T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DERİN ÖĞRENME YAKLAŞIMLARINI KULLANARAK VİDEOLARDA VE CANLI YAYINLARDA GERÇEK ZAMANLI ŞİDDET TESPİTİ

Osama ALKAYAL

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ocak-2023
KONYA

TEZ KABUL VE ONAYI

Osama ALKAYAL tarafından hazırlanan “Derin Öğrenme Yaklaşımlarını Kullanarak Videolarda ve Canlı Yayınlarda Gerçek Zamanlı Şiddet Tespiti” adlı tez çalışması 13/01/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Doç. Dr. Rahime CEYLAN	……..
Danışman Dr. Öğr. Üyesi Levent CİVCİK	……..
Üye Doç. Dr. Humar KAHRAMANLI ÖRNEK	……..

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Osama ALKAYAL

Tarih:13.01.2023

ÖZET

YÜKSEK LİSANS

DERİN ÖĞRENME YAKLAŞIMLARINI KULLANARAK VİDEOLARDA VE CANLI YAYINLARDA GERÇEK ZAMANLI ŞİDDET TESPİTİ

Osama ALKAYAL

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Levent CİVCİK

2023, 51

Jüri
Dr. Öğr. Üyesi Levent CİVCİK
Doç. Dr. Rahime CEYLAN
Doç. Dr. Humar KAHRAMANLI ÖRNEK

Video ve canlı yayınlarda bilgisayar görme ve yapay zekâ yaklaşımları kullanılarak şiddet tespiti dünya çapında birçok araştırmacının ilgisini çeken güncel bir alandır. Videolardaki şiddeti tespit etmek zordur, çünkü videolar yalnızca uzamsal veya zamansal özellikleri olan diğer veri türleriyle karşılaştırıldığında analiz edilmesi zor olan uzamsal-zamansal özelliklere sahiptir. Derin öğrenme, böylesine zorlu bir göreve birçok çözüm sunan derin yapıları ve hiyerarşik öğrenme yaklaşımlarını kullanan bir yapay zekâ alt sınıfıdır. Girdi ve çıktı katmanları arasında birçok katmana sahip olması ve ağ içeren yapıları nedeniyle derin öğrenme yaklaşımları, örüntü sınıflandırma ve uzamsal-zamansal özellikleri çıkarmada oldukça başarılıdır. Bu çalışma, videolarda şiddet tespiti gerçekleştirmek için derin öğrenme yaklaşımlarını kullanarak bir model oluşturmayı amaçlamaktadır. Bu kapsamda çalışmada, videolarda şiddet tespiti için ConvLSTM ile MobileNets'ten oluşan hibrit bir model kullanılmıştır. Yapılan çalışma, bu kombinasyonun şiddet tespiti alanında kullanılmasında öncüdür. MobileNets, ardışık video karelerinden uzamsal özellikleri çıkarmak için kullanılmıştır. Öte yandan ConvLSTM ile, yerel uzamsal özellikleri korunup, kareler arasındaki ilişkiler zamansal olarak analiz edilmiştir. Geliştirilen sinir ağını eğitmek ve test etmek için yaklaşık 2000 video klip kullanılmıştır. Gerçekleştirilen model ile %96'lık bir test doğruluğuna ulaşılmıştır. Edilen sonuçlar, bu konuda yapılan birçok çalışmanın üzerinde bir başarı oranını yakalanmıştır.

Anahtar Kelimeler: ConvLSTM, Derin Öğrenme, MobileNets, Şiddet, Uzamsal-Zamansal Özellikler, Yapay Zekâ

ABSTRACT

MS THESIS

REAL-TIME VİOLENCE DETECTION IN VIDEOS USING DEEP LEARNING APPROACHES

Osama ALKAYAL

Konya Technical University
Institute of Graduate Studies
Department of Electrical-Electronic Engineering

Advisor: Asst. Prof. Dr. Levent CİVCİK

2023, 51

Jury
Asst. Prof. Dr. Levent CİVCİK
Assoc. Prof. Dr. Rahime CEYLAN
Assoc. Prof. Dr. Humar KAHRAMNLI ÖRNEK

Violence detection in videos and live broadcasts using computer vision and artificial intelligence approaches has become a hot research area attracting the attention of many researchers worldwide. Detecting violence in videos is a challenge because videos have spatiotemporal features that are difficult to analyze when compared to other types of data that only have spatial or temporal features. Deep learning is a subclass of artificial intelligence that uses deep structures and hierarchical learning approaches that offer many solutions to such a challenging task. Due to their structure, which contains many layers or networks between the input and output layers, deep learning approaches successfully classify patterns and extract spatiotemporal features. This study aims to build a neural network using deep learning approaches to perform violence detection in videos. In this context, a hybrid model consisting of ConvLSTM and MobileNets was used for violence detection in videos. This work is the pioneer in using this combination in the field of violence detection. MobileNets has been used to extract spatial features from successive video frames. On the other hand, ConvLSTM has been used to analyze the relations between these frames in time manners while maintaining the local spatial features. Around 2000 video clips were used to train and test the developed neural network. A test accuracy of 96% has been achieved using the proposed model. The results have reached a successful test rate higher than many studies on this subject.

Keywords: ConvLSTM, Deep Learning, MobileNets, Violence, Spatiotemporal Features, Artificial Intelligence

ÖNSÖZ

Tez çalışmasını gerçekleştirme aşamasında yardımlarını benden eksik etmeyen, beni yönlendiren danışmanım ve değerli hocam Dr. Öğr. Üyesi Levent CİVCİK'e ve desteklerinden ötürü bana maddi ve manevi her türlü desteği sağlayan aileme teşekkürlerimi sunarım.

Osama ALKAYAL
KONYA-2023

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Kısaltmalar

BiLSTM	: Çift Yönlü Uzun Kısa Süreli Bellek
CNN	: Konvolüsyonel Sinir Ağı
ConvLSTM	: Konvolüsyonel Uzun Kısa Süreli Bellek
FN	: Yanlış Negatif
FP	: Yanlış Pozitif
LSTM	: Uzun Kısa Süreli Bellek
MPEG	: Hareketli Görüntü Uzmanları Birliği
ReLU	: Doğrultulmuş Lineer Birim
RGB	: Kırmızı Yeşil Mavi
RNN	: Tekrarlayan Sinir Ağı
SNN	: Darbeli Sinir Ağı
TN	: Doğru Negatif
TP	: Doğru Pozitif
2B	: 2 Boyutlu
3B	: 3 Boyutlu

1. GİRİŞ

Şiddet en iyi şekilde "Zarar veren veya zarar vermeyi amaçlayan fiziksel bir güç eylemidir" diyen Kristine M. Jacquin (2017) tarafından tanımlanmıştır. Şiddetin verdiği zarar fiziksel, psikolojik veya her ikisi de olabilir. Aile İçi Şiddete Karşı Ulusal Koalisyon'a göre, her 4 kadından 1'i ve her 9 erkekten 1'i yaşamları boyunca bir tür aile içi veya kişiler arası şiddete maruz kalmaktadır (Anonymous, 2020). Ancak, Adalet İstatistikleri Bürosu tarafından yapılan bir ankette, aile içi şiddet veya kişilerarası şiddet vakalarının sadece %47'si polise bildirilmektedir (Rennison ve Welchans, 2018). Şiddeti polise bildirecek kimsenin olmadığı sokaklar, metro istasyonları, marketler gibi boş yerler ihbarsız şiddetin yaşandığı yerler arasındadır. Şiddet olaylarının derhal ilgili makamlara bildirilmesiyle, hem şiddete maruz kalanlara ivedilikle yardım edilecek, hem de failler tutuklanarak durumun daha iyi ele alınması sağlanacaktır.

Videolarda şiddet tespiti, dünya çapında birçok araştırmacının ilgisini çeken sıcak bir araştırma alanı haline gelen, iyi bilinen insan eylemi tanıma görevinin özel bir senaryosudur. İnsan eylemlerini tanıma, bir kişi veya bir grup insan tarafından gerçekleştirilen eylemleri belirlemeyi amaçlarken, şiddet algılama, gerçekleştirilen eylemlerin şiddet içerikli olup olmadığını belirlemeyi amaçlar.

İnsan eylemleri genellikle, onları iyi anlamak için toplu olarak analiz edilmesi gereken bir dizi olaydır. Başka bir deyişle, bir insan eylemini, eylemi en iyi temsil eden tüm görüntü dizisini (video) düşünmek yerine tek bir görüntüyü (kare) dikkate alarak analiz etmek, hatalı sonuçlara yol açabilir. Teknik açıdan bakıldığında, görüntüler yalnızca uzamsal özellikler içerirken, videolar hem zaman alanı hem de uzay alanı analizini gerektiren uzamsal-zamansal özellikler içerir ve bu da eylem tanımayı zorlu bir görev haline getirir. Derin öğrenme, böylesine zorlu bir göreve birçok çözüm sunan yapay zekanın bir alt sınıfıdır.

1950'lerden bu yana, genellikle makine öğrenme olarak adlandırılan yapay zekanın bir alt kümesi, birçok uygulama alanında devrim niteliğinde bir rol oynamaktadır. Yapay sinir ağları, derin öğrenmeyi doğuran, beyinden ilham alan makine öğrenmenin bir alt alanıdır. Denetimli derin öğrenme, sınıflandırma işlemleri için derin sinir ağlarını ve etiketlenmiş verileri kullanan yapay zekâ teknolojilerinden biridir (Najafabadi ve ark., 2015). "Derin" terimi, girdi ve çıktı katmanları arasındaki işlem katmanlarının sayısını ifade eder. Derin sinir ağları, özellik öğrenme ve örüntü sınıflandırması için kullanılan doğrusal olmayan hesaplama birimlerinin birçok aşamasından oluşan hiyerarşik

mimarilere sahiptir. Daha derin gizli katmanlara sahip olmak, son zamanlarda özellikle örüntü tanıma ve sınıflandırma görevlerinde farklı alanlarda klasik yöntemlerin performansını aşmaya başlamıştır (Dixit ve ark., 2018). Derin sinir ağlarının genel odak noktası, girdi verilerinden örüntü öğrenme ve öğrenilen örüntülerin gelecekte görünmeyen verilerde kullanılmak üzere genelleştirilmesidir. Diğer bir deyişle öğrenme, eğitilen modelin belirli bir görevi yerine getirebilmesi için modelin parametrelerinin optimize edilmesinden oluşan bir süreçtir.

Bu çalışmada, videolarda şiddet tespiti görevinde kullanılmak üzere derin öğrenme yaklaşımlarını kullanan bir model oluşturulması amaçlanmaktadır. Güvenlik sistemlerindeki son gelişmelerle birlikte; sokaklar, tren istasyonları, trenler, otobüsler, parklar, stadyumlar, yurtlar, üniversiteler, alışveriş merkezleri, marketler, asansörler, merdivenler gibi birçok yerde güvenlik kameraları bulunmakta ve yaşanan bir şiddet olayı sonrasında bu kameraların görüntüleri polis tarafından incelenmektedir. Gerçek zamanlı otomatik şiddet tespiti için bu kameraların görüntülerine derin öğrenme teknikleri ve algoritmalarının gerçek zamanlı olarak uygulanması, bildirilmeyen şiddet olaylarının miktarını azaltmaya ve dolayısıyla halkın güvenliğini korumaya yardımcı olacaktır. Bu kapsamda, geliştirilecek sistemin performansını ve hızını en yüksek seviyeye çıkarmak adına sistemin çeşitli parametreleri deneysel olarak geliştirilecektir.

Yapılan tezin bölümleri şu şekildedir; ilk olarak, bölüm 2'de önceki çalışmalardan bahsedilmektedir. Bundan sonra, bölüm 3'te problem tanıtılmaktadır. Bölüm 4'te ise yapay zekâ ve derin öğrenmeye genel bir bakış sunulmaktadır. Daha sonrası, bölüm 5'te önerilen modelin yapısı ve parametreleri gösterilmektedir. Ayrıca, bölüm 6'da önerilen modelin deneysel olarak geliştirilmesi için yapılan deneysel çalışmalar tartışılmaktadır. Ondan sonra bölüm 7'de, bu çalışmanın bulguları ve elde edilen sonuçlarının önceki çalışmalarla karşılaştırılması öne sürülmektedir. Son olarak, bölüm 8'de de tezin genel sonuçları ve önerilerinden bahsedilmektedir.

2. KAYNAK ARAŞTIRMASI

Derin öğrenme kullanarak videolarda şiddeti tespit etmenin temel bir yöntemi, zamansal özelliklere önem vermeden yalnızca videonun uzamsal özelliklerini analiz eden 2B (2 Boyutlu) CNN (Konvolüsyonel Sinir Ağı) kullanmaktır. Bu yaklaşımı kullanan Mumtaz ve ark. (2018), Hokey veri setindeki her bir video klipten elde edilen görüntüler (kareler) ile 2B CNN'i eğitmiş ve %99.28 doğruluğu elde ettiğini iddia etmişlerdir.

Mohtavipour ve ark. (2021), şiddet tespit sürecine zamansal özellikleri dahil etmek için, videoların ham kareleriyle birlikte videolardan çıkarılan el yapımı özellikleri kullanmışlardır. Araştırmacılar, üç paralel 2B CNN'den oluşan derin bir sinir ağı kurdular. İlk CNN uzamsal akış (ham kareler) kullanılarak eğitilmiş, ikinci CNN zamansal akış kullanılarak eğitilmiş ve üçüncü CNN uzamsal-zamansal akış kullanılarak eğitilmiştir. Uzamsal akış, videonun genel ortamını analiz eder. Zamansal akış, birden fazla ardışık kareden oluşturulan optik akış görüntüsünü kullanarak hareketli hedefler için hareket hızını dikkate alır. Uzamsal-zamansal akışta, şiddet davranışının anlaşılmasına yardımcı olan eylemlerin şekli, birbirini takip eden birden çok kareden diferansiyel hareket enerji görüntüsü oluşturularak elde edilir. Yukarıda bahsedilen 3 CNN'nin çıktısı daha sonra şiddet davranışlarını tespit etmek için bir sınıflandırıcıya beslenir. Araştırmacılar, Hokey ve Violent Flow veri setinde sırasıyla %100 ve %99.35 doğruluk elde ettiklerini iddia etmişlerdir.

Benzer şekilde, el yapımı özellikler Baba ve ark. (2019) tarafından da kullanılmış ancak video karelerinin görsel görünümünü tamamen ihmal etmişlerdir. Araştırmacılar, şiddet davranışının doğrudan hareketle ilişkili olduğu varsayımına bağlı olarak MPEG (Hareketli Görüntü Uzmanları Birliği) videosundan elde edilen hareket vektörleri ile bir 2B CNN sınıflandırıcıyı eğitti. Bir video klipteki her 2 kare, şiddet veya şiddetsizlik olarak sınıflandırılacak bir hareket vektörü oluşturur. CNN sınıflandırıcısından sonra, CNN sınıflandırıcısından elde edilen ikili diziyi analiz etmek için iki zaman alanlı filtre uygulanmıştır. BEHAVE veri setinde bu yöntemle elde edilen doğruluk %86 olmuştur.

Peixoto ve ark. (2018), insan denetimi olmayan videolardan şiddeti tespit etmenin sadece teknik değil, aynı zamanda kavramsal bir sorun olduğunu belirtmişlerdir. Buna göre, 7 CNN'den oluşan derin bir sinir ağı oluşturmuşlar ve her biri kan, yumruk, kavga ve ateşli silah gibi farklı bir şiddet kavramını anlamak için kullanmışlardır. Bu CNN'lerin çıktıları daha sonra video hakkında bir karar üretmek için füzyon ağına beslenmiştir. Hem uzamsal hem de zamansal özellikleri dahil etmek için araştırmacılar CNN'leri kare akışı

ve videolardaki hareketleri tanımlayan videolardan manuel olarak çıkarılan diğer üç akış olmak üzere dört farklı girdiyle beslemişlerdir. Araştırmacılar, MediaEval 2013 veri setinde %78'lik bir doğruluk oranı elde etmişlerdir.

Videolarda insanların davranışlarını sınıflandırmada zamansal özelliklerin önemi nedeniyle, çoğu araştırmacı farklı CNN çeşitlerini kullanmışlardır. 2B CNN'lerin kendileri yalnızca uzamsal özellikleri analiz etmek için kullanılırken, Abdali ve Al-Tuma (2019), 2B CNN'yi LSTM (Uzun Kısa Süreli Bellek) adı verilen bir ağla birleştirdi. LSTM genellikle önceden eğitilmiş özelliklerin bir kısmını (bu durumda ardışık kareler) yeniden değerlendirmek için kullanmış, bu da tüm ağın hem uzamsal hem de zamansal özellikleri işlemesine izin vermiştir. Araştırmacılara göre, bu yöntem, bellek kaynaklarının miktarı sınırlı olduğunda çok faydalıdır. Bu modelin Hokey veri kümesindeki test sonucu, 131 kare/sn. hızında %98'dir.

Sumon ve ark. (2020), transfer öğrenme stratejilerini uygulamış ve ResNet50 adlı önceden eğitilmiş bir CNN modelinden ve bir LSTM ağından oluşan bir model oluşturmuştur. ResNet50, binlerce veri ile önceden eğitilmiş 50 katmandan oluşan ve uzamsal özellik çıkarıcı olarak kullanılan 2B CNN'dir. 30 ardışık kareden gelen uzamsal özellikler daha sonra zamansal analiz için LSTM ağına beslenir. Araştırmacılar, 220 gerçek hayat videosundan oluşturdukları veri seti kullanarak bu modeli yeniden eğitti ve %97'lik bir test doğruluğu elde etti.

Benzer bir şekilde, Halder ve Chatterjee (2020), 2B CNN'den gelen belirli bir karenin özellik haritasını hem sonraki hem de önceki karelerin özellik haritalarıyla işleyen BiLSTM (Çift Yönlü Uzun Kısa Süreli Bellek) kullanmıştır. Bu yöntem, şiddet davranışlarına ilişkin bilgilerin çoğunu içeren videonun zamansal özelliklerini daha derin bir şekilde analiz eder. Araştırmacılar tarafından Hockey Fights, Movies ve Violent Flow veri setleri üzerinde elde edilen test sonucu sırasıyla %99.27, %100 ve %98.64'tür.

Peixoto ve ark. (2020) tarafından yürütülen araştırmada, araştırmacılar videolardaki şiddeti tespit etmek için hem işitsel hem de görsel özelliklerin birleştirilmesini incelediler. Araştırmacılar işitsel özellikleri analiz etmek için bir sinir ağı tasarladılar ve bunu bir füzyon sinir ağı kullanarak görsel özellikleri (hem uzamsal hem de zamansal) analiz eden CNN-LSTM ile birleştirdiler. Araştırmacılar, MediaEval 2013 veri setinde %78'lik genel sınıflandırma doğruluğu elde ettiklerini iddia ettiler.

Yukarıda bahsedilen CNN-LSTM modelleri, ağın LSTM kısmında tam bağlantılı yapıya sahiptir. Sudhakaran ve Lanz'a (2017) göre bu, ağın videoların yerel özelliklerini değil, küresel uzamsal-zamansal özelliklerini çıkarmasını sağlar. Bu da şiddet

davranışlarının büyük bir kısmı yerel özelliklerde bulunduğundan dolayı modelin performansını düşürmüştür. Bunun üstesinden gelmek için, LSTM ağı içinde konvolüsyonel katmanları kullanan ConvLSTM (Konvolüsyonel Uzun Kısa Süreli Bellek) adlı bir model kullanmışlardır. ConvLSTM, yerelleştirilmiş uzamsal-zamansal özellikleri yakalayabilen ve böylece videoda yer alan yerel hareketi analiz etmeyi sağlayan tek bir ağdır. Bunun yanı sıra, videoların girdisi olarak karelerin kendisini kullanmak yerine, iki ardışık kare arasındaki farkı kullandılar. Araştırmacıların Hokey veri setinde elde ettikleri doğruluk oranı %97.1'dir.

