



**DİJİTAL OYUNLARDA SİBER TERÖRÜN YAPAY ZEKÂ İLE TESPİTİ
VE ERİŞİME ENGELLENMESİ**

Ahmet Edip ÇAPANOĞLU

**DOKTORA TEZİ
ADLİ BİLİŞİM ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

MART 2024

Ahmet Edip ÇAPANOĞLU tarafından hazırlanan “DİJİTAL OYUNLARDA SİBER TERÖRÜN YAPAY ZEKÂ İLE TESPİTİ VE ERİŞİME ENGELLENMESİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Adli Bilişim Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Nursel YALÇIN

Bilgisayar ve Öğretim Teknolojileri Eğitimi, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Başkan: Prof. Dr. Ahmet KARAARSLAN

Elektrik-Elektronik Mühendisliği, Ankara Yıldırım Beyazıt Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Cemal KOÇAK

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Murat YILMAZ

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Muhammet Abdullah BÜLBÜL

Bilgisayar Mühendisliği, Ankara Yıldırım Beyazıt Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Tez Savunma Tarihi: 06/03/2024

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Aslıhan TÜFEKÇİ

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
 - Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
 - Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
 - Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
 - Bu tezde sunduğum çalışmanın özgün olduğunu,
- bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza

Ahmet Edip ÇAPANOĞLU

06/03/2024

DİJİTAL OYUNLARDA SİBER TERÖRÜN YAPAY ZEKÂ İLE TESPİTİ VE ERİŞİME ENGELLENMESİ

(Doktora Tezi)

Ahmet Edip ÇAPANOĞLU

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

MART 2024

ÖZET

Dijital oyunların eğitim, dikkat güçlendirme, strateji kurma, eğlenme, sosyalleşme gibi farklı faydalı amaçları olsa da içerdikleri şiddet, kumar, cinsellik, küfür, uyuşturucu gibi zararlı içeriklerle de kullanıcıları olumsuz yönde etkilemektedir. Dijital oyunlar içerisinde en yaygın ve tehlikeli içeriklerin başında şiddet gelmektedir. Özellikle küçük yaştan itibaren siber şiddete maruz kalan çocuklarda bilinçaltına yerleşen şiddet, ilerleyen yaşlarında hayatlarına olumsuz yön vermektedir. Bu şiddetin iletişim eksikliği, yalnızlık, depresyon, saldırganlık, suç işleme, intihar gibi birçok yönü yapılan çalışmalar ile gösterilmiştir. Şiddet içerikleri silah, bıçak, kavga, ateş, kan vb. görüntülerle dijital oyunlarda çok yaygın karşımıza çıkmaktadır. Bu şiddet unsurlarının gerçek zamanlı tespiti ve analizi problemlere erken müdahale etmemize olanak sağlayacağından oldukça önemlidir. Çalışmada ilk olarak dijital oyunlar içerisinde barınan şiddet unsurları incelenerek farklı oyunlardan elde edilen 6256 görüntüden oluşan bir şiddet veri seti oluşturuldu. Veri seti şiddet unsurlarından silah, bıçak, bomba, ateş, kan, kavga olmak üzere 6 farklı sınıfa ayrıldı. Daha sonra 6256 şiddet görüntüsü etiketlenerek 18704 adet şiddet içeriği derin öğrenme modellerinde kullanılmak üzere kaydedildi. Çalışma kapsamında gerçek zamanlı şiddet tespiti, analizi ve gerektiğinde erişim engellemesi yapan derin öğrenme tabanlı model geliştirildi. Geliştirilen şiddet tespit modelinin omurgası, görüntü özelliklerini daha hızlı çıkarabilmek ve daha hızlı nesne algılama yapabilmek için parametre ve katman sayısı azaltılarak hafifletildi. Geliştirilen yöntem altı farklı derin öğrenme yöntemi ile karşılaştırıldı. Test sonuçlarına bakıldığında geliştirilen yöntemin yüksek tespit hızı ve doğruluk performansı ile gerçek zamanlı dijital

oyunlar için kullanılabileceği gösterildi. Ayrıca geliştirilen yöntem gerçek zamanlı ekran analizi yapabildiğinden kullanıcının bilgisayarında kayıtlı görüntü, video, dijital oyun vb. görsellerden veya Youtube, Instagram, Facebook vb. sosyal medya içeriklerden şiddet analizi yapabildiği de gösterildi. Çalışmada son olarak tespit edilen şiddet unsurlarının analizi gerçekleştirildi. Tanımlanan üç şiddet eşik değeri ile bu şiddet eşik değerlerinden fazla şiddete maruz kalan kullanıcılar için kullanıcı uyarıları, ebeveyn uyarıları ve erişim engellenmesi gerçekleştirildi.

Bilim Kodu : 92432

Anahtar Kelimeler : Dijital oyunlarda şiddet, Derin Öğrenme, Silah Tespiti, Bıçak Tespiti, Bomba Tespiti, Kavga Tespiti, Kan Tespiti, Ateş Tespiti.

Sayfa Adedi : 87

Danışman : Doç. Dr. Nursel YALÇIN

DETECTION AND PREVENTION OF CYBER TERRORISM IN DIGITAL GAMES USING DEEP LEARNING

(PH. D. THESIS)

Ahmet Edip ÇAPANOĞLU

GAZİ UNIVERSITY
INSTITUTE OF INFORMATICS

March 2024

ABSTRACT

Despite their beneficial objectives, such as educational enhancement, attention empowerment, strategic skill development, entertainment, and facilitation of socialization, digital games pose significant adverse influences on users. These negative impacts stem from the inclusion of violence, gambling, obscenity, abuse, and drug-related content within these games. Notably, violence is among the most prevalent and harmful elements. It tends to embed itself in the subconscious of children who are exposed to cyber-violence, especially at early developmental stages, subsequently affecting their lives adversely in later years. Empirical studies have documented various outcomes of exposure to such violence, including diminished communication skills, loneliness, depression, aggression, criminal behavior, and increased suicide risk. In digital games, the portrayal of violence through weapons, knives, physical altercations, fire, and blood is widespread. Given the potential for intervention, the real-time detection and analysis of these violent elements are of paramount importance. This study embarks on identifying and categorizing violent elements in digital games. It constructs a dataset comprising 6,256 images depicting violence, which is classified into six classes based on the nature of the violent element: weapons, knives, bombs, fire, blood, and physical altercations. Subsequently, 6,265 images containing 18,704 instances of violence were annotated to train deep learning models. Within the ambit of this research, a model was developed capable of real-time detection, analysis, and, if deemed necessary, restriction of access to violent content. The model's architecture was optimized

by minimizing the number of parameters and layers, thus accelerating feature extraction and object detection processes. This novel method was evaluated against six alternative deep learning approaches. The comparative analysis revealed that the proposed model exhibits superior detection speed and accuracy performance, rendering it suitable for real-time application in digital gaming environments. Additionally, it was demonstrated that the developed method can perform real time screen analysis, allowing the user to conduct violence analysis from digitally stored images, videos, digital games etc., as well as from social media content on platforms such as YouTube, Instagram, Facebook, etc. The study also includes an analysis of detected violent content, wherein user and parental warnings are issued based on three predefined violence threshold levels, with access restrictions imposed on users exceeding these thresholds.

Science Code : 92432

Key Words : Deep Learning, Violence in digital games, Weapon Detection, Knife Detection, Bomb Detection, Fight Detection, Blood Detection, and Fire Detection.

Page Number : 87

Supervisor : Assoc. Prof. Dr. Nursel YALÇIN

TEŞEKKÜR

Doktora çalışmalarımın her aşamasında katkılarıyla beni motive eden, yol gösteren, akademik çalışmalarında her zaman bana destek olan değerli danışmanım Doç. Dr. Nursel YALÇIN'a, tez izleme sürecimde değerlendirmeler ve tavsiyeler ile tezimin şekillenmesine destek olan Prof. Dr. Ahmet KARAARSLAN ve Doç. Dr Cemal KOÇAK hocalarıma, çalışmalarına ilişkin fikir alışverişinde bulunduğum arkadaşlarıma, hayatım boyunca desteklerini eksik etmeyen değerli annem, babam, kardeşlerim, sevgili eşim ve çocuklarıma teşekkürlerimi sunarım.



İÇİNDEKİLER

	Sayfa
ÖZET	v
ABSTRACT	vii
İÇİNDEKİLER	x
ÇİZELGELERİN LİSTESİ	xii
ŞEKİLLERİN LİSTESİ	xiii
GÖRÜNTÜLERİN LİSTESİ	xv
SİMGELER VE KISALTMALAR.....	xvi
1. GİRİŞ.....	1
1.1 DİJİTAL OYUNLARIN OLUMLU VE OLUMSUZ YÖNLERİ.....	2
1.2 PROBLEM.....	3
1.3 TEZİN KONUSU VE AMACI.....	4
1.4 YAŞ GRUPLARI.....	5
1.5 TANIMLAR.....	5
1.6 SINIRLILIKLAR.....	6
2. YAKLAŞIM	7
2.1 DERİN ÖĞRENME.....	7
2.1.1 Derin Öğrenme Tarihsel Gelişimi	7
2.1.2 Derin Öğrenme Yapısı.....	9
2.1.2.1 Giriş katmanı	11
2.1.2.2 Evrişim katmanları.....	12
2.1.2.3 Aktivasyon fonksiyonları	12
2.1.2.4 Havuzlama katmanları.....	14
2.1.2.5 Düzleştirme katmanı	15
2.1.2.6 Tam bağlantılı katmanlar.....	15
2.1.2.7 Düğüm silme katmanı	15

2.1.2.8 Kayıp fonksiyonu.....	16
2.1.2.9 Dolgulama	17
2.1.3 Nesne Tanıma Algoritmaları.....	18
2.1.3.1. YOLO.....	19
2.1.3.2. YOLO V4	19
2.1.3.3. YOLO V4 Tiny	20
2.1.3.4 YOLO V5	20
2.1.3.5 YOLO V7	21
2.1.3.6 YOLO V8.....	22
2.1.3.7 Faster R-CNN	24
2.1.3.8 SSD	25
3. YÖNTEM	27
3.1 VERİ SETİ OLUŞTURMA.....	27
3.2 VERİ AYIKLAMA	28
3.3 GÖRÜNTÜ ETİKETLEME.....	28
3.4 VERİ ÇOĞALTMA	30
3.5 KONFIGÜRASYON	31
4. DENEYSEL SONUÇLAR	33
4.1 PERFORMANS KRİTERLERİ	33
4.2 PERFORMANS SONUÇLARI.....	35
4.2.1 YOLO V4 Tiny Deneysel Sonuçları	35
4.2.2 YOLO V5s Deneysel Sonuçları	37
4.2.3 YOLO V7 Deneysel Sonuçları.....	39
4.2.4 SSD Deneysel Sonuçları	41
4.2.5 Faster R-CNN Deneysel Sonuçları.....	42
4.2.6 YOLO V8n ve YOLO V8 Geliştirilen Model Deneysel Sonuçları.....	44
5. ŞİDDET ANALİZİ	53
6. SONUÇ VE ÖNERİLER	57
KAYNAKLAR	59

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1 YOLOV4 Tiny deneysel sonuçları	37
Çizelge 4.2 YOLOV5 modeli deneysel sonuçları.....	38
Çizelge 4.3YOLO V7 modeli deneysel sonuçları.....	41
Çizelge 4.4 SSD modeli deneysel sonuçları	42
Çizelge 4.5 Faster R-CNN modeli deneysel sonuçları.....	44
Çizelge 4.6 YOLO V8n modeli deneysel sonuçları.....	48
Çizelge 4.7 Geliştirilen YOLO V8n modeli deneysel sonuçları.....	49
Çizelge 4.8 Test edilen modellerin performans değerleri	49

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1 Dünya genelinde yıllara göre dijital oyun oynayan kullanıcı sayıları.....	1
Şekil 2.1 Sinir hücresi yapısı	10
Şekil 2.2 Evrişimli sinir ağ mimarisi.....	11
Şekil 2.3 Evrişim katmanında uygulanan bir filtre örneği	12
Şekil 2.4 Farklı aktivasyon fonksiyonlarının istatistiksel sonuçları	14
Şekil 2.5 2x2 Maksimum ve ortalama havuzlama örneği	15
Şekil 2.6 a) Standart sinir ağı b) Düğüm silinmiş sinir ağı.....	16
Şekil 2.7 Dolgulama yöntemleri	17
Şekil 2.8 Nesne tanıma algoritmaları	18
Şekil 2.9 YOLO sürümlerinin yıllara göre gelişimi.....	19
Şekil 2.10 YOLO V7 de sunulan birleştirme tabanlı model	21
Şekil 2.11 YOLO V8 mimarisi.....	23
Şekil 2.12 YOLO V8'in performans karşılaştırılması	24
Şekil 2.13 Faster R-CNN ağ yapısı.....	25
Şekil 2.14 SSD ağ yapısı	25
Şekil 3.1 Nesne tanıma modellerinin genel akış şeması	27
Şekil 3.2 Veri setinde kullanılan görüntülerden örnekleri	28
Şekil 3.3 LabelImg görüntü etiketleme işlemi.....	29
Şekil 3.4 Veri setinde etiketlenen şiddet içeriklerinin miktarı	29
Şekil 3.5 YOLO V8n eğitim aşamasında mozaik veri çoğaltma örneği	30
Şekil 4.1 YOLO V4 Tiny eğitim grafiği	36
Şekil 4.2 YOLO V5s eğitim sonuçları	37
Şekil 4.3 YOLO V5 Eğitim sürecine ait görüntüler	38
Şekil 4.4 YOLO V7 eğitim sonuç grafikleri.....	39
Şekil 4.5 YOLO V7 karışıklık matrisi	40
Şekil 4.6 YOLO V7 modeli CPU ortamında gerçek zamanlı çalışma örneği	41
Şekil 4.7 Faster R-CNN modeline ait epoch – kayıp fonksiyon grafiği	43
Şekil 4.8 Faster R-CNN modeline ait epoch – mAP grafiği	43
Şekil 4.9 YOLO V8 modeli ağın derinlik, genişlik ve kanal parametre değerleri.....	44
Şekil 4.10 a) YOLO V8n omurgası b) Geliştirilen YOLO V8 model omurgası ...	45

Şekil 4.11 a) YOLO V8n kesinlik grafiği b) Geliştirilen model kesinlik grafiği....	45
Şekil 4.12 a) YOLO V8n duyarlılık grafiği b) Geliştirilen model duyarlılık grafiği .	46
Şekil 4.13 a) YOLO V8n F1 değeri grafiği b) Geliştirilen model F1 değeri grafiği ..	46
Şekil 4.14 YOLO V8n karışıklık matrisi.....	47
Şekil 4.15 YOLO V8 geliştirilen model karışıklık matrisi.....	48
Şekil 5.1 Modelin analiz ekran görüntüsü.....	53
Şekil 5.2 Şiddet analizi aktivite diyagramı	54
Şekil 5.3 Birinci şiddet eşik değeri için uyarı mesajı.....	55
Şekil 5.4 Aile bilgilendirme e-postası	55



GÖRÜNTÜLERİN LİSTESİ

Görüntü	Sayfa
Görüntü 4.1 Modelin doğru tespit örnekleri	50
Görüntü 4. 2 Modelin başarısız tespit örnekleri	50
Görüntü 4. 3 Modelin farklı oyunlardaki başarılı tespit örnekleri	51
Görüntü 4.4 Modelin gerçek dünya görüntülerinden doğru tespit örnekleri	52
Görüntü 4.5 Modelin sosyal medya görüntülerinden doğru tespit örnekleri.....	52



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

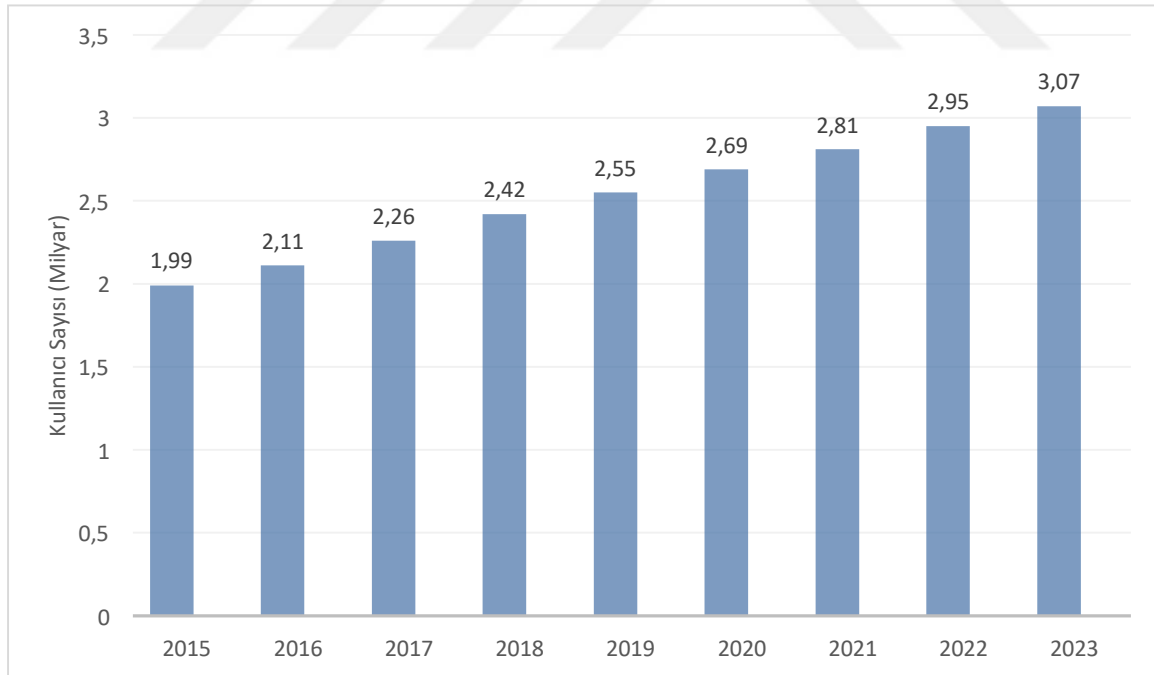
Simgeler

Açıklamalar

YSA	Yapay Sinir Ağı
ESA	Evrişimli Sinir Ağları
RNN	Tekrarlayan Sinir Ağları
LSTM	Uzun-Kısa Vadeli Hafıza Ağları
YOLO	Sadece Bir Kez Bak
SSD	Tek Atış Nesne Tespiti
PEGI	Pan Avrupa Oyun Bilgilendirme
GPU	Grafik İşlem Birimi
CPU	Merkezi İşlem Birimi
GAN	Çekişmeli Üretici Ağlar
SPPF	Hızlı Uzaysal Piramit Havuzlama
DP	Doğru Pozitif
DN	Doğru Negatif
YP	Yanlış Pozitif
YN	Yanlış Negatif

1. GİRİŞ

Dijital oyun tanım olarak; bilgisayar, konsol, cep telefonu, tablet, atari vb. elektronik cihazlarla bir yazılıma karşı veya internet üzerinden dünya genelinde bir ağa bağlanarak yüzlerce insanla aynı anda oynanabilen hatta üretilen sanal cihazlarla oyunun bir parçası haline gelecek kadar ilerleyen dijital araç ve ortamlarda oynanan tüm oyunları içermektedir [1] [2]. Donanım ve yazılım alanlarındaki teknolojik gelişmelerle internetin hayatın her alanına girmesi, dijital ortamların büyüyerek yoluna devam etmesi dijital oyun tanımının da genişletilmesini zorunlu kılmaktadır. Bu kapsamda dijital oyunlar yeni dijital dünyada etkileşimlik, sanallık, değişkenlik, modülerlik, dijitallik ve birçok yeni özellikleri barındıran bir iletişim ortamına dönüşmüş durumdadır[3]. Bu yeni özellikler ile daha fazla ilgiyi üzerinde toplayan dijital oyunlar gün geçtikçe artan kullanıcı sayıları ile dikkat çekmektedir. Öyle ki dünya genelinde yaklaşık 3 milyar insanın aktif bir dijital oyuncu olduğu rapor edilmiştir[4].



Şekil 1.1 Dünya genelinde yıllara göre dijital oyun oynayan kullanıcı sayıları

Dijital oyunların oyuncular üzerinde etki bırakıp davranış biçimlerini etkileyebileceği üzerine çeşitli çalışmalar yapılmıştır[5]. Bu etkilerin kullanım durumlarına göre olumlu ve olumsuz yönde kullanıcılara yansıyabileceği belirtilmiştir.

