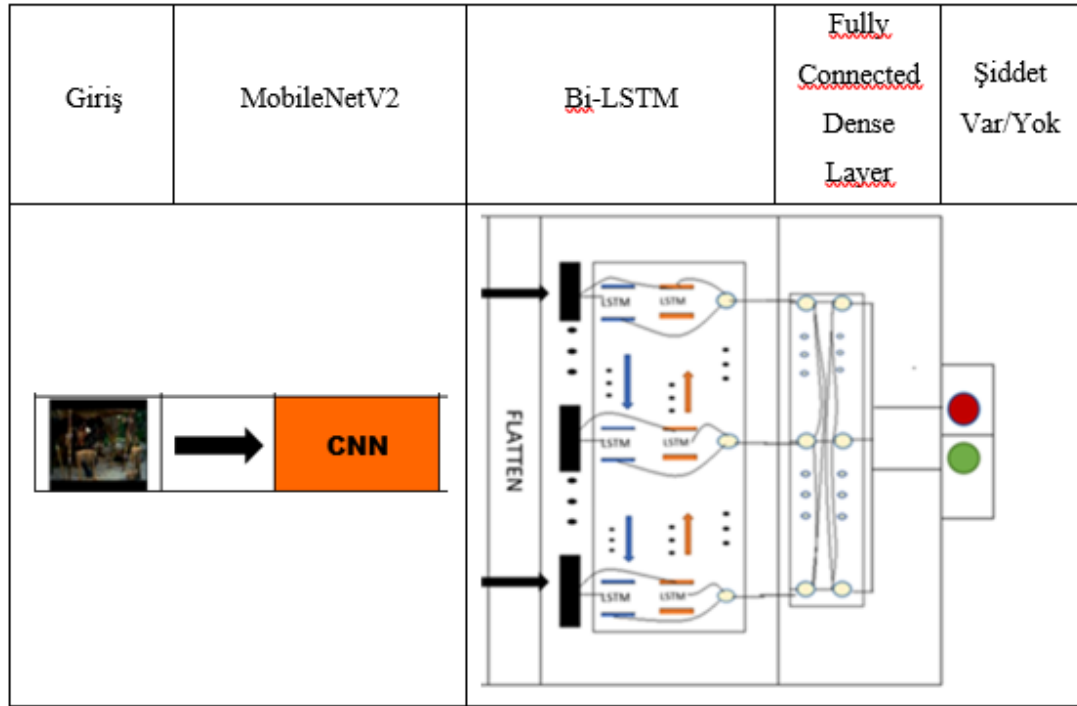
MobileNetV2 hafif derin konvolüsyonlar oluşturmak için derinlemesine ayrılabilir konvolüsyonlar kullanan aerodinamik bir mimariye dayanmaktadır (Howard vd. 2017). Derinlemesine Ayrılabilir Konvolüsyon (Depthwise Separable Convolutions) tekniğini kullandığı için standart konvolüsyon işlemine nazaran sekiz veya dokuz kat daha az parametreyle özellik çıkarımı yapılabilmektedir.

Tablo 3: MobileNet Genel Mimarisi (Howard vd. 2017)

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Bidirectional, LSTM sıralı bir işlem olduğundan kullanma ihtiyacı duyulmuştur. Zamansal açıdan yaklaşan ya da tersten yaklaşan karelerin özelliklerinin tespitinde kullanılmıştır. Şekil 25'te Bidirectional LSTM gösterilmiştir.

Dense Layers Derin Öğrenme söz konusu olduğunda bu katmanlar her yerde kullanılmaktadır. Burada, tam bağlantılı yoğun katmanlar rastgele ağırlıkların eklenmesine yardımcı olur. Şekil 24'te Dense Layer gösterilmiştir.



Şekil 24: Model Diyagramında Dense Layer

Yüksek doğruluk oranı ve hızlı bir şiddet tespiti detektörü hedeflerken önceki çalışmalardan güçlü ve avantajlı bölümlerden esinlenerek yeni bir model oluşturuldu. Bu modelin blok diyagramı yeniden tasarlandı. Eğitim ve test veri seti olarak Real-Life Violence Situations veri seti kullanıldı.

3.2.3 Verinin Hazırlanması

Bu çalışmada iki sınıf bulunduğu için veri kümesi iki dizinden oluşmaktadır: Şiddet içeren ve şiddet içermeyen. Bu çalışmada 3 adet veri seti kullanılmıştır. Bunlardan biri hockey fight veri seti (Bermejo Nievas vd. 2011: 3), Ulusal Hokey Ligi'nden 500'ü şiddet içeren ve 500'ü şiddet içermeyen 1000 adet aksiyon içeren klipten oluşmaktadır. Bir diğeri movies dataset (Shilaskar vd. 2023: 1) veri kümesi, filmlerden toplanan dövüş sahnelerinden oluşmaktadır. Toplamda 200 adet video

klipten oluşan bu veri seti, 100 adet şiddet içeren ve 100 adet şiddet içermeyen video şeklinde ikiye ayrılmıştır. Bir diğer veri seti de Real-Life Violence Situations (Soliman vd. 2019: 82) veri setidir. Veri kümesi, eşit sayıda şiddet içeren ve şiddet içermeyen film kliplerini içermektedir. Her dizinde 1000 adet klip bulunmaktadır. Toplamda 2000 adet kısa video bulunmaktadır. Şiddet klipleri sokak, hapishane ve okul gibi birçok farklı alandaki kavgaları içerir; şiddetsizlik videoları ise oyun, futbol, basketbol, tenis, yüzme ve yemek yeme gibi diğer insan eylemlerini içermektedir. Bu veri kümesi videolarının bir kısmı manuel olarak çekilmiştir. Ancak kişi ve kurum fazlalığını önlemek için bazı videolar YouTube'dan toplanmıştır (Soliman vd. 2019: 83).

Model eğitiminde kullanılan veri seti, modelin performansını sınırlayabilmektedir. Bazı durumlarda veri seti içerisindeki görüntüleri öğrenmek yerine, veri setindeki kalıpları ezberlemeye doğru gitmektedir. Bu tip veri setleri özel olarak hazırlandığından görüntüler ve kalıplar birbirine yakın olmaktadır. Gerçek hayatta burada görülen performansın altında bir performans beklenmektedir.

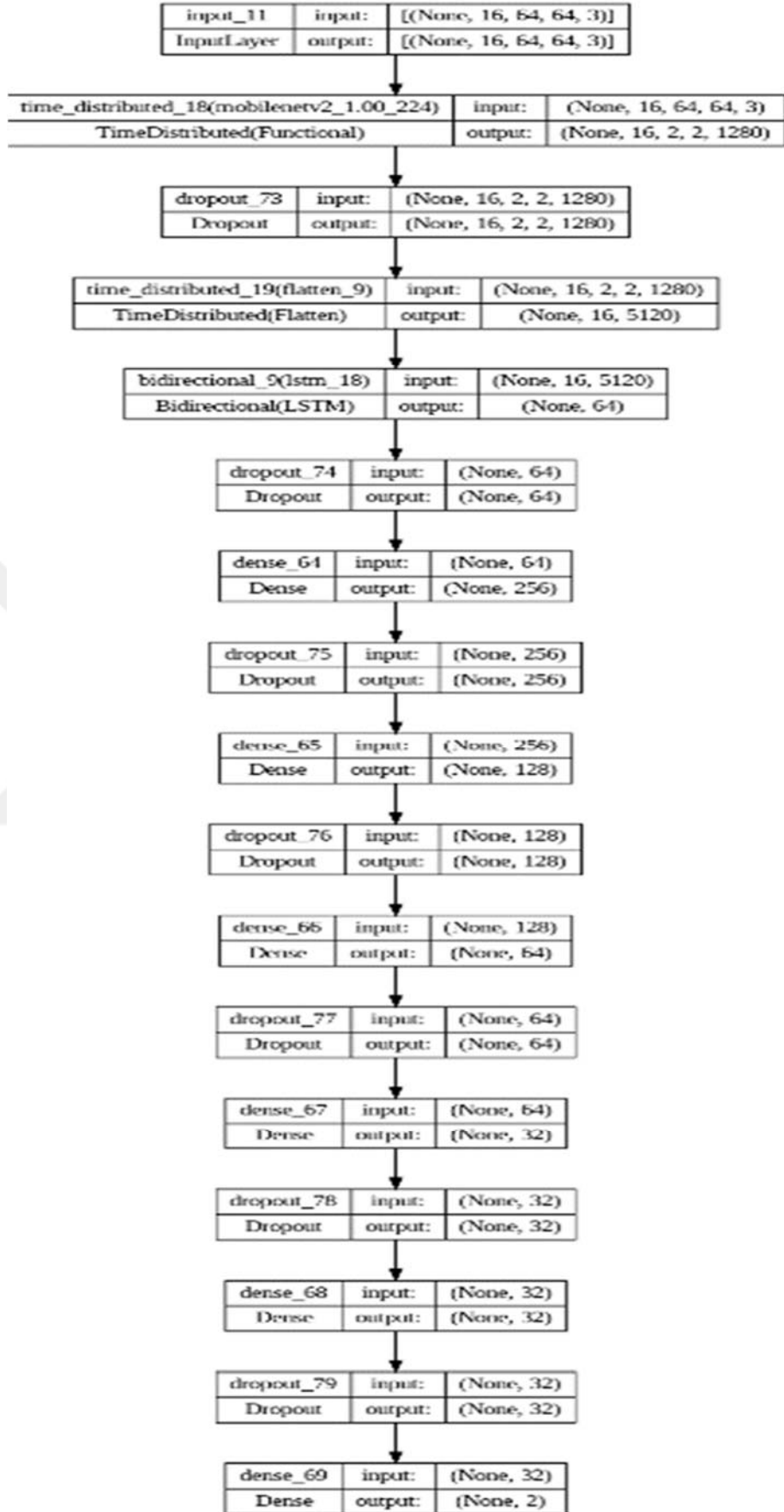
Videoları dizinlere ayırdıktan sonra çerçevelere ayırmak gerekmiştir. Videoları çerçevelere ayırırken 16 kare/sn. hız kullanılmıştır. Bu çerçeveler yeniden boyutlandırılıp normalizasyon işleminden geçirilmiştir. Burada amaç, değerleri 0 ile 1 arasına sığdırmaktır. Daha sonra bu dosyalar npy formatında kaydedilmiştir.

Bu çerçevelerin ikili (binary) olarak temsil edilmesi için One Hot Encoding tekniği kullanılmıştır. Tek seferlik kodlamada, kategorik verinin bir özelliği her seviye için bir yeni değişken ve her kategori 0 veya 1 içeren ikili bir değişken ile eşleştirilmiştir (Dahouda ve Joe, 2021: 114382).

Etiketler One Hot Encoding tekniği ile vektöre dönüştürüldükten sonra eğitim ve test verileri olmak üzere ikiye ayrılmıştır. Veriler; önce %80 eğitim, %20 test verisi şeklinde ayrılmış ve bu verilere göre deney yapılmıştır. Daha sonra veriler, %90 eğitim, %10 test verisi şeklinde yeniden ayrılmış, deneyler bir kez de bu verilere göre yapılmıştır.

3.2.4 Modelin Oluşturulması

Model, mekânsal özelliklerine ek olarak iki yönlü zamansal süreçleri işlemektedir. Bu da modelin aynı zamanda daha doğru daha az hesaplama süresi tüketmesine yol açmaktadır. Bu durumdan dolayı düşük maliyetlidir. Modeli oluşturmak için açık kaynaklı TensorFlow kullanılmıştır.