Modelin yapısına odaklanmak yerine, Soliman ve ark. (2019) çalışması, modeli eğitmek için kullanılan veri kümelerinin önemini ve ünlü veri kümeleriyle eğitilen modellerin gerçek hayat verilerine uygulandığında nasıl düşük doğrulukları elde ettiklerini gösterir. Genel olarak bu veri kümeleri, benzer şiddet davranışları ve benzer ortamları içeren videolardan oluşur. Araştırmacılar, bu veri setleri üzerinde bir CNN-LSTM modeli eğitti. Daha sonra bu modeli, kendi oluşturdukları Real-Life Violence Situations aldı olan gerçek hayat videoları içeren veri seti üzerinde test ettiler ve elde ettikleri en iyi doğruluk oranı %71.5'tir. Daha sağlam bir model oluşturmak için, 2000 videodan oluşturdukları gerçek hayat veri seti ile bir CNN-LSTM ağını transfer öğrenme teknikleri kullanarak yeniden eğittiler ve %88.2 test doğruluğunu elde ettiler.

Traore ve Akhloufi (2020), girdi olarak iki akışı olan bir CNN-LSTM modeli oluşturmuşlardır. Bu durumda araştırmacılar, videoların ihtiva ettiği RGB (Kırmızı, Yeşil, Mavi) kareleriyle birlikte, karelerdeki hareketli nesneler ve hızları hakkında bilgi içeren optik akış görüntülerini kullanmışlardır. Bu iki akışta kullanılan CNN modeli EfficientNet-B0'dır. Bu iki akışın çıktıları, ekleme katmanı kullanılarak birbirine eklenir ve daha sonra zamansal analiz için LSTM ağına beslenir. Araştırmacılar Hokey, Violent Flow ve Real-Life Violence Situations veri kümelerinde sırasıyla %99, %93.75 ve %96.74 doğruluk elde etmiştir.

Real-Life Violence Situations veri seti Lima ve Figueiredo (2021) tarafından da kullanılmıştır. Araştırmacılar, bir LSTM ağını 10 akışlı MobileNets ile birleştirmiş ve %91 doğruluk elde etmiştir.

Accattoli ve ark. (2020) tarafından önerilen yöntem, videolardaki şiddeti tespit etmek için bir uzamsal-zamansal özellik çıkarıcı olarak 3B (3 Boyutlu) CNN ve ikili bir sınıflandırıcı olarak destek vektör makinesinin kullanılmasını önermektedir. 3B filtreleri kullanan 3B CNN, birden fazla kareyi tek bir girdi olarak işleyerek hem uzamsal hem de zamansal özellikleri analiz edebilir. Bundan dolayı araştırmacılar, ağın şiddet

davranışlarını algılaması için 16 ardışık karenin yeterli olduğunu öne sürdüler. Sinir ağı, Hockey Fight ve Crowd Violence veri kümeleri kullanılarak eğitildi, test edildi ve sırasıyla %98.51 ve %99.29 doğruluk elde edildi.

3B CNN'lerin ana avantajı, herhangi bir ek ağ veya videoların ön işlenmesini gerektirmemeleridir. Bununla birlikte, Li ve arkadaşların (2019) göre iki büyük dezavantajı vardır. İlk olarak, çok sayıda parametre hesaplama açısından verimsizdir. İkinci olarak, gereksiz parametreler, özellikle küçük veri kümelerinde, modelin fazla uydurmasına ve genelleme yeteneğinin azalmasına neden olabilir. Bu iki dezavantajın üstesinden gelmek için araştırmacılar, özelliklerin yeniden kullanımını ve kanal etkileşimini destekleyen yoğun şekilde birbirine bağlı bir mimari geliştirdiler ve bu mimarinin uzamsal-zamansal özellikleri yakalayabildiğini ve on kat daha az parametre gerektirdiğini gösterdiler. Araştırmacılar tarafından Mix veri setinde elde edilen doğruluk %99.32'dir.

Gerçek zamanlı şiddet tespitinde çok sayıda işe yaramaz karenin işlenmesini azaltmak ve dolayısıyla kaynakların kullanımını azaltmak için Ullah ve ark. (2019), 2B CNN'den ve ardından 3B CNN'den oluşan bir model tasarlanmışlardır. 2B CNN, karelerdeki kişileri algılamak için kullanılır ve şiddet tespiti için yalnızca insanları içeren kareler 3B CNN'ye beslenir. Önerilen model, Hockey Fight ve Crowd Violence veri kümeleri kullanılarak eğitilmiş ve test edilmiş ve sırasıyla %96 ve %98 doğruluk oranlarına ulaşılmıştır.

Gkountakos ve ark. (2020), transfer öğrenmeyi uygulanmış ve mekansal-zamansal özellikleri öğrenme görevi için Hara ve ark. (2017) tarafından geliştirilen 3B-ResNet adlı önceden eğitilmiş bir 3B CNN modelini kullanmışlardır. Bu ağ sığ bir yapıya sahiptir, milyonlarca video ile eğitilmiştir ve bu da fazla uydurma probleminin önüne geçmektedir. Fazla uydurma, geliştirilen modelin yeni verilere genelleyemediğinde gerçekleşir, bu da gerçek hayattaki videolara uygulanmasının güvenilir olmadığı anlamına gelir. Araştırmacılar, transfer öğrenmeyi uygular, Violent Flow veri setini kullanarak 3B-ResNet parametrelerini ince ayarlar ve %99.31'lik doğruluğu elde etmişlerdir.

Tang ve ark. (2021), 3B CNN'lerin görünüm ve kısa süreli davranışlar gibi yerel uzamsal-zamansal özellikleri öğrendiğini, ConvLSTM'lerin ise uzun süreli davranışlar gibi küresel özellikleri öğrendiğini belirtmektedir. Bu iki ağın avantajlarını birleştirmek için ConvLSTM ile paralel olarak 3B CNN'den oluşan bir model oluştururlar. Bu iki ağın çıktıları daha sonra hem küresel hem de yerel özellikleri birleştiren ve her ağın katkısına

karar veren bir ağa beslenir. Araştırmacılar bu yöntemi insan davranışlarını tanımak için uygulamış ve %99.63 doğruluk elde etmiştir.

İnsan eylemi tanıma ve videolarda örüntü sınıflandırması gibi bazı benzer araştırma alanlarında, bazı araştırmacılar, insan beyninden ilham alan SNN'ler (Darbeli Sinir Ağı) ile birleştirilmiş CNN'ler gibi derin öğrenme mimarilerini kullanmaya başladılar. SNN, olay tabanlı sinir ağlarıdır; bu, zaman alanında belirli bir olay paterni gerçekleştiğinde ateşleyen darbeli nöron modeli nedeniyle zamansal özellikleri öğrenme yeteneğine sahip oldukları anlamına gelir. Örneğin, Samadzadeh ve ark. (2021), sıradan CNN'deki tüm nöronları darbeli nöronlar ile değiştirerek konvolüsyonel bir darbeli sinir ağı tasarladı, modellerinin performansının 3B CNN, CNN-LSTM ve diğer modellerin performansının üstesinden geldiğini gösterdi. Araştırmacılar, SNN'nın diğer yaklaşımlara kıyasla düşük hesaplama maliyeti avantajına sahip olduğunu ekledi.

Jeba ve Meal (2020), iki 2B CNN ve ardından zaman alanı sınıflandırıcısı olarak bir SNN'den oluşan insan eylemleri tanıma için bir model önerdi. İlk CNN, eylem dizilerinin uzamsal görünümünü yakalar ve ikinci CNN, hareket vektörlerini çıkarmak için kullanılır. Bu iki CNN, videonun tüm karelerinin uzamsal özelliklerini çıkarır ve bir dizi uzamsal özellik oluşturur. SNN sınıflandırıcısı, bu dizileri karşılıklı olarak analiz etmek ve bu dizilerdeki davranış türlerini sınıflandırmak için alır. Araştırmacılar, yöntemlerinin en son tekniklerden daha iyi performans gösterdiğini ve birkaç durumda karşılaştırılabilir olduğunu belirtmişlerdir.

Xing ve ark. (2020), zamansal özellikleri daha derinden analiz etmek için bir 3B darbeli CNN önerdi. Bu çalışma, aynı anda birden fazla kare grubunu dikkate alan, yukarıda bahsedilen 3B CNN'ye çok benzer. Ancak, 3B darbeli CNN'ler, mevcut kare grubunu analiz ederken önceki kare grubundan gelen bilgileri kullanmalarına izin veren bir tür belleğe sahiptir. Bu hafıza, olayların oluşum sırasına bağlı olarak artan veya azalan zar potansiyeli tarafından en iyi şekilde modellenir. Önerilen yöntem, DVS el hareketi veri kümesinde doğrulanmıştır ve %96.59'luk bir 10 sınıf el hareketi tanıma doğruluğu elde etmiştir.

Yalçın (2018), tezinde insan etkinliğini tanımaya yönelik farklı derin öğrenme yaklaşımlarını incelemiştir. Bu çalışmada araştırmacı, iskelet tabanlı yaklaşımlar ve RGB tabanlı yaklaşımlar olmak üzere iki yaklaşım kategorisi kullanmıştır. İskelet tabanlı yaklaşımlar kategorisinde, videolardaki insan vücudu, bir RGB-D sensörü kullanılarak insan iskeletini taklit eden bitişik çizgilerle temsil edilmiştir. Araştırmacı, 3B CNN, CNN-LSTM, CNN-BiLSTM ve iki akış CNN dahil olmak üzere çeşitli derin öğrenme

yaklaşımlarını incelemişlerdir. Bu kategoride en iyi tanıma sonucu 3B CNN yaklaşımı kullanılarak elde edilmiştir. RGB tabanlı yaklaşımlar kategorisinde araştırmacı, optik akış görüntüleri ile beslenen 3B CNN, CNN-LSTM ve 3B CNN'yi incelemiştir. Araştırmacı, RGB tabanlı yaklaşımlar kategorisi için en iyi sonucun optik akış görüntüleri ile beslenen 3B CNN kullanılarak elde edildiğini belirtmiştir.

Tanberk (2020), satranç oyunundaki tüm insan hareketlerini içeren bir veri seti kullanarak bir satranç oyuncusunun hareketlerini taklit eden bir robot geliştirdi. Araştırmacı, video kliplerle beslenen bir 3B CNN, optik akış görüntüleri tarafından beslenen bir 3B CNN ve bir LSTM ağını birleştiren bir derin öğrenme ağı tasarladı. Bu üç ağın çıktısı, bir veri birleştirme algoritması kullanılarak birleştirilir ve ardından SVM sınıflandırıcısına beslenir. Tüm hareketler için genel sınıflandırma doğruluğu %95.2 idi.

Abdi (2021), insanların koşma veya yürüme sırasındaki davranışlarından acil durumları tespit eden bir derin öğrenme yaklaşımı geliştirmiştir. Geliştirilen sistem, bir seferde bir görüntü ile ilgilenen bir 2B CNN modeli kullanmaktadır. Araştırmacı, birçok video akışını kullanarak 2000 görüntüden oluşan bir veri seti oluşturdu. Bu görüntüler önerilen CNN modelini eğitmek için kullanılmış ve %86 doğruluk elde edilmiştir.

3. PROBLEM TANITIMI

Bu çalışmada, videolarda şiddet tespitini bir sınıflandırma problemi olarak ele alınmıştır. Sınıflandırma problemlerinin soyut bir şekilde anlaşılması, bu çalışmanın ne hakkında olduğuna dair iyi bir genel bakışa sahip olmak için önemlidir. Genel olarak sınıflandırma, bir modelin belirli bir girdi verisinin, girdi sınıflarının hangisine ait olduğunu tahmin etmesi gerektiği bir görevdir. Şiddet tespiti durumunda, modelin giriş videolarının şiddet içerikli olup olmadığını tahmin etmesi gerekmektedir. Dolayısıyla, modelin şiddet içeren video veya şiddet içermeyen video olmak üzere yalnızca iki olası çıktısı vardır.

İnsan davranışları, bu davranışları gerçekleştirmek için yapılan eylemler dizisi toplu olarak analiz edildiğinde genellikle daha iyi anlaşılır. Bunu anlamak için, şekil 3.1'deki eylem dizisini ele alalım. Bu zaman adımlarının her biri, kendi başına ele alındığında davranışa ilişkin farklı bir gösterge verebilir. Ancak, bu zaman adımları birbirini takip etmese de bunları toplu olarak ele almak, davranışın şiddet içerikli olduğunu açıkça gösterir.

Şekil 3.1. Şiddet içerikli bir davranışın zaman adımları

Her bir zaman adımında (karede) bulunan özelliklere uzamsal özellikler denir. Uzamsal özellikler, arka plan ortamı, kenarlar ve her karede bulunan nesneler hakkında bilgi içerir. Bir videonun her saniyesi 30 kare içerir. Öte yandan, karelerin arasındaki ilişkiler ve karelerin akışının anlamı zamansal özelliklerde bulunur.

Derin öğrenme yaklaşımlarını kullanarak şiddet tespiti görevi hem uzamsal hem de zamansal özellikleri çıkarabilen bir yaklaşım gerektirir. Manuel özellik çıkarma veya verilerin çok fazla ön işlenmesini gerektiren yaklaşımlar yavaştır ve daha fazla hesaplama gerektirir, bu da modeli gerçek zamanlı uygulamalar için uygunsuz hale getirir.

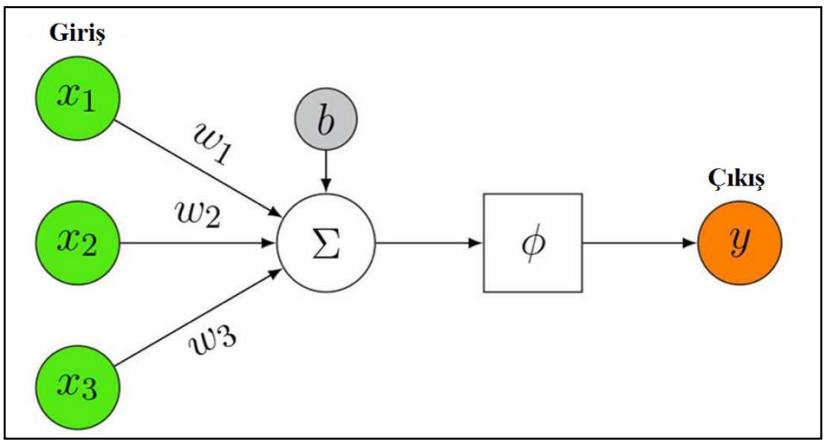
4. YAPAY SİNİR AĞLARI VE DERİN ÖĞRENME

Yapay sinir ağları, biyolojik sinir ağlarından ilham alan bir makine öğrenme sınıfıdır. Yapay sinir ağlarının arkasındaki fikir, basitleştirilmiş matematiksel modeller kullanarak dendritler, hücre gövdeleri ve aksonlar gibi nöronların belirli kısımlarını taklit etmenin mümkün olmasıdır (Nagyfi, 2018). Her bir yapay sinir ağı, bağlantılar (aksonlar ve dendritlere benzer) aracılığıyla diğer düğümlerle iletişim kuran düğümler (hücre gövdelerine benzer) içerir. Biyolojik sinir ağları, bir çıktıyı ateşlemek için nöronlar arasındaki güçlendirici sinapslara sahiptir; benzer şekilde, yapay sinir ağındaki düğümler arasındaki bağlantılar, istenen sonucu sağlama yeteneklerine göre ağırlıklandırılır. (Choi ve ark., 2020).

4.1. Yapay Nöron Modeli

Yapay sinir ağının temel yapı taşı yapay nöron modelidir. Şekil 4.1'de yapay bir nöronun modeli gösterilmektedir. Bir yapay nöron modelinin temel fikri, bir *x* girdisinin *w* ile ağırlıklandırılması ve ardından bir aktivasyon fonksiyonunun argümanını oluşturmak için bir bias *b* ile toplanmasıdır. Hem *x* hem de *w*, *n* büyüklüğünde vektörlerdir, burada *n* girdinin boyutunu temsil eder (Svozil ve ark., 1997). Aktivasyon fonksiyonunun çıktısı *y*, nöronun çıktı sinyalidir ve aşağıdaki gibi gösterilebilir:

$$y = \phi(z) = \phi\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{4.1}$$



Şekil 4.1. Yapay nöron modeli (Emmert-Streib ve ark., 2020)

4.2. Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları, nöronun çıkışını belirlemek için kullanılır. Aktivasyon fonksiyonları, fonksiyona bağlı olarak 0 ile 1 veya -1 ile 1 arasında bir çıktı verir. Aktivasyon fonksiyonlarının lineer ve lineer olmayan olmak üzere iki ana kategorisi vardır. Tablo 4.1'de en sık kullanılan aktivasyon fonksiyonlarını özetlenmektedir (Sharma, 2017).

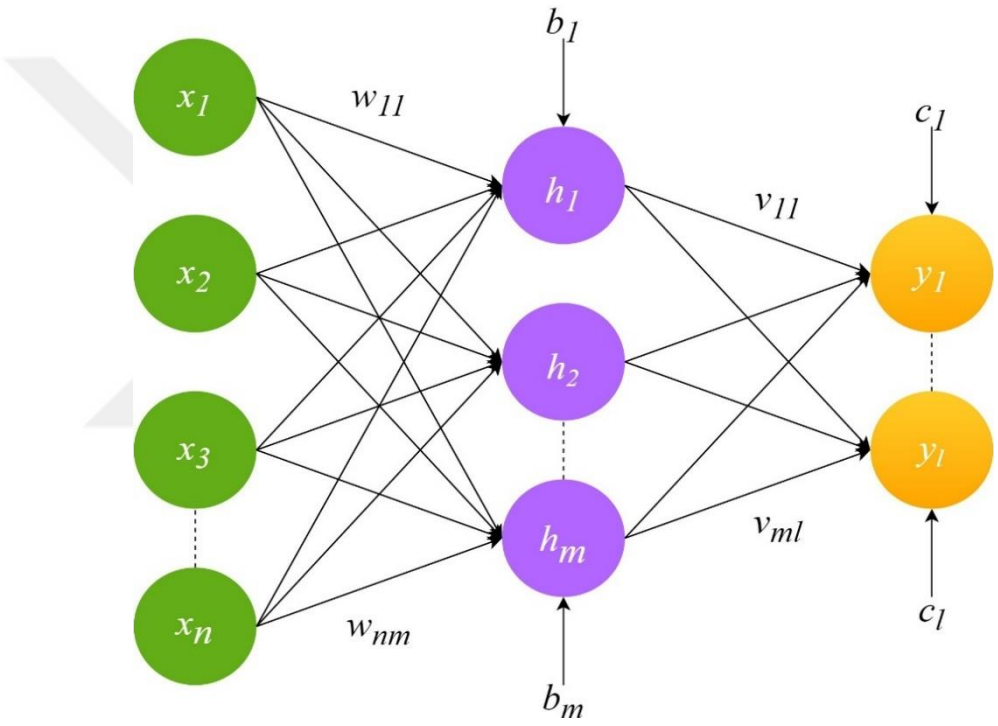
Tablo 4.1. En sik kullanılan aktivasyon fonksiyonları

Aktivasyon Fonksiyonu	$\phi(z) = z$		Çıkış aralığı
Doğrusal Aktivasyon fonksiyonu	$\phi(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	(4.2)	$(-\infty, \infty)$
Hiperbolik tanjant	$\phi(z) = s(z) = \frac{1}{1 + e^{z}}$	(4.3)	$(-1,1)$
Sigmoid	$\phi(z) = sgn(z) = \begin{cases} -1 & eğer\ z < 0\ ise \\ 0 & eğer\ z = 0\ ise \\ 1 & eğer\ z > 0\ ise \end{cases}$	(4.4)	$(0,1)$
Signum	$\phi(z) = R(z) = \begin{cases} 0 & eğer\ z < 0\ ise \\ 1 & eğer\ z \geq 0\ ise \end{cases}$	(4.5)	$[-1,1]$
Basamak fonksiyonu	$\phi(z) = H(z) = \begin{cases} 0 & eğer\ z \leq 0\ ise \\ z & eğer\ z > 0\ ise \end{cases}$	(4.6)	$[0,1]$
ReLu (Doğrultulmuş Lineer Birim) fonksiyonu	$\phi(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$	(4.7)	$[0, \infty]$
Softmax	$\phi(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$	(4.8)	$(0,1)$

Softmax aktivasyon fonksiyonu, bir sonraki bölümde ele alınacak olan çok katmanlı yapay sinir ağlarında esas olarak son katmanda kullanılır. Sınıflandırma görevleriyle ilgili sinir ağlarında kullanılır. Yani, sinir ağının çıktıları, belirli bir girdi grubunun veri kümesinin hangi sınıfına ait olduğunu söyleyen sınıflar şeklindedir. Softmax aktivasyon fonksiyonu, *n* boyutlu bir *z* vektörünü, $\sum_i \phi(z_i) = 1$ özelliğine sahip olan *n* boyutlu bir $\phi(z_i)$ vektörüne dönüştürmektedir. Softmax katmanın çıktılarının toplamı 1 olacaktır. Bu çıktıların her biri, her bir sınıfın olasılığını temsil etmektedir. Ondan sonra olasılığı en büyük olan çıkışın değeri 1 olacak ve diğer sınıfların değeri 0 olacaktır. Bu işlemeye "Sıcak Kodlama" denilmektedir (Emmert-Streib ve ark., 2020).