1.1 Dijital Oyunların Olumlu ve Olumsuz Yönleri

Dijital oyunların günümüzde en faydalı kullanıldığı alanlardan bir tanesi dijital oyun tabanlı eğitim sistemleridir [6]. Özellikle küçük yaşta çocukların eğlendirerek öğrenme yöntemi ile daha verimli öğrendikleri gözlemlenmiştir[7]. Dijital oyun tabanlı öğrenmenin işbirlikçi öğrenme ortamı sunarak, içsel motivasyonu artırdığı belirtilmiştir [8]. Karar verme ve hızlı düşünme becerisini artırdığı da vurgulanmıştır[9]. Öğrenme sürecine aktif katılım sağladığı, çocuklarda problem çözme ve görsel dikkat becerilerinin de gelişimine olumlu etkilerinin olduğu belirtilmiştir [10]. Ayrıca otistik veya diğer özel ihtiyaçları olan çocuklara beceri öğretme, dikkat eksikliği bozukluğu olan çocuklara beyin dalgası geri bildirimi kullanarak dikkat odaklamayı sağlama ve fiziksel rehabilitasyona yardımcı olmak için araştırma veya ölçüm aracı olarak da kullanılmaktadır [11].

Dijital oyunların strateji geliştirme, zekâ güçlendirme, stres atma, sosyalleşme gibi birçok olumlu yanlarının yanı sıra kullanıcıları çok çeşitli yönlerden tehdit eden içeriklerle karşılaşmak mümkündür. Dijital oyunların odaklanma, dikkat dağınıklığı gibi problemlere yol açtığı için okul ve çalışma hayatında performans düşüklüğüne neden olduğu yapılan çalışmalarda incelenmiştir [12] [13]. Ayrıca nefret, cinsellik, islamifobi gibi içerikleri bilinçaltı mesajlar ile kullanıcıları sunularak etkilediği de görülmektedir [14] [15] [16]. Irkçılık nekro politika ve biyopolitik söylemi de dijital oyunlarda kullanıcıları tehdit eden unsurlar arasında sayabiliriz[17, 18]. Dijital oyunlarda ki sayılabilecek tehditlerin başında ise şiddet gelmektedir. Şiddet içerikli oyunlar küçük yaştan itibaren oynanmakta ve bu şiddet içerikleri hayatın bir parçası haline gelerek içselleştirilmektedir [19]. Şiddetin içselleştirilmesindeki en önemli sebeplerden birisi de dijital oyunların uzun süre oynanarak oyuna bağımlı kalınmasıdır[20]. Zamanla hayatın normal akışı içerisinde varmış gibi bilinçaltına yerleşen şiddet saldırganlığı artırmaktadır[21]. Saldırgan bir yapıda olan insanlar ise çeşitli suçlar işletecek duruma gelebilmektedir [22]. Şiddet içerikli dijital oyunların oynanma süresi ile günlük hayatta karşılaşılan sadistlikle yani kişilerin başkalarına şiddet uygulama isteğinin artırması arasında güçlü bir bağ olduğu da belirtilmiştir [23]. Dijital oyunların bir diğer olumsuz etkisi ise gençlerin dış dünya ile sosyal bağını kopararak yalnızlaşmasıdır[24]. Yalnızlığa sürüklenen gençlerin anksiyete bozukluğu ve daha kolay depresyona girdiği gözlemlenmiştir[25]. Yalnızlaşan ve depresyona giren gençlerin ise intihar girişimde bulunabileceği belirtilmiştir [26]. Bunun sonuçlarından bir tanesi Mavi Balina örneğidir. Mavi Balina oyun içerisinde ağına düşürdüğü gençleri şiddet, yalnızlık, depresyon gibi birçok psikolojik durumun içerisine sokup sanal tutsak haline getirerek

intiharla sonuçlanan bir sürecin içerisinde dâhil etmektedir. Bu tarz oyunlara siber terör denilmesinin sebepleri arasında, dünya genelinde etkili olması, arkasında savunmasız gençleri intihara teşvik eden bir grubun olması, şiddet içermesi, kişileri esir alması gibi etkenler yer almaktadır. Mavi balina örneğine bakacak olursak oyunun arkasında emir veren grup yöneticileri günlük görevleri kullanıcılara bildirmekte ve 50 gün içerisinde kişilerin bu görevleri tamamlamasını istemektedir. 50. günün sonunda ise gençlerden intihar etmesi emredilmektedir [27]. Öyle ki TBMM araştırma komisyonu 2020 yılında ülkemizde yaklaşık 150 civarında gencin intiharıyla bağlantılı olduğu düşünülen ve sanal ölüm tuzağı olarak bilinen Mavi Balina oyununun sosyal medya üzerinden hızla yayıldığını, daha fazla can kaybının olmaması için gerekli tedbirler alınması gerektiğini, çocukların ve ailelerin bu gibi intihara sürükleyen, psikolojilerine zarar veren ve şiddete yönlendiren dijital oyunlara karşı bilinçlendirilmeleri gerektiği belirtilmiştir [28].

Ohio Eyalet Üniversitesi tarafından 8-12 yaş arası 220 çocuk üzerinde yapılan bir araştırmada şiddet içerikli dijital oyunları oynayan çocukların gerçek hayatta silahlarla ilgilenme olasılığının daha yüksek olduğunu belirtilmiştir[29]. Suudi Arabistan'ın Riyad şehrinde üniversite öğrencileri üzerinde yapılan bir çalışma da ise gençlerde siber şiddet davranışını etkileyen faktörler ve intiharla ilişkisi araştırılmıştır. Araştırmada dijital oyunlar sosyal medya ve teknolojik farkındalıkların siber şiddete yönelik davranışsal niyet ve intihar üzerindeki etkisinin olduğu gösterilmiştir [30]. Amerika Birleşik Devletleri'nde 15624 lise öğrencisi üzerinde yapılan bir başka çalışmada ise medya ve dijital oyunu oynama sürelerinin aşırı olması gençlerin şiddet içeriklerine daha fazla maruz kalabileceğinden gerçek hayatta kişilerarası şiddet deneyimlerine ve intihar riskine katkıda bulunabileceği belirtilmiştir[31]. Benzer şekilde 228 üniversite öğrencisi üzerinde yapılan bir çalışmada şiddet içerikli dijital oyunları saatlerce oynayan öğrencilerin kendilerine zarar verme ve intihara daha yatkın olduğu gösterilmiştir [32].

Dijital oyunların kullanıcılara etkisini tetikleyen yan faktörler ise yaş, cinsiyet, aile ortamı, kişilik, sosyoekonomik durum, toplum yapısı vb. birçok faktör gösterilmektedir [33].

1.2 Problem

Dijital oyunlara gençler ve çocuklar oldukça ilgi duymaktadır. Çocuklar ilerleyen yaşlarına adım atarken ebeveynlerin kontrolü azalmakta ve zararlı birçok farklı içeriklerdeki oyunlara yönelebilmektedirler[34]. Şiddet, savaş, suç, soygun, dövüş gibi içeriklerin oynandığı dijital oyunlar zamanla bu içeriklerin kişilerde normalleşmesine neden olarak kavga, kin,

düşmanlık, saldırganlık, depresyon, intihar gibi davranışları tetiklemektedir [35] [36] [37] [38] [39] [40] . Ayrıca kişiler, oyun bağımlılığı ile sosyal yaşamdan uzaklaşarak yalnız kalma eğilimi gösterebilmektedir[41]. Yalnızlaşan gençler zamanla hayatı anlamsız bularak kötü niyetli kişilerin yönlendirmesi ile intihar girişimde bulunabilmektedirler[42]. Bu yüzden dijital oyunlar içeriğinin analiz edilmesi çok önemlidir. İçerisindeki şiddet unsurlarının tespiti toplumun ileride karşılaşılabilecek problemlere de erken müdahale imkânı sunmaktadır. Ayrıca çocukların ne tür oyunlar oynadığı hakkında da genel bilgi sahibi olmamıza olanak sağlamaktadır. Dijital oyunlarda kullanıcıları olumsuz yönde etkileyecek bu zararlı içeriklerin neler olduğuna dair birçok çalışma yapılmıştır. Fakat içerisinde ki zararlı içeriklerin tespiti, analizi, sınıflandırılması ve erişim engeli gibi çalışmalar mevcut değildir.

1.3 Tezin Konusu ve Amacı

Gençlerin ve çocukların karşılaşılabileceği risk ortamlarından birisi de dijital oyun dünyasıdır. Dijital oyunlardaki riskler gençleri ve çocukları fiziksel, duygusal, ahlaki ve zihinsel olarak olumsuz bir biçimde etkileyebilmektedir. Kişilere veya farklı nesnelere zarar vermeyi, yaralamayı veya yok etmeyi amaçlayan fiziksel bir aktivite olarak tanımlanan şiddet, dijital oyun oynayan çocuk ve gençlerin psikolojik davranışlarını etkileme potansiyeli nedeniyle oldukça önemlidir. Özellikle ateşli silahların veya bıçakların kullanıldığı oyunlarda gençlerin bu suç unsurlarını benimsemesi ve şiddeti içselleştirmesi toplumda suç oranının artmasına sebep olmaktadır[43]. Dijital oyun içerisinde kavga, ölüm ve kanlı sahneler ile vahşiliğinin normalleşmesi, şiddetin sosyal hayata yansması, kültürel şiddeti hayata geçirebilmektedir[44].

Şiddet içeren oyunların erken tespiti ve gerekli önlemlerin alınması toplumun geleceği ve güvenliği için büyük önem taşımaktadır. Mevcut durumda ebeveyn gözetim ve kontrolü yeterli olmadığı için bu tehlikelere çözümler de yeterli değildir. Şiddet içeriklerinin tespiti insan eli ile yapılması ise zaman ve performans olarak etkili sonuçlar vermez. Bu tarz problemlerde yapay zekâ kullanarak şiddet içeriklerinin tespitini sağlamak daha verimli sonuçlar elde etmemizi sağlar.

Sonuç olarak çalışma kapsamında şiddet içeren dijital oyunlarda kullanılmak üzere gerçek zamanlı silah, bıçak, bomba, kan, yangın ve kavga tespiti yapabilen derin öğrenme tabanlı bir model geliştirilmiştir. Geliştirilen model altı farklı derin öğrenme modeli ile

karşılaştırılarak dijital oyunlar için kullanılabileceği gösterilmiştir. Ayrıca çalışmamızda kullanıcıları etkileyen şiddet riskleri incelenerek şiddetin analizi yapılmıştır. Belirlenen eşik değerleri üzerinde şiddete maruz kalan kullanıcılar için önce kendisi daha sonra ebeveyn bilgilendirmesi mesaj ve mail yoluyla yapılmaktadır. Belirlenen son eşik değerinin de aşıldığı durumlarda ise erişim engellemesi yapılarak kullanıcının dijital oyun oynaması sonlandırılmaktadır.

1.4 Yaş Grupları

PEGI, dijital oyunların güvenli ve kullanımını desteklemek için yaş sınıflandırması, yaş derecelendirmesi ve oyun içeriğinin belirli yaştaki oyuncular için uygunluğunu denetleyen Avrupa'nın 38 ülkesinde kullanılan bir sistemdir [45]. Çalışmamız PEGI standartlarına uygun 16 yaş ve altı ebeveyn kontrolüne ihtiyaç duyan çocuklar için yapılmıştır.

1.5 Tanımlar

Şiddet, siber şiddet ve siber terör kavramları problem türüne göre farklı tanımlar içeren geniş bir konudur. Çalışma kapsamında ele alınan tanımları ise şu şekildedir:

Şiddet: Tür Dil Kurumun genel tanımına göre kaba kuvvet kullanma, kaba güç anlamına gelmektedir[46]. Bireysel ve yapısal olarak tanımlanan şiddet cinayet, yaralama, rehin alma, kan, kavga, yakma, işgal, savaş, terör, çete ve mafya gibi içeriklerle karşımıza çıkmaktadır [47]. Bu şiddet içeriklerinin insan beyni tarafından öğrenebilir olması ise dikkat edilmesi gereken bir noktadır. Yapılan bir çalışmada suçlu örneklerdeki deneklerin %80'e varan oranı çocukluklarında veya ergenlik döneminde şiddete tanık olduklarını bildirmişlerdir [48].

Siber Şiddet: İnternet ve dijital ortamlarda kullanıcıları şiddet içeriklerine maruz bırakarak psikolojik, sosyal veya fiziksel zarar görmesine neden olan şiddet türüdür[49]. Siber şiddetin gerçek dünyadaki şiddete benzer düzeyde korku ve sıkıntıya yol açabileceği belirtilmiştir[50]. Dijital oyunlarda siber şiddet yazılı, görsel, işitsel veya psikolojik olarak kullanıcıları etkileyebilmektedir[3]. Çalışmamız kapsamında görsel içeriklerde bulunan şiddet ele alınmıştır.

Siber Terör: Siber şiddetin etkisinde kalarak şiddeti hayatın normal akışı içerisinde kullanarak insanlara veya kendine fiziki zarar verme hatta ölüm ve intihara kadar sonuçlanan durumlar için kullanabiliriz. Fransa’da Julien Barreaux isimli gencin Counter-Strike oyunu sırasında oyun karakterini bıçak ile öldüren oyuncuyu altı ay boyunca izledikten sonra göğsünden bıçaklayarak öldürmek istemesini siber teröre örnek olarak gösterilebilir[51]. Siber terörün tanımları arasında kişilere veya mallara karşı şiddete yol açan, korku yaratmaya yetecek kadar zarar veren bir saldırı türü olduğu belirtilmiştir[52]. Siber terörün sonuçları arasında kişilere şiddet uygulanması, acı çekmesi, ciddi şekilde yaralanması veya ölümle sonuçlanması şeklinde gösterilmiştir[53].

1.6 Sınırlılıklar

Dijital oyun dünyası çeşitlilik bakımından binlerce oyun içerdiğinden bütün dijital oyunlardan görüntüler içeren bir veri seti hazırlanması oldukça zordur. Çalışmamız gerçek dünya görüntülerine benzer görüntüler içeren ve dünya üzerinde yaygın oynanan dijital oyunlar üzerine gerçekleştirilmiştir. Çizgi film, animasyon vb. görüntüleri içeren dijital oyunlar bu aşamada çalışma kapsamına alınmamıştır.

Dijital oyunlarda barınan şiddet içeriklerinin başarılı bir şekilde tespit edilmesi için bu şiddet içeriklerinin derin öğrenme modellerine öğretebilecek, genel kullanıma açık bir veri seti bulunmamaktadır. Bu kapsamda veri seti çalışma sürecinde oluşturulmuştur.

2. YAKLAŞIM

Tez çalışmasının ikinci bölümünde derin öğrenme yapısı detaylı incelenmiştir. Evrişimli Sinir Ağları (ESA) temelli yaklaşımlar araştırılarak nesne tanıma modelleri üzerinde durulmuştur.

2.1 Derin Öğrenme

Derin öğrenme, temelinde makine öğrenmesi ve yapay sinir ağları gibi alanları barındıran bir kavramdır. Derin Öğrenme ile ilgili çalışmalar eski yıllara dayansa da son yıllarda geliştirilen teknoloji, alt yapı, veri miktarının artması ve bu verilerin internet ortamına aktarılması, veri transferini çok daha kolay hale getirerek bu alanda yapılan çalışmaların gün geçtikçe artırmasına olanak sağladı. Derin öğrenme, birden fazla katmana sahip olan, her katmanda daha yetenekli özellikler öğrenilerek çok fazla özelliği girdi setinden çıkarma yeteneğine sahip karmaşık yapay sinir ağları kullanılarak makine öğrenmesi işleminin gerçekleştirilmesine için kullanılmaktadır [54]. Oluşturulan çok derin katmanlı modelleri hesaplayabilecek grafik işlemcilerin geliştirilmesi ve bu modelleri uygulayabilecek büyük veri setlerinin oluşturulup erişime açılması görüntü işleme[55], siber güvenlik[56], doğal dil işleme[57], bankacılık[58], sağlık[59], eğitim[60], reklamcılık[61] ve birçok önemli alanda etkili sonuçların elde edilmesine imkân sağlamıştır.

2.1.1 Derin Öğrenme Tarihsel Gelişimi

Derin öğrenmenin tarihsel gelişimine kısaca bakacak olursak 1943 yılında McCulloch-Pitts tarafından ilk yapay sinir hücreleri geliştirilmiş [62] ve Walter Pitts tarafından geliştirilen bu hücrelerin matematiksel modeli oluşturularak yapay sinir ağlarının temeli atılmıştır [63]. Maccharty.J 1955 yılında yapay zeka terimini kullanan ilk araştırmacı, 1956 yılında yapılan Dartmouth Conference ise ilk yapay zeka konferansı olmuştur [64]. 1957 yılında Rosenblatt .F yapay zekâ temel yapı taşı olan algılayıcıların tanımını ortaya koymuştur [65]. 1962 yılına gelindiğinde Rosenblatt tarafından tek katmanlı sinir ağı modeli tasarlanmıştır [66]. Minsky ve Papert ise yazdıkları Perceptrons kitabında tek katmanlı ağların yeterli olmadığına vurgu yaparak çok katmanlı derin ağların mümkün olabileceğinden bahsetmişlerdir [67].

1986 yılında Hinton.G tarafından geriye yayılım algoritması yayınlanmıştır [68]. 1989'da George Cybenko, yalnızca bir gizli katmana sahip bir sinir ağının her zaman çok değişkenli ve sürekli bir fonksiyona tahmin yeteneği olduğunu kanıtlamıştır. Çok katmanlı algılayıcılarla herhangi bir fonksiyonun temsil edilebileceğini göstererek tam bağlantılı ağların öğrenme mimarisine sahip olduğu hipotezine destek vermiştir [69]. 1997 yılında ise dünya satranç şampiyonu Garry Kasparov, IBM tarafından oluşturulan Deep Blue ya yenilerek dünya gündemine oturmuştur. Deep Blue, Grandmaster oyun veritabanını etkili bir şekilde kullanarak, çoklu paralel seviyelere sahip ve tek çipli etkili arama motoru sayesinde saniyede 2 - 2.5 milyon pozisyon arasında arama hızlarına ulaşabilen derin düşünme yapısına sahiptir. Oyun sonunda veri tabanını güncelleştirme ve iyileştirmesi sayesinde başarısını artırabilmektedir [70].

1998 yılında LECun.Y derin öğrenmenin de temeli olan Evrişimli Sinir Ağları ile rakamları sınıflandıran bir model geliştirmiştir. Çek, senet, vb. yerlerdeki el yazılarının tanımlanmasına yönelik olan bu çalışma ESA'nın, minimum ön işlemle doğrudan piksel görüntülerinden görsel desenleri tanımak için tasarlanmıştır [71].

2000 li yıllara kadar ki dönemlerde sınırlı veri ve donanımsal yetersizlikler gibi sebeplerden derin öğrenme yerine makine öğrenmesi yaygın olarak kullanılmıştır. 2000 li yıllardan sonra daha güçlü Grafik İşlem Birimleri (GPU), gelişimleri ile birlikte yeniden çalışmaların yaygınlaştığı bir alan haline gelmiştir. 2009 yılında Fei.Li ve ekibi dünyanın en büyük görüntü veri seti olan Imagenet'i ücretsiz olarak yayınlamıştır. Bu sayede birçok çalışma için ihtiyaç duyulan veriler sağlandığından yapay zekânın da önü açılmıştır [72]. 2012 yılındaki ImageNet yarışmasını kazanan AlexNet derin öğrenme adına dünyada büyük ilgi odağı haline gelmiştir [73]. Bu dönemde derin öğrenme mimarileri büyük başarılar elde etmiştir.