Şekil 25: Önerilen Model

3.2.5 Eğitim Ayarları

Oluşturulan modeli eğitmek için kayıp fonksiyonu categorical crossentropy, ayrıca bu bir sınıflandırma problemi olduğundan one-hot encoding kullanılmıştır. Batch size 8, epoch 50 olarak ayarlanmıştır. Dropout değeri 0.32 olarak girilmiştir.

3.2.6 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları seçimi için gizli katmanlarda Relu, sınıflandırma katmanında Softmax kullanılmıştır.

ReLU aktivasyon fonksiyonu negatif değerlerde sıfır ve pozitif değerlerde fonksiyonun değeri ne ise onu verir. Derin öğrenmede yaygın kullanılmış ve popüler hâle gelmiştir. Bunun nedeni, bu fonksiyonun hesaplama verimliliği ve kaybolan gradyan probleminden kaçınma yeteneğidir. Bu fonksiyon, doğrusal olmayan bir fonksiyondur ve diğerlerine göre daha verimlidir.

Softmax aktivasyon fonksiyonunun yaygın olarak çok sınıflı sınıflandırma problemleri çözümünde ve yapay sinir ağlarında kullanılan son aktivasyon fonksiyonu olarak kullanılması tercih edilir. Bu fonksiyon, girdileri birden fazla sınıf üzerinde olasılık dağılımına dönüştürülerek sınıflar arasındaki ilişkiyi belirler.

3.2.7 Dropout

Dropout katmanı, ağ içindeki bazı bağlantıların dışarıdan girilen değerlere göre kaldırılmasıyla eğitim performansı artabilmektedir. Tam bağlı katmanlarda belli eşik değerinin altındaki düğümlerin düşürülmesinin başarıyı artırdığı gözlenmiştir. Dropout katmanına 0 ile 1 arasında bir değer verilmektedir. Ağ zayıf bilgi içeren alanları azaltmaktadır. Dropout değerini tüm katmanlarda aynı vermek zorunluluğu yoktur. Bu çalışmada, dropout değeri tek kullanılmış ve 0.32 verilmiştir.

3.2.8 Başarının Değerlendirilmesi

Sınıflandırma algoritmalarında başarı oranının belirlenmesinde karmaşıklık (confusion) matrisi kullanılır. Model oluşturulduktan sonra veri seti eğitim verisi ve test verisi diye iki farklı gruba ayrılır. Model, eğitim veri setiyle eğitildikten sonra test veri setiyle başarı değerlendirmesi yapılır. Eğitim verisi ile performans değerlendirmesi yapılması yanlış bir sonuç alınmasına neden olur. Bu da performansı doğru değerlendirmiş olmaz. Bu yüzden performans değerlendirilmesi test verisi ile yapılır. Karmaşıklık matrisi, verideki var olan durum ile sınıflama modelimizin doğru

ve yanlış tahminlerinin sayısını gösterir. Sınıflandırmada başarı ölçütü belirlemede kullanılan karmaşıklık matrisinde Doğruluk (accuracy), Duyarlılık (precision), Kesinlik (recall) ve F1 değerleri bulunmaktadır. Karmaşıklık matrisinin sınıflandırma problemlerine göre boyutu değişebilmektedir. Bu çalışmada ikili sınıflandırma problemi olduğundan karmaşıklık matrisi 2X2 boyutunda bir matristir.

Tablo 4: Karmaşıklık Matrisi

		Mevcut Durum	
		Pozitif	Negatif
Tahmin	Pozitif	DP(TP)	YN(FN)
	Negatif	YP(FP)	DN(TN)

Tablo 4'te satırlar tahminleri gösterirken sütunlar gerçek durumu göstermektedir. Bu tabloda:

DP (TP) : Modelin tahmini doğru iken, gerçekte de doğru olan durumdur.

YN (FN): Modelin tahmini yanlış iken, gerçekte doğru olan durumdur.

YP (FP): Modelin tahmini doğru iken, gerçekte yanlış olan durumdur.

DN (TN): Modelin tahmini yanlış iken, gerçekte yanlış olan durumdur.

Karmaşıklık matrisinin terimleri kullanılarak Doğruluk (Accuracy), Duyarlılık (Precision) ve Kesinlik (Recall) değerleri hesaplanır.

3.2.8.1 Doğruluk (Accuracy):

Sistemde doğru ve yanlış olarak yapılan doğru tahminlerin tüm tahminlere oranıdır. Eğer veri dengesiz ise tek başına bu değer yanılgıya düşürebilir. Dengesiz veri olduğu durumlarda Duyarlılık-Kesinlik ve F1 değerlerine bakılır. Doğruluk gerçekte olduğu gibi model tarafından tahmin edilen değerlerin toplam değerlere oranıdır. Bu çalışma da şiddet içeren veya içermeyen sahnelerden doğru bilinenlerin tüm sahnelere oranıdır. Doğruluk değerini hesaplamak için (3.1) formülü kullanılır.

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

3.2.8.2 Duyarlılık (Precision):

Pozitif olarak tahminde bulunulan bir durumdaki başarıyı gösteren durumdur. DP değerlerinin modelin yaptığı toplam doğru tahmin sayısına bölünmesiyle Duyarlılık hesaplanır. Önerilen modellerin amaçları doğrultusunda bu, doğru bir şekilde şiddet içermeyen olarak sınıflandırılan değerlerin tüm şiddet içermeyenlerin miktarına bölünmesi anlamına gelmektedir.

Duyarlılık, adından da anlaşılacağı üzere bir modelin doğruluğunun ölçüsüdür. Çok sayıda YN vakası varsa yani yanlışlıkla dövüş görüntüsü olarak etiketlenen dövüş dışı videolar varsa, hassasiyet değeri düşecektir. Duyarlılık hesaplama formülü (3.2)'de verilmiştir.

$$\text{Duyarlılık (Precision)} = \frac{DP}{DP + YN} \quad (3.2)$$

3.2.8.3 Kesinlik (Recall):

Pozitif durumların ne kadar doğru tahmin edildiğini gösterir. DP değerlerinin veri kümesindeki toplam pozitif örnek sayısına bölünmesiyle geri çağırma hesaplanır. Önerilen modeller için bu, toplam şiddet videosu sayısının doğru bir şekilde şiddet olarak kategorize edilen şiddet sayısına bölünmesi anlamına gelir.

Modelin geri çağırılması, pozitif örneklerin ne kadar iyi tahmin edebildiğinin bir göstergesi olarak hizmet eder.

Çok sayıda YP vakası varsa modelin geri çağırma oranı düşük olacaktır. Kesinlik hesaplamak için (3.3) formülü kullanılır.

$$\text{Kesinlik (Recall)} = \frac{DP}{DP + YP} \quad (3.3)$$

3.2.8.4 F-Ölçütü

Duyarlılık (Precision) ve Kesinlik (Recall) metriklerin harmonik ortalamasıdır. Bu, 0 ila 1 aralığında birleştirilmiş bir Duyarlılık ve Kesinlik metriğidir. Çoğu durumda F-ölçütü değeri 1'e yakınsa model büyük bir Duyarlılık ve Kesinlik sergiler. F-ölçütü hesaplama formülü (3.4)'te verilmiştir.

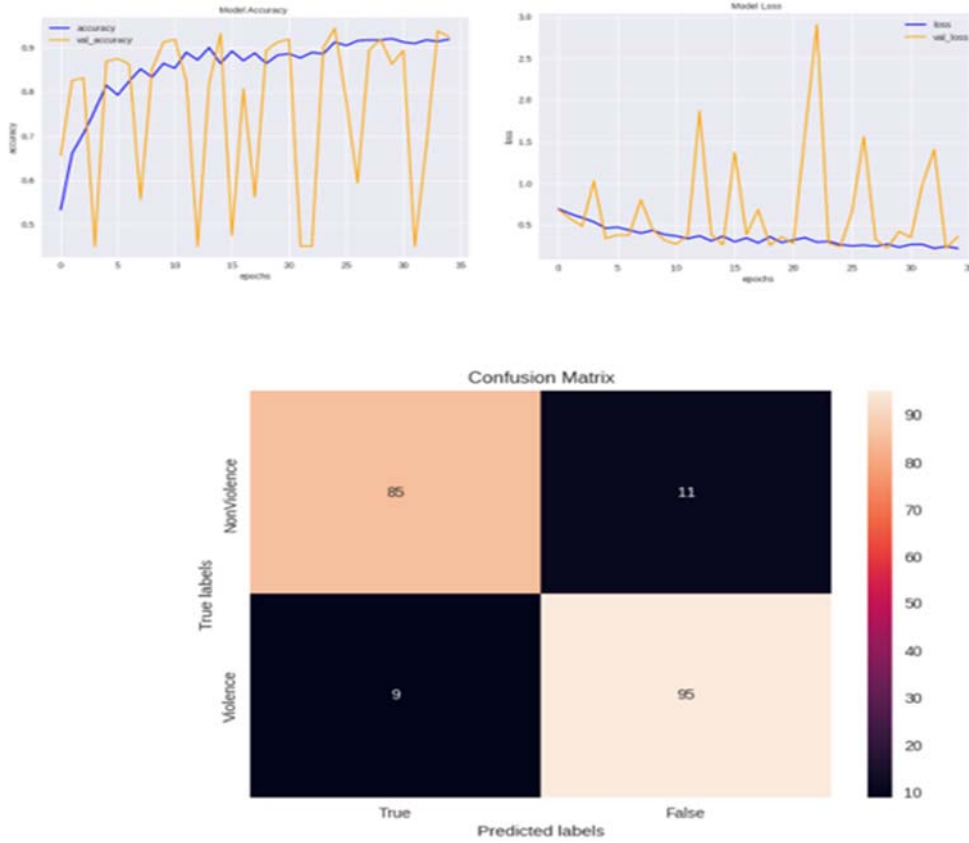
$$F - \text{ölçütü} = \frac{2 * \text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (3.4)$$

3.2.9 Deneyler ve Sonuçları

Bu bölümde hedeflenen doğruluğu yüksek ve hızlı model oluşturmak için yapılan deneyler sunulacaktır. İlk olarak veri seti %80 eğitim verisi, %20 test verisi olarak ayrılmıştır. Katmanlar ve bazı dışarıdan girilen parametreler yeniden oluşturulmuştur. Bu şartlar altında ResNet50, MobileNetV2, MobileNetV3Small, MobileNetV3Large transfer öğrenme modelleri ayrı ayrı kullanılarak Real-Life Violence Situations, Movies ve Hockey Fight veri setleri ile deneyler yapılmıştır. Daha sonra katmanlar ve dışarıdan girilen parametreler değiştirilmeden veri seti %90 eğitim, %10 test olarak yeniden ayrılmıştır. Bu durumda tekrar ResNet50, MobileNetV2, MobileNetV3Small, MobileNetV3Large transfer öğrenme modelleri için deneyler tekrar Real-Life Violence Situations, Movies ve Hockey Fight veri setleri için yapılmış ve %99,37 oranında başarı elde edilmiştir. Deney ayrıntıları ve sonuçları aşağıda verilmiştir:

3.2.9.1 Deney 1 ve Sonuçları

Deney 1’de veri seti eğitim %80, test %20 olarak seçilmiştir. Kullanılan önceden eğitilmiş ağ ResNet50 seçilmiş olup, eğitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine göre deneyler yapılmıştır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Şekil 26’da gösterilmiştir.