4.3. İleri Beslemeli Çok Katmanlı Yapay Sinir Ağları

Yapay nöron modelini tanıttığımıza göre, artık çok katmanlı yapay sinir ağlarının yapısını tanıtmaya geçiyoruz. Çok sayıda nörondan oluşan bir sinir ağı oluşturmak için nöronların birbirine bağlı olması gerekir. Standart bir "çok katmanlı" yapay sinir ağı, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katmandan oluşur. Belirli bir katmandaki her nöron, bir sonraki katmandaki tüm nöronlarla birbirine bağlıdır. *i* ve *j* nöronları arasındaki bağlantı w_{ij} ağırlık katsayısı ile karakterize edilir (Svozil ve ark.,1997). Şekil 4.2'de çok katmanlı sinir ağlarının modeli gösterilmektedir.



Şekil 4.2. Çok katmanlı sinir ağlarının modeli

Ve bu durumda gizli katmanın çıktıları aşağıdaki gibidir:

$$h_m = \phi_1\left(\sum_{i=1}^{n} x_i w_{im} + b_m\right) \tag{4.9}$$

Benzer şekilde sinir ağının çıktıları da aşağıdaki gibidir:

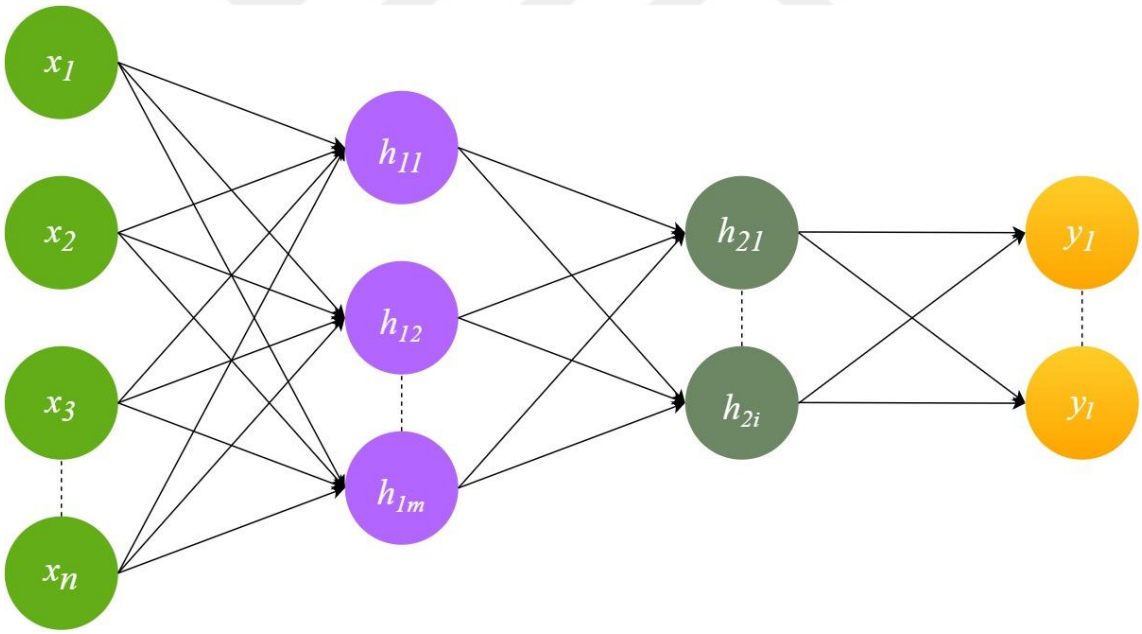
$$y_l = \phi_2\left(\sum_{j=1}^{m} h_j v_{jl} + c_l\right) \tag{4.10}$$

$\phi_1$ve ϕ_2 aktivasyon fonksiyonlarında mevcut olan alt notları, aynı sinir ağı içinde farklı aktivasyon fonksiyonlarının kullanılabileceğini gösterir. Örneğin, sınıflandırma

görevlerinde birçok araştırmacı gizli katmanlarda ReLU aktivasyon fonksiyonunu ve çıktı katmanında Softmax aktivasyon fonksiyonunu kullanır. Sinir ağının optimal parametrelerini (ağırlıklar ve biaslar) hesaplamak için bir optimizasyon algoritması ile bir hata fonksiyonu tanımlanmalıdır. Bu konu bölüm 4.7'de tartışılmaktadır.

4.4. Derin Sinir Ağlarının Genel Yapısı

Genel olarak, bir ağın derinliği, giriş ve çıkış katmanları arasındaki doğrusal olmayan dönüşümlerin sayısını, yani doğrusal olmayan aktivasyon fonksiyonlarını içeren katmanların sayısını belirtir. Örneğin, bölüm 4.3'te tartışılan ve şekil 4.2'de gösterilen sığ mimari 2 derinliğe sahipken, aşağıda Şekil 4.3'te gösterilen sinir ağı 4 derinliğe sahiptir (toplam katman sayısı eksi bir (giriş katmanı)). İleri beslemeli sinir ağı mimarisini derin olarak adlandırmak için gereken sayı tartışmalıdır, ancak ikiden fazla gizli katmana sahip mimariler genellikle derin olarak kabul edilir (Emmert-Streib ve ark., 2020).

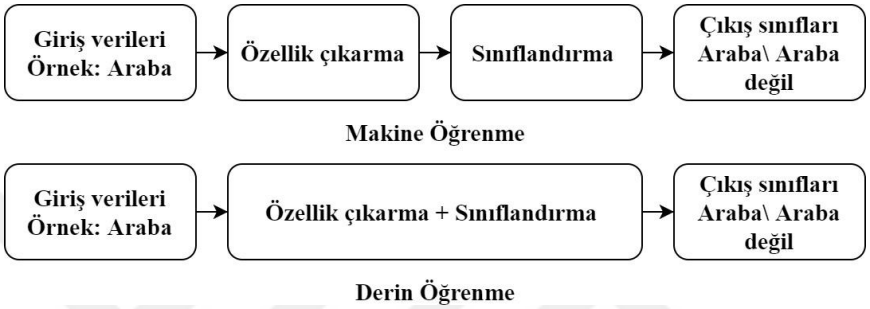


Şekil 4.3. Derin sinir ağların genel yapısı

4.5. Derin Öğrenmede Özellik Çıkarma

Bölüm 4.3'te tartışılan sinir ağları gibi sıradan makine öğrenimi yaklaşımlarında, ham veriler, sınıflandırma veya tespit modeline sunulmadan önce kullanıcı tarafından bir miktar ön işleme ve hazırlamaya maruz kalır. Bu işlemeye özellik çıkarma denir. Örneğin,

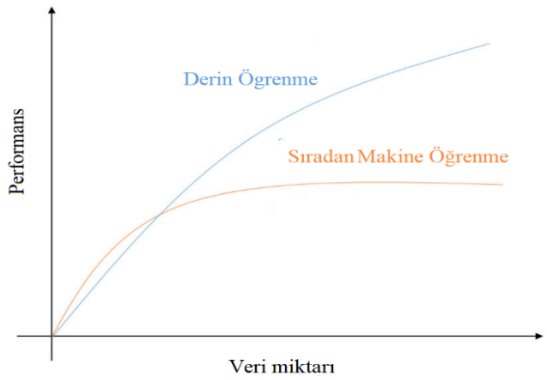
görüntü sınıflandırmada özellik çıkarıma, filtreleme, kenar algılama ve segmentasyon gibi görüntü işleme yaklaşımlarından bazılarını içerebilir (Yadav ve ark., 2013). Öte yandan, derin öğrenmede, doğrusal olmayan işlemenin birden çok aşaması, özellik çıkarma görevini otomatikleştirmeye yardımcı olur. Diğer bir deyişle derin öğrenme, sistemin karmaşık algılama görevlerini maksimum doğrulukla ve daha az ön işleme veya veri hazırlama ile anlamasına yardımcı olan bir yaklaşımdır (Dragan ve ark., 2019). Şekil 4.4'te, derin öğrenme ile sıradan makine öğrenme yaklaşımları arasındaki fark gösterilmektedir.



Şekil 4.4. Sıradan makine öğrenme ile derin öğrenme arasındaki fark

4.6. Veri Miktarının Performans Üzerindeki Etkisi

Öğrenme yaklaşımının performansı, ele alınan problemin girdi verilerinin miktarı ile ilgilidir. Daha az miktarda veri için, sıradan makine öğrenimi yaklaşımları daha iyi bir performans gösterir. Veri miktarı belirli bir sayının üzerine çıktıkça, sıradan makine öğrenme yaklaşımlarının performansı sabit hale gelirken; derin öğrenme yaklaşımlarının performansı, veri miktarının artışına göre artar. Veri miktarı ile öğrenme yaklaşımının performansı arasındaki ilişki şekil 4.5'te gösterilmektedir (Dixit ve ark., 2018).



Şekil 4.5. Veri miktarı ile öğrenme yaklaşımının performansı arasındaki ilişki

4.7. Parametrelerin Öğrenilmesi ve Optimizasyonu

Sinir ağlarında farklı öğrenme yaklaşımlarının temel hedefi, eğitim sırasında sınıflandırma modelinin hatasını en aza indirecek şekilde optimum parametreleri (ağırlıklar ve biaslar) elde etmektir. Bunu gerçekleştirmek için, bir kayıp fonksiyonu (hata fonksiyonu) belirlemek ve tekrarlamalı eğitim süreci sırasında bu kayıp fonksiyonunu optimize etmek için bir optimizasyon algoritması belirlemek gerekir.

4.7.1. Kayıp fonksiyonları

Sinir ağının performansını ölçmek için kayıp veya hata fonksiyonları kullanılır. Temel olarak bir sınıflandırıcının veya tahmin edicinin ne kadar iyi olduğu hakkında bir fikir verirler. Kayıp fonksiyonlarının iki ana kategorisi vardır (Parmar, 2018). Sinir ağının görüntü sınıflandırma gibi sınıflandırma görevleri için kullanılması durumunda, kayıp fonksiyonlarına sınıflandırma kayıpları denir. Buna karşılık, sinir ağı, belirli bir matematiksel fonksiyonun çıktısını tahmin etmek gibi belirli bir değeri tahmin etmek için kullanılıyorsa, o zaman kayıp fonksiyonlarına regresyon kayıpları denir.

4.7.1.1. Sınıflandırma kayıp fonksiyonları

En yaygın olarak kullanılan sınıflandırma kaybı fonksiyonlarından biri çapraz entropidir. Bölüm 4.2'de Softmax'ın her bir çıktı sınıfı için bir olasılık verdiğinden bahsedilmiştir. Bir sinir ağının çıktıları olasılık biçiminde olduğunda çapraz entropi kullanılır (Zhu, 2020). Buna göre çapraz entropi kaybı fonksiyonu aşağıdaki gibi tanımlanır:

$$E = -\frac{1}{N}\sum_{i=1}^{N} y_i^T \log(\hat{y}_i) \tag{4.11}$$

Burada N çıkış sayısı, y_i^T hedeflenen çıkış ve $\hat{y}_i$, bu çıkış için Softmax aktivasyon fonksiyonunun elde ettiği olasılıktır.

4.7.1.2. Regresyon kayıp fonksiyonları

Tablo 4.2'de bazı regresyon kaybı fonksiyonları özetlenmektedir (Parmar, 2018). Bu tabloda y_i belirli bir girdi için hedef çıktı, $\hat{y}_i$ sinir ağı tarafından elde edilen çıktı ve n çıktıların sayısıdır.

Tablo 4.2. Regresyon kayıp fonksiyonları

Kayıp fonksiyonu	Matematiksel modeli	Açıklama
Ortalama kare hatası	$E = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ (4.12)	Kare değeri almak, bazı tahmin edilen değerler (aykırı değerler) karşılık gelen hedef değerlerden çok uzak olduğunda toplam hatanın büyük ölçüde artmasına neden olur.
Ortalama mutlak hatası	$E = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$ (4.13)	Karelerden yararlanmadığı için aykırı değerlere karşı daha dayanıklıdır.
Ortalama önyargı hatası	$E = \frac{1}{n}\sum_{i=1}^{n} y_i - \hat{y}_i$ (4.14)	Daha az doğrudur çünkü negatif hatalar pozitif hataları ortadan kaldırır.

4.7.2. Optimizasyon algoritmaları

Bu bölümde bahsedilen tüm optimizasyon algoritmaları, gradyan inişinin varyantları olarak kabul edilir. Standart gradyan inişi algoritması, kaybın negatif gradyanı yönünde her iterasyonda küçük adımları atarak kayıp fonksiyonunu en aza indirmek için ağ parametrelerini (ağırlıkları ve biasları) aşağıdaki gibi güncellemektedir:

$$\theta_{l+1} = \theta_l - \alpha \nabla \mathrm{E}(\theta_l) \quad (4.15)$$

Burada θ parametre vektörü, l itrasyon sayısı, α öğrenme oranı ve $\Delta \mathrm{E}(\theta_l)$, optimize edilmesi istenen kayıp fonksiyonunun gradyanıdır. Ayrıca, $\Delta \mathrm{E}(\theta_l)$ terimi, kayıp fonksiyonunun θ_l parametresine göre kısmi türevinden başka bir şey değildir ve aşağıdaki gibi gösterilir:

$$\nabla \mathrm{E}(\theta_l) = \frac{\partial E}{\partial \theta_l} \quad (4.16)$$

Bölüm 4.3'teki 4.9 ve 4.10 denklemlerinden, her katmanın çıktısının önceki katmanların tüm parametrelerinin bir fonksiyonu olduğunu hatırlayalım. Buna göre, kayıp fonksiyonu da bu parametrelerin bir fonksiyonudur. Böyle karmaşık bir türevin hesaplanması, geriye yayılım algoritması tarafından gerçekleştirilir (Patrikar, 2019).

4.7.2.1. Toplu gradyan inişi

Bu algoritma, tüm veri kümesi için kayıp fonksiyonunun gradyanını hesaplar ve ardından sinir ağının parametrelerini günceller. Bu, bir güncelleme yapmak için veri setindeki tüm eğitim örnekleri için gradyanların hesaplanması gerektiği ve bu gradyanların ortalamasının parametreleri güncellemek için kullanılması gerektiği anlamına gelir. Bu algoritma, büyük veri kümeleri için büyük miktarda bellek gerektirir (Patrikar, 2019).

4.7.2.2. Stokastik gradyan inişi

Bu algoritmada, sinir ağının parametreleri, toplu gradyan inişi ile karşılaştırıldığında daha sık güncellenir. Sinir ağının parametreleri, mini-batch adı verilen orijinal veri kümesinin bir alt kümesinden sonra güncellenir. Her alt kümeyi geçmek ve parametreleri bir kez güncellemek itrasyon olarak adlandırılır. Eğitim algoritmasının tüm eğitim seti üzerinde mini-batch'lar kullanarak tam geçişi "1" epoch'dur. Stokastik gradyan inişi gürültülüdür, çünkü bir mini-batch kullanılarak hesaplanan parametre güncellemeleri eğitim performansında salınımlara neden olur. (Patrikar, 2019).

4.7.2.3. Momentum ile stokastik gradyan inişi

Stokastik gradyan iniş algoritmasında, en dik iniş yolu boyunca optimum değere doğru salınım olabilir. Parametre güncellemesine bir momentum terimi eklemek, bu salınımı azaltmanın bir yoludur. Momentum ile stokastik gradyan inişi kullanılarak parametre güncellemesi aşağıdaki gibidir:

$$\theta_{l+1} = \theta_l - \alpha \nabla \mathrm{E}(\theta_l) + \gamma(\theta_l - \theta_{l-1}) \quad (4.17)$$

Burada γ, bir önceki itrasyondaki gradyan adımının yeni itrasyona katkısını belirler (Anonymos, 2018).

4.7.2.3. Adam optimizasyon algoritması

Adam optimizasyon algoritması, parametrelere göre farklılık gösteren ve optimize edilmekte olan kayıp fonksiyonuna otomatik olarak adapte edilebilen öğrenme oranlarını kullanarak ağ eğitimini iyileştirmeye çalışmaktadır. Aynı zamanda, en dik iniş yolu

boyunca optimum seviyeye doğru salınımı azaltmanın bir yolu olarak parametre güncellemesine bir momentum terimi ekler.

Her parametre için benzersiz bir öğrenme oranı elde etmek için bu algoritma, öğrenme oranının ölçekleme parametresi aşağıdaki gibi tanımlar:

$$v_l = \beta_2 \, v_{l-1} + (1 - \beta_2)[\nabla E(\theta_l)]^2 \tag{4.18}$$

Momentum terimi de aşağıdaki gibidir:

$$m_l = \beta_1 \, m_{l-1} + (1 - \beta_1)\nabla E(\theta_l) \tag{4.19}$$

Ve en sonunda parametre güncellemeleri aşağıdaki gibidir:

$$\theta_{l+1} = \theta_l - \frac{\alpha m_l}{\sqrt{v_l} + \epsilon} \tag{4.20}$$

Burada $\beta_1 , \beta_2 \in [0,1)$ ve ϵ sıfıra bölmeyi önlemek için eklenen küçük bir sabittir. Birçok iterasyonun gradyanları benzerse, parametre güncellemeleri belirli bir yönde bir momentum kazanmaktadır. Gradyanlar tutarlı değilse, gradyanın hareketli ortalaması küçülmekte ve bu nedenle parametre güncellemeleri de yavaşlamaktadır (Kingma ve Ba, 2015).

4.8. Konvolüsyonel Sinir Ağları

Konvolüsyonel sinir ağları (CNN), esas olarak görüntü sınıflandırma ve hedef tanıma için kullanılan gözetimli derin öğrenme yaklaşımlarından biridir. CNN'ler ileri beslemeli derin yapay sinir ağlarıdır. Bu ağ yapısı Fukushima (1988) tarafından önerilmiştir. Ancak, bu ağ, eğitmek için gerektirdiği büyük hesaplama gücü nedeniyle yaygın olarak kullanılmamıştır. LeCun ve ark. (1998) CNN'lere gradyan tabanlı bir öğrenme algoritması uygulamışlar ve elle yazılmış rakam sınıflandırma problemi için başarılı sonuçlar elde etmişlerdir. CNN'lerin, insan görsel işleme sistemine daha çok benzemesi, 2B ve 3B görüntülerin işlenmesi için yüksek düzeyde optimize edilmesi ve 2B özelliklerin öğrenilmesi ve çıkarılmasında etkili olma gibi diğer derin yapay sinir ağlarına göre çeşitli avantajları vardır (Alom ve ark. 2019).