2014 yılında Min Lin ve arkadaşları giriş verileri soyutlamak için daha karmaşık yapılara sahip mikro sinir ağları inşa ederek “Ağ İçinde Ağ” modelini tanıtmışlardır [74]. 2014 yılında Szegedy ve arkadaşları “Başlangıç” modeli derin evrişimli sinir ağı mimarisini kullanarak GoogleNet modelini önermişlerdir. Bu mimarinin temel özelliği, ağ içindeki değişkenleri daha verimli kullandığı gösterilmiştir. Böylelikle mevcut hesaplama kapasitesi ile ağı derinliğini ve genişliğini artırmaya olanak sağlanmıştır. 2014 yılında GoogLeNet, ImageNet yarışmasında birincilik elde etmiştir [75]. 2014 yılında Ian Goodfellow ve ekibi Çekişmeli Üretici Ağ (GAN) modelini tasarlayarak yapay veri üretme ile bu alanda yapılacak önemli çalışmaların temelini atmıştır [76]. Simonyan ve arkadaşları 2015 yılında VGG 16 mimarisini tasarlayarak ImageNet veri seti üzerinde 3×3 boyutunda evrişimli

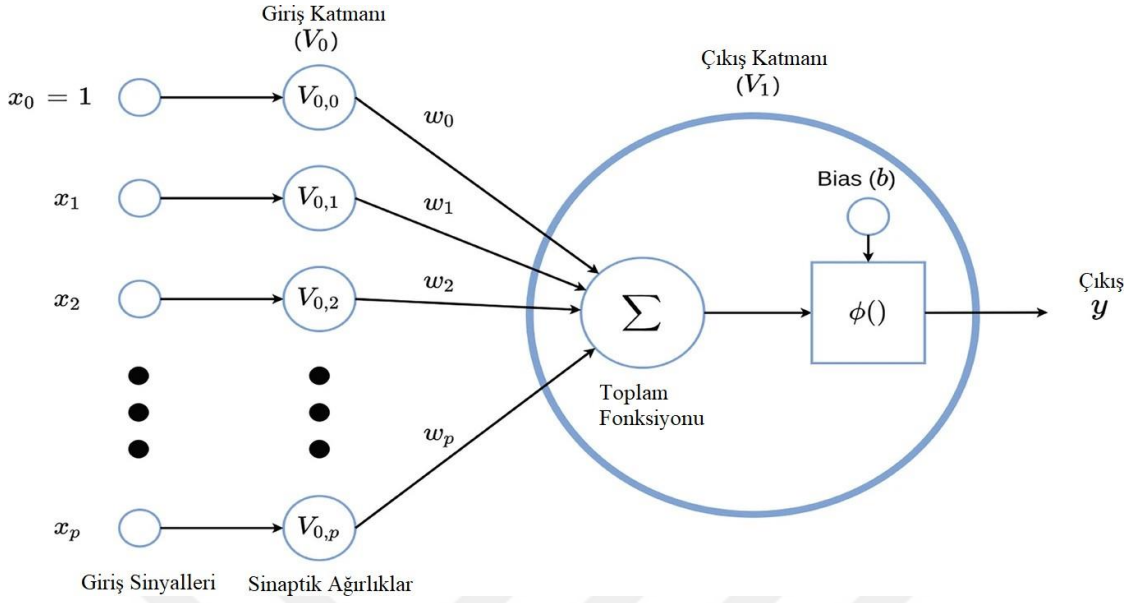
filtrelere sahip bir mimari ile derinliği artan ağlarda kapsamlı bir değerlendirme yapmışlardır [77]. 2016 yılında Google DeepMind tarafından tasarlanan ALPHAGO yazılımı GO oyununda dünya şampiyonunu yenmiştir [78]. He ve arkadaşları daha önce kullanılan yöntemlerden farklı olarak çok daha derin ağların eğitimi kolaylaştırmak için "Artık Öğrenme Çerçevesi" adında bir yöntem geliştirerek ResNet'i tasarlamışlardır [79]. 2017 yılında Hinton.G ve ekibi kapsül ağları tanıtmışlardır [80]. 2022 yılında OpenAI sohbet yoluyla insan benzeri konuşma cevapları üretme odaklı olarak tasarlanmış ChatGPT modelini tanıtılmıştır [81].

Derin öğrenme alanında yapılan bu başarılı çalışmaların ardından günümüzde çalışma yapılan alan sayısında ciddi artışlar meydana gelmektedir. Buda çalışma yapılmayan diğer alanlara da cesaret vererek derin öğrenmenin farklı çalışmalarda kullanılmasına olanak sağlamaktadır.

Derin öğrenme içerisinde problemin türüne göre Evrişimli Sinir Ağları (ESA), Tekrarlayan Sinir Ağları (RNN), Uzun-Kısa Vadeli Hafıza Ağları (LSTM), Derin Oto Kodlayıcılar (Autoencoders), Çekişmeli Üretici Ağlar vb. birçok farklı mimari geliştirilmiştir. ESA özellikle görüntü işleme alanında önemli başarılar elde ettiğinden tez kapsamında ESA tabanlı nesne tanıma modelleri üzerinde çalışmalar yapılmıştır. Evrişim ve havuzlama katmanları ile veri üzerinde özelliklerin hiyerarşik olarak çıkarılmasına olanak tanıyan bu mimari, özellikle görüntü işleme gibi alanlarda büyük veri setleri ile eğitildiğinde yüksek başarı oranları elde edilmektedir.

2.1.2 Derin Öğrenme Yapısı

Derin Öğrenme; görüntü, ses ve metin gibi verileri anlamlandırmaya yardımcı olan birçok doğrusal olmayan bilgi işleme katmanından yararlanarak özellik çıkarma, özellikleri dönüştürme, sınıflandırma gibi işlemlerden geçirebilen ve analiz etme yeteneğine sahip denetimli veya denetimsiz öğrenebilen, büyük ölçüde sinir ağlarına dayalı, yapay zekânın alt dalı olan bir makine öğrenme tekniğidir [82]. Bu teknik, insan beyninin yapısı ve işleyişi tarafından ilham alınarak tasarlanmıştır[83]. Derin öğrenme modelleri, girdi verilerinden özellikleri otomatik olarak öğrenip temsil edebilmek için çok sayıda katmanlı bağlantılı düğümden bir diğer adı ile sinir hücresinden oluşan yapay sinir ağları kullanır [84]. Şekil 2.1 de sinir hücresi yapısı gösterilmiştir.



Şekil 2.1 Sinir hücresi yapısı[85]

x : Nörona dışardan gelen giriş verisi veya bağlantılı olduğu diğer nöronlardan gelen girdiyi ifade eder.

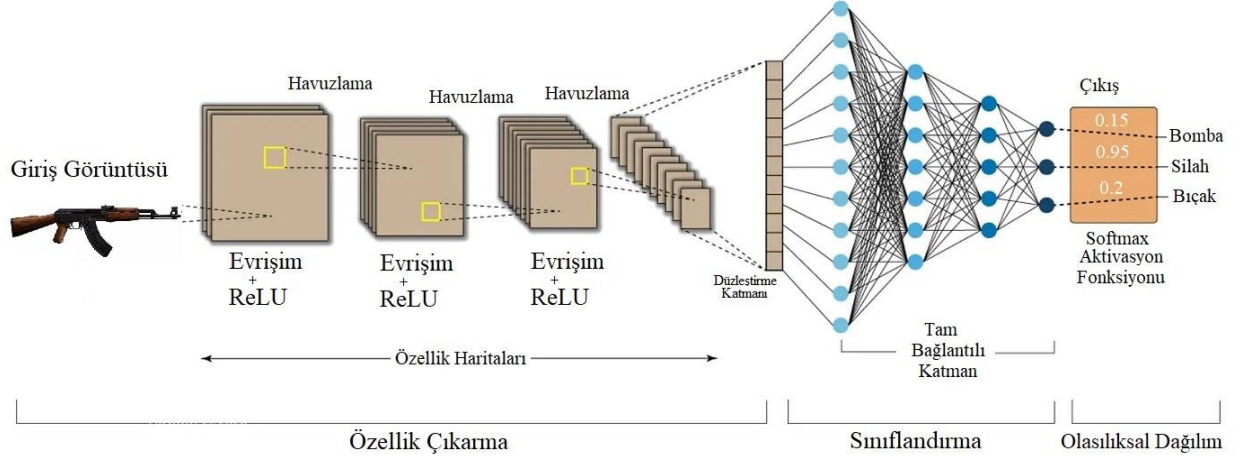
w : Nöronlar arasındaki sinapsı taklit ederek alınan bilgiyi değiştiren sinaptik ağırlıkların vektörüdür.

b : Nöronun kesişim veya eşik değeri olarak bilinen Bias değeridir.

y : Çıkış değeri olarak tanımlanır.

Derin öğrenme modelleri, yapay sinir ağlarının çok sayıda katmanını içeren derin sinir ağlarından oluşur. Ağların derin olarak adlandırılmasının sebebi çok sayıda gizli katmanın bulunmasından kaynaklanmaktadır. Bu gizli katmanlar modele giderek daha karmaşık ve soyut özellikleri öğrenme yeteneği kazandırır. Düşük seviye katmanlar kenar veya renk bulma gibi basit özellikleri yakalayabilirken, yüksek seviye katmanlar metin anlamları, nesne şekilleri gibi daha karmaşık özellikleri yakalayabilirler. Ancak derin öğrenme modelleri her ne kadar iyi tasarlansa da başarılı bir eğitim süreci olmadığında başarılı sonuçlar veremezler. Bu yüzden modeller eğitim aşamasında büyük veri kümelerine ihtiyaç duyarlar ve eğitim sırasında nöronlar arasındaki ağırlıkları ayarlayarak verilerden gerekli bilgileri öğrenirler. Ayrıca derin öğrenme modelleri veri setinde ki ham verilerden gerekli özellikler çıkararak daha anlamlı bilgiler elde etme yeteneğine sahiptirler. Elde edilen yeni özellikler tavsiye sistemleri, otonom araçlar, görüntü, ses ve doğal dil işleme gibi birçok

alanda başarılı bir şekilde kullanılmaktadır [86]. Şekil 2.2 de evrişimli sinir ağı mimarisi gösterilmiştir.



Şekil 3.2 Evrişimli sinir ağı mimarisi

Evrişimli Sinir Ağları temel olarak şu katmanlardan oluşmaktadır:

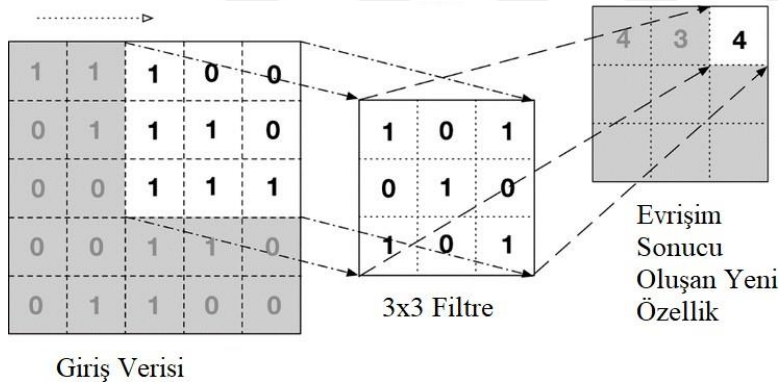
- Giriş Katmanı
- Evrişim Katmanı
- Havuzlama Katmanı
- Düzleştirme Katmanı
- Tam Bağlantılı Katman
- Düğüm Silme Katmanı
- Sınıflandırma Katmanı
- Çıkış Katmanı [87].

2.1.2.1 Giriş katmanı

Görüntülerin işlendiği ilk katman olan giriş katmanı eğitimden önceki bir ön işlem katmanıdır. Veri türlerine göre, giriş katmanının çalışma yapısı da farklılık göstermektedir. Önceden tanımlanmış filtreler ile ham giriş görüntüsünün kenarlarını bulma, gradyanlarını hesaplama veya kontrast çıkarma gibi özellikleri ağı kazandırarak ağın tanıma yeteneğine yardımcı olmaktadır [88].

2.1.2.2 Evrişim katmanları

Evrişimli Sinir Ağlarının temel işlemlerinin yapıldığı katmandır. Evrişim katmanının rolü, önceki katmandaki özelliklerin yerel birleşimlerini saptamaktır. Bu katmanda matrisler üzerinden 2 boyutlu işlemler yapılır. Evrişim katmanı giriş katmanından gelen matrise belirlenen boyutlarda filtrelerin uygulandığı ve elde edilen matrisin bir aktivasyon fonksiyonuna tabi tutularak yeni özelliklerin çıkarıldığı katmandır. Özelliklerin en iyi şekilde çıkarılabilmesi için filtre boyutu ve ağırlık değerleri oldukça önemlidir. Evrişim işleminin her adımında hata oranına bakılarak yeni ağırlık değerleri belirlenir ve evrişim döngüsü yeni değerler üzerinde tekrarlanır. Görüntüdeki köşeler, kenarlar, renk değişimleri ve diğer önemli özelliklerin tespiti sağlanarak özellik haritaları oluşturulur ve daha karmaşık özelliklerin elde edilebilmesi için havuzlama katmanlarına iletilir [89]. Şekil 2.3 de evrişim katmanında uygulanan bir filtre örneği gösterilmiştir.



Şekil 4.3 Evrişim katmanında uygulanan bir filtre örneği

2.1.2.3 Aktivasyon fonksiyonları

Derin öğrenme modellerinin çok önemli bir bileşeni olan aktivasyon fonksiyonları genellikle evrişim katmanlarının ardından kullanılır. Aktivasyon işlevi, derin öğrenme modelinin çıktısını, doğruluğunu, birleştirme yeteneğini, hızını ve modeli eğitmenin hesaplama verimliliğini belirler. Aktivasyon fonksiyonları farklı etki alanlarındaki derin ağlarının çıktılarını kontrol etmek için doğrusal veya doğrusal olmayabilir [90]. Derin öğrenme

modellerinde nöronlarda gerçekleşen ağırlık ve giriş değerinin çarpımının sonucu aktivasyon fonksiyonuna tabi tutularak çıkış değeri elde edilir. Çıkış değerleri gerçek problemlerde doğrusal olmayabilir. Doğrusal olmayan çıkış değerleri ile yapay sinir ağını eğitmek için aktivasyon fonksiyonlarının kullanılması gerekmektedir. Aktivasyon fonksiyonları her bir nöronun çıktısını 1 ile 0 veya -1 ile 1 arasında normalleştirmeye yardımcı olur. Sigmoid, ReLU, Leaky ReLU, Hyperbolic Tangent, Softmax, Swish gibi aktivasyon fonksiyonları yaygın olarak kullanılmaktadır. Aşağıda sık kullanılan aktivasyon fonksiyonlarının hesaplanması gösterilmiştir [91].

$$f_{\text{sigmoid}}(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

$$f_{\text{ReLU}}(x) = \max(0, x) \quad (2.2)$$

$$f_{\text{Tanh}}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

$$f_{\text{Softplus}}(x) = \log(1 + e^x) \quad (2.4)$$

$$f_{\text{SSsh}}(x) = x \times \text{Sigmoid}(\beta \times x) \quad (2.5)$$

$$f_{\text{Msh}}(x) = x \times \text{Tanh}(\text{Softplus}(x)) \quad (2.6)$$

[92].

Aktivasyon fonksiyonları tasarlanan modelin ihtiyacına göre farklı seçilebilmektedir. Örneğin negatif giriş değerine sahip verilerin olduğu durumlarda ReLU gibi fonksiyonlar, negatif değerın türevini sıfır hesapladığından modelin eğitim başarısını azaltmaktadır [93]. Bu gibi durumlarda negatif değerlerde de geri yayılımı mümkün kılan ve modelin eğitim performansını artırmak amaçlı Leaky ReLU, Swish gibi fonksiyonları tercih edilmektedir [94]. Aktivasyon işlemleri, her veri için binlerce hatta milyonlarca nöron arasından hesaplandığından hesaplama maliyeti açısından da verimli olmaları gerekmektedir. ReLU, Swish gibi aktivasyon fonksiyonları performans ihtiyacından dolayı geliştirilmiş fonksiyonlardır. Çıkış katmanında tercih edilen Softmax ise 0 ile 1 arasındaki her sınıf için çıkış değerlerini normalleştirir ve giriş değerinin belirli bir sınıfta olma olasılığını hesaplar. Giriş değerinin birden çok kategoriye ayırması gereken modellerde yalnızca çıktı katmanı

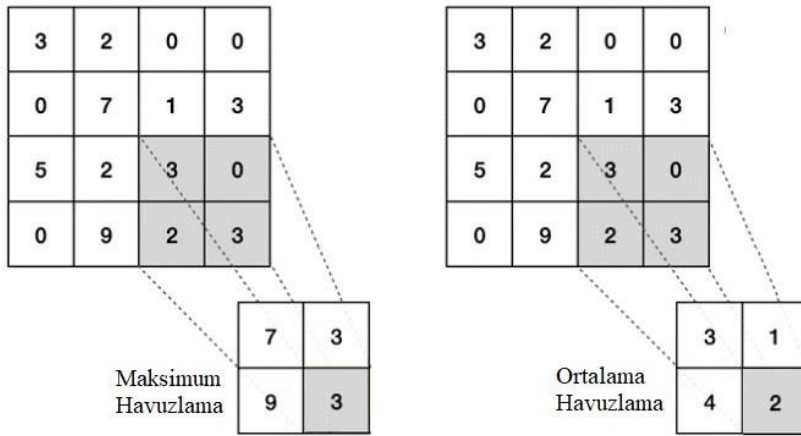
için kullanılır [95]. Şekil 2.4 te farklı aktivasyon fonksiyonlarının istatistiksel sonuçları gösterilmiştir.

Aktivasyon Fonksiyonu	μ_{acc}	μ_{loss}	σ_{acc}
Mish	87.48%	4.13%	0.3967
Swish	87.32%	4.22%	0.414
GELU	87.37%	4.339%	0.472
ReLU	86.66%	4.398%	0.584
ELU	86.41%	4.211%	0.3371
Leaky ReLU	86.85%	4.112%	0.4569
SELU	83.91%	4.831%	0.5995
SoftPlus	83%	5.546%	1.4015
SReLU	85.05%	4.541%	0.5826
ISRU	86.85%	4.669%	0.1106
TanH	82.72%	5.322%	0.5826
RReLU	86.87%	4.138%	0.4478

Şekil 5.4 Farklı aktivasyon fonksiyonlarının istatistiksel sonuçları [96]

2.1.2.4 Havuzlama katmanları

Havuzlama katmanının rolü, anlamsal olarak benzer özellikleri tek bir ögede birleştirerek özelliklerin konumunun ve ölçeğinin daha iyi öğrenilmesine yardımcı olmaktır. Havuzlama katmanı ortalama, maksimum, minimum, ortanca değer atama gibi işlemlerle giriş matrisini alt bölgelerle özetleyerek özellik matrisinin boyutunu azaltır. Seçilen filtre boyutu ve adım kaydırma değeri hassasiyete doğrudan etki etmektedir. Örneğin 3x3 boyutunda uygulanan bir maksimum havuzlama işleminde 9 pikselden en yüksek değere sahip olan değer, 3x3 boyutundaki alanı özetlediğinden filtre boyutunun doğru seçilmesi oldukça önemlidir [97]. Şekil 2.5 te 2x2 boyutunda maksimum ve ortalama havuzlama işlemleri örneklendirilmiştir.



Şekil 6.5 2x2 Maksimum ve ortalama havuzlama örneği

2.1.2.5 Düzleştirme katmanı

Düzleştirme katmanı evrişim ve havuzlama katmanından çıkan iki veya üç boyutlu dizileri tek boyutlu doğrusal vektöre dönüştürmek için kullanılır. Oluşan bu vektördeki veriler tam bağlantılı katmanın giriş verileri olarak kullanılır[98].