Şekil 26: Deney 1'in Model Accuracy, Model Loss, Confusion Matrix Grafikleri

Bu deney üç farklı deney kümesi ile yapılmıştır. Bu veri setleri Real-Life Violence Situations, Movies ve Hockey Fight'tır. En iyi sonuç Hockey Fight very seti ile elde edilmiş olup doğruluk oranı 91 çıkmıştır. Verilerin dağılımına bakınca dengeli bir dağılım olduğundan doğruluk oranı, bir anlam ifade etmektedir. Modelin performansını daha iyi yorumlamak için karmaşıklık matrisinin değerlerinin oranlarına bakmak uygun olacaktır.

Bu deneyde formülleri yukarıda verilen doğruluk, kesinlik, hassasiyet ve F1 değerlerine bakınca sırasıyla yüzde olarak 90, 90, 91, 90 oranları elde edilmiştir.

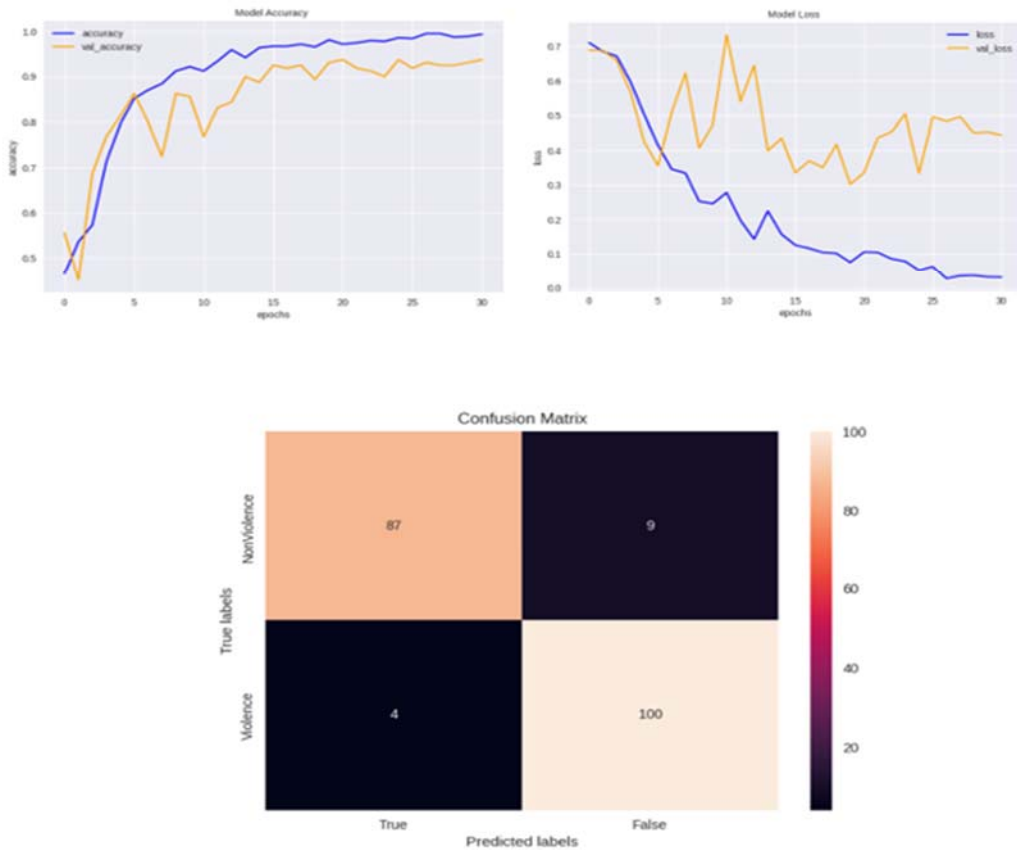
Verideki 85 tane şiddet içermeyen video ve 95 tane şiddet içeren video doğru tahmin edilmiştir. 11 tane şiddet içermeyen video model tarafından hata yapılarak şiddet içeren video şeklinde tahmin edilmiştir. 9 adet de şiddet içeren video model tarafından hata yapılarak şiddet içermeyen şeklinde tahmin edilmiştir. Model bu senaryoda şiddet içermeyen video tahmininde daha çok zorluk çekmiştir. Bir başka ifadeyle model, bu senaryoda şiddet içeren videoları daha iyi tahmin etmiştir. Bu deneyle ilgili veri setlerine göre elde edilen sonuçlar Tablo 5'te verilmiştir.

Tablo 5: Deney 1’den Elde Edilen Sonuçlar

Train/Test Split 80/20							
ResNet50							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Acuracy Test	Precision	Recall	F1
Real-Life Violence Situations	80	20	68,67	70	69	73	71
Movies	80	20	70,01	53	48	64	55
Hockey Fight	80	20	91	90	90	91	90

3.2.9.2 Deney 2 ve Sonuçları

Deney 2’de veri seti eğitim %80, test %20 olarak seçilmiştir. Kullanılan önceden eğitilmiş ağ olarak MobileNetV2 seçilmiş, eğitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine göre deneyler yapılmıştır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Şekil 27’de gösterilmiştir.



Şekil 27: Deney 2’nin Model Accuracy, Model Loss, Confusion Matrix grafikleri

Bu deneyde doğruluk oranı 99.37 çıkmıştır. Verilerin dağılımına bakınca dengeli bir dağılım olduğundan doğruluk oranı bir anlam ifade etmektedir. Bu deneyde

formülleri yukarıda verilen Doğruluk, Kesinlik, Hassasiyet ve F1 değerlerine bakınca sırasıyla yüzde olarak 94, 92, 96, 94 oranları elde edilmiştir. Bu deneyde en iyi sonuç Hockey Fight veri seti ile elde edilmiştir. Real-Life Violence Situations veri seti accuracy oran değeri bu değere yakın çıkmıştır.

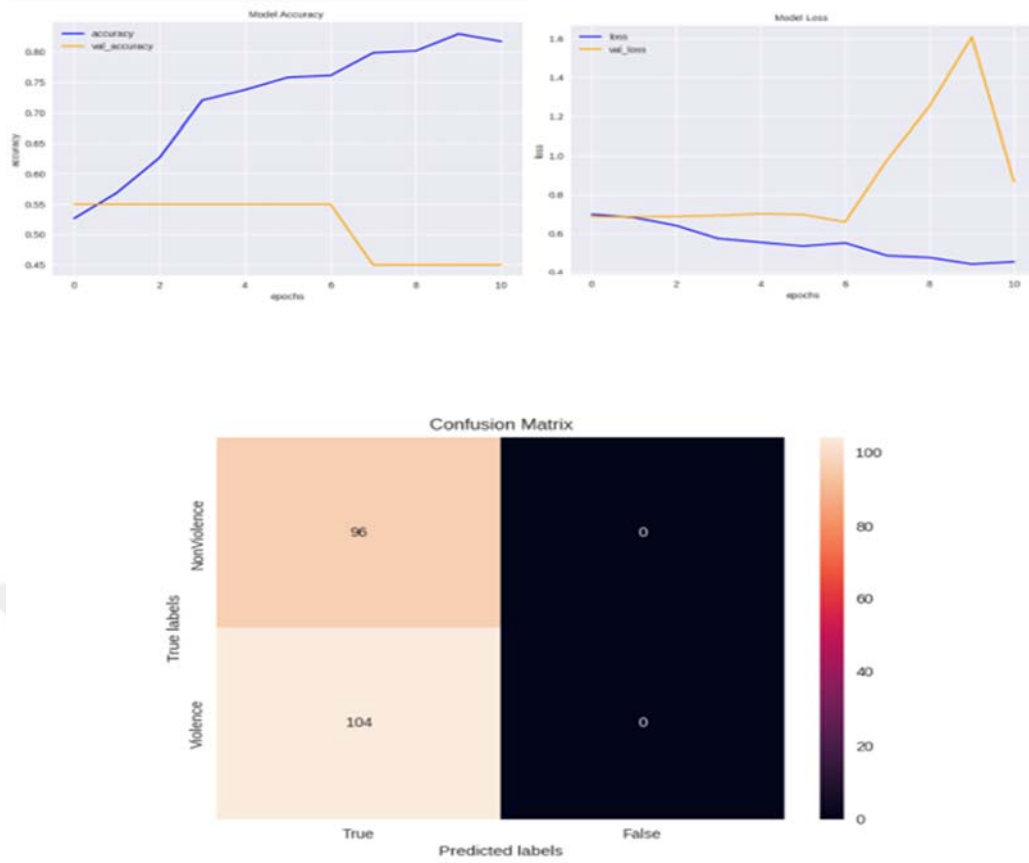
Verideki 87 tane şiddet içermeyen video ve 100 tane şiddet içeren video doğru tahmin edilmiştir. 9 tane şiddet içermeyen video model tarafından hata yapılarak şiddet içeren video olarak tahmin edilmiştir. 4 adet de şiddet içeren video model tarafından hata yapılarak şiddet içermeyen şekilde tahmin edilmiştir. Model, bu senaryoda şiddet içermeyen video tahmininde daha çok zorlanmıştır. Bir başka ifadeyle model, bu senaryoda şiddet içeren videoları daha iyi tahmin etmiştir. Bu deneyle ilgili veri setlerine göre elde edilen sonuçlar Tablo 6’da verilmiştir.

Tablo 6: Deney 2’den Elde Edilen Sonuçlar

Train/Test Split 80/20							
MobilnetV2							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
Real-Life Violence Situations	80	20	99,06	92	89	96	92
Movies	80	20	50,64	45	45	95	61
Hockey Fight	80	20	99,37	94	92	96	94

3.2.9.3 Deney 3 ve Sonuçları

Deney 3’te veri seti eğitim %80, test %20 olarak seçilmiştir. Kullanılan önceden eğitilmiş ağ olarak MobileNetV3Small seçilmiş, eğitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine göre deneyler yapılmıştır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Şekil 28’de gösterilmiştir.



Şekil 28: Deney 3'ün Model Accuracy, Model Loss, Confusion Matrix Grafikleri

Bu deneyde en iyi doğruluk oranı 81.72 çıkmıştır. Verilerin dağılımına bakınca dengeli bir dağılım olduğundan doğruluk oranı bir anlam ifade etmemektedir. Bu deneyde formülleri yukarıda verilen Doğruluk, Kesinlik, Hassasiyet ve F1 değerlerine bakılınca sırasıyla yüzde olarak 69, 48, 1, 65 oranları elde edilmiştir. Bu deneyde en iyi sonuç Hockey Fight veri seti ile elde etmiştir.