Bu bölümde, (Albawi ve ark., 2017), (Alom ve ark., 2019) ve (Emmert-Streib ve ark., 2020)'de açıklanan CNN yapısından bahsedilecektir.

4.8.1. Konvolüsyonel sinir ağlarının yapısı

Herhangi bir CNN en azından bir konvolüsyonel katmanı, bir alt örnekleme katmanı ve bir sınıflandırma kısmından oluşmaktadır. Konvolüsyonel katmanı ve alt örnekleme katmanı beraber özellik çıkarma kısmını oluşturmaktadır. Aynı CNN'de 1'den fazla konvolüsyonel katmanı ve alt örnekleme katmanı eklenebilmektedir. Özellik çıkarma kısmında, iki katman arasındaki yalnızca birbirine yakın nöronlar birbirine bağlıdır. Sınıflandırma kısmında ise iki katman arasındaki tüm nöronlar birbirine bağlıdır.

4.8.1.1. Konvolüsyonel katmanı

Konvolüsyonel katmanında giriş katmanından veya önceki katmanlardan gelen veriler "kernel" denilen bir kare matris ile arasında konvolüsyon işlemi yapılmaktadır. Bu katmandaki kernel'ler rastgele değerlerden oluşturulmakta ve bu değerler eğitim esnasında öğrenme algoritmaları kullanılarak güncellenmektedir. Her kernel tek bir özellik çıkarıcı olarak kabul edilmekte ve 1'den fazla kernel kullanılabilmektedir. Bu işlemde üç önemli faktör vardır. Bunlar kernel'lerin boyutu N, her çarpmadan sonra atılan adım S ve giriş verilerin boyutu bozmamak için yapılan sıfır doldurma P. Girdinin boyutu $WxHxZ$ ise çıktının boyutları $W_{out}xH_{out}xZ$ olmakta ve aşağıdaki gibi hesaplanmaktadır

$$H_{out}, W_{out} = \frac{(H, W - N + 2P)}{S + 1} \tag{4.21}$$

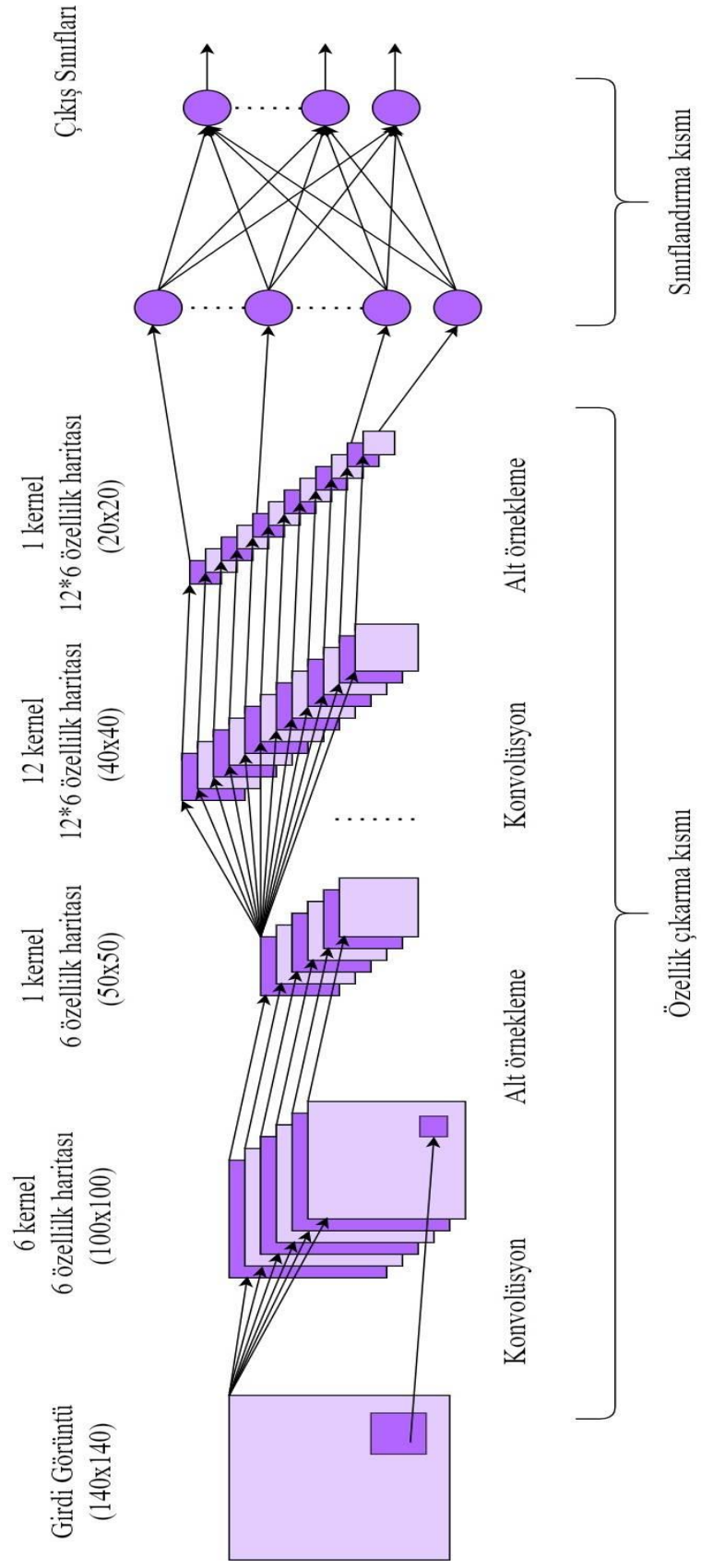
Daha sonra bu veriler bir aktivasyon fonksiyonundan geçirilmektedir. Bu aktivasyon fonksiyonunun çıktılarına özellik haritaları denilmektedir. Araştırmacıların çoğu bu aşamada ReLU aktivasyon fonksiyonunu kullanmaktadır.

4.8.1.2. Alt örnekleme katmanı

Bu katmanda aynı konvolüsyonel katmanındaki gibi giriş verileri "kernel" denilen bir matris ile arasında konvolüsyon işlemi yapılmaktadır. Ancak bu katmandaki kernel'ler 1'lerden oluşacak ve bu kernel'ler eğitim esnasında güncellenmemektedir. Bu katmanın çıkış özellik haritalarının sayısı önceki katmana göre aynı sayıda ama boyutları daha küçüktür. Kernel'in boyutu 2 ise çıkış özellik haritasının boyutu 2'ye bölünür. Konvolüsyondaki çarpma işlemelerinin toplamını almaktansa, ya en büyük sonuç (max-pooling) ya da sonuçların ortalaması (average-pooling) alınmaktadır.

4.8.1.3. Sınıflandırma kısmı

Bu kısımda iki katman arasındaki bütün nöronlar birbirine bağlıdır. Bu kısımda en azından bir gizli katman bulunmakta ve bu durumda bu kısım basit bir ileri beslemeli yapay sinir ağını temsil etmektedir. Bu kısmın girdileri özellik çıkarma kısmında gelen özellik haritalarıdır. Çıktıları ise, bir girdi görüntüsünün veri kümesinin hangi alt sınıfına ait olduğunu gösteren etiketlerdir. Şekil 4.6'da CNN'lerin genel yapısı gösterilmektedir.



Şekil 4.6. CNN'lerin genel yapısı

4.9. Transfer Öğrenme

Transfer öğrenmeyi anlamak için iyi bir örnek; bisiklet sürmeyi zaten bildiği göz önüne alındığında, bir kişinin motosiklet kullanmayı öğrendiğini düşünmektir. Bu durumda kişinin motosikleti sıfırdan öğrenmesine gerek yoktur. Bunun yerine kişi, özellikle motosiklet sürmekle ilgili bazı yeni özelliklerin veya görevlerin öğrenilmesiyle, bisiklet sürme konusundaki bilgisine dayanarak motosiklet kullanmayı öğrenir. Benzer şekilde derin öğrenmede de sıfırdan bir model oluşturup onu başlangıç rastgele parametrelerle eğitmek yerine, çok büyük miktarda veri ile eğitilmiş ve olağanüstü bir performans elde etmiş bazı modellerden faydalanabiliriz (Sarkar, 2018). Önceden eğitilmiş modelleri kullanmanın birçok nedeni vardır. İlk olarak, büyük modelleri büyük veri kümeleri üzerinde eğitmek çok pahalı hesaplama gücü gerektirir. İkincisi, büyük modelleri eğitmek birkaç hafta sürebilir. Yeni modellerin önceden eğitilmiş ağırlıklarla eğitilmesi yakınsamayı hızlandırabileceği gibi ağın genelleştirilmesine de yardımcı olabilir (Alom ve ark., 2019). CNN'lerde milyonlarca parametre içerebilen çok karmaşık yapılara sahip ve binlerce etiket ve binlerce görüntü içeren çok büyük veri setleri ile eğitilmiş birçok model bulunmaktadır. AlexNet, LeNet-5, GoogLeNet, ResNet ve MobileNets önceden eğitilmiş CNN mimarilerinin çok ünlü örnekleridir (Emmert-Streib ve ark., 2020).

Transfer öğrenmenin uygulanması, uygulama alanına ve önceden eğitilmiş modelde kullanılacak veri kümesinin boyutuna bağlıdır. Uygulama alanına ve veri setinin boyutuna göre transfer öğrenme yaklaşımları aşağıdaki gibidir (Alom ve ark., 2019) ve (Jimoh, 2019).

4.9.1. Durum 1: Küçük ve benzer veri kümesi

1. Adım: Özellik çıkarma kısmındaki ağırlıklar dondurulacak ve sınıflandırma kısmı kaldırılacaktır.
2. Adım: Yeni veri setindeki sınıf sayısı kadar çıktı boyutuna sahip yeni bir sınıflandırma kısmı eklenecektir.
3. Adım: Sınıflandırma kısmındaki ağırlıklar rastgele başlatılacaktır.
4. Adım: Ağ, yalnızca sınıflandırma kısmındaki ağırlıkları güncellemek üzere eğitilecektir.

4.9.2. Durum 2: Küçük ama farklı veri kümesi

1. Adım: Özellik çıkarma kısmında alt düzey katmanlardaki ağırlıklar dondurulacak ve sınıflandırma kısmı kaldırılacaktır.
2. Adım: Yeni veri setindeki sınıf sayısı kadar çıktı boyutuna sahip yeni bir sınıflandırma kısmı eklenecektir.
3. Adım: Sınıflandırma kısmındaki ağırlıklar rastgele başlatılacaktır.
4. Adım: Ağ, sınıflandırma kısmındaki ağırlıkları güncellemek ve önceden eğitilmiş modelin özellik çıkarma kısmında sadece üst düzey katmanlardaki ağırlıklara ince ayar yapmak için eğitilecektir.

4.9.3. Durum 3: Büyük ama farklı veri kümesi

1. Adım: Sınıflandırma kısmı kaldırılacaktır.
2. Adım: Yeni veri setindeki sınıf sayısı kadar çıktı boyutuna sahip yeni bir sınıflandırma kısmı eklenecektir.
3. Adım: Sınıflandırma kısmındaki ağırlıklar rastgele başlatılacaktır.
4. Adım: Ağ, sınıflandırma kısmındaki ağırlıkları güncellemek ve önceden eğitilmiş modelin özellik çıkarma kısmındaki tüm ağırlıklara ince ayar yapmak için eğitilecektir.

4.10. Geriye Yayılım Algoritması

Geriye yayılım algoritması, sinir ağları için verimli bir eğitim algoritması olarak Rumelhart ve ark. (1986) tarafından tanıtıldı. Ana fikri, ağ parametrelerini, ağın gerçek çıktısı ile istenen çıktı arasındaki farkı en aza indirecek şekilde ayarlamaktır. Diğer bir deyişle, ağın kaybını en aza indirmek için parametrelerini ayarlayarak ağı eğitiriz. Bölüm 4.3'te, bir ileri beslemeli sinir ağının çıktısını hesaplama yolunu gösterdik. Bölüm 4.7'de, ağın hatasını hesaplamak için kullanılan kayıp fonksiyonlarını ele aldık. Bir $E(x_1, x_2, \dots, x_n)$ fonksiyonumuz varsa, bu fonksiyonun gradyanı şu şekilde hesaplanır:

$$\frac{\partial E}{\partial x} = [\frac{\partial E}{\partial x_1}, \frac{\partial E}{\partial x_2}, \dots, \frac{\partial E}{\partial x_n}] \tag{4.22}$$

Bu türevi hesaplamak, bize $E(x)$ fonksiyonunun $x_1, x_2, \dots, x_n$ parametrelerini değiştirerek değerinin nasıl değişeceği hakkında bilgi verir. Başka bir deyişle, kayıp fonksiyonunu $E(x)$ en aza indirmek için $x_1, x_2, \dots, x_n$ parametrelerinin pozitif veya

negatif yönde ne kadar değişmesi gerektiğini gösterir. Bölüm 4.3'teki sinir ağını düşünün. Eğer w_{ij}^k, *k-1* katmanındaki *i* nöronu *k* katmanındaki *j* nöronuna bağlayan ağırlık, b_j^k, *j* nöronunun bias'ı ve o_i^{k-1}, *k-1* katmanındaki *i* nöronunun çıkışı ise, o zaman *k* katmanındaki *j* nöronunun girişi olan Z_j^k, aşağıdaki gibi hesaplanır:

$$Z_j^k = \sum_{i=1}^{n} O_i^{k-1} w_{ij}^k + b_j^k \tag{4.23}$$

Ve *k* katmanındaki *j* nöronunun çıktısı şu şekilde hesaplanır:

$$O_j^k = \sigma(Z_i^k) = \sigma\left(\sum_{i=1}^{n} O_i^{k-1} w_{ij}^k + b_j^k\right) \tag{4.24}$$

Bu durumda σ aktivasyon fonksiyonudur. *E* fonksiyonu, sinir ağındaki tüm parametrelerin bir fonksiyonudur ve bunu tüm *k* katmanlar ve *i* ve *j* nöronları $E(w_{ij}^k, b_j^k)$ olarak yeniden yazabiliriz. Zincirleme kuralı kullanılarak w_{ij}^k ve b_i^k parametrelerinin gradyanı şu şekilde hesaplanabilir:

$$\frac{\partial E}{\partial w_{ij}^k} = \frac{\partial E}{\partial Z_j^k} \cdot \frac{\partial Z_j^k}{\partial w_{ij}^k} \tag{4.25}$$

$$\frac{\partial Z_j^k}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k}\left(\sum_{i=1}^{n} O_i^{k-1} w_{ij}^k + b_j^k\right) = O_i^{k-1} \tag{4.26}$$

Böylece, w_{ij}^k parametresinin gradyanı aşağıdaki gibidir:

$$\Delta E(w_{ij}^k) = \frac{\partial E}{\partial w_{ij}^k} = \frac{\partial E}{\partial Z_j^k} \cdot O_i^{k-1} \tag{4.27}$$

Kısmi türevin bu ayrışımı, temel olarak, w_{ij}^k ağırlığına bağlı olarak hata fonksiyonundaki değişikliğin, Z_j^k aktivasyonu nedeniyle hata fonksiyonu *E*'deki değişimin çarpı w_{ij}^k ağırlığına bağlı olarak Z_j^k aktivasyonundaki değişimin olduğunu söyler. Benzer şekilde, b_i^k parametresinin gradyanı aşağıdaki gibidir:

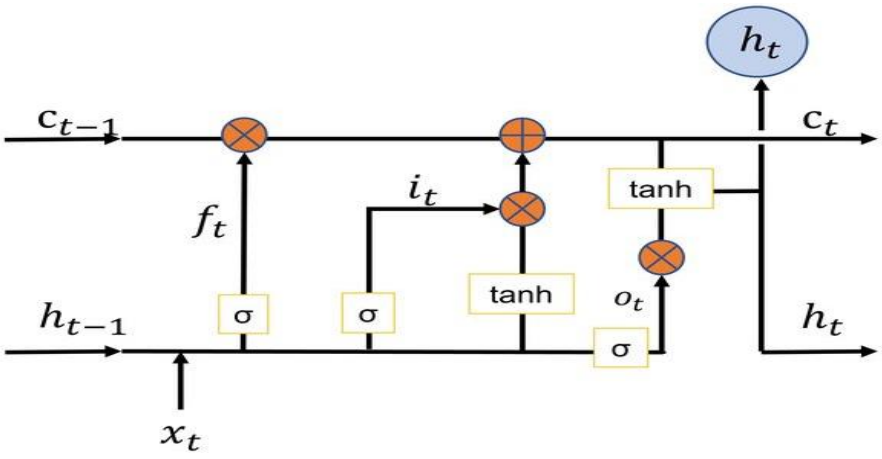
$$\frac{\partial Z_j^k}{\partial b_i^k} = \frac{\partial E}{\partial Z_j^k} \cdot \frac{\partial Z_j^k}{\partial b_i^k} \tag{4.28}$$

$$\frac{\partial Z_j^k}{\partial b_i^k} = \frac{\partial}{\partial b_j^k}\left(\sum_{i=1}^{n} O_i^{k-1} w_{ij}^k + b_j^k\right) = 1 \tag{4.29}$$

$$\Delta E(b_i^k) = \frac{\partial E}{\partial b_j^k} = \frac{\partial E}{\partial Z_j^k} \tag{4.30}$$

4.11. Tekrarlayan Yapay Sinir Ağları

İnsan beyni olayları zamansal olarak işler; diğer bir deyişle insan beyni, bir andaki olayla ilgili bilgileri işledikten sonra bu bilgileri ihmal edip bir sonraki olayı sıfırdan işlemeye başlamaz. Örneğin, bir makaleyi okurken, okumaya devam ettikçe o makaleye ilişkin anlayışımız gelişir, hatta bazı durumlarda değişir. CNN'ler gibi standart ileri beslemeli sinir ağları, zamansal olarak gelişen verileri işleyemez. RNN'ler (Tekrarlayan Sinir Ağı); metinler, ses kayıtları ve videolar gibi zamansal özellikleri öğrenmede güçlüdür. Standart RNN'lerde karşılaşılan ana sorun, zaman içinde geri yayılım uygulanırken ağın parametreleri güncelleyememesine neden olan kaybolan gradyanlardır. Bu, ağın yalnızca yeni girdileri hatırlamasına, eski girdileri hatırlamamasına neden olur. Başka bir deyişle, ağ kısa süreli bir hafızaya dönüşür. Bunu çözmek için Hochreiter ve Schmidhuber (1997), bir hücre durumu ve çeşitli kapılar içeren LSTM hücreleri geliştirdi. Hücre durumu, önceki zaman adımlarındaki bilgileri son zaman adımına (LSTM hücreleri) taşıyan bir köprü görevi görür. Şimdiki giriş vektörü x_t ve önceki gizli durum vektörü h_{t-1}, önceki hücre durumu c_{t-1}'in muhafaza edilip edilmeyeceğine karar vermek için unutma kapısında kullanılır. Bu, hücre durumunun bilgiyi tutması gereken zamanın önceden tanımlanmadığı anlamına gelir. Ancak, ağ unutmayı öğrenir. Unutma kapısının çıktısını giriş kapısının çıktısına eklemek, yeni bir hücre durumu c_t ile sonuçlanacaktır. Çıkış kapısında, yeni hücre durumu c_t, şimdiki giriş x_t ve önceki gizli durum h_{t-1}, yeni gizli durum h_t'yi oluşturmak için kullanılır. Hem yeni hücre durumu c_t hem de yeni gizli durum h_t bir sonraki LSTM hücresine (zaman adımı) geçirilecektir. LSTM'nin yapısı şekil 4.7'de gösterilmektedir.