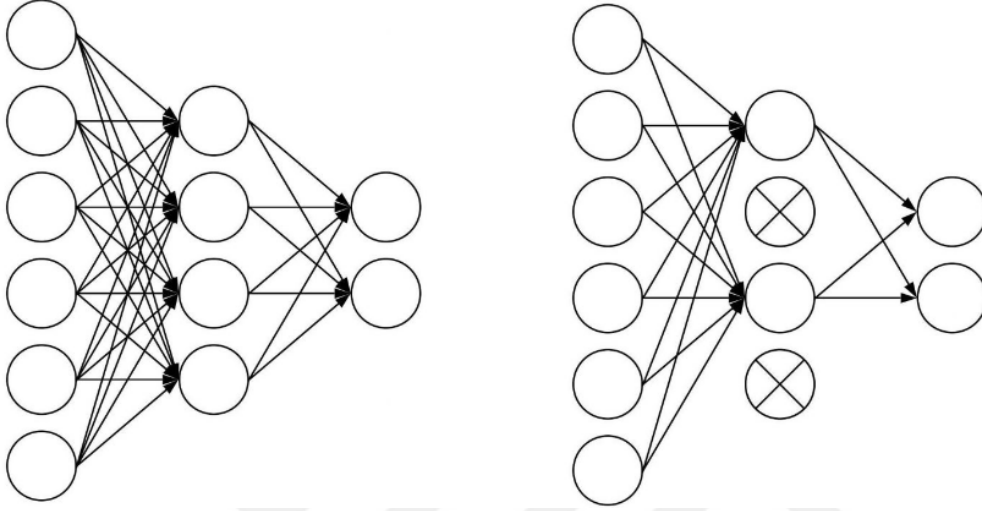
2.1.2.6 Tam bağlantılı katmanlar

Evrişim ve havuzlama katmanlardan gelen veriler düzleştirme katmanından sonra tam bağlantılı katmanlara giriş verisi olarak aktarılır. Tam bağlantılı katman derin öğrenmenin temel unsurlarındandır. Orta katman veya gizli katman olarak ta adlandırılan bu katman nöronların önceki katmandaki tüm aktivasyonlarla tam bağlantılarını sağlar. Tam bağlantılı katmanın en büyük avantajı, giriş verisi hakkında bir varsayımda bulunulmasına gerek kalmadan bağımsız işlem yapmasıdır. Ayrıca önceki katmanlardan gelen özellikleri kullanarak sınıflandırma veya regresyon görevlerini gerçekleştirerek orta ve yüksek seviye özellikler çıkarılmasını sağlar. Bu katmanda çok sınıflı sınıflandırma problemleri için genellikle Softmax aktivasyon fonksiyonu tercih edilir [99].

2.1.2.7 Düğüm silme katmanı

Düğüm silme tekniği gizli katmanlarda ki nöronun çıktısını 0,5 olasılıkla sıfır olarak ayarlanarak yapılır. Böylelikle sıfıra ayarlanan nöronlar ileri ve geri yayılıma katılmazlar[100]. Bu teknikle genellikle ağırlık eğitimi sırasında aşırı öğrenmesinin önüne

geçilmek hedeflenir. Bir düğümün silinmesi, aktivasyon fonksiyonuna katkısının sıfır olarak ayarlandığı anlamına gelir. Aktivasyon katkısı olmadığı için silinen düğümlerin gradyanları da sıfır olarak ayarlanır [101]. Şekil 2.6 da yapay sinir ağının standart yapısı ve rastgele düğüm silme işlemi sonrasındaki durumu gösterilmiştir.



Şekil 7.6 a) Standart sinir ağı

b) Düğüm silinmiş sinir ağı

Derin sinir ağında bir nöronun doğru şekilde öğrendiği durumlarda ağın daha derinlerindeki diğer nöronlar bilgi ihtiyacında o nörona güvenmeyi hızla öğrenmektedirler. Ağ genel bir kuralı öğrenmek yerine bu nöron tarafından öğrenilen özelliklere aşırı derecede bağlı kalarak ağın kırılgan hale gelmesine sebep olur. Düğüm silme bu tarz ezberlemelerin önüne geçerek öğrenme aşamasında bu gibi nöronlar silinebileceğinden herhangi bir nörona bağlı olmadan öğrenme devam eder. Bu durumda diğer nöronlarda eksik özellikleri öğrenmeye zorlanır [102]. Düğüm silme veri çeşidi fazla olan veri setlerinde modelin yeni veriler üzerindeki tahmin gücünü artırma etkisi de vardır. Modelin tahmin yaptığı durumlarda düğüm silme işlemi durdurulmalıdır. Aksi halde tahminin daha gürültülü ve kullanışsız veriler içermesine yol açabilir [103].

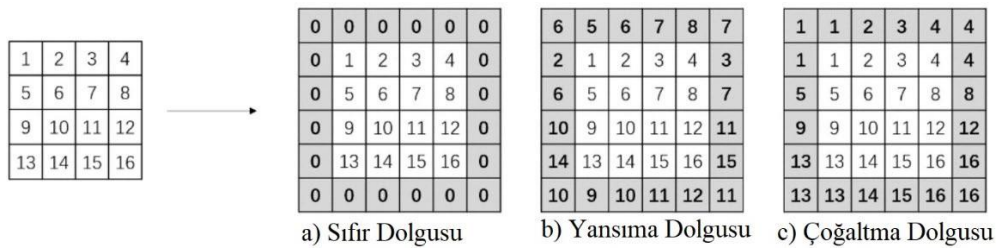
2.1.2.8 Kayıp fonksiyonu

Maliyet fonksiyonu olarak da adlandırılan kayıp fonksiyonu, ileriye doğru yayılma ile ağın tahminlerini gerçek etiketlerle karşılaştırır ve eğitim sürecini yönlendirir. Problemin çeşitliliğine göre farklı kayıp fonksiyonları kullanılabilir. Örneğin çok sınıflı sınıflandırmalar için yaygın olarak kullanılan kayıp fonksiyonu çapraz entropidir [104].

Sürekli değerlerin olduğu durumlarda ise ortalama hata karesi (mean squared error) regresyonlara uygulanabilir[105]. Sınırlayıcı kutular için CIOU[106] ve DFL[107] gibi kayıp fonksiyonları başarılı sonuçlar vermektedir. Softmax fonksiyonunun çıkış katmanında kullanıldığı ağ yapılarında ortalama hata karesi kaybolma eğilimi göstermektedir. Her yineleme sonrası çapraz entropinin geri yayılım hatası ortalama hata karesine göre daha düşüktür[108]. Ayrıca ortalama hata karesi kullanımı eğitim aşamasını çok yavaşlattığı için çok sınıflı sınıflandırma problemlerinde çapraz entropi daha popüler bir kayıp fonksiyonu haline gelmiştir[109].

2.1.2.9 Dolgulama

Dolgulama giriş verilerinin boyutlarını yönetmek için kullanılan önemli bir tekniktir. Giriş matrisine gerçek verilerin dışında kenar çevresine sıfır katmanları veya farklı değerler eklenerek yapılır. Dolgulama temel olarak giriş matrisine filtreler uygulandıktan sonra çıkış matrisinin aynı boyutta kalması veya ayarlanmak istenilen boyuta getirilme amacı ile yapılmaktadır. Böylelikle çıkış özellik haritasının boyutu kontrol altına alınmış olur. Dolgulama yapılmadığı durumlarda evrişim ile üretilen özellik haritasının boyutu her katman sonunda küçülür. Buda derin ağlarda özellik haritasının hızla azalmasına ve önemli bilgilerin kaybolmasına neden olur. Dolgulamanın bir diğer önemli özelliği ise yeterince kullanılmayan kenar piksellerin kullanımını sağlayarak bu pikselde kaybolabilecek bilgilerin tutulması sağlanır. Genel kullanılan dolgulama yöntemleri sıfır, çoğaltma ve yansıma dolgulamadır. Sıfır dolgulama giriş matrisinin etrafını sıfır sayıları ile doldurarak yapılır. Çoğaltma yönteminde giriş matrisinin sınırları boyunca değerleri çoğaltarak genişletilir. Yansıma yönteminde ise giriş değerleri sınır eksenini boyunca aynı değerler ile yansıtılır[110]. Şekil 2.7 de dolgulama yöntemleri örnekle gösterilmiştir.

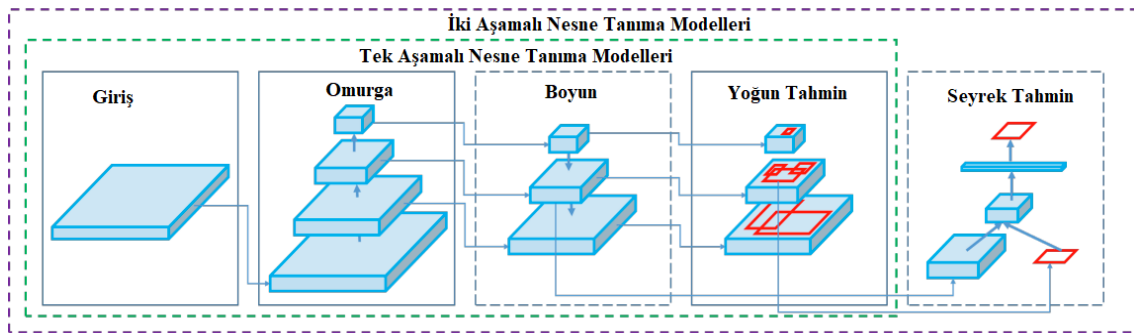


Şekil 8.7 Dolgulama yöntemleri

Dolguilama için gereken veri miktarı uygulanacak filtrenin ve istenilen çıkış matrisinin boyutuna bağılıdır. Dolgulamanın türünün ve miktarının düzgün yapılmadığı durumlarda modelin gerçek verilerin yerine doldurulmuş verileri öğrenmesine yol açabilir. Buda modelin tahmin yeteneğini düşürür. Diğer taraftan genel kullanılan yöntemlerin dinamik olmama gibi bir dezavantajı da vardır. Bu nedenle, dolguilama değeri her zaman statik olduğundan model eğitimi sırasında giriş matrisinin sınırlarına göre en doğru tahmin edilebilecek şekilde optimize edilemeyebilir. Daha başarılı sonuçlar elde edebilmek için modele özgü dolguilama modelleri geliştirilen çalışmalarda mevcuttur [111].

2.1.3 Nesne Tanıma Algoritmaları

Nesne tanıma, görüntü içerisindeki bir nesnenin sınıfı, şekli, koordinatlarıyla birlikte yükseklik ve genişlik bilgilerini belirlenerek gerçek konumunu tespit etme işlemidir [112]. Nesne tanıma modellerinin büyük kısmı ESA tabanlıdır ve iki farklı kategoriye ayrılmıştır. Bunların ilki bölgesel öneri ve sınıflandırmanın sıralı olarak yapıldığı R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN gibi iki aşamada modellerdir. İkincisi ise bölgesel öneri ve sınıflandırma işlemlerini eş zamanlı olarak gerçekleştiren tek aşamalı modellerdir. Tek aşamalı yöntemlerin en başarılıları YOLO versiyonları, SSD, vb modellerdir. Bu bölümde çalışma kapsamında test edilen nesne tanıma modelleri hakkında bilgi verilmiştir. Şekil 2.8 de tek aşamalı ve iki aşamalı modellerin yapısı gösterilmiştir.



Giriş: {Görüntü, Bölge, Görüntü Piramidi, ... }

Omurga: {VGG16 [77], ResNet-50 [79], ResNeXt-101[113], Darknet53 [114], ... }

Boyun: { FPN [115], PANet [116], Bi-FPN[117], ... }

Baş:

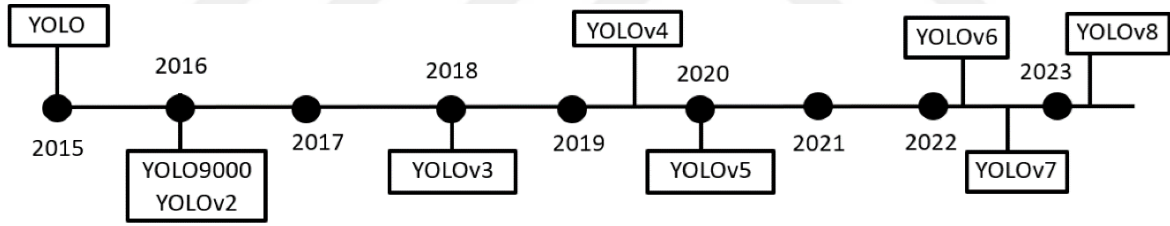
Yoğun Tahmin: {RPN[118], YOLO [119] [120] [114], SSD [121], RetinaNet [122], FCOS [123] }

Seyrek Tahmin: { Faster R-CNN [118], R-FCN [124], ... }

Şekil 9.8 Nesne tanıma algoritmaları [125]

2.1.3.1. YOLO

YOLO, görüntü işleme alanında en yaygın kullanılan nesne tespit algoritmalarından bir tanesidir. Bu algoritma, ilk olarak giriş görüntüsünü 448×448 boyutunda yeniden boyutlandırdıktan sonra çok ölçekli bölgelere böler. Daha sonra görüntü üzerinde tek bir evrişim ağı çalıştırır. Sınırlayıcı kutular kullanarak sınıf olasılıklarını tahmin ederek eşik değerinin üzerinde güvenilirliğe sahip tespitleri gerçek zamanlı sonuçlandırır. YOLO diğer nesne tanıma modellerine göre algılama performansı yüksek ve daha az hesaplama maliyetine sahiptir. Sınıflandırıcı tabanlı yaklaşımlardan farklı olarak YOLO, modelin tamamını doğrudan bir kayıp fonksiyonu üzerinden eğiterek daha hızlı sonuçlar elde edebilmektedir. 2015 yılında Joseph Redmon ve ekibi tarafından ilk tanıtıldığı zaman Fast R-CNN gibi iki aşamalı nesne tanıma mimarilerinden daha düşük tahmin performansı göstermesine rağmen zaman içerisinde mimarisindeki çeşitli iyileştirmelerle farklı modelleri tanıtılarak hız ve tahmin başarısı çok daha başarılı seviyelere çıkarılmıştır [119]. Şekil 2.9 da YOLO nesne tanıma modellerinin yıllara göre gelişimi gösterilmiştir.



Şekil 10.9 YOLO sürümlerinin yıllara göre gelişimi

2.1.3.2. YOLO V4

YOLO V4, CSPDarknet53 ile tasarlanmış bir omurga[126], SPP [127] ve PAN[116] dan oluşan boyun kısmı ve YOLO V3 [114] ile aynı özelliklere sahip bir baş kısmı ile tasarlanmıştır. YOLO V4'ün çeşitli önemli yenilikleri ile hız ve doğruluk oranlarında başarılı sonuçlar verdiği gösterilmiştir. Bu yeniliklerin ilki Cutmix ve mozaik veri çoğaltma yöntemidir. Bir diğer yenilik ise Mish aktivasyon fonksiyonu kullanılarak eğitim aşamasında doğruluk oranının artırıldığı gösterilmiştir. Ağ aktivasyonlarının ortalama ve varyanslarına göre normalleştirilmesi için Yığın Normalizasyonu veya Çapraz Yinelemeli Yığın Normalizasyonu yerine Çapraz Mini Toplu Normalleştirme uygulanmıştır. Böylelikle

istatistikleri tek bir mini grup içinde toplamak yerine tüm gruplar içindeki istatistikler toplanmaktadır. YOLO V4, 2020 yılında tanıtıldığı zaman MS COCO veriseti üzerinde eğitilmiş ve test sonuçlarına göre mevcut tüm nesne tanıma modellerinden daha hızlı ve daha doğru sonuç verdiği belirtilmiştir. Ayrıca modelin 8-16 GB VRAM'e sahip geleneksel bir GPU üzerinde eğitilebildiği için herkes tarafından rahatlıkla kullanılabileceği ifade edilmiştir [125].

2.1.3.3. YOLO V4 Tiny

Çalışma kapsamında test edilen modellerden bir tanesi de YOLO V4 Tiny modelidir. YOLO V4 modelinin daha hafif bir versiyonu olan YOLOV4 Tiny daha hızlı nesne algılama performansına sahip olması için tasarlanmıştır. Mimari açıdan bakıldığında omurga ağı olarak YOLO V4 Tiny, CSPDarknet53 ağı yerine CSPDarknet53 Tiny ağını kullanır. CSPDarknet53 Tiny ağı, artık ağıdaki ResBlock modülü yerine CSPBlock modülünü kullanmaktadır. CSPBlock modülü, özellik haritasını iki bölüme ayırır ve farklı bir modül ile birleştirir. Bu özellik hesaplama maliyetini artırsa da %10 ile %20 arasında doğru tahmin oranını artırdığı belirtilmiştir. Bir diğer farklılık ise hesaplama işlemini daha da basitleştirmek için YOLO V4 Tiny aktivasyon fonksiyonu Mish yerine LeakyReLU kullanılmaktadır. YOLO V4 Tiny'in düşük GPU özelliklerine sahip mobil ve gömülü sistemler için de kullanılabileceği gösterilmiştir[128].

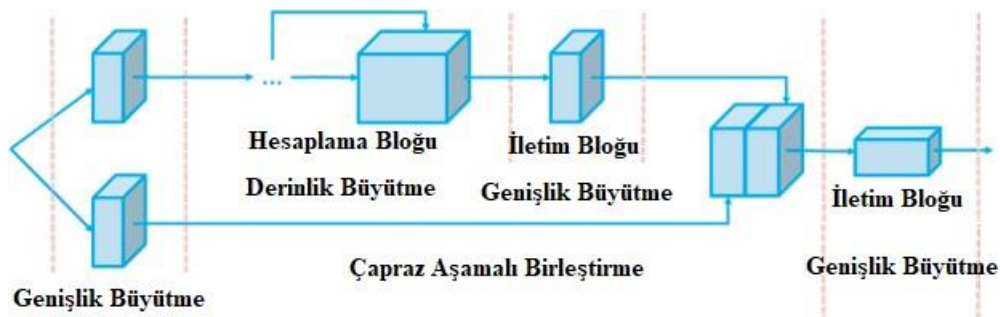
2.1.3.4 YOLO V5n

YOLO V5 Ultralytics şirketi tarafından 2020 yılında YOLO V4'den 1 ay sonra tanıtılmıştır. Ultralytic, YOLO'nun önceki sürümlerini derin öğrenme alanındaki en ünlü framewoklerden biri olan Python dilinde yazılmış PyTorch'a dönüştüren şirkettir. YOLO V5 önceki sürümlerden farklı olarak herhangi bir akademik yayın ile tanıtılmamıştır. Ağ mimarisi YOLO V4'ün mimarisine oldukça benzerdir [129]. PyTorch ın getirdiği avantajlar ile YOLO V5'in çoklu ağ mimarisinin kullanımı oldukça esnektir ve çok hafif bir model boyutuna sahiptir. Doğruluk performansı açısından karşılaştırıldığında YOLO V4 ile benzer sonuçlar vermektedir[130]. Çalışma kapsamında YOLO V5'in en hafif modeli olan YOLO V5n test edilmiştir.

2.1.3.5 YOLO V7

2022 yılında tanıtılan YOLO V7 modeli diğer YOLO modellerinden farklı olarak sunulan yeniliklerin ilki E-ELAN kullanmasıdır. Genişletilmiş verimli katman toplama ağı olarak adlandırılan E-ELAN, orijinal gradyan yolunu bozmadan ağıın öğrenme yeteneğini sürekli olarak geliştirebilmek için genişletme, karıştırma ve birleştirme işlemleri yapar. Böylelikle grup evrişimini kullanarak hesaplama bloklarının kanalını ve önem derecesini genişletmek amaçlanmaktadır. Mimari açıdan bakıldığında ise E-ELAN yalnızca hesaplama bloğundaki mimariyi değiştirir. Geçiş katmanlarının mimarileri tamamen değiştirmez. Orijinal ELAN tasarım mimarisini korumakla birlikte E-ELAN, daha çeşitli özellikleri öğrenmek için farklı hesaplama blok gruplarına da rehberlik edebilir. Bu sayede farklı özellik haritaları tarafından öğrenilen özellikler, parametreler ve hesaplamaların kullanım şekli geliştirilebilir.

YOLO V7 ile sunulan bir diğer yenilik ise “Birleştirme Tabanlı Model” kullanmasıdır. Yapay sinir ağının derinliğinde ölçek büyütme veya küçültme işlemi gerçekleştirildiğinde, birleştirme tabanlı bir hesaplama bloğunun hemen ardından gelen çeviri katmanının derecesinin de azalacağını veya artacağını görülmektedir. Bu durum bir sonraki iletim katmanının giriş genişliğinin de artmasına veya azalmasına neden olmaktadır. Bir hesaplama bloğunun derinlik faktörünü ölçeklendirildiğinde o bloğun çıkış kanalındaki değişimin de hesaplaması gerekmektedir. YOLO V7 de bu problemin çözümü, model ölçeklendirme işlemi gerçekleştirirken yalnızca hesaplama bloğundaki derinliğin ölçeklendirilmesi ile yapılmaktadır. İletim katmanının geri kalanına karşılık gelen kısımda genişlik ölçeklendirme gerçekleştirilmektedir. Şekil 2.10 da YOLO V7 de sunulan birleştirme tabanlı model gösterilmiştir.