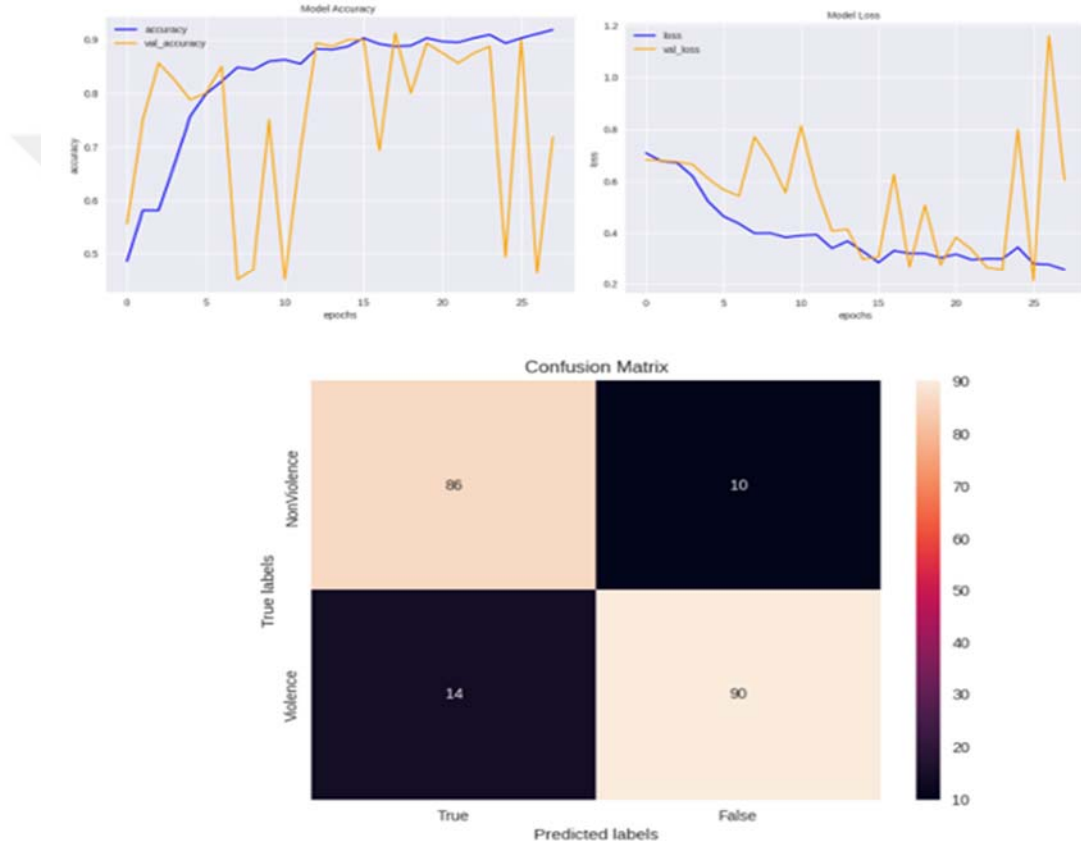
Bu model, şiddet içeren 104 adet videoya şiddet yok demiştir. Şiddet içermeyen 96 adet videoya ise şiddet yok diyerek doğru tahmin etmiştir. Bu deneyle ilgili veri setlerine göre elde edilen sonuçlar Tablo 7'de verilmiştir.

Tablo 7: Deney 3'ten Elde Edilen Sonuçlar

Train/Test Split 80/20							
MobileNetV3Small							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
Real-Life Violence Situations	80	20	69,51	53	52	96	67
Movies	80	20	60,26	51,02	48	1	65
Hockey Fight	80	20	81,72	69	48	1	65

3.2.9.4 Deneý 4 ve Sonuları

Deneý 4'te veri seti eęitim %80, test %20 olarak seilmiřtir. Kullanılan nceden eęitilmiř aę olarak MobileNetV3Large seilmiř, eęitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine gre deneýler yapılmıřtır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Őekil 29'da gsterilmiřtir.



Őekil 29: Deneý 4'ün Model Accuracy, Model Loss, Confusion Matrix Grafikleri

Bu deneýde doęruluk oranı 91,87 çıkmıřtır. Bu deneýde de iyi sonu Hockey Fight veri seti ile elde edilmiřtir. Verilerin daęılımına bakınca dengeli bir daęılım olduęundan doęruluk oranı anlam ifade etmektedir. Bu deneýde formlleri yukarıda verilen Doęruluk, Kesinlik, Hassasiyet ve F1 deęerlerine bakılınca sırası ile yüzde olarak 88, 90, 87, 88 olarak elde edilmiřtir. Bu modelde deęerlere gre řiddet iermeyen videoları doęru tahmin etme oranı yksektir. řiddet iermeyen videolar daha kolay tahmin edilmektedir. nceki deneýlerle karřılařtırıldıęında bu model

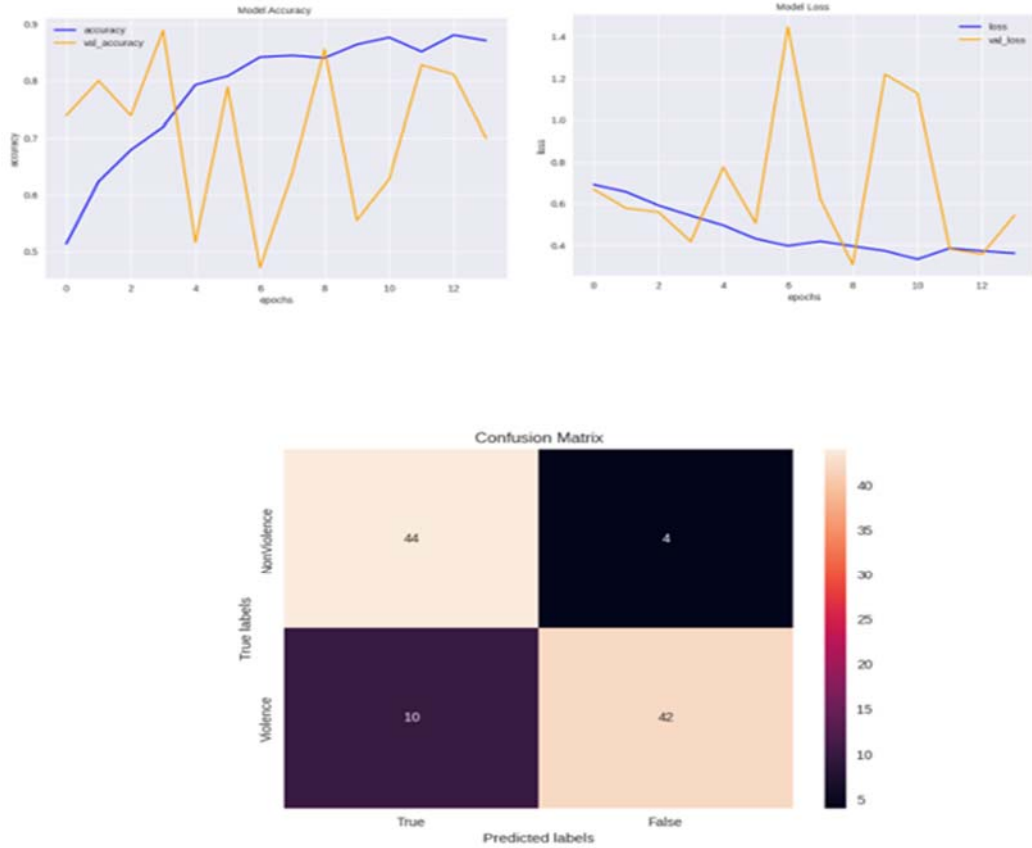
Deney1 ve Deney 2’deki modellerden daha iyi değildir. Bu deneyle ilgili veri setlerine göre elde edilen sonuçlar Tablo 8’de verilmiştir.

Tablo 8: Deney 4’ten Elde Edilen Sonuçlar

Train/Test Split 80/20							
MobileNetV3Large							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
Real-Life Violence Situations	80	20	62,66	59,50	57	75	65
Movies	80	20	51,92	45	45	1	62
Hockey Fight	80	20	91,87	88	90	87	88

3.2.9.5 Deney 5 ve Sonuçları

Deney 5’te veri seti eğitim %90, test %10 olarak seçilmiştir. Kullanılan önceden eğitilmiş ağ olarak ResNet50 seçilmiş, eğitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine göre deneyler yapılmıştır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Şekil 30’da gösterilmiştir.



Şekil 30: Deney 5’in Model Accuracy, Model Loss, Confusion Matrix Grafikleri

Bu deneylerde en iyi doğruluk oranı 88 çıkmıştır. En iyi sonucu Hockey Fight veri seti vermiştir. Verilerin dağılımına bakınca dengeli bir dağılım olduğundan doğruluk oranı bir anlam ifade etmektedir. Bu deneyde formülleri yukarıda verilen Doğruluk, Kesinlik, Hassasiyet ve F1 değerlerine bakılınca sırasıyla yüzde olarak 86, 91, 81, 86 oranları elde edilmiştir.

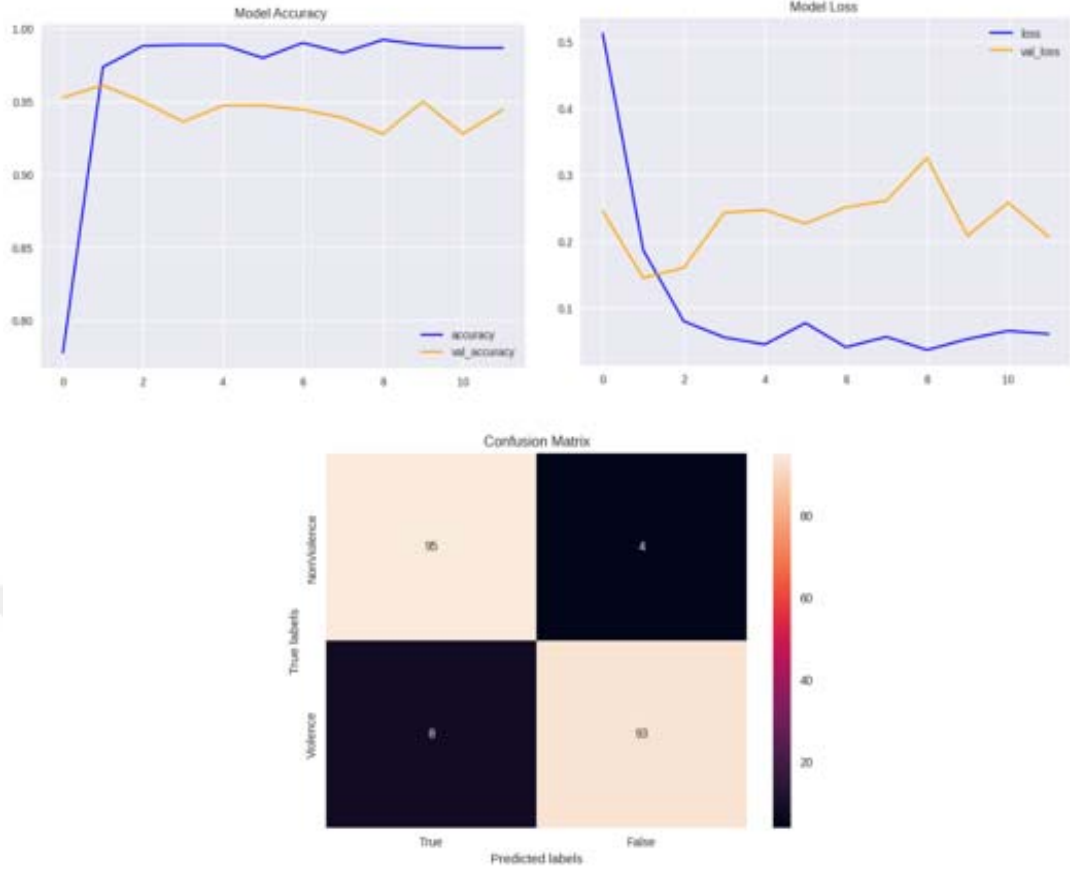
Verideki 44 tane şiddet içermeyen video ve 42 tane şiddet içeren video doğru tahmin edilmiştir. 4 tane şiddet içermeyen video, model tarafından hata yapılarak şiddet içeren video şeklinde tahmin edilmiştir. 10 adet de şiddet içeren video model tarafından hata yapılarak şiddet içermeyen video şeklinde tahmin edilmiştir. Model, bu senaryoda şiddet içeren video tahmininde daha çok zorlanmıştır. Bir başka ifadeyle model, bu senaryoda şiddet içermeyen videoları daha iyi tahmin etmiştir. Deney 1 ile karşılaştırıldığında bu modelin daha başarısız olduğu söylenebilir. Bu deneyle ilgili veri setlerine göre elde edilen sonuçlar Tablo 9’da verilmiştir.