Şekil 4.7 LTSM hücre yapısı (Shi ve ark., 2022)

Hücre durumu c_t ve gizli durum h_t şu şekilde hesaplanır:

$$i_t = \sigma(w_{xi} \cdot x_t + w_{hi} \cdot h_{t-1} + b_i) \quad (4.31)$$

$$f_t = \sigma\big(w_{xf} \cdot x_t + w_{hf} \cdot h_{t-1} + b_f\big) \quad (4.32)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ tanh(w_{xc} \cdot x_t + w_{hc} \cdot h_{t-1} + b_c) \quad (4.33)$$

$$o_t = \sigma(w_{xo} \cdot x_t + w_{ho} \cdot h_{t-1} + b_o) \quad (4.35)$$

$$h_t = o_t \circ tanh\,(c_t) \quad (4.36)$$

w_{xi}, w_{xf}, w_{xo} ve w_{xc} giriş vektörünü giriş kapısına, unutma kapısına, çıkış kapısına ve hücre durumuna bağlayan ağırlık matrisleridir. w_{hi}, w_{hf}, w_{ho} ve w_{hc}önceki gizli durum vektörünü giriş kapısına, unutma kapısına, çıkış kapısına ve hücre durumuna bağlayan ağırlık matrisleridir.

5. ÖNERİLEN MODEL

Bu bölümde, önerilen modelin kavramsal tasarımını ve pratik uygulaması verilmiştir. Kavramsal tasarımda, önerilen modelin blok diyagramını, literatür taramasından birçok modelin güç alanlarını birleştiren hibrit bir modeli temsil edecek şekilde tasarlanmıştır. Pratik uygulama kısmında ise önerilen modelin uygulama detaylarını gösterilmiştir.

5.1. Kavramsal Tasarım

Bu tezin amacı, videolardan şiddeti en iyi doğrulukla ve mümkün olan en hızlı sürede tespit edebilen bir model oluşturmaktır. Literatür taramasındaki çalışmaları inceledikten sonra, güvenilir bir şiddet tespit modeli oluşturmak için araştırmacıların odaklandığı bazı yönlerin olduğunu görülmüştür. Bu yönleri tartıştıktan sonra, önerilen modelin ana bileşenlerini içeren blok diyagramı oluşturulmuştur.

5.1.1. Bir şiddet tespit modeli tasarlamanın önemli yönleri

İyi bir model tasarlama süreci, hipotezler oluşturmaya, bu hipotezlere dayalı modelleri uygulamaya ve bu modelleri test etmeye bağlıdır. Örneğin, bazı araştırmacılar ağın derinliğini artırmanın modelin genel doğruluğunu artıracağını öne sürmekte ve modellerini buna göre oluşturmaktadırlar. Bu bölümde, bir şiddet tespit modeli tasarlarken araştırmacıların başarılı hipotezler oluşturdukları alanları ele alınmıştır.

5.1.1.1. Modelin yapısı

Literatür taramasında yer alan tüm çalışmalarda modelin yapısı şiddet tespit modeli tasarlamanın önemli bir yönü olarak görülmektedir. Sudhakaran ve Lanz'a (2017) göre, şiddet içeren davranışlar yerel hareketler olarak kabul edildiğinden, performans açısından en iyi şiddet tespit modelleri, yerel uzamsal-zamansal özellikleri analiz ederek yerel hareketleri yakalama yeteneğine sahiptir. Bu görevi yapmak için doğrudan 3B CNN kullanılabilir, ancak bu modelin gerektirdiği büyük hesaplama gücünden kaçınmak için yerel uzamsal-zamansal özellikleri analiz edebilen daha hafif bir model olan ConvLSTM'nin kullanılmasını önermişlerdir.

5.1.1.2. Modeli eğitmek için kullanılan veri seti

Modeli eğitmek için kullanılan veri kümesi, modelin performansını sınırlayabilir ve bazı durumlarda eğitilen model, veri kümesi içindeki kalıpları analiz etmek yerine veri kümesini ezberler. Araştırmacıların büyük bir kısmı, modellerini eğitmek için benzer ortamlara ve benzer şiddet içerikli eylemlere sahip videolar içeren veri kümelerini kullanmışlardır. Bu, gerçek hayattaki eylemlere uygulandığında modelin performansını düşürür. Hokey Fights veri kümesi, benzer videoların bulunduğu veri kümeleri arasında yer almaktadır. Bu veri seti, hokey profesyonel maçları sırasında eşit miktarda şiddet içeren ve içermeyen eylemlerden oluşur: Genellikle, yakın vücut etkileşimine katılan iki oyuncunun sahneleri. Başka bir örnek de Violent Flow veri kümesidir, bu veri kümesindeki şiddet içeren sahnelerin çoğu futbol maçlarına aittir.

Soliman ve ark. (2019), örneklere göre değişen bir veri kümesine sahip olmak için gerçek hayattaki şiddet eylemlerine ilişkin iki bin video içeren bir veri kümesi oluşturmuşlardır. Şiddet içeren klipler; sokaklar, hapishaneler ve okullar gibi birçok farklı ortamda yaşanan kavgaları içerir. Şiddet içermeyen videolar ise; futbol oynamak, basketbol oynamak, tenis oynamak, yüzmek ve bir şeyler yemek gibi diğer insani eylemleri içerir.

5.1.1.3. Transfer öğrenmeyi kullanmak

Sumon ve ark. (2020), Gkountakos ve ark. (2020) ve Lima ve Figueiredo (2021) gibi birçok araştırmacı, daha iyi performanslar elde etmek için transfer öğrenmeyi uygulamıştır. Transfer öğrenmeyi uygulamak, daha iyi performans elde etmek için iyi bir yaklaşımdır. Ancak, çok büyük bir hesaplama gücü gerektirmeyen, hafif önceden eğitilmiş bir model seçmek önemlidir. Büyük hesaplama gücü gerektiren önceden eğitilmiş modeller, modeli yavaşlatabilir.

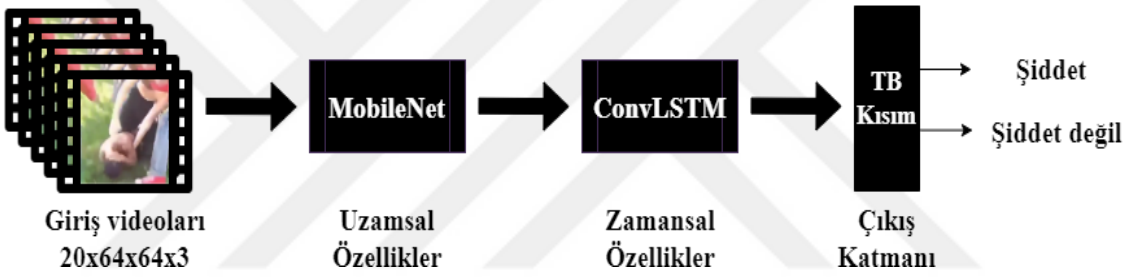
5.1.2. Önerilen modelin blok diyagramı

Geçen bölümde, yüksek performanslı bir şiddet tespit modeli tasarlamanın önemli yönlerini ele aldık. Bu bölümde, önceki çalışmaların avantajlarını ve güçlü alanlarını birleştirerek modelimizin blok diyagramının tasarımı verilmiştir. Modeli eğitmek ve

değerlendirmek için Real-Life Violence Situations veri seti kullanılmıştır. Daha önce de belirtildiği gibi, gerçek hayat eylemleri algılamak açısından modeli daha güvenilir hale getirecektir.

MobileNets CNN modeli, modelin genel performansını artırmak için faydalı olacak transfer öğrenmeyi uygulayarak uzamsal özellikleri çıkarmak için kullanılacaktır. Zamansal özelliklerin çıkarılması için, katmanları arasında konvolüsyonel bağlantılara sahip olan ConvLSTM kullanılmıştır. Bu, modeli yerel uzamsal-zamansal özellikleri çıkarabilecek kapasitede yapar.

Literatür taramasındaki üç çalışmanın bu güçlü noktalarını birleştirerek önerilen hibrit modelimizi oluşturduk. Önerilen modelin blok diyagramı şekil 5.1'de gösterilmektedir. Blok diyagramında gösterdiğimiz hibrit modeli, Real-Life Violence Situations veri setine uygulayan ilk araştırmacılar olduğumuzu not etmek önemlidir.



Şekil 5.1. Önerilen modelin blok diyagramı

5.1.2.1. MobileNets

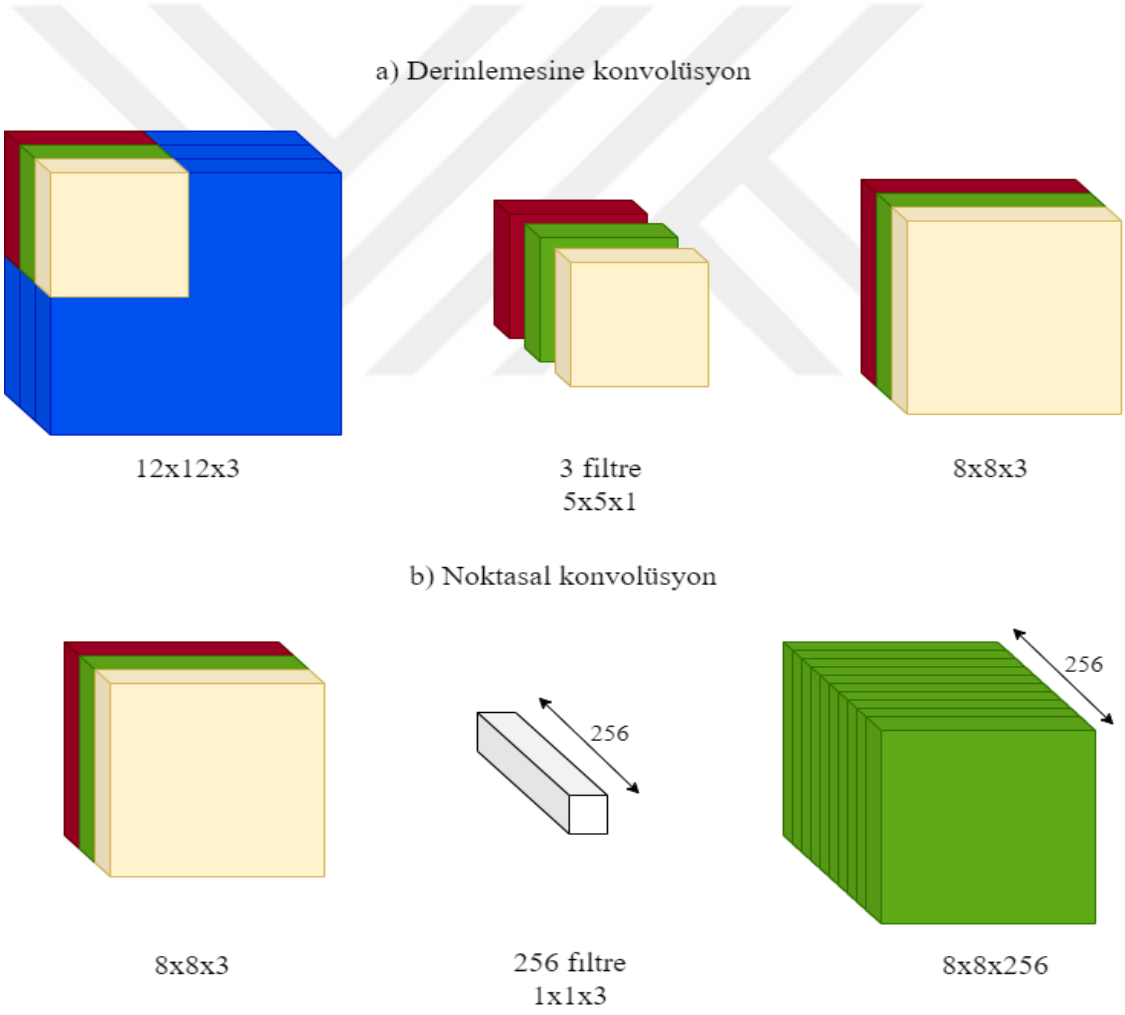
MobileNets'in ana görevi uzamsal özellikleri çıkarmaktır. MobileNets, Howard ve ark. (2017) tarafından Google firmasında mobil görme uygulamaları için geliştirilmiş bir CNN modelidir. MobileNets, yapısında ağın ihtiyaç duyduğu işlem gücünü azaltmada ve dolayısıyla modeli daha hafif ve daha hızlı hale getirmede çok avantajlı olan derinlemesine ayrılabilir konvolüsyonu kullanır. Bu, MobileNets modelimizde uzamsal özellik çıkarıcı olmaya uygun bir aday yapar.

Derinlemesine ayrılabilir konvolüsyon fikri, *HxWx3* boyutlu bir giriş RGB görüntüsünü *KxKx3* boyutunda *n* sayıda filtre ile konvolüsyon etmek yerine, *HxW* boyutunda her bir giriş kanalı *KxK* boyutunda bir filtre ile konvolüsyon edilmesidir. Bundan sonra, derinlemesine ayrılabilir konvolüsyonun sonuçlarına *1x1x3* boyutunda n

filtre ile noktasal konvolüsyon uygulanır. Derinlemesine ayrılabilir konvolüsyon şekil 5.2'de gösterilmiştir.

Bu iki işlem, normal konvolüsyon ile elde edilen çıktıya eşdeğer bir çıktıya yol açar, ancak çok daha az işlem gücü gerektirir. Örneğin, bir giriş RGB görüntüsüne *5x5x3* boyutunda 256 filtre uygulamak istenirse, normal konvolüsyon 5*5*3*256 = 19200 parametre gerektirir. Ancak derinlemesine konvolüsyon durumunda toplam parametre sayısı (5*5*1*3) + (1*1*3*256) = 843'tür. Parametre sayısındaki bu büyük azalma ağı önemli ölçüde hızlandırır.

MobileNets 55 derinliğe sahiptir ve toplam parametre sayısı 4.3 milyondur. Bu ağ, ImageNet veri setinde %89.5'lik doğruluğuna ulaşmıştır. MobileNets'in genel yapısı tablo 5.1'de gösterilmiştir.



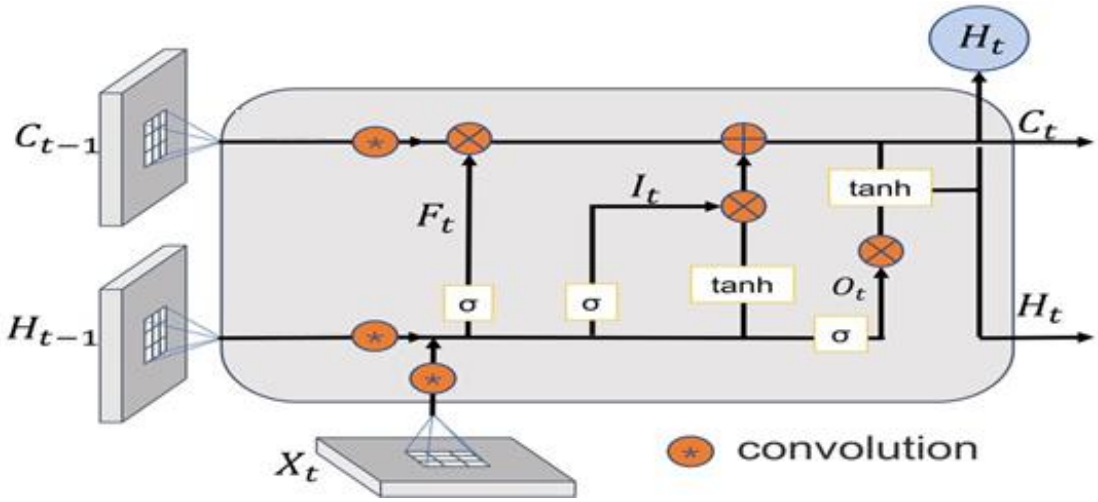
Şekil 5.2. Derinlemesine ayrılabilir konvolüsyon aşamaları, a) Derinlemesine konvolüsyon b) Noktasal konvolüsyon

Tablo 5.1. MobileNets genel yapısı (Howard ve ark., 2017)

Konvolüsyon Türü / Adım	Filtre Boyutu	Giriş Boyutu
Konv / 2A	3x3x3x32	224x224x3
D Konv / 1A	3x3x32 D	112x112x32
Konv / 1A	1x1x32x64	112x112x32
D Konv / 2A	3x3x64 D	112x112x64
Konv / 1A	1x1x64x128	56x56x64
D Konv / 1A	3x3x128 D	56x56x128
Konv / 1A	1x1x128x128	56x56x128
D Konv / 2A	3x3x128 D	56x56x128
Konv / 1A	1x1x128x256	28x28x128
D Konv / 1A	3x3x256 D	28x28x256
Konv / 1A	1x1x256x256	28x28x256
D Konv / 2A	3x3x256 D	28x28x256
Konv / 1A	1x1x256x512	14x14x156
5X D Konv / 1A	3x3x512 D	14x14x512
5X Konv/1A	1x1x512x512	14x14x512
D Konv / 2A	3x3x512 D	14x14x512
Konv / 1A	1x1x512x1024	7x7x512
D Konv / 2A	3x3x1024 D	7x7x1024
Konv / 1A	1x1x1024x1024	7x7x1024
Alt örnekleme	7x7	7x7x1024
Tam bağlantılı katman	1024x1000	1x1x1024
Softmax	Sınıflandırıcı	1x1x1000

5.1.2.2. ConvLSTM

ConvLSTM, daha önce bölüm 4.11'de bahsedilen LSTM'nin bir çeşididir. LSTM yalnızca zamansal özellikleri analiz edebilir. LSTM'leri CNN'lerle birleştirirsek, küresel uzamsal-zamansal özellikleri analiz edebilen bir CNN-LSTM ağı elde ederiz. LSTM'ler tamamen bağlantılı bir yapıya sahip oldukları için yalnızca global özellikleri çıkarabilirler. LSTM'yi bir CNN'nin yardımı olmadan kendi başına uzamsal-zamansal özellikleri analiz edebilecek hale getirmek için, Shi ve ark. (2015) şimdiki girişi X_t, önceki hücre durumu C_{t-1}'i ve önceki hücrenin gizli durumu H_{t-1}'i şimdiki hücreye bağlamak için konvolüsyonel bağlantılar kullandılar. Bir ConvLSTM uzamsal-zamansal özellikleri kendi başına çıkarabilse de önceden eğitilmiş ConvLSTM modelleri olmadığı için transfer öğrenmeyi uygulamak istiyorsak ConvLSTM kendi başına kullanılamaz. Sonuç olarak hem transfer öğrenmeyi uygulamak hem de yerel uzamsal-zamansal özelliklerinden faydalanmak için MobileNets ile ConvLSTM kullanıyoruz. ConvLSTM'nin yapısı şekil 5.2'de gösterilmektedir.



Şekil 5.2. ConvLSTM hücre yapısı (Shi ve ark., 2022)

Herhangi bir ConvLSTM hücresi, hücre gizli durumu H_t ve hücre durumu C_t olmak üzere iki adet çıktısı vardır. X_t, t zaman adımındaki giriş verileri, H_{t-1} ve C_{t-1} sırasıyla bir önceki zaman adamında hücre gizli durumu ve hücre durumu ise, hücre gizli durumu H_t ve hücre durumu C_t aşağıdaki denklemlere göre hesaplanır.

$$I_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \quad (5.1)$$

$$F_t = \sigma\big(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f\big) \quad (5.2)$$

$$C_t = F_t \circ C_{t-1} + I_t \circ tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \quad (5.3)$$

$$O_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_{t-1} + b_o) \quad (5.4)$$

$$H_t = O_t \circ tanh\,(C_t) \quad (5.5)$$

Bu denklemleri bölüm 4.11'de bulunan denklemlerle karşılaştırsak, (·) çarpma işlemi yerine (∗) konvolüsyon işlemi kullanılmaktadır.