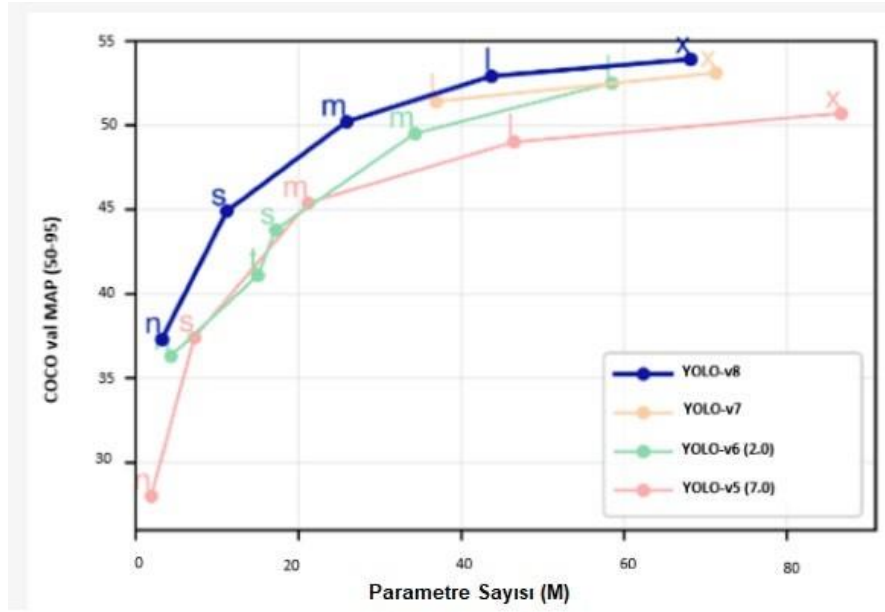
Şekil 11.10 YOLO V7 de sunulan birleştirme tabanlı model

Geliştirilen birleştirme tabanlı model, modelin ilk tasarımıda sahip olduğu özellikleri korumasını sağlayarak ağ yapısının bozulmasını önlemektedir[131]. YOLO V7 bir diğer güncellemesi gradyan akış yayılma yolu kullanılarak yeniden parametrelendirilebilen evrişim işlemi yapabilmesidir. Ayrıca, öğrenme sürecinde dinamik ayarlanabilen etiketler ile daha iyi sonuçlar elde edilebildiği gösterilmiştir[132].

2.1.3.6 YOLO V8

YOLO V5 gibi Ultralytics şirketi tarafından 2023 yılında tanıtılan YOLO V8, yeni özellikler ve optimizasyonlar ile önceki YOLO sürümleri geliştirilerek tanıtılmıştır. Benzer sayıda parametreyle tüm YOLO modellerine göre daha verimli çalıştığı belirtilen YOLO V8'in donanım açısından da yeni mimari reformlara işaret ettiği vurgulanmıştır [133]. Ayrıca YOLO V8'in sınırlı donanıma sahip cihazlarda da yüksek çıkarım hızına ulaşabileceği ifade edilmiştir. YOLOV8'deki Hızlı Uzaysal Piramit Havuzlama (SPPF) katmanı farklı ölçeklerdeki özellikleri sabit boyutlu bir özellik haritasında bir araya getirerek ağın hesaplamasını hızlandırmak için tasarlanmıştır[134].

YOLO V8'in bir diğer önemli özelliği önceki modellerdeki bağlantı kutularını içermeyen bir model olmasıdır. Böylelikle nesnenin bağlantı kutusundan uzaklığı yerine doğrudan nesnenin merkezini tahmin edebilme yeteneğine sahiptir[135]. Bağlantı kutularının olmaması tahmin kutularının sayısını da azalttığı için ağın belirli bir çıkarım sonrasında aynı nesneyi temsil eden gereksiz veya örtüşen sınırlayıcı kutu tahminlerini eleyerek yalnızca en güvenilir ve doğru olanı seçen Maksimum Olmayan Bastırma (Non-Maximum Suppression) işlemini de hızlandırmaktadır[136]. YOLO V8, sınırlayıcı kutular için CIoU ve DFL kayıp fonksiyonlarını kullanırken sınıflandırma işlemi için ikili çapraz entropi kayıp fonksiyonunu kullanır. Bu fonksiyon seçimleri nesne algılama performansını artırdığı gibi küçük nesneleri algılanma da daha başarılı sonuçlar vermektedir. Şekil 2.11 de YOLO V8 mimarisi gösterilmiştir.

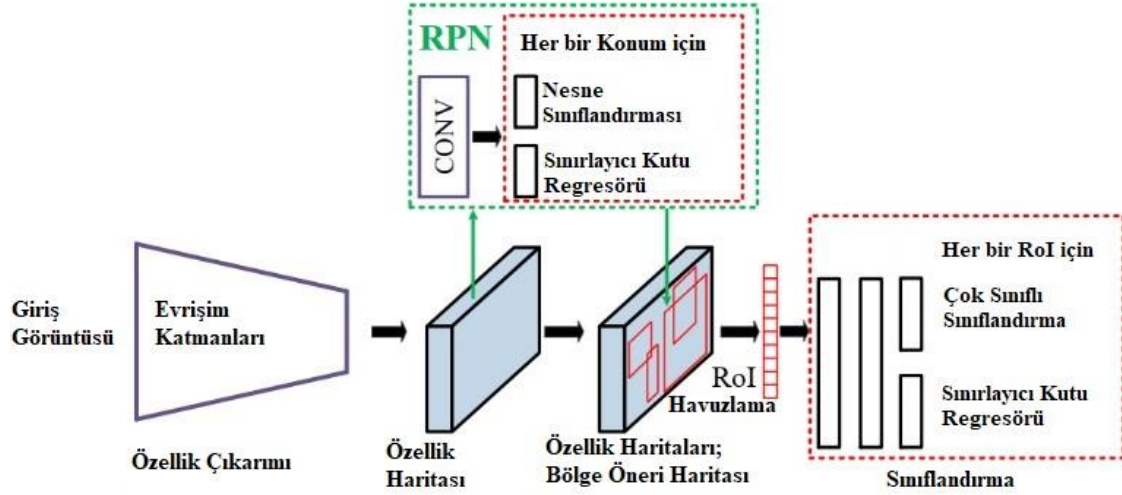


Şekil 13.12 YOLO V8'in performans karşılaştırılması [138].

2.1.3.7 Faster R-CNN

Faster R-CNN bölge öneri ağı ve nesne tespit ağından oluşan iki aşamalı nesne tanıma algoritmalarındandır[118]. İlk olarak giriş görüntüsü evrişim ağından geçerek özellik haritaları çıkarılır. Bölge öneri ağı, tespit edilecek nesnelerin görüntü üzerinde bulunabilecek yerleri için öneriler oluşturur. Daha sonra özellik haritaları üzerinde tahmin noktaları ile farklı boyut ve konumlarda birçok tahmin kutusu oluşturulur. Bu tahminlerin sonuçlandırılması için nesne tespit ağında hangi sınıfta olduğunu belirleyen sınıflandırma yapılır. Belirlenen eşik değeri üzerinde tahmine sahip nesneler sonuçlandırılır[139].

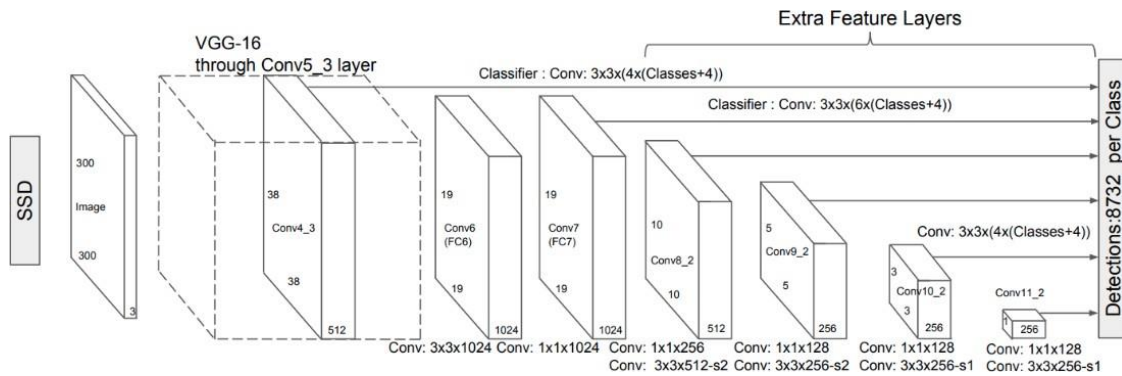
Faster R-CNN nesne tanıma modelleri içerisinde başarılı sonuçlar verse de öneri ağı modülünün CNN'den bağımsız olarak ayrı bir modülde gerçekleştirilmesi nedeniyle öğrenme ve yürütme hızında verimsizliklere sahiptir [140]. Şekil 2.13 te Faster R-CNN ağ yapısı gösterilmiştir.



Şekil 14.13 Faster R-CNN ağ yapısı

2.1.3.8 SSD

Tek bir derin sinir ağı kullanarak nesne tespiti yapabilen SSD sınırlayıcı kutuların alanını, özellik haritasını, konumunu ve farklı en boy oranları üzerinden varsayımlar yapabilen bir modeldir. Bu model nesnenin tahmini sırasında sınırlayıcı kutularda çeşitli ayarlar yapabilme özelliğine sahiptir. Ayrıca birden fazla özellik haritasından gelen tahminleri birleştirerek nesnenin kategorisini puanlayarak tahmin eder. Tüm hesaplamayı tek bir ağda yaptığı için birçok modele göre daha hafif ve hızlı tespit yapabilmektedir[121]. Şekil 2.14 te SSD ağ yapısı gösterilmiştir.

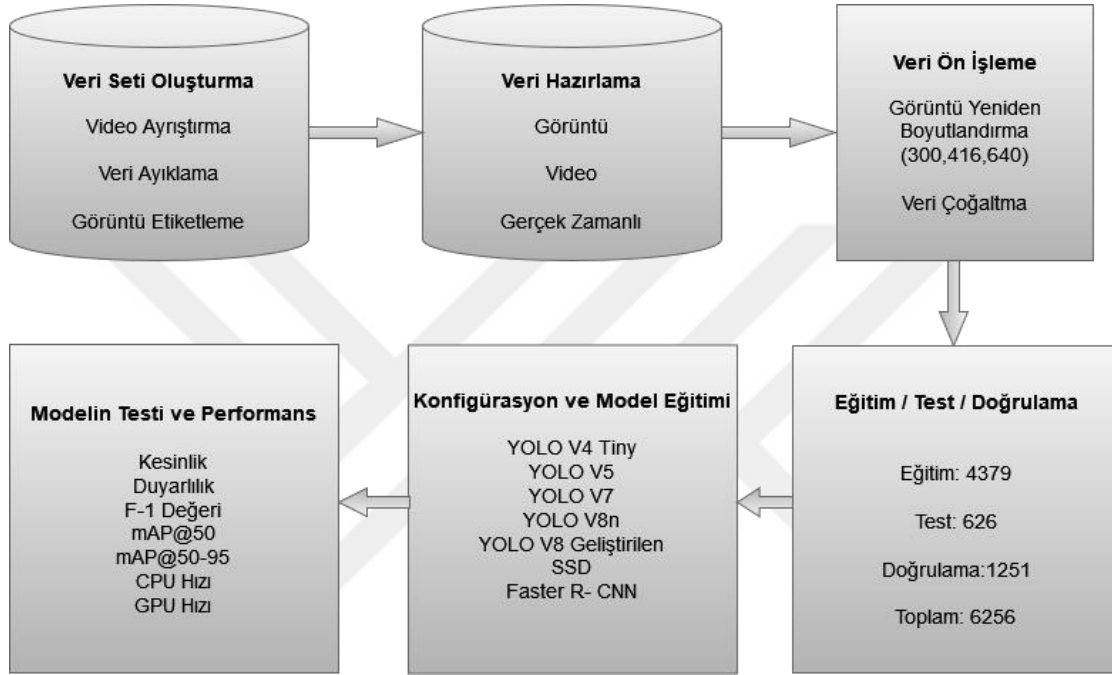


Şekil 15.14 SSD ağ yapısı



3. YÖNTEM

Derin öğrenme tabanlı nesne tanıma modelleri temel bazı adımlar izlenerek oluşturulmaktadır. Çalışmamızda oluşturulan yöntemlerin izlediği adımlar şekil 3.1 de gösterilmiştir.



Şekil 16.1 Nesne tanıma modellerinin genel akış şeması

3.1 Veri Seti Oluşturma

Dijital oyunlarda şiddet içeriklerinden oluşturulmuş kullanıma açık bir veri seti bulunmamaktadır. Tez çalışmamız kapsamında farklı dijital oyunlardan şiddet içerikli görüntüleri içeren bir veri seti oluşturulmuştur. Veri setimizi oluşturan görüntüler CSGo, PubG, GTA, Valorant, Half Life vb. dünya üzerinde yaygın oynanan oyunlarından 100 den fazla video karelerine ayırarak elde edilen görüntülerden oluşturulmuştur. Bu kapsamda video görüntülerini karelerine ayıran ve saniye başına 1 görüntü karesi kaydedecek bir yazılım geliştirilmiştir. Veri setimizde her sınıfa ait en az 1000 görüntünün bulunduğu yaklaşık 6256 görüntü bulunmaktadır. Modelin eğitimi ve testi için kullanılan bu görüntüler silah, bıçak, bomba, yangın, kan ve kavga sınıflarına ait görüntülerden oluşmakta ve modelin daha başarılı çalışması için farklı açılardan elde edilmiş görüntüleri içermektedir.

Modellerin eğitimi için 4379, testi için 626 ve doğrulama için 1251 görüntü kullanılmıştır. Modeller ayrıca dijital oyun ortamı, dijital oyunlara ait video ve görüntüler kullanılarak da test edilmiştir. Şekil 3.2 de veri setinde kullanılan görüntülerden örnekleri gösterilmiştir.



Şekil 17.2 Veri setinde kullanılan görüntülerden örnekleri

3.2 Veri Ayıklama

Modelin eğitimi sırasında modele sunulacak görüntüler ne kadar çeşitli ve çok olursa modelin başarısı da o oranda artmaktadır. Video içerisinden görüntü çıkarma aşamasından sonra elde edilen görüntülerden, modelin eğitime katkı sağlamayacağı düşünülen görüntülerin ayıklanarak veri setinden çıkarılması gerekmektedir. Elde edilen görüntülerden bir kısmı şiddet içermediğinden, bulanıklık, çözünürlük, açı vb. durumlardan eğitim aşamasına katkı sağlayamayacağı için eğitim veri seti içerisinden çıkartılmıştır. Veri ayıklama işlemi sonucunda modelin eğitimi için kullanılacak 6256 görüntü elde edilmiştir.

3.3 Görüntü Etiketleme

Modelin eğitimi için görüntüler seçmek ve etiketlemek en önemli aşamalardan bir tanesidir. Her ne kadar başarılı bir model tasarlansa da, model doğru görüntüler ile eğitilmediğinde ortaya başarısız sonuçların çıkması muhtemeldir. Anlamsız eğitim seti görüntülerini seçmek modelin başarısını düşüreceğinden seçilecek görüntülerin açık bir şekilde şiddet görüntüleri içermesi gerekmektedir. Ayrıca görüntülerin kalitesi modelin başarısını etkilediğinden eğitim için seçilen görüntüler yeterli çözünürlükte ve net olmalıdır. Eğitim için kullanılacak her bir görüntünün sınıfı, konumu, koordinatları, genişlik ve yükseklik bilgisini sınırlayıcı çerçeveler kullanılarak etiketlenmesi gerekmektedir. Çalışmada

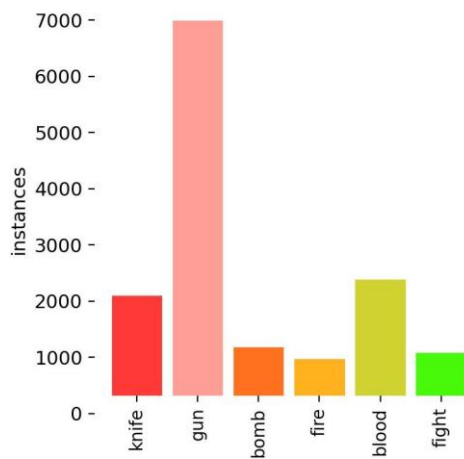
LabelImg kullanılarak görüntü etiketleme işlemi gerçekleştirilmiştir. Şekil 3.3 de LabelImg ile görüntü etiketleme örneği gösterilmiştir.



Şekil 18.3 LabelImg görüntü etiketleme işlemi

Etiketleme işlemi sonunda modelin eğitimi için kullanılacak görüntüler ve görüntü içerisindeki şiddet unsurlarının hangi sınıfa ait oldukları ve konum bilgileri YOLO ve VOC formatında kaydedilmiştir.

Bir görüntü içerisinde aynı veya farklı sınıfa ait birden fazla şiddet içeriği bulunabilir. Çalışmamızda veri setimizde bulunan 6256 görüntüden toplam 18704 şiddet içeriği etiketlenerek kaydedilmiştir. Buda her bir görüntünün ortalama 3 şiddet unsuru içerdiğini göstermektedir. Şekil 3.4 de veri setinde etiketlenen şiddet içeriklerinin miktarı grafiksel olarak gösterilmiştir.



Şekil 19.4 Veri setinde etiketlenen şiddet içeriklerinin miktarı

Etiketlenen 18704 içerikte 9922 adet silah, 2543 adet bıçak, 1191 adet bomba, 1010 adet ateş, 2934 adet kan ve 1104 adet kavgaya görüntüsü bulunmaktadır.

3.4 Veri Çoğaltma

Yapay sinir ağına eğitim aşamasında olabildiğince çeşit veriler sunmak ağı öğrenme yeteneğini artırmaktadır. Mevcut veri setinde görüntüleri çeşitli açılarda ters, yatay, dikey çevirerek, ölçekleyerek, kırparak, X-Y koordinatlarında hareket ettirerek, çeşitli filtreler kullanarak veya gürültü ekleyerek veri çoğaltma işlemi gerçekleştirilebilir[141]. Ayrıca YOLO'nun yapısında bulunan Cutmix, Mozaik gibi yöntemler sayesinde başarılı bir şekilde veri çoğaltma yapılabilir. Mozaik veri çoğaltmada ağa 4 farklı görüntü tek seferde sunularak modelin bunları aynı anda öğrenmesi istenmektedir. Ayrıca yığın normalleştirme sayesinde her katmandaki 4 farklı görüntüden aktivasyon istatistikleri hesaplanmaktadır. Şekil 3.5 de YOLO V8n eğitim aşamasında mozaik veri çoğaltma örneği gösterilmiştir.



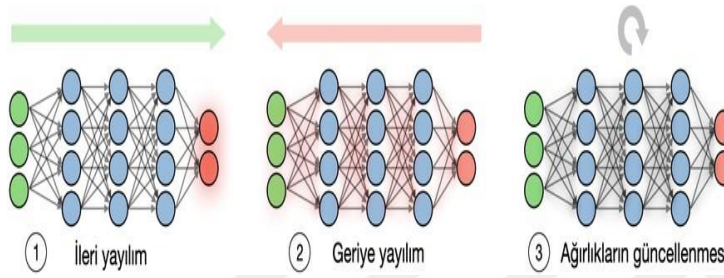
Şekil 20.5 YOLO V8n eğitim aşamasında mozaik veri çoğaltma örneği

Cutmix veri çoğaltma yönteminde ise eğitim işlemi sırasında farklı iki görüntüden etiketleri ile birlikte parçalar alınarak tek bir görüntüde birleştirme işlemidir. Elde edilen bu yeni görüntü modele sunularak modelin daha zor şartlarda öğrenmesi istenmektedir. CutMix yöntemi giriş görüntüsündeki bozulmalara karşı modelin sağlamlığını ve tespit performansını artırmaktadır [142].

3.5 Konfigürasyon

Konfigürasyon; eğitim sürecinin temellerini oluşturan ağ yapısı, epoch, grup büyüklüğü, aktivasyon fonksiyon seçimi, giriş görüntü boyutu vb. parametre değerlerinin ayarlandığı aşamadır. Bu değerlerin önemli bir kısmı şu şekildedir;

Epoch: Modelin eğitimi sırasında tüm eğitim görüntülerini ileri ve geri yayılım gerçekleştirdikten sonra ağırlıkların güncellenmesi olarak tanımlayabiliriz. Şekil 3.6 de Epoch yapısı gösterilmiştir. Modellerin belirli bir epoch döngüsü sonrasında öğrenme yeteneğinde bir ilerleme olmadığı gözlemlendiği anda eğitim işlemi durdurur.