Tablo 9: Deney 5’ten Elde Edilen Sonuçlar

Train/Test Split 90/10							
ResNet50							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
Real-Life Violence Situations	90	10	49,86	52,5	1	06	11
Movies	90	10	80,31	52	52	1	68
Hockey Fight	90	10	88	86	91	81	86

3.2.9.6 Deney 6 ve Sonuçları

Deney 6’da veri seti eğitim %90, test %10 olarak seçilmiştir. Kullanılan önceden eğitilmiş ağ olarak MobileNetV2 seçilmiş, eğitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine göre deneyler yapılmıştır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Şekil 31’de gösterilmiştir.



Şekil 31: Deneş 6'nın Model Accuracy, Model Loss, Confusion Matrix Grafikleri

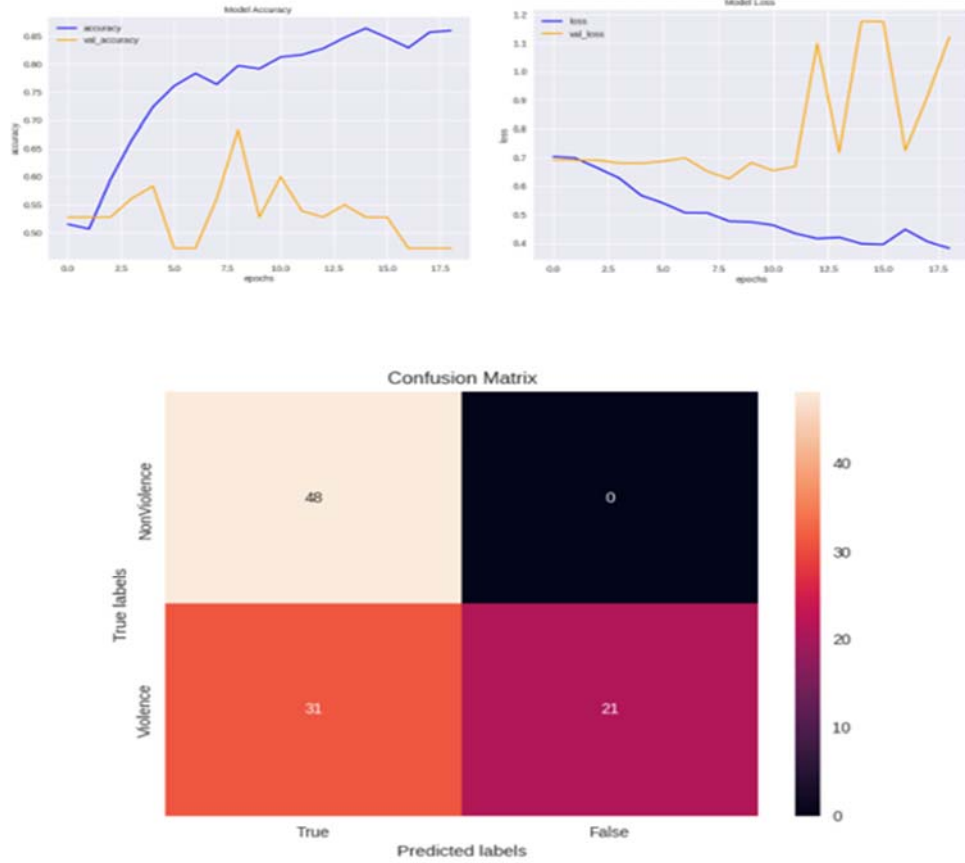
Deneş 6'da en iyi doğruluk oranı 98,68 çıkmıştır. Bu deneşde en iyi doğruluk oranını Real-Life Violence Situations veri seti vermiştir. Verilerin dağılımına bakılınca dengeli bir dağılım olduğundan doğruluk oranı bir anlam ifade etmektedir. Bu deneşde formülleri yukarıda verilen Doğruluk, Kesinlik, Hassasiyet ve F1 değerlerine bakılınca sırasıyla yüzde olarak 94, 96, 92, 94 oranları elde edilmiştir. Bu deneşle ilgili veri setlerine göre elde edilen sonuçlar Tablo 10'da verilmiştir.

Tablo 10: Deneş 6'dan Elde Edilen Sonuçlar

Train/Test Split 90/10							
MobileNetV2							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
Real-Life Violence Situations	90	10	98,68	94	96	92	94
Movies	90	10	96,02	84	90	75	82
Hockey Fight	90	10	98,61	93	94	92	93

3.2.9.7 Deneý 7 ve Sonuları

Deneý 7’de veri seti eęitim %90, test %10 olarak seilmiřtir. Kullanılan nceden eęitilmiř aę olarak MobileNetV3Small seilmiř, eęitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine gre deneýler yapılmıřtır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Őekil 32’de gsterilmiřtir.



Őekil 32: Deneý 7’nin Model Accuracy, Model Loss, Confusion Matrix Grafikleri

Deneý 7’de doęruluk oranı 85,97 çıkmıřtır. Verilerin daęılımına bakılınca dengeli bir daęılım olduęundan doęruluk oranı bir anlam ifade etmektedir. Bu deneýde formlleri yukarıda verilen Doęruluk, Kesinlik, Hassasiyet ve F1 deęerlerine bakılınca sırasıyla yüzde olarak 69, 1, 40, 58 oranları elde edilmiřtir.

Verideki 48 tane řiddet içermeyen video ve 21 tane řiddet içeren video doęru tahmin edilmiřtir. řiddet içermeyen videoların tamamını model doęru tahmin etmiřtir. 31 adet de řiddet içeren video model tarafından hata yapılarak řiddet içermeyen řeklinde tahmin edilmiřtir. Model, bu senaryoda řiddet içermeyen video tahmininde hata yapmamıřtır. řiddet içeren videolarda fazla hata yapmıřtır. Bir bařka ifadeyle

model, bu senaryoda şiddet içermeyen videoları tahmin edebilirken şiddet içeren videoları tahminde zorlanmıştır.

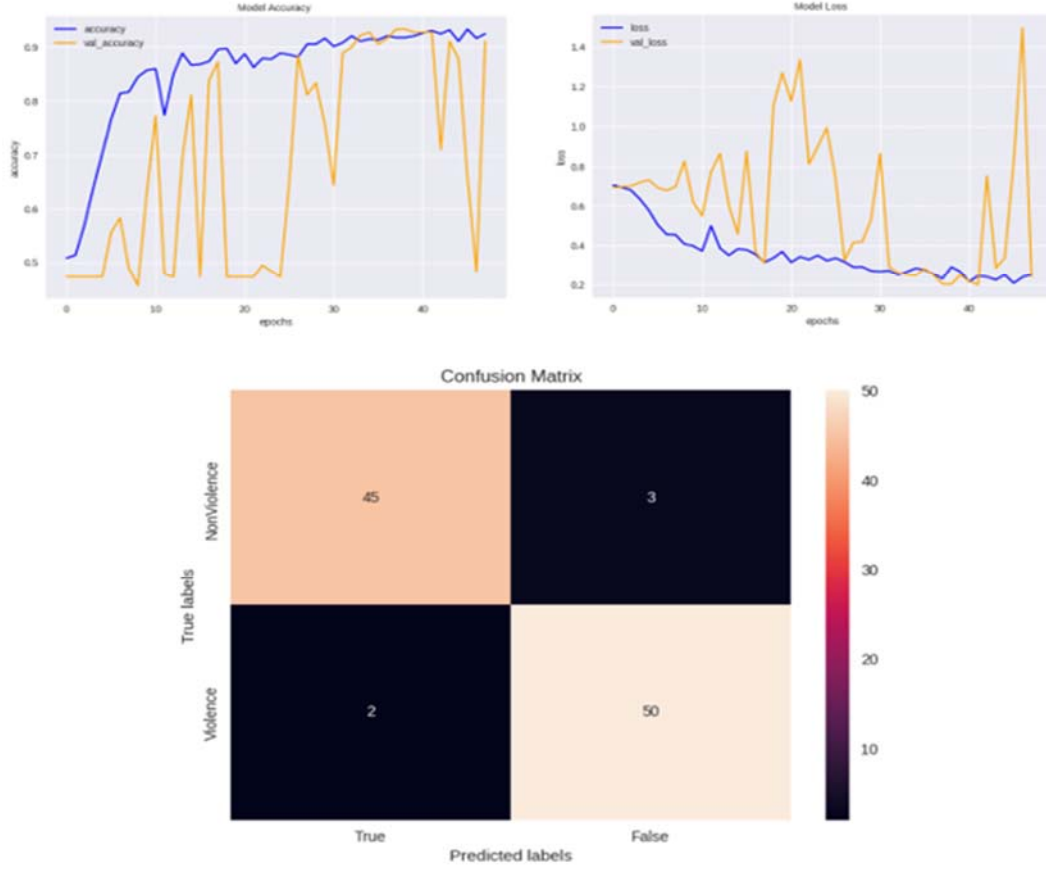
Bu deneyle ilgili veri setlerine göre elde edilen sonuçlar Tablo 11’de verilmiştir.

Tablo 11: Deney 7’den Elde Edilen Sonuçlar

Train/Test Split 90/10							
MobileNetV3Small							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
Real-Life Violence Situations	90	10	69,24	50,50	51	99	67
Movies	90	10	46,59	56	52	1	69
Hockey Fight	90	10	85,97	69	1	40	58

3.2.9.8 Deney 8 ve Sonuçları

Deney 8’de veri seti eğitim %90, test %10 olarak seçilmiştir. Kullanılan önceden eğitilmiş ağ olarak MobileNetV3Large seçilmiş, eğitim ayarlarına uygun parametreler girilip Real-Life Violence Situations, Movies ve Hockey Fight veri setlerine göre deneyler yapılmıştır. En iyi sonucun Model Accuracy, Model Loss, Confusion Matrix grafikleri Şekil 33’te gösterilmiştir.



Şekil 33: Deney 8'in Model Accuracy, Model Loss, Confusion Matrix Grafikleri

Deney 8'de doğruluk oranı 92,5 çıkmıştır. Verilerin dağılımına bakınca dengeli bir dağılım olduğundan doğruluk oranı bir anlam ifade etmektedir. Bu deneyde formülleri yukarıda verilen Doğruluk, Kesinlik, Hassasiyet ve F1 değerlerine bakılınca sırasıyla yüzde olarak 95, 94, 96,95 oranları elde edilmiştir. Bu senaryoda en iyi sonuç Hockey Fight veri seti ile alınırken en kötü sonuç movies veri seti ile alınmıştır. Bu deneyle ilgili veri setlerine göre elde edilen sonuçlar Tablo 12'de verilmiştir.