5.1.2.3. Sınıflandırma kısmı

Bu kısım; ağın önceki kısımlarından gelen uzamsal-zamansal özellikleri bir düzleştirme katmanı ile düzleştirildikten sonra ele alır. Bu kısımda giriş ve çıkış katmanları arasındaki tüm nöronlar birbirine bağlıdır. Bu kısmın çıktı katmanında 4.7.1.1 bölümünde bahsettiğimiz Softmax aktivasyon fonksiyonunu kullanılmıştır. Softmax, her çıktı sınıfı için bir olasılık verir ve bu olasılıkların toplamı bire eşittir. Olasılığı daha yüksek olan sınıf, videonun doğasını tanımlar. Bu durumda, videonun şiddet doğasıyla ilgili iki çıkış sınıfımız vardır. Video şiddet içeriyorsa "Şiddet" sınıfının çıktısı bir olur ve "Şiddet değil" çıktısı sıfır olur ve bunun tersi de geçerlidir.

5.2. Pratik Uygulama

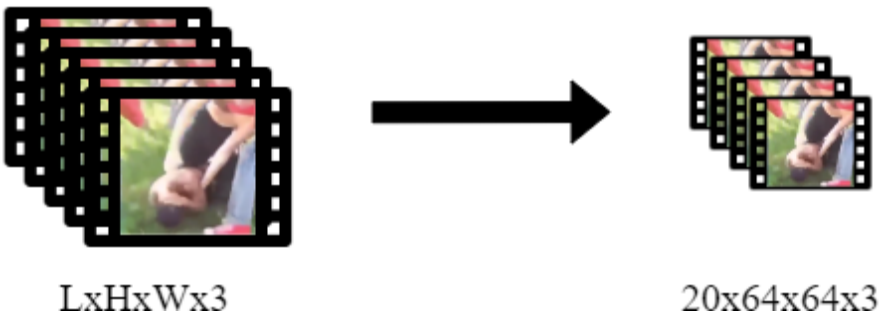
Önerilen modelin pratik uygulaması dört aşamadan oluşmaktadır: Öncelikle veri setini eğitim için hazırlık, transfer öğrenme tekniklerinin uygulanması, modeli oluşturmak için blok diyagramın bileşenlerini bir araya getirilmesi ve son olarak da modelin eğitim seçeneklerinin tanımlanmasıdır.

5.2.1. Veri setinin hazırlanması

Real-Life Violence Situations veri seti, "Şiddet değil" olarak etiketlenmiş 1000 video ve "Şiddet" olarak etiketlenmiş 1000 video içerir. Videoların uzunluğu *L*, kare yüksekliği *H* ve kare genişliği *W* olarak farklılık gösterir. Tüm videolar RGB renklidir. Bu, veri kümesindeki herhangi bir videonun genel boyutunu *LxHxWx3* yapar. Aşağıdaki denkleme göre tanımlanan bir örnekleme penceresi *n* olduğunu varsayalım:

$$n = \frac{L}{20} \tag{5.6}$$

Denklem 5.6 kullanılarak, veri setindeki tüm videolar, videolar okunurken her *n* kareden sonra bir kare alınarak *20x64x64x3* boyutuna getirilir. Bu, veri kümesindeki her videonun 20 kare içermesini ve her karenin üç renk kanallı RGB *64x64* boyutunda olmasını sağlar. Şekil 5.2'de, videolara uygulanan ön işleme gösterilmektedir. Bu çalışmada, önerilen modelin eğitilmesi için 1700 video, test edilmesi için ise 300 video kullanılacaktır.



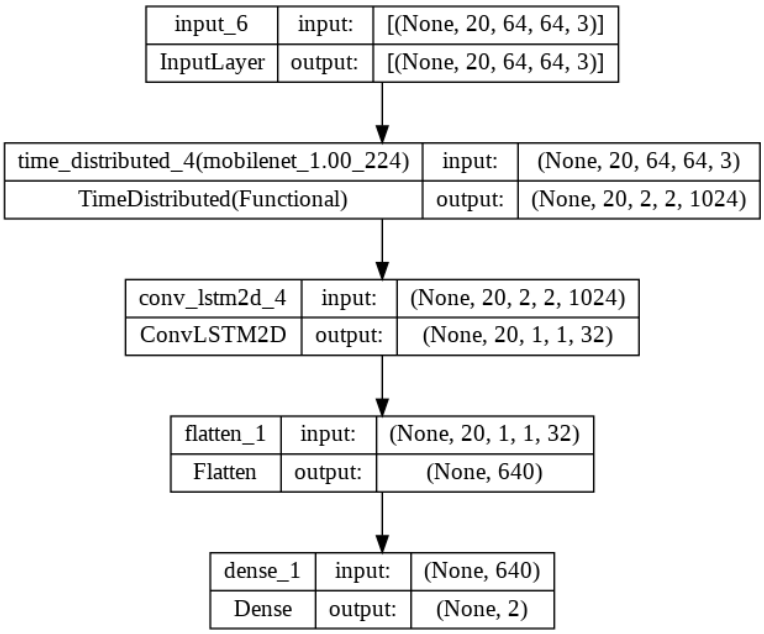
Şekil 5.2. Videolara uygulanan ön işleme

5.2.2. Transfer öğrenme tekniklerini uygulamak

Bu çalışmada transfer öğrenmeyi MobileNets modelini içe aktararak ve üst düzey katmanlarını kaldırarak, yani sınıflandırma bölümünü kaldırarak ve uzamsal özellikler çıkarma bölümünü koruyarak uygulanmıştır. Bundan sonra, ağın bir sonraki parçası olan ConvLSTM parçasına bağlanacaktır. MobileNets ağırlıkları dondurulacaktır. Bu, eğitim sürecinin yalnızca ConvLSTM içindeki parametreleri ve sınıflandırma bölümlerini içereceği anlamına gelir. Bu durumda, MobileNets özellik çıkarıcı olarak kullanılır.

5.2.3. Model oluşturmak

Önerilen modeli oluşturmak için Python'da açık kaynaklı TensorFlow (Abadi ve ark., 2015) kitaplığı kullanılmıştır. Giriş ve çıkış şekilleri gibi ağın parametreleri tanımlanmıştır. Daha önce veri hazırlama bölümünde tartıştığımız gibi, giriş videolarının boyutu 20x64x64x3'tür. MobileNets, 4 boyutlu girdileri, yani videoları kabul edemez. Bunu çözmek için, MobileNets'in videoları girdi olarak almasını sağlayan, zamana göre dağıtılmış bir katmanla kullanılmıştır. Daha sonrası, MobileNets'i ağın bir sonraki kısmı olan ConvLSTM kısmıyla birleştirilmiştir. ConvLSTM kısmının çıktısı, bir düzleştirme katmanı kullanılarak düzleştirilir. Bundan sonra, iki çıkış sınıfına sahip bir sınıflandırma kısmı ile birleştirilmiştir. Şekil 5.3 önerilen modelin yapısını göstermektedir.



Şekil 5.3. Önerilen modelin Python'da oluşturulması

5.2.4. Eğitim seçeneklerini tanımlama

Bu modeli eğitmek için, kayıp fonksiyonu olarak kategorik çapraz entropiyi ve optimizasyon algoritması olarak Adam algoritması kullanılmış, Minibatch'i 20 olarak tanımlanmıştır. Bölüm 4.7.2.2'de tanımlandığı şekliyle minibatch'lar, eğitim sürecini yinelemelere bölmek için kullanılan veri kümesinin küçük alt gruplarıdır; öyle ki ağ parametreleri her eğitim örneğinden veya veri kümesinin bütün örneklerinden değil, her küçük alt gruptan sonra güncellenir. Maksimum epoch sayısını 20 olarak tanımlarız. Yinelemeler veya itrasyonlar, modelin küçük alt gruplarını kaç kez gördüğünü tanımlarken; epoch'lar, modelin tüm veri kümesini kaç kez gördüğünü tanımlar. Gereksiz eğitimden kaçınmak için, model doğruluğunda 10 epoch'dan sonra herhangi bir gelişme olmazsa eğitimin duracağı şekilde bir erken durdurma kriteri kullanmıştır.

6. ÖNERİLEN MODELİN PERFORMANSININ DENEYSEL OLARAK İYİLEŞTİRİLMESİ

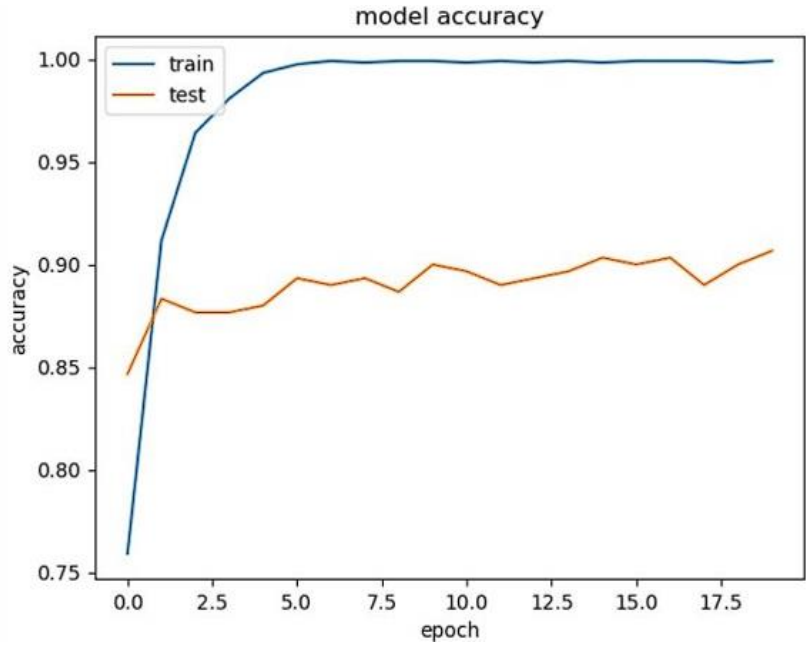
Bu bölüme, 5. bölümde önerilen modelin eğitim ve test sonuçlarını göstererek başlıyoruz. Hedef, %95'in üzerinde bir test doğruluğu elde etmektir. Bu nedenle, 5. bölümde önerilen modelin doğruluğu %95'i geçmiyorsa, doğruluğu artırmak için bazı model parametrelerini değiştirerek birkaç deney yapılacaktır. En iyi sonucu sağlayan parametreler, önerilen modeli yeniden oluşturmak için kullanılacaktır.

6.1. Deney1: Önerilen Modelin Test Edilmesi

Bu deneyde, 5. bölümde önerilen modeli hiçbir parametreyi değiştirmeden test ediyoruz. Daha önce belirtildiği gibi, bu model transfer öğrenme tekniklerini kullanır. Böylece sadece zamansal özellik çıkarma kısmı ve sınıflandırma kısmı eğitilecektir. Maksimum epoch sayısı 20 olarak tanımlanır.

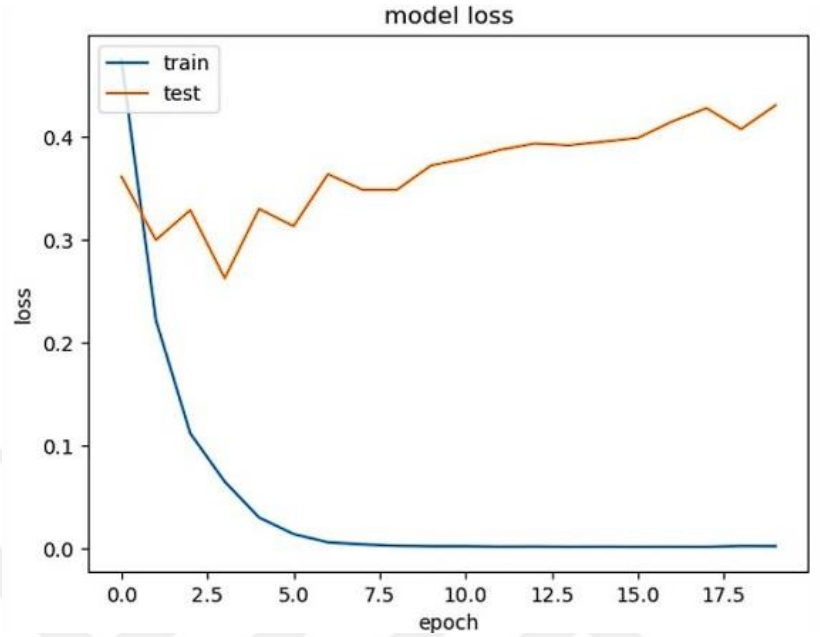
6.1.1. Deney1 sonuçları

Önerilen modelin eğitim ve test doğruluk eğrileri şekil 6.1'de gösterilmektedir. Önerilen modelin test doğruluğu %91'dir.



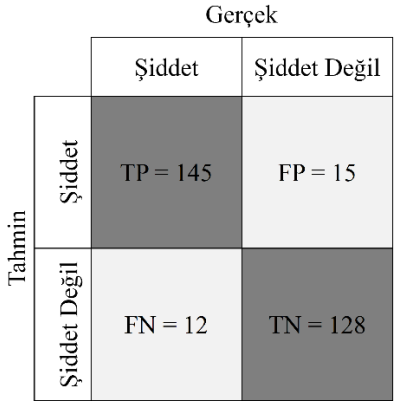
Şekil 6.1. Deney1'de elde edilen doğruluk

Şekil 6.1'de görüldüğü gibi bu sonuç 18 epoch'dan sonra elde edilmiştir. Modelin kaybı şekil 6.2'de gösterilmektedir.



Şekil 6.2. Deney1'deki modelin kaybı

Şekil 6.3'te deney1'de test edilen modelin karışıklık matrisi gösterilmektedir. Karışıklık matrisinde (Confusion Matrix) görülen TP (Doğru Pozitif), gerçekte şiddet içeren ve model tarafından "Şiddet" olarak sınıflandırılan videoların sayısıdır. FP (Yanlış Pozitif), gerçekte şiddet içermeyen ama model tarafından "Şiddet" olarak sınıflandırılan videoların sayısıdır. TN (Doğru Negatif), gerçekte şiddet içermeyen ve model tarafından "Şiddet Değil" olarak sınıflandırılan videoların sayısıdır. FN (Yanlış Negatif) ise, gerçekte şiddet içeren ama model tarafından "Şiddet Değil" olarak sınıflandırılan videoların sayısıdır.



Şekil 6.3. Deney1'de elde edilen karışıklık matrisi

Modelin performansını daha detaylı bir şekilde ölçmek için, karışıklık matrisinde bulunan sayıları ve aşağıdaki denklemleri kullanarak Kesinlik, Duyarlık ve Özgüllük gibi metrikler elde edilebilir.

Duyarlık: Doğru bir şekilde tahmin edilen şiddet içeren videoların, tüm şiddet içerikli videolara oranıdır. Buna Şiddet tespitin gerçek doğruluk oranı da diyebiliriz ve bu oran yüksek olmalıdır ve aşağıdaki gibi hesaplanır.

$$Duyarlık = \frac{TP}{TP+FN} \tag{6.1}$$

Kesinlik: Doğru bir şekilde tahmin edilen şiddet içeren videoların, "Şiddet" olarak tahmin edilen tüm videolara oranıdır. Bu oran da yüksek olmalıdır ve aşağıdaki gibi hesaplanır.

$$Kesinlik = \frac{TP}{TP+FP} \tag{6.2}$$

Özgüllük: Doğru bir şekilde tahmin edilen şiddet içermeyen videoların, tüm şiddet içermeyen videolara oranıdır. Ayrıca, şiddet içermeyen videoların tespitinin gerçek doğruluk oranı olarak da tanımlanabilir. Bu oran da yüksek olmalıdır ve aşağıdaki gibi hesaplanır.

$$Özgüllük = \frac{TN}{TN+FP} \tag{6.3}$$

Bu denklemleri kullanarak deney1 için Duyarlık, Kesinlik ve Özgüllük hesaplanmıştır ve sırasıyla %92.36, %90.63 ve %89.51 olarak elde edilmiştir.

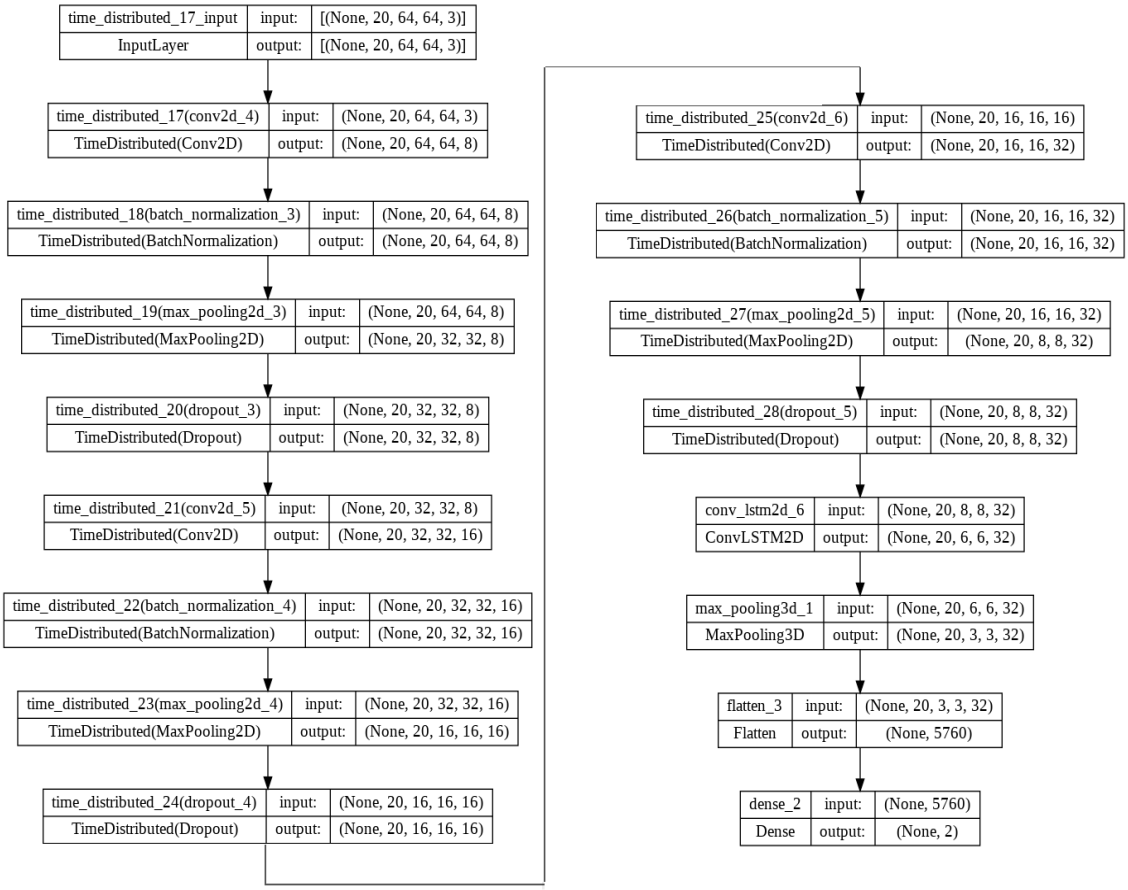
6.2. Deney2: Sıfırdan bir Model Oluşturmak

Bu deneyde, önceden eğitilmiş model MobileNets'ten faydalanmadan özellik çıkarma kısmını sıfırdan oluşturarak modelin performansını iyileştirme hedeflenmiştir. Böylece, MobileNets'i bir CNN ile değiştirilmiştir. Bu modelin pratik uygulanması şekil 6.4'te gösterilmektedir. Modelleri sıfırdan eğitmek daha uzun sürdüğü için bu deneyde maksimum epoch sayısını 40 olarak belirlenmiştir.