Şekil 3.6 Bir epoch sırasında yapılan işlemler

Grup (Batch): Modelin eğitimi sırasında GPU, CPU donanım yapılarına bağlı olarak tüm eğitim verilerini tek seferde ağa verilemediğinden belirli miktarda veriyi gruplar halinde modele aktarma işlemidir. Grup boyutu 32, 64, 128 vb. değerler ile ayarlanmaktadır.

Alt Gruplar (Subdivisions): Grupların eşit olarak bölündüğü daha küçük alt gruplardır. Düşük belleğe sahip donanımlarda, alt grup sayısı daha yüksek ayarlanarak modele yüklenen veri miktarı azaltılır. Böylelikle eğitim işlemi daha uzun süreye yayılır. Benzer şekilde yüksek belleğe sahip donanımlarda alt grupların sayısı daha düşük ayarlanarak her yinelemede daha fazla veri yüklenmesi sağlanarak eğitim süreci hızlandırılır.

Görüntü Boyutu: Modele sunulan giriş görüntüsünün yükseklik ve genişlik değerleridir. Nesne tanıma modellerinde 300, 416, 512, 640 gibi farklı görüntü boyutlarını tercih edilmektedir.

Aktivasyon Fonksiyonu: Aktivasyon fonksiyonunun seçimi yapay sinir ağının performansı üzerinde büyük etkiye sahiptir. Gizli katmanlar ve çıktı katmanları arasında çeşitli

hesaplamalar gerçekleştirmek için aktivasyon fonksiyonlarını kullanır[143]. Aktivasyon işlemi, sinir ağındaki bir katmanın çıktısını belirli değerler arasına dönüştürerek anlamsız değerlerin önüne geçer. Kayıp fonksiyonu ve öğrenme hızıyla birlikte aktivasyon fonksiyonunun seçimi, bir sinir ağının nasıl öğrendiğini ve davrandığını belirlemede büyük rol oynar[144]. Ağ yapısına göre Sigmoid, Tanh, ReLU, Leaky ReLU, Swish, Mish vb. aktivasyon fonksiyonları tercih edilmektedir.

Lr0: İlk öğrenme oranıdır. Modelin ağırlıklarını ne kadar hızlı güncelleneceğini etkilediğinden optimizasyon süreci için önemlidir.

Optimizer: Optimizer algoritmaları modelin parametrelerini güncellemek ve eğitim sırasında kayıp fonksiyonunu optimize etmek için kullanılır. Ayrıca eğitim boyunca öğrenme oranını ayarlar. SGD, Adam, AdamW, vb. optimizasyon algoritmaları yaygın kullanılır[145].

Filtre Boyutu: Kernel olarak adlandırılan filtre, evrişim işleminin temelini oluşturur. Giriş verilerindeki belirli kalıpları veya özellikleri tespit etmek için kullanılmaktadır[146]. Derin öğrenme modellerinde 3x3, 5x5 ve 7x7 vb. filtre boyutları daha yaygın kullanılmaktadır. Filtre boyutunun küçük seçilmesi performans artışı ve eğitim süresinin daha kısa olmasını sağlar[147].

Conf: Tespitler için minimum güven eşik değeridir. Bu eşik altında güvenle tespit edilen nesneler dikkate alınmaz.

IoU: Modelin doğrulama aşamasında kullandığı eşik değeridir. Belirlenen IoU eşik değeri ve üstünde uyuşan nesneler model tarafından sonuçlandırılır.

4. DENEYSEL SONUÇLAR

4.1 Performans Kriterleri

Performans kriterleri modelin ne kadar başarılı olduğunu anlamak için çeşitli değerlendirme kriterleriyle sonuçlarımızı analiz etmemizi sağlayan yöntemlerdir. Çalışmamızda AP, mAP, Doğruluk, Kesinlik, Duyarlılık ve F1 performans kriterleri hesaplanarak modellerin başarısı test edilmiştir [148].

Ortalama Kesinlik (AP) : Tahmin edilen sonucun ne kadar doğru olduğunu gösterir. AP, bir modelin kesinlik ile duyarlılık eğrisini değerlendiren bir ölçümdür. Formül 4.1 de ortalama kesinlik hesaplanması gösterilmiştir.

$$AP = \int_0^1 P(R) dR \quad (4.1)$$

Ortalama Kesinlik Değerlerinin Ortalaması (mAP) : Tahmin edilen sonucun ne kadar doğru olduğunu gösteren AP'nin ortalama değeridir. Formül 4.2 de mAP hesaplanması gösterilmiştir.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP(i) \quad (4.2)$$

Birleşim Üzerinden Kesişim (IoU): Modelin nesne tanıma için tahmin ettiği sınırlayıcı alanın, nesnenin gerçek yerinin bulunduğu sınırlayıcı alana bölünerek hesaplanır. IoU eşik değeri, başarılı bir tespitin hangi düzeyde olacağına karar veren ve önceden modele tanımlanan bir değerdir. Bu eşik değerinin üstünde kalan tahminler sonuçlandırılırken, altında kalan tahminler gösterilmez. Formül 4.3 de IoU hesaplanması gösterilmiştir.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (4.3)$$

mAP50: IoU eşiklerinin 0,50 ayarlandığında hesaplanan mAP değerini ifade eder. Modelin daha basit tespitlerini dikkate alan doğruluk ölçüsüdür.

mAP 50-95: 0,50 ile 0,95 arasında değişen IoU eşik değerlerinde hesaplanan mAP değerini ifade eder. Modelin farklı tespit zorluk seviyelerindeki performansının kapsamlı ölçümünü sağlar. mAP 50 - 95 aralığı seçildiğinde, IoU eşik değeri 0,50 den başlayarak 0,05 miktar artan adımlarla 0,95 değerine kadar ilerler ve modelin tahminleri değerlendirilir.

Kesinlik (Precision) : Doğru şekilde tespit edilen tüm tespit sonuçlarının yüzdesini ifade eder. Doğru pozitif tahminlerin, pozitif tahminlere oranı ile hesaplanır. Formül 4.4 de kesinlik hesaplanması gösterilmiştir.

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (4.4)$$

Doğruluk (Accuracy): Modelin doğru tahmin ettiği nesnelerin toplam tespit sayısına oranı ile hesaplanmaktadır. Formül 4.5 de doğruluk hesaplanması gösterilmiştir.

$$\text{Doğruluk} = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (4.5)$$

Duyarlılık (Recall): Tespit edilmesi gereken nesnenin olduğu görüntülerde modelin ne kadar başarılı tahmin yaptığını bulmak için kullanılır. Doğru pozitif tahmin edilen ilgili görüntülerin, gerçek tüm pozitif görüntü miktarına oranı ile hesaplanmaktadır. Formül 4.6 da duyarlılık hesaplanması gösterilmiştir.

$$\text{Duyarlılık} = \frac{TP}{(TP+FN)} \quad (4.6)$$

F1 Değeri: F1 değeri ise Kesinlik ve Duyarlılık hesaplamalarının harmonik ortalamasıdır. Bu değer hem yanlış pozitifleri hem de yanlış negatifleri dikkate alarak modelin performansının dengeli bir değerlendirmesini sağlar. F1 değeri 0 ile 1 arasında bir değere sahiptir. Bu değer ne kadar yüksek olursa, bir nesnenin tespit doğruluğu da o kadar yüksek demektir. Formül 4.7 de F1 değeri hesaplanması gösterilmiştir.

$$\text{F1 - değeri} = \frac{2 \times \text{Kesinlik} \times \text{Duyarlılık}}{(\text{Kesinlik} + \text{Duyarlılık})} \quad (4.7)$$

Karışıklık Matrisi (Confusion Matrix): Her bir sınıf için doğru pozitiflerin, doğru negatiflerin, yanlış pozitiflerin ve yanlış negatiflerin sayılarını göstererek sonuçları ayrıntılı biçimde görmemizi sağlar[149].

Formüllerde belirtilen değerleri detaylandırarak olursak:

TP (Doğru Pozitif): Bir nesnenin tahmin edilen yeri ile gerçek yerinin kesişim alanı, ayarlanan IoU eşik değerinden büyük olduğu durumlarda gerçek pozitifler belirlenir.

FP (Yanlış Pozitif): Bir nesnenin tahmin edilen yeri ile gerçek yerinin kesişim alanı, ayarlanan IoU eşik değerinden küçük olduğu durumlarda model nesneyi tespit etmez. Yanlış Pozitif, modelin tespit etmesi gereken ancak tespit edemediği nesne sayısı olarak ifade edilir.

TN (Doğru Negatif): Görüntü içerisinde tespit edilecek bir nesne olmadığı ve modelinde doğru bir şekilde bir şey tespit etmediği durumların sayısıdır.

FN (Yanlış Negatif) : Nesnenin yanlış sınıf etiketi ile algılandığı durumların sayısını ifade eder.

4.2 Performans Sonuçları

Bu bölümde tez kapsamında oluşturulup test edilen derin öğrenme modellerinin performans değerleri karşılaştırılmıştır. Performans deneylerimiz 2 farklı donanımda test edilmiştir.

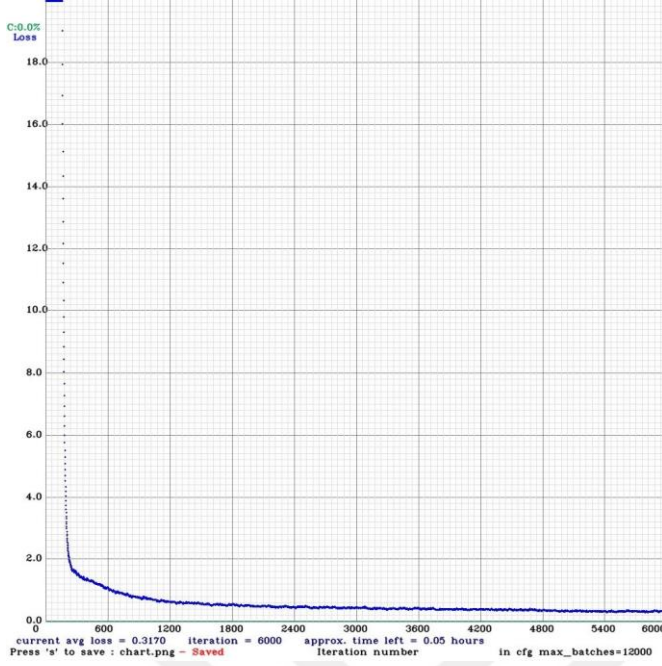
1.Donanım: Intel i5 2. Nesil 2.2 GHz CPU, 2GB GeForce GT 540M GPU ve 8 GB RAM

2.Donanım: Intel i7 12.Nesil 3.6 GHz CPU, RTX 3060 12GB GPU ve 16 GB RAM

Deney sonuçları tablolarında bulunan CPU değerleri 1. donanım ile test edilen CPU değerleri, GPU değerleri ise 2. donanım ile test edilen GPU değerlerini göstermektedir.

4.2.1 YOLO V4 Tiny Deney Sonuçları

YOLO V4 Tiny modeli Google Colab sanal ortamında yaklaşık 15 saat sürede eğitilmiştir. Öğrenme işlemi 6000 iterasyon sonunda durma noktasına geldiği için eğitim bu noktada tamamlanmıştır. İterasyon sayılarına göre kayıp fonksiyonu değerleri şekil 4.1 de gösterilmiştir.



Şekil 21.1 YOLO V4 Tiny eğitim grafiği

Eğitim öncesi ayarlanan bazı önemli konfigürasyon değerleri şu şekildedir:

Width, Height: 416

Epoch: 100

Batch: 64

Subdivisions: 16

Saturation: 1.5

Exposure: 1.5

Filtre Sayısı: Genel kullanımda (sınıf sayısı + 5) x 3 olarak hesaplanmaktadır. Çalışmamızda 6 sınıf olduğu için filtre sayısı 33 olarak ayarlanmıştır.

Max_Batches: Genel kullanımda (sınıf sayısı x 2000) olarak ayarlanmaktadır. Çalışmamızda 12000 olarak ayarlanmıştır.

Steps: Max_Batches değerinin %80 ve %90 ı olarak ayarlanmaktadır. Çalışmamızda 9600, 10800 olarak ayarlanmıştır.

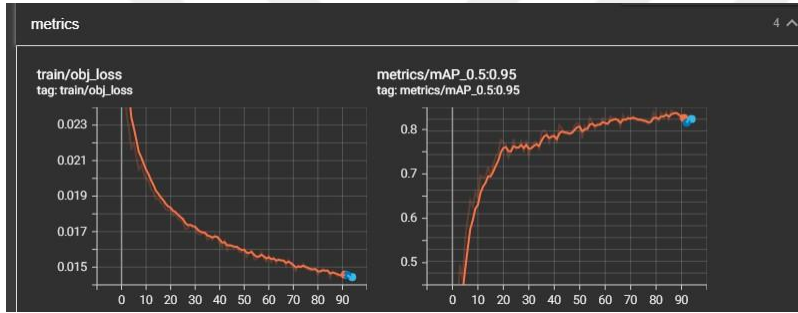
Çizelge 4.1 de YOLO V4 Tiny modeline ait sonuçlar gösterilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
YOLO V4 Tiny	6.1	81	91	85.7	79	46.2	410	25

Çizelge 4.1 YOLO V4 Tiny deneysel sonuçları

4.2.2 YOLO V5s Deneysel Sonuçları

YOLO V5s Google Colab sanal ortamında yaklaşık 17 saat sürede eğitilmiştir. Öğrenme işlemi 103 epoch sonunda durma noktasına geldiği için eğitim bu noktada tamamlanmıştır. Epoch sayılarına göre mAP ve kayıp fonksiyonu değerleri şekil 4.2 de gösterilmiştir.



Şekil 22.2 YOLO V5s eğitim sonuçları

YOLO V5s modeli ağ yapısında parametre sayısı, katman sayısı, görüntü boyutu, yığın boyutu, aktivasyon fonksiyonları gibi değerler YOLO V4 Tiny modelinden farklılık göstermektedir. Eğitim ortamında belirli önemli adımların ekran görüntüleri birleştirilerek Şekil 4.3 de gösterilmiştir. Ekran görüntüleri eğitimin başlangıç aşamasında, ilk epoch, 50. Epoch ve 103. Epoch sonunda alınarak birleştirilmiştir.

```
python train.py --img 640 --batch 16 --epochs 500 --data /content/drive/MyDrive/yolov5_thesis/data.yaml --cfg /content/drive/MyDrive/yolov5_thesis/configs/default.yaml
```

Model Summary: 270 layers, 7030417 parameters, 7030417 gradients

Starting training for 500 epochs...

Epoch	gpu_mem	box	obj	cls	labels	img_size
0/499	3.74G	0.09233	0.03629	0.03455	26	640: 100% 239/239 [03:19<00:00, 1.20it/s]
	Class	Images	Labels	P	R	mAP@.5 mAP@.5:.95: 100% 16/16 [00:08<00:00, 0.264 0.0871]
	all	486	1444	0.519	0.258	
1/499	4.64G	0.06504	0.02875	0.01733	23	640: 100% 239/239 [03:16<00:00, 1.22it/s]
	Class	Images	Labels	P	R	mAP@.5 mAP@.5:.95: 100% 16/16 [00:08<00:00, 0.642 0.641 0.637 0.257]
	all	486	1444	0.642	0.641	
49/499	4.64G	0.02951	0.01701	0.001236	45	640: 100% 239/239 [03:16<00:00, 1.22it/s]
	Class	Images	Labels	P	R	mAP@.5 mAP@.5:.95: 100% 16/16 [00:08<00:00, 0.849 0.479]
	all	486	1444	0.912	0.803	
50/499	4.64G	0.02911	0.01693	0.001393	28	640: 100% 239/239 [03:16<00:00, 1.22it/s]
	Class	Images	Labels	P	R	mAP@.5 mAP@.5:.95: 100% 16/16 [00:08<00:00, 0.866 0.477]
	all	486	1444	0.905	0.833	
103/499	0G	0.02535	0.0151	0.0008633	26	640: 100% 239/239 [1:05:41<00:00, 16.49s/it]
	Class	Images	Labels	P	R	mAP@.5 mAP@.5:.95: 100% 16/16 [02:04<00:00, 7.79s/it]
	all	486	1444	0.949	0.826	0.869 0.494

Şekil 23.3 YOLO V5 Eğitim sürecine ait görüntüler

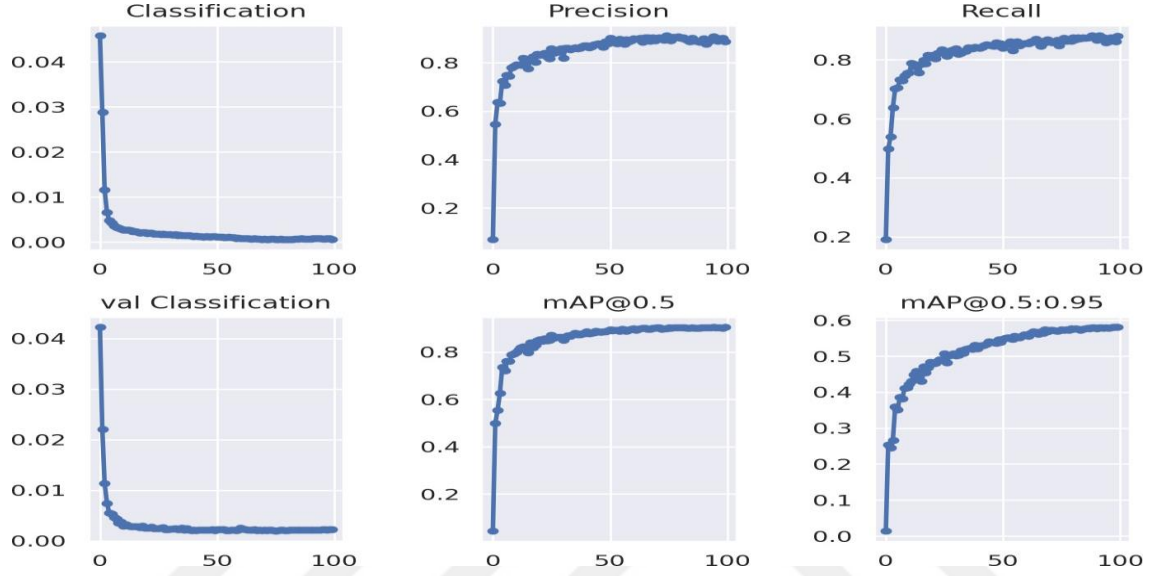
YOLO V5 modeli YOLO V5n, YOLO V5s, YOLO V5m, YOLO V5l ve YOLO V5x olmak üzere 5 farklı versiyon ile sunulmuştur. Çalışmamızda YOLO V5 in 270 katmanlı ve 7 milyon parametrelili olan YOLO V5s modeli test edilmiş ve modele ait sonuçlar Çizelge 4.2 de gösterilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
YOLOV5s	7	82.6	94.9	88,3	86.9	49.4	430	33

Çizelge 4.2 YOLO V5s modeli deneysel sonuçları

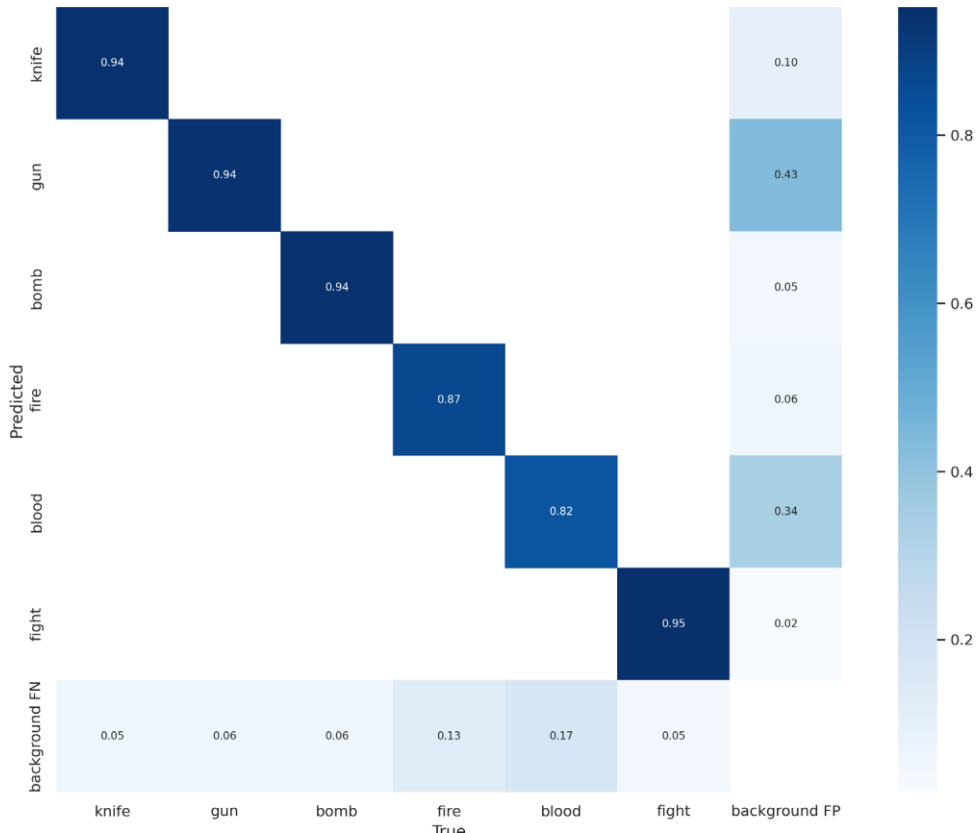
4.2.3 YOLO V7 Deneysel Sonuçları

YOLO V7 Google Colab sanal ortamında yaklaşık 25 saat sürede eğitilmiştir. Öğrenme işlemi 100 epoch sonunda durma noktasına geldiği için eğitim bu noktada tamamlanmıştır. Epoch sayılarına göre eğitim sonuçları şekil 4.4 de gösterilmiştir.