Tablo 12: Deney 8'den Elde Edilen Sonuçlar

Train/Test Split 90/10							
MobileNetV3Large							
Veri Seti	Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
Real-Life Violence Situations	90	10	85,69	70	64	95	76
Movies	90	10	49	52	50	33	40
Hockey Fight	90	10	92,5	95	94	96	95

Yapılan çalışmada üç farklı veri seti kullanılmıştır. Bunlar Real-Life Violence Situations, Movies ve Hockey Fight veri setleridir. Bununla beraber dört farklı önceden eğitilmiş ağ kullanılmıştır. Bunlar ise ResNet50, MobilNetV2, MobileNetV3Small, MobileNetV3Large'tır. Eğitim ve veri seti ayırmada iki farklı grup oluşturulmuştur: Biri %80 eğitim, %20 test, diğeri ise %90 eğitim, %10 testtir. Bu şartlar altında önce veri setleri %80 eğitim, %20 test olacak şekilde ayrılıp sırayla seçilen önceden eğitilmiş ağlar ile her veri seti için deneyler yapılmıştır. Aynı şekilde veri seti değiştirilerek %90 eğitim, %10 test olarak ayarlanmıştır. Yine önceden eğitilmiş ağları sırayla seçerek her veri seti için deneyler yapılmıştır. Böylece toplamda 24 adet deney yapılmıştır. 24 adet deneyin sonucunda Hockey Fight veri seti ile MobileNetV2 transfer öğrenme modeli kullanılarak 99,37 değeri ile en iyi sonuç elde edilmiştir. Veri setlerine göre elde edilen sonuçlar Tablo 13, 14, 15'te verilmiştir.

Tablo 13: Real-Life Violence Situations Veri Seti İçin Deneylerde Elde Edilen Sonuçlar

Real-Life Violence Situations						
ResNet50						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	68,67	70	69	73	71
90	10	49,86	52,5	1	06	11
MobileNetV2						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	99,06	92	89	96	92
90	10	98,68	94	96	92	94
MobileNetV3Small						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	69,51	53	52	96	67
90	10	69,24	50,50	51	99	67
MobileNetV3Large						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	62,66	59,50	57	75	65
90	10	85,69	70	64	95	76

Tablo 14: Movies Veri Seti İçin Deneylerde Elde Edilen Sonuçlar

Movies						
ResNet50						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	70,01	53	48	64	55
90	10	80,31	52	52	1	68
MobileNetV2						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	50,64	45	45	95	61
90	10	96,02	84	90	75	82
MobileNetV3Small						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	60,26	51,02	48	1	65
90	10	46,59	56	52	1	69
MobileNetV3Large						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	51,92	45	45	1	62
90	10	49	52	50	33	40

Tablo 15: Hockey Fight Veri Seti İçin Deneylerde Elde Edilen Sonuçlar

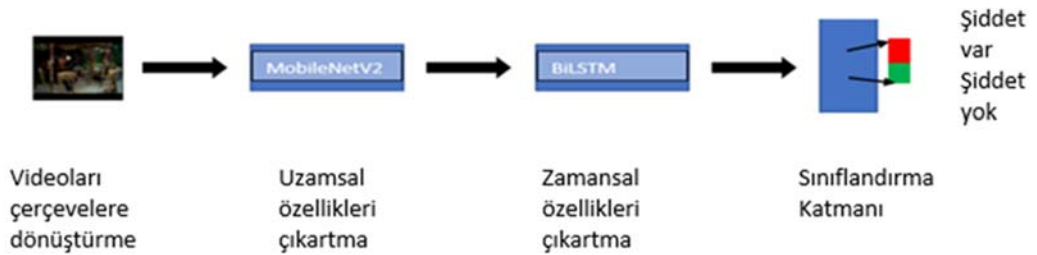
Hockey Fight						
ResNet50						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	91	90	90	91	90
90	10	88	86	91	81	86
MobileNetV2						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	99,37	94	92	96	94
90	10	98,61	93	94	92	93
MobileNetV3Small						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	81,72	69	48	1	65
90	10	85,97	69	1	40	58
MobileNetV3Large						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	91,87	88	90	87	88
90	10	92,5	95	94	96	95

3.3 BULGULAR

Her geçen gün boyutu ve çeşitliliği artan veri, bir taraftan çok değerli hâle gelirken diğer taraftan onu mevcut yöntemlerle analiz etmek ve yönetmek imkânsız hâle gelmektedir. Bu veri çeşitlerinden bir tanesi de video verileridir. Toplumsal alanların hemen her yerinde kurulan video kayıt sistemlerinin biriktirdiği verileri yönetmek ve analiz etmek ihtiyacı bulunmaktadır. Bu verileri otomatik bir şekilde yapmak gerekmektedir. Bunun için bilgisayar sistemlerindeki donanımın gelişmesi neticesinde derin öğrenme yöntemiyle bir çözüm aranmıştır. Videolarda bulunan şiddet tespiti için derin öğrenme tabanlı şiddet detektörü oluşturularak bu alanda çözüm üretilmiştir.

Şiddet tespiti, bilgisayarlı görünüm altında incelenmektedir ve videolardaki şiddet tespitini bu yöntemle belirlemek oldukça zordur. Dolayısıyla bu soruna derin öğrenme yöntemleriyle çözüm aranmıştır.

Bu çalışma, ikili sınıflandırma problemlerini gidermek için çözüm üretmeyi amaçlamıştır. Veri seti üzerinde özellik çıkarımı ve seçimi aşamalarında kullanılan yöntemler ise videolar çerçevelere dönüştürüldükten sonra önce uzamsal olarak özellikler çıkartılmış, daha sonra sıralı karelerde zamansal olarak özellikler çıkartılmıştır. Bunun için transfer öğrenme ve LSTM ağları birleştirilerek hibrit bir yapı oluşturulmuştur. Sınıflandırıcı olarak LSTM'nin bir çeşidi olan BILSTM sınıflandırıcısı kullanılmıştır. BILSTM'yi kullanmaktaki en büyük avantaj, değişik uzunluktaki girdilerin sınıflandırılabilmesine imkân vermesidir (Keçeli ve Kaya 2018). Derin öz nitelik çıkartmak için MobileNetV2 kullanılmıştır. Bu derin öz nitelikler ile BILSTM modeli eğitilmiştir. Derin öz nitelik çıkartmada transfer öğrenme modeli kullanmanın sınıflandırma yapma üzerinde olumlu etkisi olmuştur. Şiddet detektörü blok diyagramı aşağıda gösterilmiştir.



Şekil 34: Şiddet Detektörü Blok Diyagramı

Deney sonuçları incelendiğinde Real-Life Violence Situations, Movies ve Hockey Fight veri setleri ile beraber dört farklı önceden eğitilmiş ağ kullanıldığı görülmektedir. Bunlar ResNet50, MobilNetV2, MobileNetV3Small, MobileNetV3Large'tır. Eğitim ve veri seti ayırmada iki farklı grup oluşturulmuştur: Biri %80 eğitim, %20 test, diğeri ise %90 eğitim, %10 testtir. Deneylerde farklı parametrelerin ve algoritmaların şiddet algılama başarısını nasıl etkilediği incelenmiştir. Dropout değeri 0.32 olarak girilmiştir. Bu problem bir sınıflandırma problemi olduğundan one-hot encoding kullanılmıştır. Bunun dışında kayıp fonksiyonu categorical crossentropy seçilmiştir. Gizli katmanlarda Relu aktivasyon fonksiyonu, sınıflandırma katmanında ise Softmax aktivasyon fonksiyonu kullanılmıştır. Son olarak da Batch size 8, epoch 50 olarak ayarlanmıştır. Bu şartlar altında her veri seti için deneyler ayrı ayrı yapılmıştır.

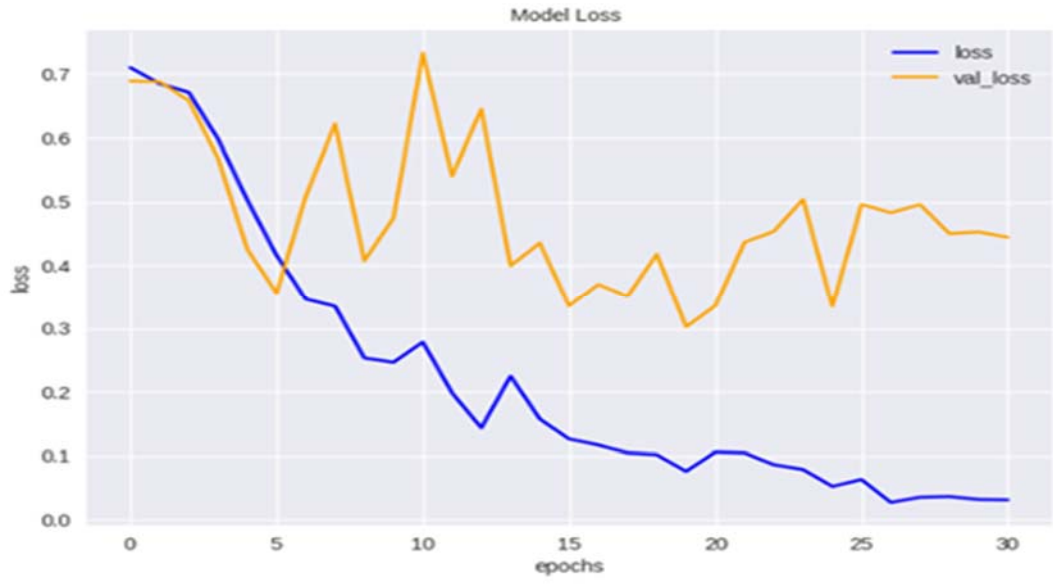
Bu durumda en iyi sonucu, Hockey Fight veri seti kullanılarak eğitim verisini %80, test verisini %20 seçmekle birlikte MobileNetV2 vermiştir.