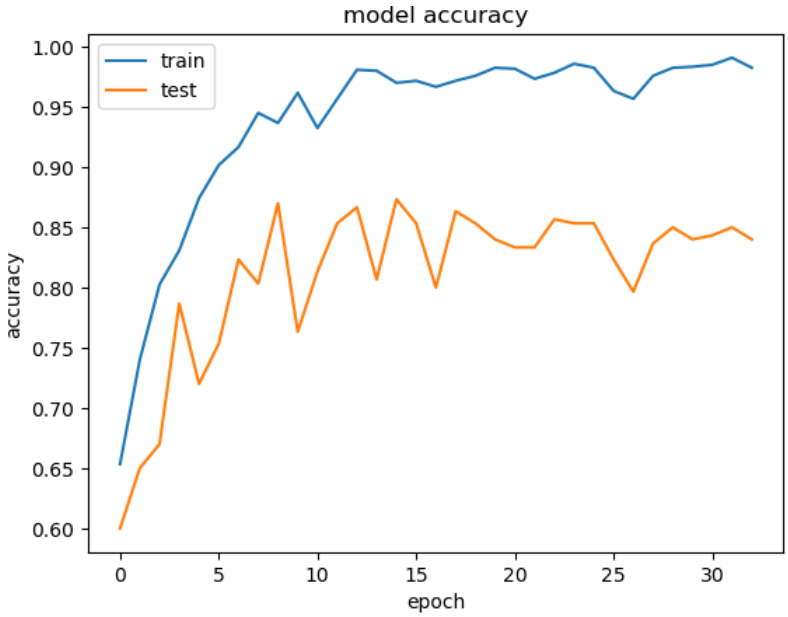
6.2.1. Deney2 sonuçları

Bu modelin performansı şekil 6.5 ve 6.6'da gösterilmektedir. 10 epoch boyunca test doğruluğunda herhangi bir gelişme olmadığı için eğitim 35 eopch sonra durduruldu. Şekil 6.5'te görüldüğü gibi, bu modelin test doğruluğu, bölüm 5'te önerilen modelin doğruluğundan daha düşük olan %87.33'tür. Ayrıca, bu modelin eğitiminin kararlı

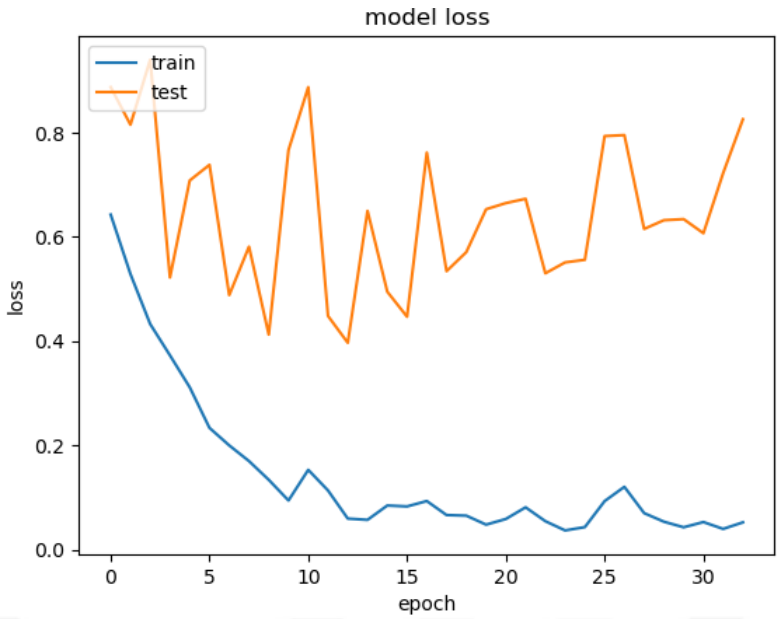
olmadığı şekil 6.5 ve 6.6'den de görülebilir. Bu sebeplerden dolayı bu model ihmal edilmiştir.



Şekil 6.4. Deney2'de sıfırdan oluşturulan model

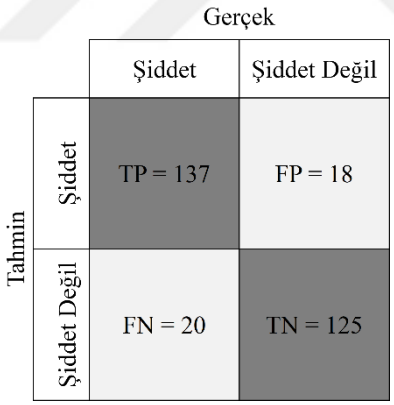


Şekil 6.5. Deney2'de elde edilen doğruluk



Şekil 6.6. Deney2'deki modelin kaybı

Deney2'de elde edilen karışıklık matrisi şekil 6.7'de gösterilmektedir. Bu matristen Duyarlık, Kesinlik ve Özgüllük değerleri hesaplanmıştır ve sırasıyla %87.26, %88.39 ve %87.41 olarak elde edilmiştir.

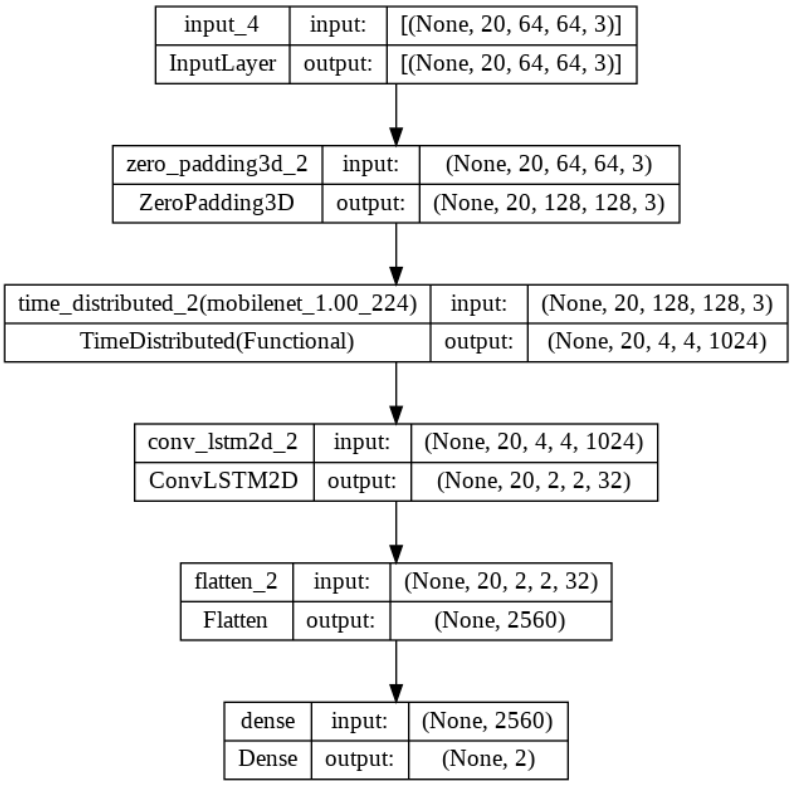


Şekil 6.7. Deney2'de elde edilen karışıklık matrisi

6.3. Deney3: Önerilen Modele Sıfır Doldurma Katmanı Eklemek

Şekil 5.3'te gösterilen önerilen modelin yapısına tekrar bakarsak, MobileNets'in çıktısının önemli ölçüde küçüldüğünü fark ederiz. Bu, MobileNets içinde gerçekleşen çoklu boyut küçültmelerinden kaynaklanmaktadır. MobileNets çıktısı *2x2* boyutunda olduğundan ve ConvLSTM modelindeki filtreler de bir filtrenin minimum boyutu olan *2x2* boyutunda olduğundan ConvLSTM kullanımı anlamsız hale gelmiştir. Bu sorunu

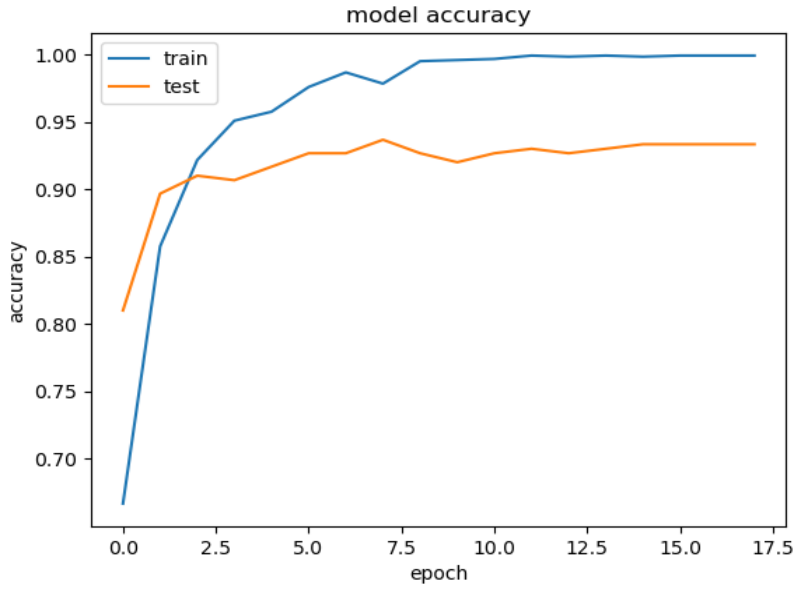
çözmek için bir çözüm, giriş videolarını *20x128x128x3* boyutuna getirmektir. Ancak bu çözümü uygulamaya çalıştığımızda, giriş videoları *20x64x64x3* boyutundayken bu çözüm dört kat daha fazla bellek gerektirdiğinden bellek sınırlamalarıyla karşılaşılmıştır. Bu sınırlamanın üstesinden gelmek için, önerilen modele sıfır doldurma katmanı eklenmiştir. Bu katman, giriş karelerinin kenarlarına sıfırlar ekleyerek giriş karelerini *64x64* boyutundan *128x128* boyutuna dönüştürür. Değiştirilen modelin yapısı şekil 6.8'de gösterilmiştir. Sıfır doldurma katmanının eklenmesinin MobileNets'in çıktı boyutunu *4x4*'e yükseltmeye yardımcı olduğu bu şekilde açıkça görülmektedir.



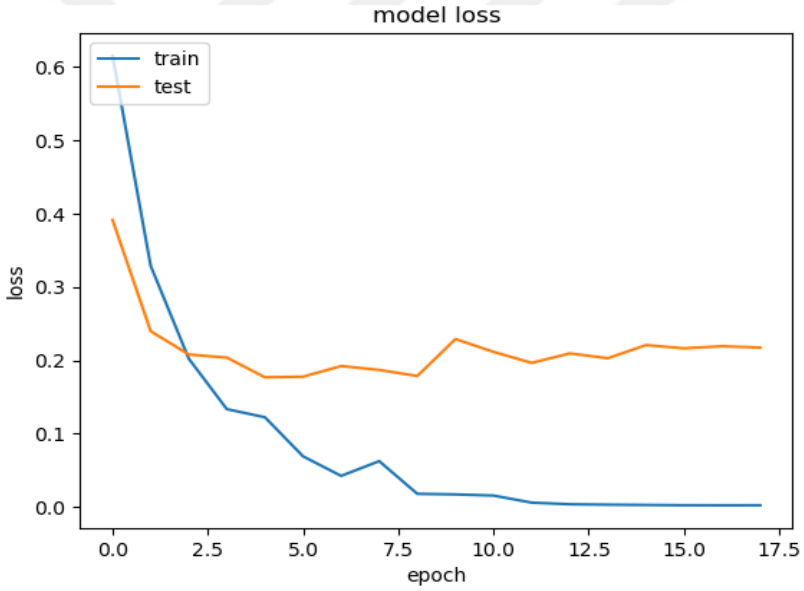
Şekil 6.8. Deney3'te yeniden oluşturulan model

6.3.1. Deney3 sonuçları

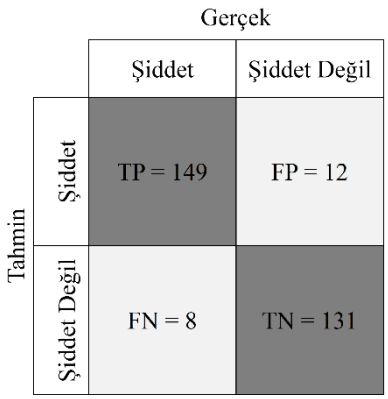
Değiştirilen modelin eğitim ve test doğruluk eğrileri şekil 6.9'da modelin kaybı ise şekil 6.10'da gösterilmektedir. Bu modelin test doğruluğu %93,33'tür. Şekil 6.9'da görüldüğü gibi bu sonuç 18 epoch'dan sonra elde edilmiştir. Şekil 6.10'da görüldüğü gibi bu modelin kaybı %20 civarındadır. Bu sonuç hem doğruluk hem de kayıp açısından deney1 ve deney2'de elde edilen sonucu aşmıştır.



Şekil 6.9. Deney3'te elde edilen doğruluk



Şekil 6.10. Deney3'taki modelin kaybı

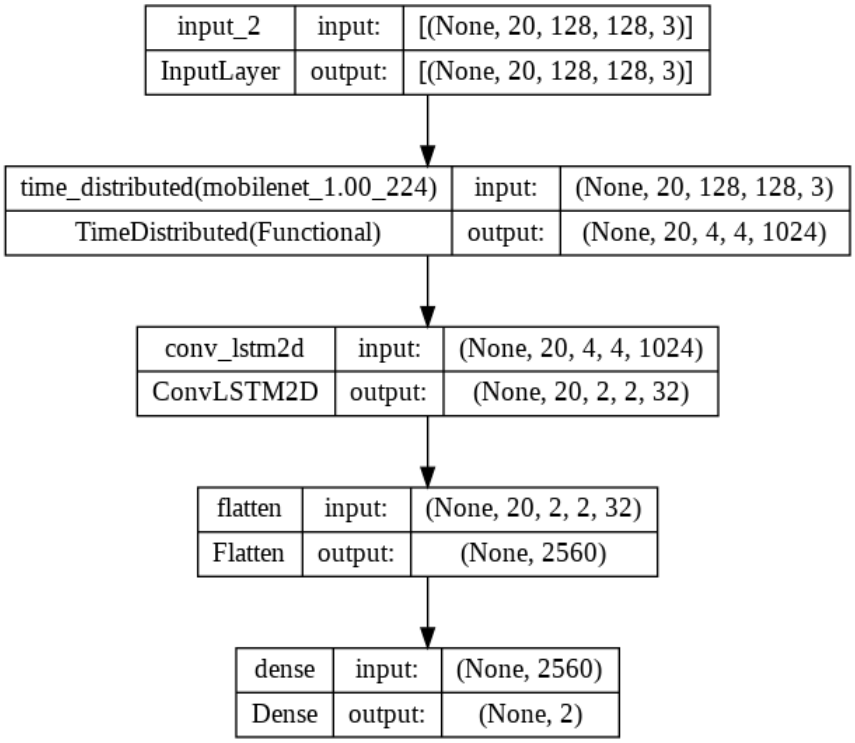


Şekil 6.11. Deney3'te elde edilen karışıklık matrisi

Deney3'te elde edilen karışıklık matrisi yukarıda gösterilen şekil 6.11'de verilmiştir. Bu matristen Duyarlık, Kesinlik ve Özgüllük oranları sırasıyla %94,90, %92.55 ve %91.61 olarak elde edilmiştir. Bu sonuçlar, deney1 ve deney2'de elde edilen sonuçlarla karşılaştırıldığında, deney3'te modelin performansının arttığı görülmektedir.

6.4. Deney4: Giriş Videolarının Yeniden Boyutlandırılması

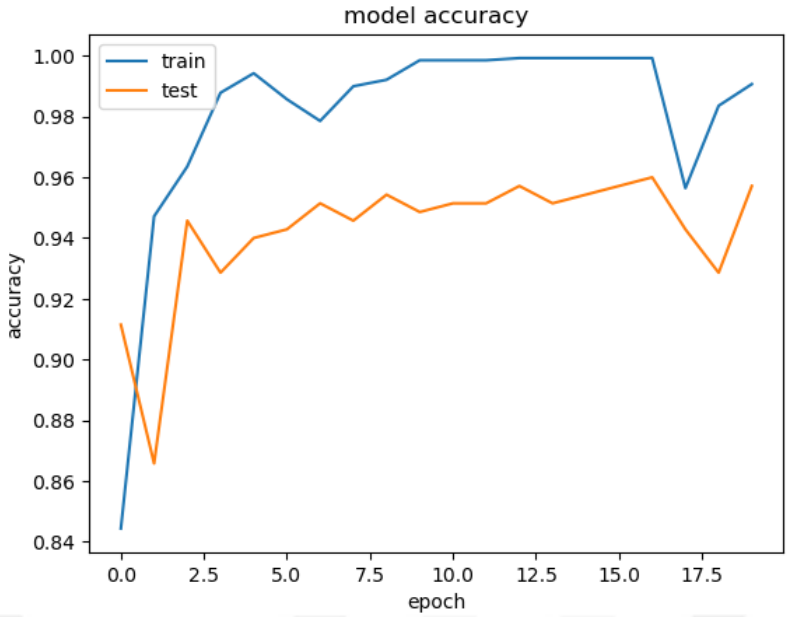
Deney3'ün olağanüstü sonuçları, bellek sınırlamalarının üstesinden gelmemiz ve giriş videolarını *20x128x128x3* boyutunda yeniden boyutlandırmamız için bize ilham vermiştir. Başlangıçta 16 GB RAM'e sahip bir bilgisayar kullanılmış ve bellek sınırlamalarını aşmak için RAM boyutunu 32 GB'a çıkarılmıştır. Videoları *20x128x128x3* boyutuna getirdikten sonra modelimizi buna göre yeniden oluşturduk. Değiştirilen modelin yapısı şekil 6.12'de gösterilmektedir.



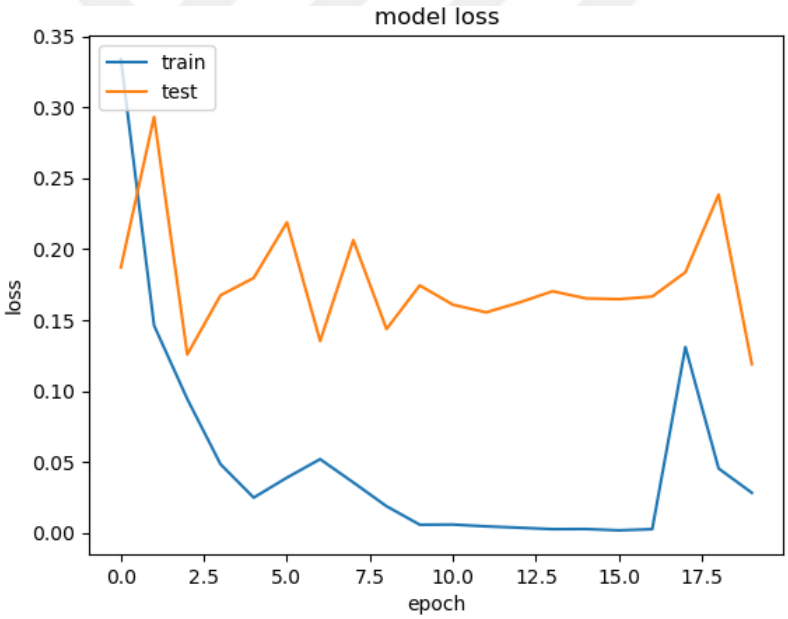
Şekil 6.12. Deney4'te yeniden oluşturulan model

6.4.1. Deney4 sonuçları

Bu deneyde, erken durdurma kriteri olmadan maksimum epoch sayısını 20'ye ayarlanmıştır. Bu deneyin sonuçları şekil 6.13 ve 6.14'te gösterilmektedir.