Şekil 24.4 YOLO V7 eğitim sonuç grafikleri

Modelin her bir sınıf için tahmin çıkarımını gösteren karışıklık matrisi Şekil 4.5 te gösterilmiştir. Silah, bıçak, bomba ve kavga sınıfları için tespit yeteneği yüksek gözükürken ateşli ve kanlı sahneler için tespit yeteneği de yeterli düzeydedir. Bu sınıfların eğitim sırasında veri sayısı ve çeşitliliği artırılarak benzer sonuçların elde edilmesi mümkündür.



Şekil 25.5 YOLO V7 karışıklık matrisi

YOLO V7 modeli YOLO V7 Tiny, YOLO V7, YOLO V7 – X, YOLO V7 – P6, YOLO V7 – W6, YOLO V7 – E6, YOLO V7- D6 ve YOLO V7 – E6E versiyonları ile sunulmuştur. Bu modellerin katman ve parametre sayısı arttığında modelin doğru tespit başarısı artarken tespit süresi de uzamaktadır. Çalışmamızda YOLO V7 in 36.5 milyon parametrelili olan YOLO V7 temel modeli test edilmiştir. Örnek bir video da yapılan test ortamı şekil 4.6 te gösterilmiştir.


```

video 1/1 (262/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (2096.0ms) Inference, (0.0ms) NMS
video 1/1 (263/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (3723.9ms) Inference, (1.0ms) NMS
video 1/1 (264/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (1876.0ms) Inference, (1.0ms) NMS
video 1/1 (265/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (1758.6ms) Inference, (1.0ms) NMS
video 1/1 (266/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (3634.3ms) Inference, (2.0ms) NMS
video 1/1 (267/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (2082.8ms) Inference, (1.0ms) NMS
video 1/1 (268/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (1781.1ms) Inference, (1.0ms) NMS
video 1/1 (269/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (3125.4ms) Inference, (2.0ms) NMS
video 1/1 (270/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (2618.2ms) Inference, (1.0ms) NMS
video 1/1 (271/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (1760.0ms) Inference, (1.1ms) NMS
video 1/1 (272/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (3744.1ms) Inference, (1.0ms) NMS
video 1/1 (273/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (2773.9ms) Inference, (1.0ms) NMS
video 1/1 (274/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (1870.4ms) Inference, (1.0ms) NMS
video 1/1 (275/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (3084.1ms) Inference, (1.0ms) NMS
video 1/1 (276/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (2595.8ms) Inference, (2.0ms) NMS
video 1/1 (277/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (1773.4ms) Inference, (1.0ms) NMS
video 1/1 (278/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (3482.9ms) Inference, (5.3ms) NMS
video 1/1 (279/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (3123.6ms) Inference, (1.0ms) NMS
video 1/1 (280/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (1695.8ms) Inference, (0.0ms) NMS
video 1/1 (281/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (4671.0ms) Inference, (6.0ms) NMS
video 1/1 (282/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 2 bloods, Done. (2210.2ms) Inference, (2.0ms) NMS
video 1/1 (283/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (4518.0ms) Inference, (1.0ms) NMS
video 1/1 (284/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (3691.3ms) Inference, (2.0ms) NMS
video 1/1 (285/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (4121.6ms) Inference, (5.0ms) NMS
video 1/1 (286/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (3470.0ms) Inference, (1.0ms) NMS
video 1/1 (287/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (1891.4ms) Inference, (1.0ms) NMS
video 1/1 (288/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (4624.0ms) Inference, (1.0ms) NMS
video 1/1 (289/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (1997.4ms) Inference, (0.0ms) NMS
video 1/1 (290/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (2718.7ms) Inference, (1.0ms) NMS
video 1/1 (291/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (3864.2ms) Inference, (1.0ms) NMS
video 1/1 (292/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (2589.9ms) Inference, (1.0ms) NMS
video 1/1 (293/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, Done. (3720.2ms) Inference, (2.2ms) NMS
video 1/1 (294/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (2257.1ms) Inference, (1.0ms) NMS
video 1/1 (295/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 blood, 1 fight, Done. (1877.6ms) Inference, (1.0ms) NMS
video 1/1 (296/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (3637.2ms) Inference, (2.0ms) NMS
video 1/1 (297/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: 1 fight, Done. (2706.3ms) Inference, (1.0ms) NMS
video 1/1 (298/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (1923.6ms) Inference, (1.0ms) NMS
video 1/1 (299/597) C:\Users\Edip\Desktop\yolov7\test_images\test2.mp4: Done. (3577.5ms) Inference, (0.0ms) NMS

```

Şekil 26.6 YOLO V7 modeli CPU ortamında gerçek zamanlı çalışma örneği

Modelin parametre sayısı YOLO V7 Tiny gibi hafif modellere göre 6 kat fazla olduğu için daha yüksek çıkarım yapabilmekte ancak bir görüntü için CPU ortamında yaklaşık 2-4 saniyeye aralığında süreye ihtiyaç duymaktadır. Buda gerçek zamanlı oyunlarda donanımda donmalara neden olmaktadır. YOLO V7 modele ait sonuçlar Çizelge 4.3 de gösterilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
YOLO V7	36.5	88	88	88	91	58	2500	150

Çizelge 4.3YOLO V7 modeli deneysel sonuçları

4.2.4 SSD Deneysel Sonuçları

SSD MobileNet 300 modeli RTX 3060 GPU kullanılarak yaklaşık 18 saatte eğitilmiştir.

Eğitim aşamasında ayarlanan bazı konfigürasyonlar şu şekildedir:

Epoch: 100

Görüntü boyutu: 300

Batch boyutu: 32

Derinlik: 128

Num_steps: 50000

min_score_thresh: 0.5

iou_theshold: 0.6

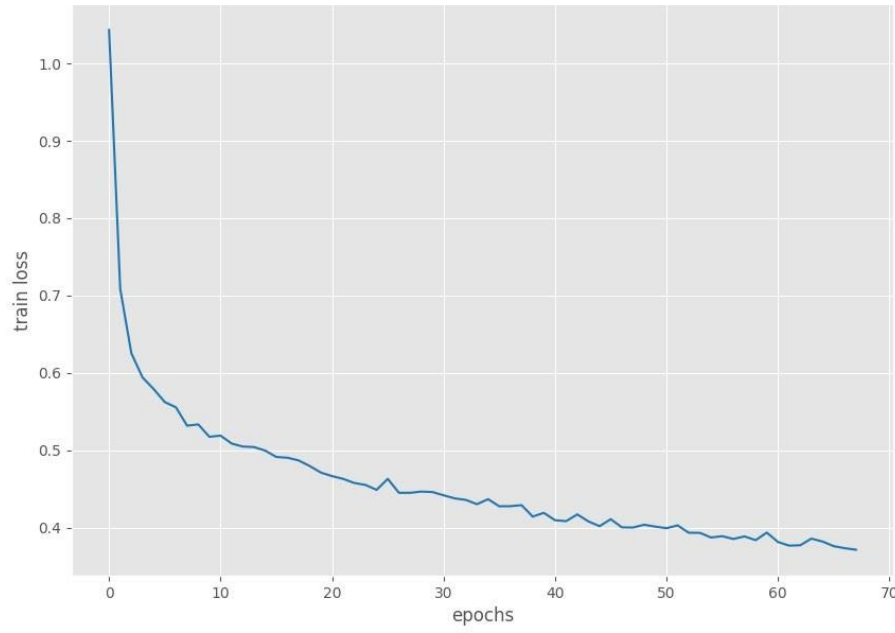
SSD MobileNet 300 modeli 267 katman ve 15 milyon parametre içermektedir. SSD; VGG16 AlexNet vb. omurga yapıları ile de kullanılmaktadır. MobileNet tercih edilmesinin en önemli sebebi daha hafif bir ağ yapısına sahip olmasıdır. MobilNet ağ yapısı derinlemesine ayrılabilir evrişimler kullanabildiğinden aynı derinliğe sahip evrişim yapısına sahip diğer ağlar ile karşılaştırıldığında parametre sayısında önemli ölçüde azalmalar görülmektedir[150]. Hızlı çıkarım yapabilen SSD modeli için deneysel sonuçlar Çizelge 4.4 te verilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
SSD	15	80	83	81,4	77,6	44,1	380	37

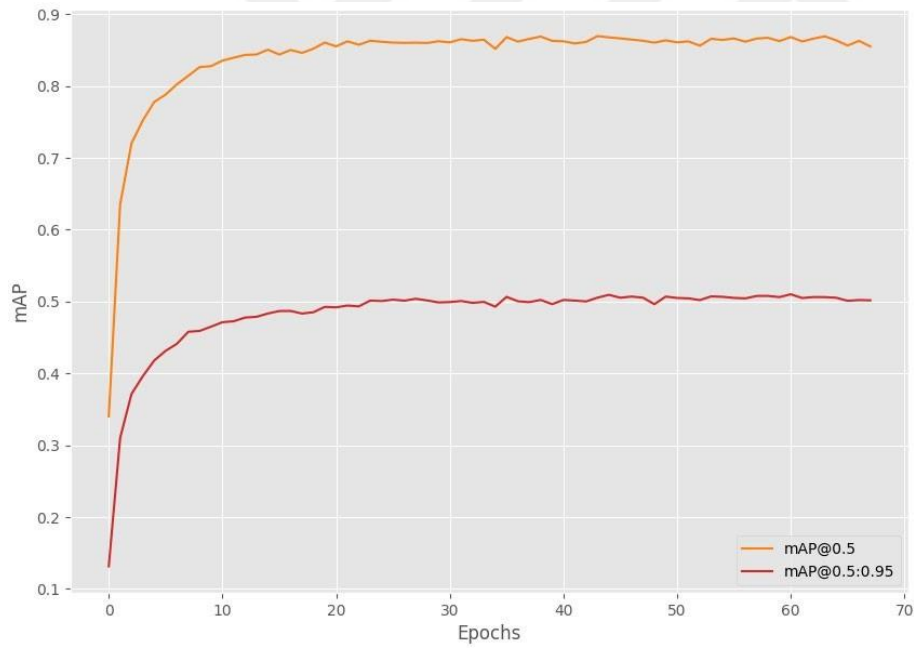
Çizelge 4.4 SSD modeli deneysel sonuçları

4.2.5 Faster R-CNN Deneysel Sonuçları

Faster R-CNN modeli RTX 3060 GPU kullanılarak yaklaşık 22 saatte eğitilmiştir. Eğitim işlemi 68 epoch sonunda durma noktasına geldiği için sonlandırılmıştır. Eğitim aşamasına ait mAP ve kayıp fonksiyon grafikleri Şekil 4.7 ve Şekil 4.8 de gösterilmiştir.



Şekil 27.7 Faster R-CNN modeline ait epoch – kayıp fonksiyon grafiği



Şekil 28.8 Faster R-CNN modeline ait epoch - mAP grafiği

Faster R-CNN modeli mAP 0.95 tahmini başarılı sonuçlar verse de bu model içerdiği katman sayısı, parametre sayısı ve 2 aşamalı nesne tanıma yaptığından diğer test edilen modellere göre en yavaş tespit hızına sahip modeldir. Faster R-CNN modeline ait sonuçlar Çizelge 4.5 te gösterilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
Faster R-CNN	42	85	87	86	89,9	50,6	3200	210

Çizelge 4.5 Faster R-CNN modeli deneysel sonuçları

4.2.6 YOLO V8n ve YOLO V8 Geliştirilen Model Deneysel Sonuçları

Dijital oyunlar genellikle yüksek performans gereksinime sahip donanımlara ihtiyaç duyan yazılımlardır. Bu gereksinimlerin yanı sıra derin öğrenme modelleri ile aynı anda çalıştıkları zaman donanımda zorlanma ve donmalar meydana gelmektedir. Bu sebeple incelenen ve test edilen yapay zekâ modellerinden daha az sayıda katman ve parametre içeren bir modelin tasarlanmasına karar verildi. Bu kapsamda YOLO V8'in en hafif modeli olan YOLO V8n in mimari yapısını nesne algılama performansını çok fazla etkilemeden daha az sayıda katman ve parametreye sahip bir yapıya dönüştürülmeye çalışıldı.

İlk olarak YOLO V8n versiyonunu 100 epoch döngü ile varsayılan hiperparametrelerle eğitildi. Daha sonra, doğrulama setindeki çıkarım hızı ile mAP 50-95 arasındaki denge dikkate alınarak modelin omurgasında çeşitli değişiklikler yapıldı. YOLO V8n mevcut yapısında 225 katman ve 3.157.184 adet parametre bulunmaktadır. Katman ve parametre sayısını düşürmek için YOLO V8n modelinin standart derinlik parametresi 0,33 den 0,2 ye ve genişlik parametresi 0.25 den 0.2 ye daraltıldı. Bu daraltma evrişim katmanlarında hesaplanan parametre değerlerini düşürmesine neden olmaktadır. Şekil 4.9 da YOLO V8 modelinin ağ derinlik, genişlik ve kanal parametre değerleri ile çalışmamızda ayarlanan değerler gösterilmiştir.

```
# Parameters
nc: 6 # number of classes
names: ['knife', 'gun', 'bomb', 'fire', 'blood', 'fight'] # class names
scales: # model compound scaling constants, i.e. 'model=yolov8n.yaml' will call yolov8.yaml with scale 'n'
# [depth, width, max_channels]
ours: [0.2, 0.2, 1024] # YOLOv8 ours summary: 211 layers, 1440298 parameters, 1440300 gradients,
n: [0.33, 0.25, 1024] # YOLOv8n summary: 225 layers, 3157200 parameters, 3157184 gradients,
s: [0.33, 0.50, 1024] # YOLOv8s summary: 225 layers, 11166560 parameters, 11166544 gradients,
m: [0.67, 0.75, 768] # YOLOv8m summary: 295 layers, 25902640 parameters, 25902624 gradients,
l: [1.00, 1.00, 512] # YOLOv8l summary: 365 layers, 43691520 parameters, 43691504 gradients,
x: [1.00, 1.25, 512] # YOLOv8x summary: 365 layers, 68229648 parameters, 68229632 gradients,
```

Şekil 29.9 YOLO V8 modeli ağın derinlik, genişlik ve kanal parametre değerleri

Omurga yapısında ise evrişim, C2f ve SPPF bloklarının belirlenen parametre değerleri yarıya indirilerek model 211 katmanlı ve 1.440.298 parametrelili bir yapıya dönüştürüldü. Böylelikle daha hafif bir yapı elde edildi. Şekil 4.10 da YOLO V8 omurgası ile geliştirilen model omurgası verilmiştir.

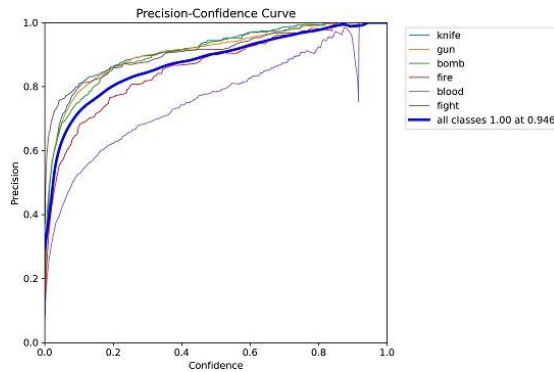
```
# YOLOv8.0n backbone
backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
  - [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]] # 3-P3/8
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, Conv, [512, 3, 2]] # 5-P4/16
  - [-1, 6, C2f, [512, True]]
  - [-1, 1, Conv, [1024, 3, 2]] # 7-P5/32
  - [-1, 3, C2f, [1024, True]]
  - [-1, 1, SPPF, [1024, 5]] # 9
```

```
# YOLOv8.0n backbone ours
backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
  - [-1, 1, Conv, [64, 3, 2]] # 1-P2/4
  - [-1, 3, C2f, [64, True]]
  - [-1, 1, Conv, [128, 3, 2]] # 3-P3/8
  - [-1, 6, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]] # 5-P4/16
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, Conv, [512, 3, 2]] # 7-P5/32
  - [-1, 3, C2f, [512, True]]
  - [-1, 1, SPPF, [512, 5]] # 9
```

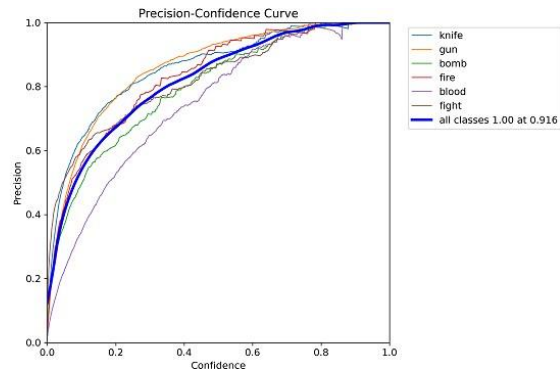
Şekil 30.10 a) YOLO V8n omurgası
omurgası

b) Geliştirilen YOLO V8 model

Eğitim aşamasında veri setinde bulunan 6256 görüntünün 4379 tanesi eğitim, 626 tanesi test ve 1251 tanesi doğrulama için kullanılmıştır. Eğitim ortamı Nvidia Geforce RTX 3060 GPU kullanılarak gerçekleştirilmiştir. Eğitim süresi yaklaşık 21 saat sürmüştür. Modellerin test işlemi gerçek zamanlı oyun ortamında, dijital oyun video ve görüntüleri kullanılarak test edilmiştir. Test sonuçlarına bakıldığında YOLO V8n %94,6 ve geliştirilen model %91,6 kesinlik değeri ile tahmin yapmaktadır. Geliştirilen modelin tespit performansı YOLO V8n den bir miktar düşük olsa da yaklaşık 2 kat hızlı tespit yeteneğine sahiptir. Şekil 4.11 de YOLO V8n ve geliştirilen modelin kesinlik grafikleri verilmiştir.