Tablo 16: Önerilen Modelin Deney Sonucu

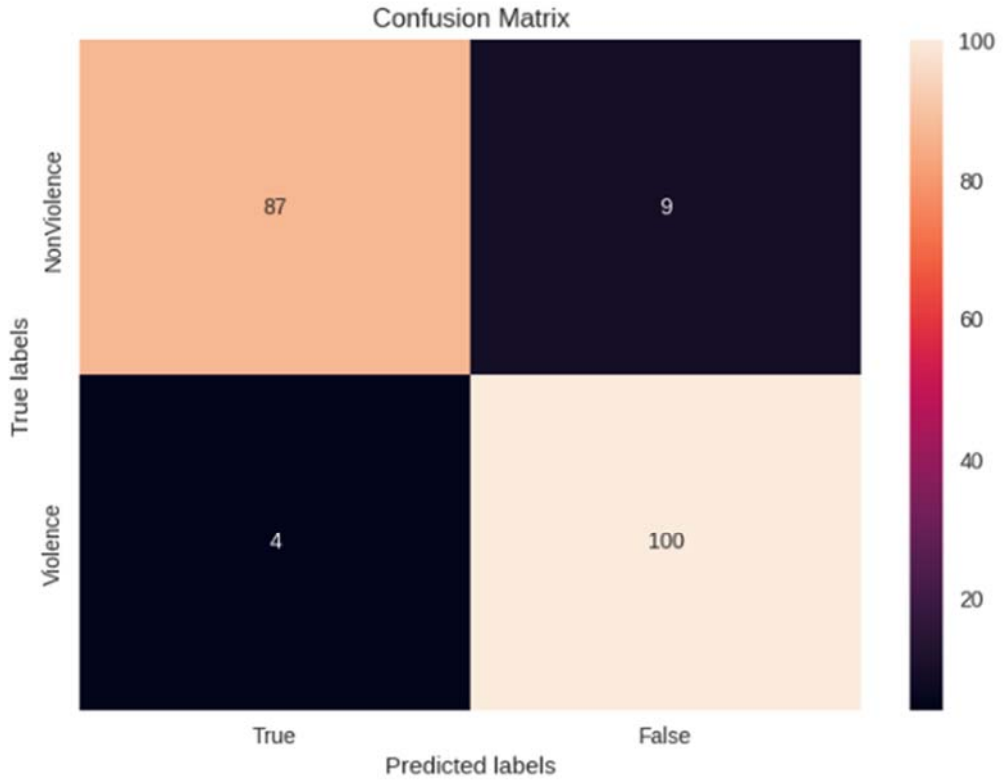
MobileNetV2						
Train Oranı	Test Oranı	Accuracy Train	Accuracy Test	Precision	Recall	F1
80	20	99,37	94	92	96	94



Şekil 35: Önerilen Modelin Accuracy Grafiği



Şekil 36: Önerilen Modelin Loss Grafiği



Şekil 37: Önerilen Modelin Confusion Matrix Grafiği

Classification Report is :				
	precision	recall	f1-score	support
0	0.96	0.91	0.93	96
1	0.92	0.96	0.94	104
accuracy			0.94	200
macro avg	0.94	0.93	0.93	200
weighted avg	0.94	0.94	0.93	200

Şekil 38: Önerilen Modelin Sonuçları

Deney, sonuçları oluşturulan modelin eğitim süreci tamamlandıktan sonra elde ettiği başarı ve performans değerleri Kesinlik (Precision), Duyarlılık (Recall), F1 skor (f1-score) ölçütlerine göre değerlendirilmiştir.

Bu tezin sonuçları literatürde diğer çalışmalarla karşılaştırılıp değerlendirilmiştir.

Dinesh Jackson vd. (2019) yaptıkları çalışmada, devasa veri tabanını işleyen gerçek zamanlı bir şiddet tespit sistemi önermektedirler. Her bir fotoğraf karesinin nitelikleri HOG (Histogram of Oriented Gradients) fonksiyonu kullanılarak elde edilir. Daha sonra fotoğraf kareleri; şiddet modeli, insan parçası modeli ve negatif model gibi niteliklere göre işaretlenir, bu kareler şiddet görüntülerinin tanınması için Çift Yönlü Uzun Kısa Süreli Bellek (BILSTM) ağında eğitilir. Şiddet içermeyen sahneler ve şiddet sahneleri içeren 409 video klip, futbol stadyumundan elde edilmiştir. Bu modelin performansı doğrulanmış ve şiddet eylemini tanıma oranı yüzde 94,5 olarak hesaplanmıştır.

Dursun ve Türkmen'in (2021) yaptıkları çalışmada, süper piksel tabanlı bir yöntem önerilerek kanlı bölge tespit edilmeye çalışılmıştır. Öncelikle, kan içeren bölgeleri tespit edebilmek için kan görüntüsü olup ve olmayan şeklinde bir veri kümesi oluşturulmuştur. Sistemin başarısını test etmek için bölütleme yönteminin üreteceği süper piksel miktarı, üç farklı üst sınır ile test edilmiştir. Bölütleme yönteminden meydana gelen parçalardan renk ve doku nitelikleri çıkarılmış, destek vektör makinesi (DVM) sayesinde kanlı bölgeleri belirleyebilecek modeller geliştirilmiştir. Farklı çekirdek fonksiyonları oluşturularak ve modellerin başarıları karşılaştırılarak test edilmiştir. Önerilen bu sistemde yaklaşık %97 doğruluk elde edilmiştir.

Videolardaki şiddet içeren görüntüleri tespit etmek için yapılan görüntü analizleri çalışmalarından biri de Ullah ve arkadaşları tarafından yapılmıştır. Belirtilen çalışmada, Evrişimsel Sinir Ağları kullanılarak şiddet içeren ya da içermeyen olarak ikili sınıflama yapılmıştır. Yani önce 2B CNN'den karelerdeki kişileri belirlemek amacıyla yararlanılmış ve ardından şiddet tespiti için sadece insanları bulunduran kareler 3B CNN'yle beslenerek bir model tasarlanmıştır. Önerilen modelde veri kümesi olarak Crowd Violence kullanılmış ve test edilmiştir. Doğruluk değeri %98 olarak elde edilmiştir.

Önerilen modelde Hockey Fight veri seti kullanılarak MobileNetV2 ve BI-LSTM hibrit yapı oluşturulmuş ve başarı oranı %99,37 olarak bulunmuştur.

Tablo 17: Önerilen Model ile Diğer Çalışmaların Başarı Değerlerinin Karşılaştırılması

	Dinesh Jackson ve Diğerleri	Dursun ve Türkmen	Ullah ve Arkadaşları	Önerilen Model
Başarı Değeri %	94,5	97	98	99,37

3.4 SONUÇ

Yapılan çalışmada üç farklı veri seti kullanılmıştır. Bunlar Real-Life Violence Situations, Movies ve Hockey Fight veri setleridir. Bununla beraber dört farklı önceden eğitilmiş ağ kullanılmıştır. Bunlar ise ResNet50, MobilNetV2, MobileNetV3Small, MobileNetV3Large'tır. Eğitim ve veri seti ayırmada iki farklı grup oluşturulmuştur: Biri %80 eğitim, %20 test, diğeri ise %90 eğitim, %10 testtir. Bu şartlar altında önce veri setleri %80 eğitim, %20 test olacak şekilde ayrılıp sırayla seçilen önceden eğitilmiş ağlar ile her veri seti için deneyler yapılmıştır. Aynı şekilde veri seti değiştirilerek %90 eğitim, %10 test şeklinde ayarlanmıştır. Yine önceden eğitilmiş ağları sırayla seçerek her veri seti için deneyler yapılmıştır. Böylece toplamda 24 adet deney yapılmıştır.

MobileNetV3Small, MobileNetV3Large önceden eğitilmiş modeller bu model için her veri setinde en düşük sonuçları vermiştir.

ResNet50, MobilNetV2 önceden eğitilmiş modeller, her veri seti için yüksek değerleri vermiştir.

Veri setlerinde ise movies veri seti hem adet sayısı düşük olmasından hem de arka plandan kaynaklı diğer veri setlerine nazaran düşük değerler üretmiştir. Bu tez

için, şiddet eylemlerini göstermesine rağmen movies veri setinin uygun olmayacağı düşünülmüştür.

Real-Life Violence Situations, Movies ve Hockey Fight veri setleri birbirine yakın değerler verse de Hockey Fight veri seti biraz daha önde değerler üretmiştir.

Belli zamanlarda birbirini tekrarlayan deneylerde birbirine yakın ya da aynı sonuçlar çıktığı gözlemlenmiştir. Bu da modelin kararlı olduğunu belirtmektedir.

Deney 2’de MobileV2 transfer öğrenme modeli seçilmiş, veri setleri %90 eğitim, %10 test şeklinde ayrılmış, tüm veri setlerinde accuracy train değeri birbirine çok yakın çıkmış ve bu değerlerin 98.68, 96.02, 98.61 olduğu gözlenmiştir. Bu da modelin doğru ve kararlı olduğu sonucuna vardırmaktadır.

Önerilen MobileNetV2- BILSTM varyantı, kullanılan veri kümesi için bildirilen en iyi sonuçları sağlamaktadır. Çalışmaya başlamadan önce hedeflenen derin öğrenme tabanlı şiddet detektörü, doğruluk oranı yüksek olması ve hızlı çalışmasının yanında düşük maliyetli bir sistem olmuştur.

Literatürde yer alan diğer çalışmalar ile bu çalışma karşılaştırıldığında önerilen videolarda şiddet içeren görüntülerin tespitinde 99,37’lik değer ve başarılı olarak en iyi sonuçlardan biri elde edilmiştir.

KAYNAKÇA

- ABDALİ Al-Maamoon R. ve AL-TUMA Rana F. (2019), “Robust Real-Time Violence Detection in Video Using CNN and LSTM”, *SCCS 2019-2019 2nd Scientific Conference of Computer Sciences*, ss. 104-108, Irak.
- AKDAĞ Ali (2017), *Derin Öğrenme Algoritmaları Kullanılarak Gerçek Zamanlı Silah Tanıma Uygulaması* (Yüksek Lisans Tezi), Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Konya.
- AKSOYLU M. Ümit (2021), *Projelerle Yapay Zekâ ve Bilgisayarlı Görü*, Birinci Basım, KODLAB Yayın, İstanbul.
- AKTAŞ Abdulsamet (2020), *Derin Öğrenme Yöntemleri ile Görüntü İşleme Uygulamaları* (Yüksek Lisans Tezi), Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- ALKAYAL Osama (2023), *Derin Öğrenme Yaklaşımlarını Kullanarak Videolarda ve Canlı Yayınlarda Gerçek Zamanlı Şiddet Tespiti* (Yüksek Lisans Tezi), Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Konya.
- ARCEDA Vicente Machaca, FABIÁN Karla Fernández ve GUTÍERREZ Juan Carlos (2016), “Real time violence detection in video”, *International Conference on Pattern Recognition Systems (ICPRS-16)*, ss. 1-7, Talca, DOI:10.1049/ic.2016.0030.
- ATALAY Muhammet ve ÇELİK Enes (2017), “Büyük Veri Analizinde Yapay Zekâ ve Makine Öğrenmesi Uygulamaları”, *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, Cilt 9, Sayı 22, ss. 155-172, DOI:10.20875/makusobed.309727.
- BİLGİN Metin (2018), *Makine Öğrenmesi*, Ed. Atınç Yılmaz, İkinci Basım, Papatya Yayıncılık, İstanbul.
- CANER Murat ve AKARSLAN Emre (2009), “Mermer Kesme İşleminde Spesifik Enerji Faktörünün ANFIS ve YSA Yöntemleri ile Tahmini”, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, Cilt 15 Sayı 2, ss. 221-226.