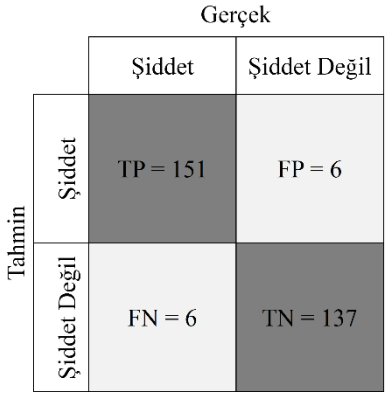


Şekil 6.13. Deney4'te elde edilen doğruluk



Şekil 6.14. Deney4'teki modelin kaybı

Şekil 6.13'te modelin 17. epoch'da %96'lık maksimum test doğruluğuna ulaşıldığı, ardından doğruluğun düşmeye başladığı ve 20. epoch'da tekrar %95.7'ye çıktığı görülmektedir. Eğitim süreci, eğitim bittikten sonra en iyi doğruluğu sağlayan parametreler geri alınacak şekilde yapıldığından dolayı bu durumda herhangi bir sıkıntı söz konusu değildir. Şekil 6.14'te görüldüğü gibi, modelin 17. epoch'daki kaybı %16'dır. Deney4'te elde edilen karışıklık matrisi şekil 6.15'te gösterilmektedir.



Şekil 6.15. Deney4 elde edilen karışıklık matrisi

Bu deneyde elde edilen karışıklık matrisinde Duyarlık, Kesinlik ve Özgüllük oranları sırasıyla %96.18, %96.18 ve %95.80 olarak elde edilmiştir. Bu sonuçlar, deney1, deney2 ve deney3'te elde edilen sonuçlarla karşılaştırıldığında, deney4'te modelin performansının oldukça arttığı görülmektedir. Aşağıda verilen tabloda deney1, deney2, deney3 ve deney4'te elde edilen sonuçlar karşılaştırılmaktadır.

Tablo 6.1. Deneylerde elde edilen sonuçların karşılaştırılması

Deney	Duyarlık "Şiddet"	Kesinlik	Özgüllük "Şiddet Değil"	Toplam Doğruluk
Deney1	%92.36	%90.63	%89.51	%91
Deney2	%87.26	%88.39	%87.41	%87.33
Deney3	%94,90	%92.55	%91.61	%93.33
Deney4	%96.18	%96.18	%95.80	%96

7. BULGULAR VE TARTIŞMA

Bu araştırmanın bulguları, bölüm 6'da yapılan deneylerin sonuçlarına dayanmaktadır. İlk olarak, deney1 ve deney2'nin sonuçları, transfer öğrenmenin uygulanmasının modelin doğruluğunu artırdığını ve eğitim sürecini de hızlandırdığını göstermektedir. Deney1'de 18 eğitim döneminden sonra transfer öğrenme kullanan model kullanılarak %91'lik bir test doğruluğu elde edilirken, deney2'de 35 eğitim döneminden sonra sıfırdan oluşturulan modelin elde ettiği maksimum doğruluk %87.33'tür. Model yapısı ve girdi boyutu açısından daha basit bir model kullanıyor olsak da deney1 sonucu, Soliman ve ark. (2019) tarafından elde edilen sonucu aşmaktadır ve Lima ve Figueiredo (2021) tarafından elde edilen sonuca eşittir. Ancak, deney1'de gösterilen model, Lima ve Figueiredo (2021) modelinden çok daha basit bir yapıya sahiptir. Bu modelde, araştırmacılar 10 akışlı bir modeli 10 giriş karesiyle besler. Bu 10 akış 10 tane MobileNets'ten oluşmaktadır.

Soliman ve ark. (2019) ve Lima ve Figueiredo (2021) çalışmalarında araştırmacılar girdi karelerinin boyutunu *224x224x3* olarak tanımlamışlardır. Bu giriş şekli, deney1'de kullanılan giriş şekli olan *64x64x3*'ün gerektirdiği belleğin yaklaşık on iki katı kadar bellek gerektirir. Ayrıca her iki çalışmada da araştırmacılar transfer öğrenme uygulamışlardır. Bu, katmanları arasında kıvrımlı bağlantılara sahip olan ConvLSTM kullanılarak elde edilen iyileştirmeleri gösterir. ConvLSTM, bir videodaki yerel hareketleri en iyi şekilde tanımlayan yerel uzamsal-zamansal özelliklerin analiz edilmesine yardımcı olmuştur. Ayrıca, 2. deneyde sıfırdan oluşturulan model bile, Soliman ve ark. (2019) çalışmasıyla kıyaslandığında, ConvLSTM uygulamanın faydalarını göstermektedir. Deney2'deki model, Soliman ve ark. (2019) tarafından elde edilen %89.22 olan doğrulukla hemen hemen eşdeğer olan %87.33'lük bir doğruluk elde eder. Bununla birlikte, deney2'deki model, toplam eğitilebilir parametre sayısına göre yaklaşık 20 kat daha hafiftir.

ConvLSTM kullanmanın faydaları 3. deneyde daha belirgin hale geldi. 1. deneyde, ConvLSTM tarafından alınan MobileNets çıktısının boyutu *2x2* idi, bu da ConvLSTM içindeki bir filtrenin mümkün olan en küçük boyutuna eşittir. Boyutları eşit olan iki değişken arasındaki konvolüsyon işlemi yalnızca bir çarpma işlemi olduğundan, bu, ConvLSTM kullanmanın faydalarını azaltır. Daha fazla bellek kullanmak zorunda kalmadan bu sorunun üstesinden gelmek için, Deney3'teki modele eğitilemez bir sıfır doldurma katmanı eklendi. Bu katman, *64x64x3* boyutunda olan girdi karelerinin

çevresine sıfırlar ekler ve bunları *128x128x3* boyutuna dönüştürür. Bu katmanın uygulanması, modelin doğruluğunun %91'den %93.33'e çıkarılmasına yardımcı olmuştur.

Deney4, giriş videolarının çözünürlüğü modelinin performansı üzerindeki önemini göstermektedir. Deney3'teki karelerin gerçek boyutu 64x64x3'tür. Bu karelerin boyutları çevrelerine sıfırlar eklenerek ikiye katlandı. Yine de karelerin çevresine sıfır eklemek karelerin boyutunu artırırken, karelerin çözünürlüğüne veya kalitesine kesinlikle hiçbir etkisi yoktur. Karenin gerçek boyutunu büyütmek daha fazla bellek gerektirse de 3. deneyin sonuçları performans uğruna belleğin bir kısmını takas edecek kadar cesaret vericiydi. Böylece veri seti, girdi karelerini *128x128x3* boyutuna yeniden boyutlandırmak için tamamen yeniden işlendi; bu, deney3'te belirtilen sıfır dolgulu karelerden daha iyi bir çözünürlüğe sahip kareleri elde etmemizi sağladı. Yeni verileri kullanılarak önerilen model yeniden eğitildi ve modelin test doğruluğu %93.33'ten %96'ya çıkarıldı. Bu sonuç Soliman ve ark. (2019) ve Lima ve Figueiredo (2021) tarafından elde edilen sonucu aşmaktadır ve Traore ve Akhloufi (2020) tarafından elde edilen sonuçla hemen hemen aynıdır.

Son olarak, Traore ve Akhloufi (2020)'nin çalışması %96,74'lük bir doğruluk elde etse de onların modeli, çalışmada sunulan modelimize kıyasıyla aşırı derecede karmaşıktır. Oluşturdukları model, EfficientNet-B0 kullanarak uzamsal özellik çıkarımı için iki akıştan oluşmaktadır. İki akış kullanmak, modelin boyutunu iki katına çıkarır. İlk akış karelerle, diğeri ise optik akış görüntüleriyle beslenir. Optik akış görüntüleri, videolardaki hareketler hakkında bilgi içerir. Optik akış görüntülerini oluşturmak için PWC-Net adlı başka bir CNN modeli kullanmışlardır. Bu karmaşıklık modeli yavaşlatır ve çok daha fazla hesaplama gücü gerektirir. Modelimiz ile önceki çalışmalardaki modeller arasındaki performans karşılaştırması tablo 7.1'de gösterilmektedir.

Tablo 7.1. Önceki çalışmalar ile modelimizin karşılaştırılması

Model	Modelin Yapısı	Girdi Karelerin Boyutu	Eğitilebilir Parametre Sayısı	Epoch Sayısı	Doğruluk
(Soliman ve ark., 2019)	Vgg16 + LSTM	224x224x3	2,163,712	2000	%88.2
(Lima ve Figueiredo, 2021)	10*(MobileNets) + LSTM	224x224x3	Bahsedilmemiş	50	%91
(Traore ve Akhloufi, 2020)	2 Akışlı Model 2*(EfficientNet-B0) + LSTM	128x128x3	Bahsedilmemiş	50	%96.74
Deney1	MobileNets + ConvLSTM	64x64x3	542,082	18	%91
Deney2	CNN + ConvLSTM	64x64x3	91,522	35	%87.33
Deney3	MobileNets + ConvLSTM	Zero-padded 128x128x3	1,221,762	18	%93.33
Deney4	MobileNets + ConvLSTM	128x128x3	1,221,762	20	%96

8. SONUÇLAR VE ÖNERİLER

Sonuç olarak, bu çalışmada elde edilen oldukça başarılı sonuçlar, videolarda şiddet tespiti görevi için derin öğrenme kullanımının güvenilirliğini ve istikrarını göstermektedir. Bu çalışmada, MobileNets ve ConvLSTM kullanılarak videolarda şiddet tespiti için hibrit bir model tanıtılmıştır. Önerilen model, Real-Life Violence Situations veri setinde, bazı durumlarda önceki çalışmaların performansını aşan ve bazı durumlarda neredeyse eşit olan %96'lık bir doğruluk elde etmiştir. Geliştirilen modelin Duyarlık, Kesinlik ve Özgüllük oranları sırasıyla %96.18, %96.18 ve %95.80'dir.

Bundan sonraki çalışmalar için iki temel öneri vardır. Birincisi giriş karelerini *224x224x3* boyutuna yeniden boyutlandırabilmek için daha fazla bellek kaynaklarını ayırmaktır. Bu boyut, MobileNets tarafından alınan standart boyuttur. Bu öneri, daha yüksek çözünürlüklü karelerle eğitilen modelin daha yüksek bir test doğruluğu elde ettiğini gösteren deneysel sonuçlara dayanarak sunulmaktadır. İkinci öneri, şiddet veya terörizm faaliyetlerini gerçekleşmeden önce de tahmin edebilmek için bu çalışmayı genişletmektir. Bu, şiddet veya terör amaçlı kişilerin davranışlarını içeren yeni bir veri kümesi oluşturarak yapılabilir. Şiddet veya terör niyeti olan kişilerin davranışlarının bazı ortak örüntüleri vardır. Örüntü öğrenmede derin öğrenme yaklaşımlarının güçlü olması bu öneriyi uygulanabilir kılmaktadır.

KAYNAKLAR

Abadi, M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. [Online], https://www.tensorflow.org/ [Ziyaret Tarihi: 30 Ekim 2022].

Abdi, M. (2021), "Event Or Emergency Case Detection By Human Running", Yüksek Lisans Tezi, *Fen Bilimleri Enstitüsü*, İstanbul Ticaret Üniversitesi.

Abdali, A. and Al-Tuma, R., 2019, Robust Real-Time Violence Detection in Video Using CNN And LSTM, *2nd Scientific Conference of Computer Sciences, SCCS-2019*, Iraq, 104-108.

Accattoli, S., Sernani, P. Falcionelli, N., Mekuria, D. and Dragoni, A., 2020, Violence Detection in Videos by Combining 3D Convolutional Neural Networks and Support Vector Machines, *Applied Artificial Intelligence*, 34(4), 329–344.

Albawi, S., Mohammed, T. and Al-Zawi, S., 2017, Understanding of a convolutional neural network, *" 2017 International Conference on Engineering and Technology, NCVPRIPG -2013*, *ICET-2017*, Antalya, 1-6.

Alom, M., Taha T., Yakop, C., Westberg, S., Sidike, P., Nasrin, M., Hasan, M., Essen, B., Awwal, A. and Asari, V., 2019, A State-of-the-Art Survey on Deep Learning Theory and Architectures, *Electronics*, 8(292).

Anonymous, 2020, National Statistics [online], National Coalition Against Domestic Violence, https://ncadv.org/STATISTICS [Ziyaret Tarihi: 10 Ekim 2022].

Anonymous, 2018, Deep Learning Toolbox, Mathworks, https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html#bu80qkw-3_head [Ziyaret Tarihi: 28 Haziran 2021].

Baba, M., Gui, V., Cernazanu, C., Pescaru, D., 2019, A Sensor Network Approach for Violence Detection in Smart Cities Using Deep Learning, *Sensors*, 19 (7).

Choi, R., Coyner, A., Kalpathy-Cramer, J, Chiang, M. and Campbell, J., 2020, Introduction to Machine Learning, Neural Networks, and Deep Learning, *Translational Vision Science & Technology*, 9(2*)*.

Dixit, M., Tiwari, A., Pathak, H. and Astya, R., 2018, An overview of deep learning architectures, libraries and its applications areas, *2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN-2018*, Greater Noida, 293-297.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S. and Dehmer, M., 2020, An Introductory Review of Deep Learning for Prediction Models With Big Data, *Frontiers in Artificial Intelligence*, 3.

Fukushima, k., 1988, Neocognitron: A hierarchical neural network capable of visual pattern recognition, *Neural Netw*, 1, 119-130.

Gkountakos, K., Ioannidis, K., Tsikrika, T., Vrochidis, S., Kompatsiaris, I., 2020, A Crowd Analysis Framework for Detecting Violence Scenes, *International Conference on Multimedia Retrieval, ICMR-2020*, Dublin, 276-280.

Halder, R., Chatterjee1, R., 2020, CNN-BiLSTM Model for Violence Detection in Smart Surveillance, *SN Computer Science*, 1(4).

Hara, K., Kataoka, H., Satoh, Y., 2017, Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition, *IEEE International Conference on Computer Vision Workshop, ICCVW-2017*, Italy.

Hochreiter, S. and Schmidhuber, J., 1997, LONG SHORT-TERM MEMORY. *Neural Computation*, 9(8).

Howard, A. G. et al., 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision, *arXiv:1704.04861*.

Jeba, B., Mala, J., 2020, Light weight convolutional models with spiking neural network based human action recognition, *Journal of Intelligent & Fuzzy Systems*, 39(1), 961-973.

Jimoh, H., 2020, How Transfer Learning works [online], Towards data scince, https://towardsdatascience.com/how-transfer-learning-works-a90bc4d93b5e [Ziyaret Tarihi: 3 Haziran 2021].

Kingma, D. and Ba, J., 2015, Adam: A Method for Stochastic Optimization, *2015 International Conference on Learning Representations, ICLR-2015, San Diego.*

Kristine, J. M., 2017, Violence [online], Britannica, https://www.britannica.com/topic/violence [Ziyaret Tarihi: 10 Ekim 2022].

Li, J., Jiang, X., Sun, T., Xu, K., 2019, Efficient Violence Detection Using 3D Convolutional Neural Networks, *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS-2019*, 1-8.

Lima., J. And Figueiredo, C., 2020, A Temporal Fusion Approach for Video Classification with Convolutional and LSTM Neural Networks Applied to Violence Detection, *INTELIGENCIA ARTIFICIAL*, 24(67), 40-50.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86(11), 2278-2324.

Mohtavipour, S., Saeidi, M., Arabsorkhi, A., 2021, A multi-stream CNN for deep violence detection in video sequences using handcrafted features, *The Visual Computer*.

Mumtaz, A., Sargano, A., Habib, Z., 2018, Violence Detection in Surveillance Videos with Deep Network using Transfer Learning, *2nd European Conference on Electrical Engineering and Computer Science, EECS-2018*, 558-563.

Nagyfi, R., 2018, The differences between Artificial and Biological Neural Networks [online], Towards data scince, https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7 [Ziyaret Tarihi: 30 Haziran 2021].

Najafabadi, M., Villanustre, F., Khoshgoftaar, T., Seliya, N., Wald, R., Muharemagic, E., 2015, Deep learning applications and challenges in big data analytics, *Journal of Big Data*, 2(1).

Parmar, R., 2018, Common Loss functions in machine learning [online], Towards data scince, https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23 [Ziyaret Tarihi: 30 Haziran 2021].

Patrikar, S., 2019, Batch, Mini Batch & Stochastic Gradient Descent [online], Towards data scince, https://towardsdatascience.com/batch-mini-batch-stochastic-gradient-descent-7a62ecba642a [Ziyaret Tarihi: 30 Haziran 2021].

Peixoto, B., Lavi, B., Bestagini, P., Dias, Z., Rocha, A., 2020, Multimodal Violence Detection in Videos, *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-2020*, 2957-2961.

Peixoto, B., Avila, S., Bestagini, P., Dias, Z., Rocha, A., 2018, Breaking down violence: A deep-learning strategy to model and classify violence in videos, *International International Conference on Availability, Reliability and Security, ARES-2018*, 1-7.

Rennison, M. and Welchans, S, 2018, Intimate Partner Violence, *Bureau of Justice Statistics*.

Rumelhart, D., Hinton, G. and Williams, R., 1986, Learning representations by back-propagating errors, *Nature*, 323, 533-536.

Samadzadeh, A., Far, F., Javadi, A., Nickabadi, A., Chehreghani, M., 2021, Convolutional Spiking Neural Networks for Spatio-Temporal Feature Extraction, *arXiv:2003.12346*.

Sarkar, D., 2018, A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning [online], Towards data science, https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a [Ziyaret Tarihi: 30 Haziran 2021].

Sharma, S., 2017, Activation Functions in Neural Networks [online], Towards data science, https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 [Ziyaret Tarihi: 30 Haziran 2021].

Shi, C., Zhang, Z., Zhang, W. and Xu, Q., 2022, Learning Multiscale Temporal–Spatial–Spectral Features via a Multipath Convolutional LSTM Neural Network for Change Detection With Hyperspectral Images, *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1-16.

Shi, X. et al., 2015, Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, *arXiv:1506.04214*.

Soliman, M., Kamal, M, El-Massih, M., Mostafa, Y., Chawky, B., Khattab, D., 2019, Violence Recognition from Videos using Deep Learning Techniques, 9*th International Conference on Intelligent Computing and Information Systems, ICICIS-2019*, 80-85.

Sudhakaran, S., Lanz, O., 2017, Learning to detect violent videos using convolutional long short-term memory, *14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS-2017*, 1-6.

Sumon, S., Goni, R., Hashem, N., Shahria, T., Rahman, R., 2020, Violence Detection by Pretrained Modules with Different Deep Learning Approaches, *Vietnam Journal of Computer Science*, 7(1), 19-40.

Svozil, D., Kvasnicka, V., Pospichal, J.,1997, Introduction to multi-layer feed-forward neural networks, *Chemometrics and Intelligent Laboratory Systems*, 39(1), 43-62.

Tanberk, S. (2020), "A New Approach Using Deep Learning Methodologies from Human Activity Recognition to Robot Grasping", Doktora Tezi, *Lisansüstü Eğitim Enstitüsü*, Doğuş Üniversitesi.

Tang, X., Yan, Z., Peng, J., Hao, B., Wang, H., Li, J., 2021, Selective spatiotemporal features learning for dynamic gesture recognition, *Expert Systems with Applications*, 168.

Traore, A., Akhloufi, M., 2020, Violence Detection in Videos using Deep Recurrent and Convolutional Neural Networks, *IEEE International Conference on Systems, Man, and Cybernetics, SMC-2020*, Toronto, 154-159.

Ullah, F., Ullah, A., Muhammad, K., Haq, I., Baik, S., 2019, Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network, *Sensors*, 19(11).

Xing, Y., Caterina, G., Soraghan, J., 2020, A New Spiking Convolutional Recurrent Neural Network (SCRNN) With Applications to Event-Based Hand Gesture Recognition, *Frontiers in Neuroscince*, 14.

Yadav, A., Dewal, M., Anand, R. and Gupta, S., 2013, Classification of hardwood species using ANN classifier, *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG-2013*, 1-5.

Yalçın, M. (2018), "Human Activity Recognition Using Deep Learning", Yüksek Lisans Tezi, *Fen Bilimleri Enstitüsü*, İstanbul Teknik Üniversitesi.

Zhu, Q, He, Z., Zhang, T. and Cui, W., 2020, Improving Classification Performance of Softmax, *Applied Scinces*, 10.