- CHOLLET François (2021), *Deep Learning with Python*, İkinci Basım, Manning Publications, Shelter Island.
- CLARİN Christine T., DİONİSİO Judith Ann M., ECHAVEZ Michael T. ve NAVAL Prospero C. (2005), “DOVE: Detection of Movie Violence Using Motion Intensity Analysis on Skin And Blood”, *Pcsc*, ss. 150-156.
- ÇALIŞKAN Duygu ve DEMİR Önder (2022), “Derin Öğrenme Yöntemleri ile Şüpheli Davranış Tespiti”. *International Periodical of Recent Technologies in Applied Engineering*, Cilt 3, Sayı 1, ss. 26-41, DOI:10.29228/porta.8.
- ÇETİNKAYA Cem (2022), *Gözetim Videolarında Anomali Tespit Yöntemlerinin Karşılaştırmalı Bir Değerlendirmesi* (Yüksek Lisans Tezi), Hacettepe Üniversitesi, Ankara.
- DAHOUDA Mwamba Kasongo ve JOE Inwhae (2021), “A Deep-Learned Embedding Technique for Categorical Features Encoding”, *IEEE Access*, Cilt 9, ss. 114381-114391, DOI:10.1109/ACCESS.2021.3104357.
- DATTA Ankur, SHAH Mubarak ve LOBO Niels Da Vitoria (2002), “Person-on-Person Violence Detection in Video Data”, *Proceedings-International Conference on Pattern Recognition*, Cilt 1, ss. 433-438. DOI:10.1109/ICPR.2002.1044748.
- DİNESH Jackson Samuel R., FENİL E., GUNASEKARAN Manogaran, VİVEKANANDA G. N., THANJAİVADİVEL T., JEEVA S. ve AHİLAN A. (2019), “Real Time Violence Detection Framework for Football Stadium Comprising of Big Data Analysis and Deep Learning Through Bidirectional LSTM”, *Computer Networks*, Cilt 151, ss. 191-200, DOI:10.1016/j.comnet.2019.01.028.
- DOĞAN Ferdi ve TÜRKOĞLU İbrahim (2019), “Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme”, *DÜMF Mühendislik Dergisi*, Cilt 10, Sayı 2, ss. 409-445, DOI:10.24012/dumf.411130.
- DOĞAN Ferdi ve TÜRKOĞLU İbrahim (2018), “Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması”, *Sakarya University Journal Of Computer and Information Sciences*, Cilt 1, Sayı 10, ss. 213-219.
- DUMAN Elvan ve ERDEM Osman Ayhan (2019), “Anomaly Detection in Videos Using Optical Flow and Convolutional Autoencoder”, *IEEE Access*, Cilt 7, ss. 183914-183923, DOI: 10.1109/ACCESS.2019.2960654.

- DURSUN Ömer Faruk ve TÜRKMEN H. İrem (2021), “Süper Piksel Tabanlı Otomatik Kanlı Bölge Tespit Sistemi”, *International Journal of Advances in Engineering and Pure Sciences*, Cilt 33, Sayı 1, ss. 28-38, DOI:10.7240/jeps.686886.
- ELMAS Çetin (2021), *Yapay Zeka Uygulamaları*, Beşinci Basım, Seçkin Yayıncılık, Ankara.
- GÉRON Aurélien (2023), *Hands-On Machine Learning with Scikit- Learn, Keras and TensorFlow*, O’Reilly Media, Sebastopol.
- GIANNAKOPOULOS Theodoros, MAKRİS Alexandros, KOSMOPOULOS Dimitrios, PERANTONİS Stavros ve THEODORİDİS Sergios (2010), “Audio-Visual Fusion for Detecting Violent Scenes in Videos”, *Hellenic Conference on Artificial Intelligence*, ss. 91-100, Athens, DOI:10.1007/978-3-642-12842-4_13.
- GONZALEZ Rafael C. ve WOODS Richard E. (2008), *Digital Image Processing*, Üçüncü Basım, Pearson, Upper Saddle River.
- HAYKİN Simon (2018), *Neural Networks and Learning Machines*, Üçüncü Basım, Pearson, Hamilton.
- HE Kaiming, ZHANG Xiangyu, REN Shaoqing ve SUN Jian (2016), “Deep Residual Learning for Image Recognition”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ss. 770-778, Las Vegas, DOI:10.1109/CVPR.2016.90.
- HOCHREİTER Sepp ve SCHMİDHUBER Jürgen (1997), “Long Short-Term Memory”, *Neural Computation*, Cilt 9, Sayı 8, ss. 1735-1780, DOI:10.1162/neco.1997.9.8.1735.
- HOWARD Andrew G., ZHU Menglong, CHEN Bo, KALENİCHENKO Dmitry, WANG Weijun, WEYAND Tobias, ANDREETTO Marco ve ADAM Hartwig (2017), “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, *October*, <http://arxiv.org/abs/1704.04861>.
- KAYTAN Mustafa, AYDİLEK İbrahim Berkan, YEROĞLU Celaleddin ve KARCI Ali (2022), “Sigmoid-Gumbel: Yeni Bir Hibrit Aktivasyon Fonksiyonu”, *Bitlis Eren Üniversitesi, Fen Bilimleri Dergisi*, Cilt 11, Sayı 1, ss. 29-45, DOI:10.17798/bitlisfen.990508.

- KEÇELİ Ali Seydi ve KAYA Aydın (2018), “Optik Akış Görüntüsü ve BILSTM ile Şiddet İçeren Hareketlerin Sınıflandırılması”, *European Journal of Science and Technology*, Sayı 14, ss. 204-208, DOI:10.31590/ejosat.460257.
- KEÇELİ Ali Seydi ve KAYA Aydın (2019), “Video Görüntülerinde Şiddet İçeren Aktivitelerin Lstm Ağı ile Tespiti”, *Dokuz Eylül Üniversitesi, Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, Cilt 21, Sayı 63, ss. 933-939, DOI:10.21205/deufmd.2019216321.
- KIZRAK Merve Ayyüce ve BOLAT Bülent (2018), “Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma”, *Bilişim Teknolojileri Dergisi*, Cilt 11, Sayı 3, ss. 263-286, DOI:10.17671/gazibtd.419205.
- KRİZHEVSKY Alex, SUTSKEVER Ilya ve HİNTON Geoffrey E. (2012), “ImageNet Classification with Deep Convolutional Neural Networks”, *Communications of the ACM*, Cilt 60, Sayı 6, ss. 84-90, DOI:10.1201/9781420010749.
- LECUN Yann, BENGİO Yoshua ve HİNTON Geoffrey (2015), “Deep Learning”, *Nature*, Cilt 521, Sayı 7553, ss. 436-444, DOI:10.1038/nature14539.
- LECUN Yann, BOTTOU Leon, BENGİO Yoshua ve HAFFNER, Patrick (1998), “Gradient-based Learning Applied to Document Recognition”, *Proceedings of the IEEE*, Cilt 86, Sayı 11, ss. 2278-2324, DOI: 10.1109/5.726791.
- MİNAEE Shervin, BOYKOV Yuri, PORİKLİ Fatih, PLAZA Antonio, KEHTARNAVAZ Nasser ve TERZOPOULOS Demetri (2022), “Image Segmentation Using Deep Learning: A Survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Cilt 44, Sayı 7, ss. 3523-3542, DOI:10.1109/TPAMI.2021.3059968.
- MOHTAVİPOUR Seyed Mehdi, SAEİDİ Mahmoud ve ARABSORKHİ Abouzar (2022), “A Multi-Stream Cnn For Deep Violence Detection in Video Sequences Using Handcrafted Features”, *Visual Computer*, Cilt 38, Sayı 6, ss. 2057-2072, DOI:10.1007/s00371-021-02266-4.
- NABİYEV Vasif (2016), *Yapay Zeka*, Beşinci Basım, Seçkin Yayıncılık, Ankara.
- NİEVAS Enrique Bermejo, SUAREZ Oscar Deniz, GARCÍA Gloria Bueno ve SUKTHANKAR Rahul (2011), “Violence Detection In Video Using Computer Vision Techniques”, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Cilt 6855, ss. 332–339, DOI:10. 1007/978-3-642-23678-5_39.

- NİLSSON Nils J. (1996), *Introduction to Machina Learning*, Stanford University, California.
- OLAH Christopher (2015), *Understanding LSTM Networks*, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, ET. 15.12.2023.
- ÖZTEMEL, Ercan (2020), *Yapay Sinir Ağları*, Beşinci Basım, Papatya Yayıncılık, İstanbul.
- PATEL Mann B. (2021), “Real-time violence detection using CNN and LSTM”, *arXiv preprint*, DOI:10.48550/arXiv.2107.07578.
- PEKER, Musa (2009), *Görüntü İşleme Tekniği Kullanılarak Gerçek Zamanlı Hareketli Görüntü Tanıma* (Yüksek Lisans Tezi), Sakarya Üniversitesi, Sakarya.
- PITTS Walter ve MCCULLOCH Warren S. (1943), “A Logical Calculus of The Ideas Immanent in Nervous Activity Warren”, *Advances in Intelligent Systems and Computing*, Cilt 5, ss. 115-133. DOI:10.1007/978-3-030-01370-7_61.
- SHARMA Simone, SHARMA Siddharth ve ANİDHYA Athaiya (2020), “Understanding Activation Functions in Neural Networks”, *International Journal of Engineering Applied Sciences and Technology*, Cilt 4, Sayı 12, ss. 310-316.
- SİT Muhammed, DEMİRAY Bekir Z., XIANG Zhongrun, EWING Gregory J., SERMET Yusuf ve DEMİR İbrahim (2020), “A Comprehensive Review of Deep Learning Applications in Hydrology and Water Resources”, *Water Science and Technology*, Cilt 82, Sayı 12, ss. 2635-2670, DOI:10.2166/wst.2020.369.
- SOLİMAN Mohamed Mostafa, KAMAL Mohamed Hussein, NASHED Mina Abd El-Masih, MOSTAFA Youssef Mohamed, CHAWKY Bassel Safwat ve KHATTAB Dina (2019), “Violence Recognition from Videos Using Deep Learning Techniques”, *İçinde, Proceedings-2019 IEEE 9th International Conference on Intelligent Computing and Information Systems, ICICIS*, ss. 80-85, IEEE, Cairo, DOI:10.1109/ICICIS46948.2019.9014714.
- SZEGEDY Christian, LIU Wei, JIA Yangqing, SERMANET Pierre, REED, Scott, ANGUELOV Dragomir, ERHAN Dumitru, VANHOUCKE Vincent ve RABİNOVİCH Andrew (2015), “Going deeper with convolutions”, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ss. 1-9, Boston, DOI:10.1109/CVPR.2015.7298594.

- ŞEKER Abdulkadir, DİRİ Banu ve BALIK Hasan Hüseyin (2017), “Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme”, *Gazi Mühendislik Bilimleri Dergisi*, Cilt 3, Sayı 3, ss. 47-64.
- TAN Ziya (2019), *Derin Öğrenme Yardımıyla Araç Sınıflandırma* (Yüksek Lisans Tezi), Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elâzığ.
- TOKMAK Mahmut ve KIRAÇ Akın (2021), “Evrişimsel Sinir Ağları ile Örümcek Kuşugillerin Bazı Türlerinin Sınıflandırılması”, *Bilge International Journal of Science and Technology Research*, Cilt 5, Sayı 1, ss. 72-79, DOI:10.30516/bilgesci.886291.
- ULLAH Fath U. Min, ULLAH Amin, MUHAMMAD Khan, HAQ Ijaz Ul ve BAİK Sung Wook (2019), “Violence Detection Using Spatiotemporal Features with 3d Convolutional Neural Network”, *Sensors*, Cilt 19, Sayı 11, ss. 1-15, DOI:10.3390/s19112472.
- ÜNLÜ Elif Işıl (2019), *Derin Öğrenme ile Görüntü Bölütleme* (Yüksek Lisans Tezi), Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elâzığ.
- XİAO Perry (2022), *Artificial Intelligence Programming with Python*, Wiley, Hoboken.
- YAĞIN Burak (2022), *Yapay Zekâ Tabanlı Görüntü İşleme Yöntemleri ile Covid-19 Tahmini* (Yüksek Lisans Tezi), İnönü Üniversitesi, Sağlık Bilimleri Enstitüsü, Malatya.
- YILMAZ Atınç ve KAYA Umut (2019), *Derin Öğrenme*, Eds. Gizem Aksan, İrem Soylu, KODLAB Yayın, İstanbul.