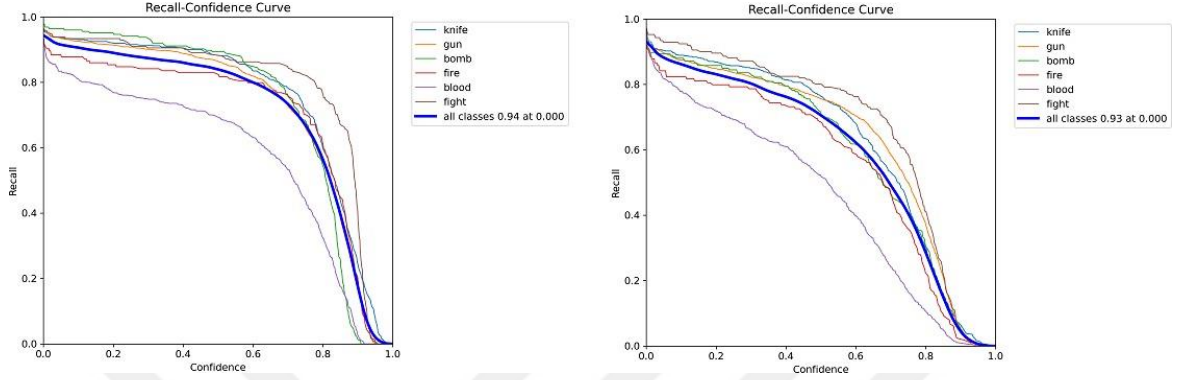


Şekil 31.11 a) YOLO V8n kesinlik grafiği



b) Geliştirilen model kesinlik grafiği

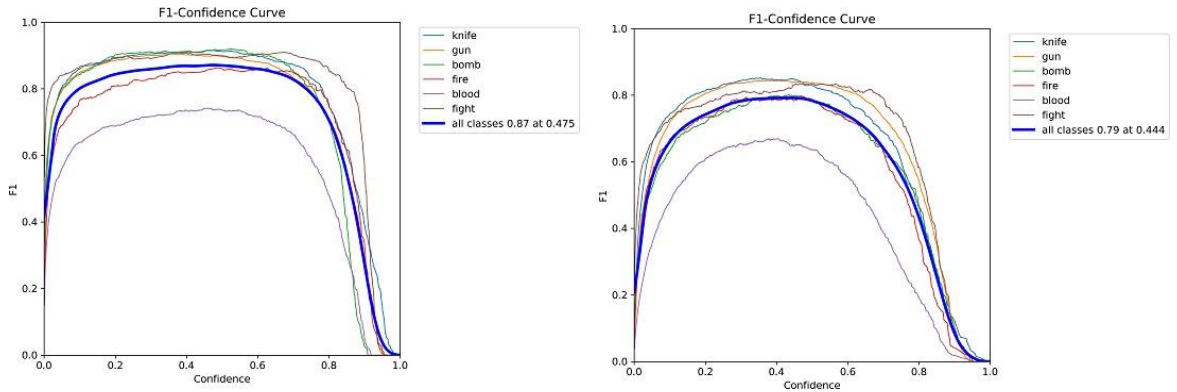
Duyarlılık sonuçlarına bakıldığında YOLO V8n %94 ve geliştirilen model % 93 duyarlılık değeri ile tahmin yapmaktadır. Şekil 4.12 de YOLO V8n ve geliştirilen modelin duyarlılık grafikleri verilmiştir.



Şekil 32.12 a) YOLO V8n duyarlılık grafiği

b) Geliştirilen model duyarlılık grafiği

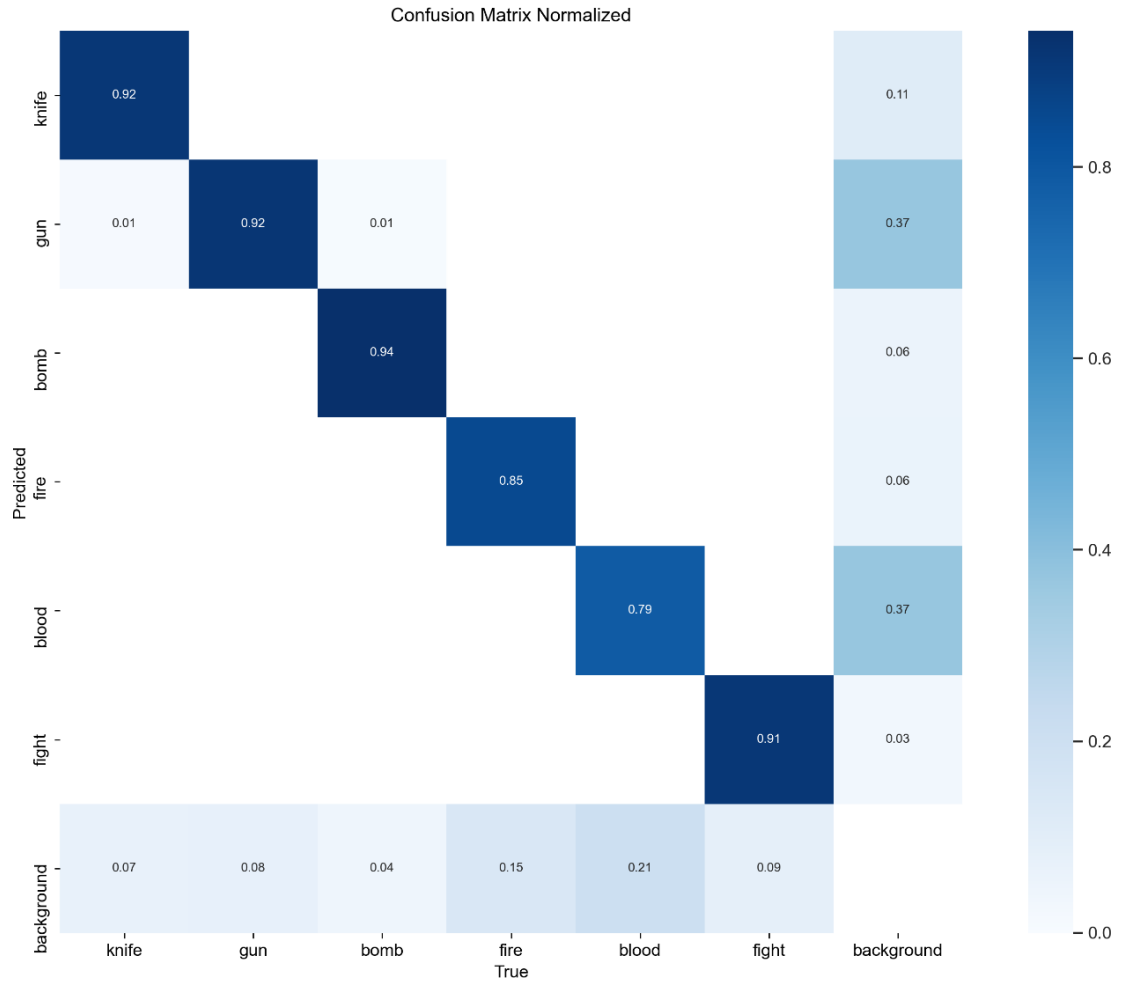
F1 değerlerine bakıldığında ise YOLO V8n %87 değerine sahipken geliştirilen model %79 F1 değerine sahiptir. Şekil 4.13 de YOLO V8n ve geliştirilen modelin F1 değer grafikleri verilmiştir.



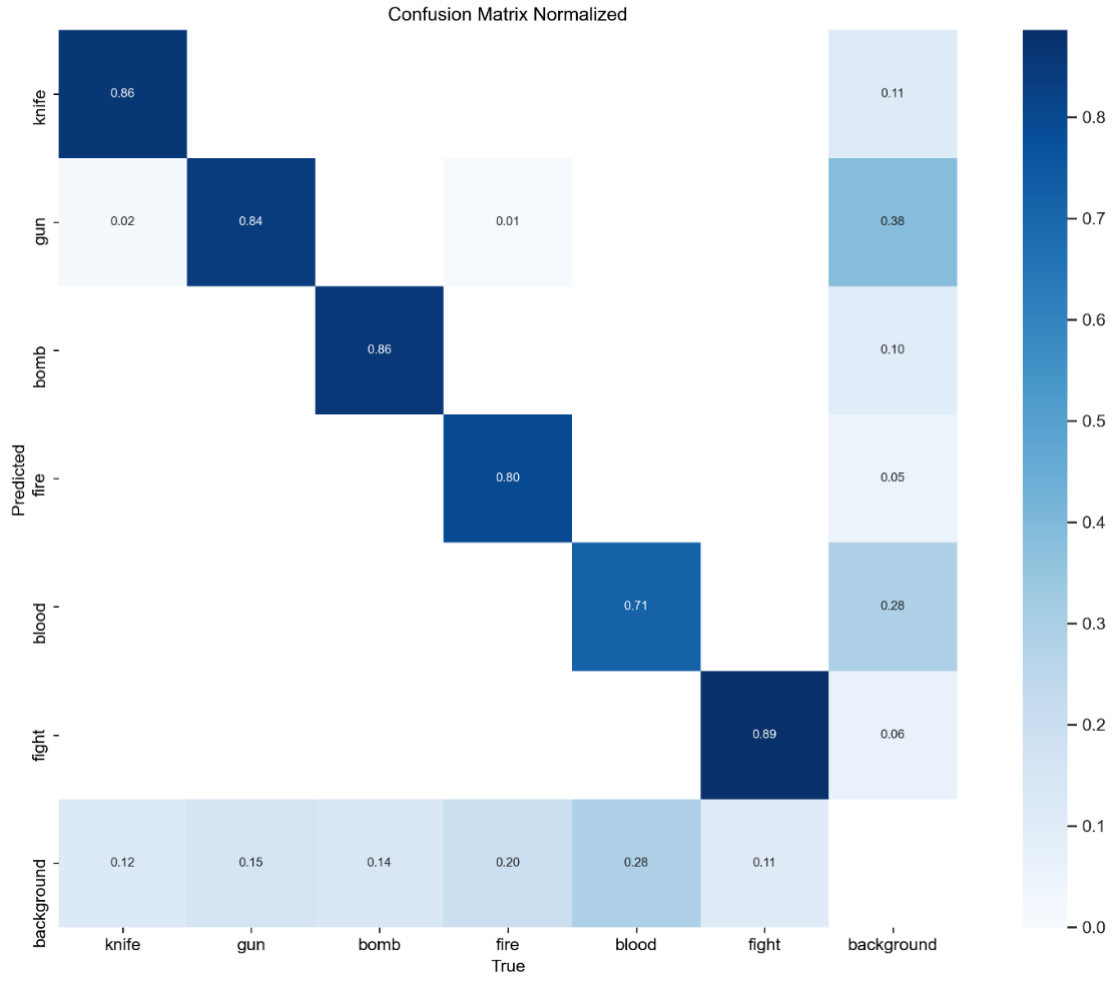
Şekil 33.13 a) YOLO V8n F1 değer grafiği

b) Geliştirilen model F1 değer grafiği

YOLO V8n ve geliştirilen modelin karışıklık matrislerine bakıldığında bütün sınıflar için yapılan tahminlerin birbirine yakın değerlere sahip olduğu gösterilmiştir. Şekil 4.14 de YOLO V8n karışıklık matrisi ve 4.15 de YOLO V8 geliştirilen model karışıklık matrisi gösterilmiştir.



Şekil 34.14 YOLO V8n karışıklık matrisi



Şekil 35.15 YOLO V8 geliştirilen model karışıklık matrisi

YOLO V8n modeline ait sonuçlar Çizelge 4.6 te gösterilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
YOLO V8n	3.15	94,6	94	87	87	50,3	330	17

Çizelge 4.6 YOLO V8n modeli deneysel sonuçları

YOLO V8 geliştirilen modeline ait sonuçlar Çizelge 4.7 te gösterilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
YOLO V8n Geliştirilen Model	1.44	91.6	93	79	83	49,5	210	9

Çizelge 4.7 Geliştirilen YOLO V8n modeli deneysel sonuçları

Çalışma kapsamında test edilen tüm modellerin performans çizelgesi Çizelge 4.8 de gösterilmiştir.

Model	Parametre Sayısı (Milyon)	Kesinlik	Duyarlılık	F-1 Değeri	mAP (0.5)	mAP (0.95)	CPU Hız (ms)	GPU Hız (ms)
YOLO V4 Tiny	6.1	81	91	85.7	79	46.2	410	25
YOLO V5s	7	82.6	94.9	88,3	86.9	49.4	430	33
YOLO V7	36.5	88	88	88	91	58	2500	150
Faster RCNN	42	85	87	86	89,9	50,6	3200	210
SSD	15	80	83	81,4	77,6	44,1	380	37
YOLO V8n	3.15	94,6	94	87	87	50,3	330	17
Geliştirilen YOLO V8 Model	1.44	91.6	93	79	83	49,5	210	9

Çizelge 4.8 Test edilen modellerin performans değerleri

Doğru Tespit Örnekleri

Geliştirilen modelin farklı test ortamlarından elde edilen test örnekleri Görüntü 4.1 de gösterilmiştir. Model eğitim veri setinde kullanılan dijital oyunlarda oldukça başarılı tahmin performansı göstermektedir.



Görüntü 4.1 Modelin doğru tespit örnekleri

Başarısız Tespit Örnekleri

Modellerin başarı performansını etkileyen unsurlar arasında çözünürlük, bulanıklık, arka plan, açı vb. nedenler ön plana çıkmaktadır. Görüntü 4.2 de modelin tespit yapamadığı görüntülere ait örnekler verilmiştir. Model gerçek zamanlı dijital oyunlarda bu görüntü karelerinde tespit edemediği şiddet unsurlarını bir sonraki görüntü karesinde tespit ederek başarısız tespitleri telafi etmektedir.



Görüntü 4. 2 Modelin başarısız tespit örnekleri

Farklı Oyunlardan Doğru Tespit Örnekleri

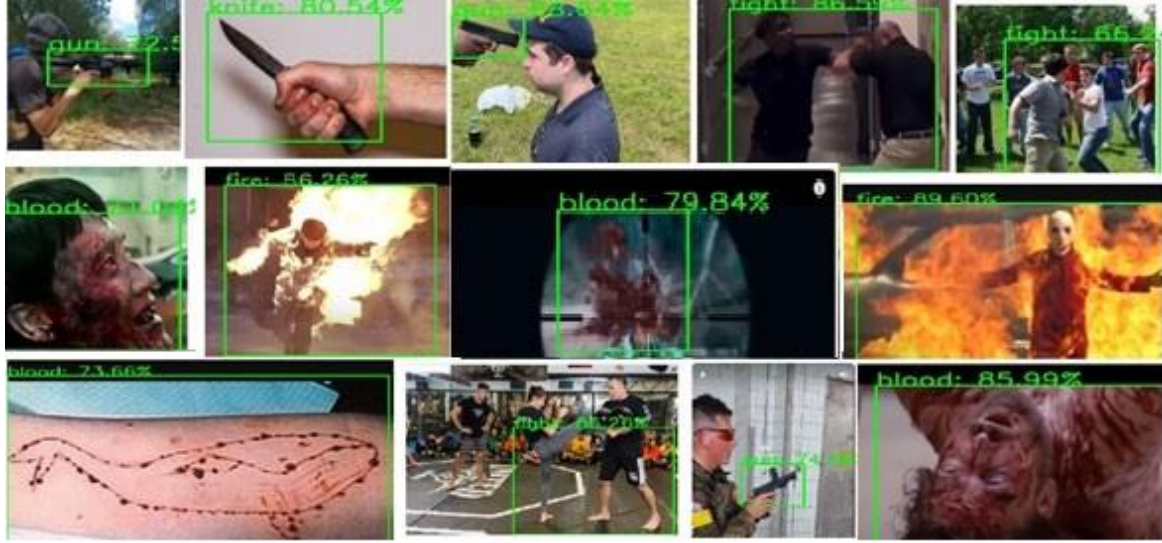
Modelin benzer şiddet görüntüleri içeren ve eğitim görüntülerinde bulunmayan farklı dijital oyunlarda da başarılı sonuçlar elde edebildiği test edilmiştir. Görüntü 4.3 de modelin farklı oyunlar üzerinde gösterdiği performans gösterilmiştir.



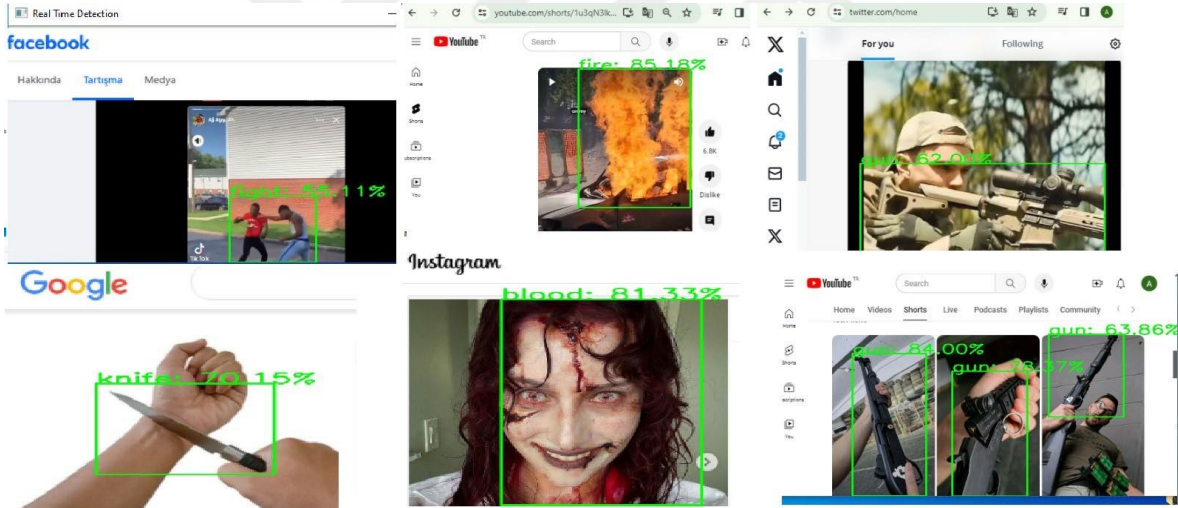
Görüntü 4. 3 Modelin farklı oyunlardaki başarılı tespit örnekleri

Gerçek Dünya Görüntülerinden Doğru Tespit Örnekleri

Çalışmamızda oluşturulan veri seti gerçek dünya görüntülerine benzer içerikli dijital oyunlardan elde edilerek oluşturulmuştur. Bu veri seti ile eğitilerek geliştirilen model gerçek zamanlı ekran analizi yapabildiğinden gençlerin ve çocukların bilgisayarında kayıtlı şiddet içerikli film, video veya Youtube, Instagram, Facebook vb. sosyal medya içeriklerini izlerken de devreye girip benzer şekilde kullanıcı ve aileleri ile etkileşime geçebilmektedir. Görüntü 4.4 de modelin gerçek dünya içeriklerde, Görüntü 4.5 de ise sosyal medya içeriklerinde performansı gösterilmiştir.



Görüntü 4.4 Modelin gerçek dünya görüntülerinden doğru tespit örnekleri



Görüntü 4.5 Modelin sosyal medya görüntülerinden doğru tespit örnekleri

5. ŞİDDET ANALİZİ

Şiddet içerikli dijital oyunlar günlük belirli süre oynatılan çocuk ve gençlerde bağımlılık, saldırganlık, yalnızlaşma, depresyon, intihar vb. birçok belirtilerin gözlemlendiği üzerine yapılan çalışmalar mevcuttur.[26] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160]. Bu kapsamda yapılacak şiddet analizi oldukça önem taşımaktadır. Özellikle çocuk yaştan itibaren dijital oyun dünyasında zaman geçiren gençlerin kontrolsüz ve ebeveyn bilgisi dışında binlerce şiddet içeriklerine maruz kalmaktadır. Bu bölümde bu kontrolsüzlüğün önüne geçerek kullanıcı ve ebeveyn bilgilendirmeleri ile farkındalık oluşturmak amaçlanmaktadır.

Şiddet analizi dijital oyun süresince tespit edilen tüm şiddet unsurlarının çeşitliliği, yoğunluğu, sınıfı, görüntü içerisindeki konumu ve zamanı kaydedilerek yapılmaktadır. Her bir görüntü içerisinde tespit edilen şiddet içerikleri kaydedildikten sonra bu içeriklerin toplam miktarı hesaplanmaktadır. Ayrıca oyun sonunda tüm bu tespit edilen şiddet içeriklerini grafiksel olarak sunmaktadır. Şekil 5.1 te örnek bir analiz gösterilmiştir.

```
C:\Windows\System32\cmd.exe - python detect.py --weights weights/best.pt --conf 0.55 --img-size 640 --source test_images/test3.mp4 --save-txt --...
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.55, device='', img_size=640, iou_thres=0.45, max_det=1, nosave=False, project='runs', save_conf=False, save_txt=True, source='test_images/test3.mp4', thresh_mail=1, update=False, view_img=False, weights=['weights/best.pt'])
YOLOv8 2022-8-11 torch 1.12.1+cu113 CUDA:0 (NVIDIA GeForce RTX 3060, 12287.375MB)
CUDA:1 (NVIDIA GeForce RTX 3060, 12287.5MB)

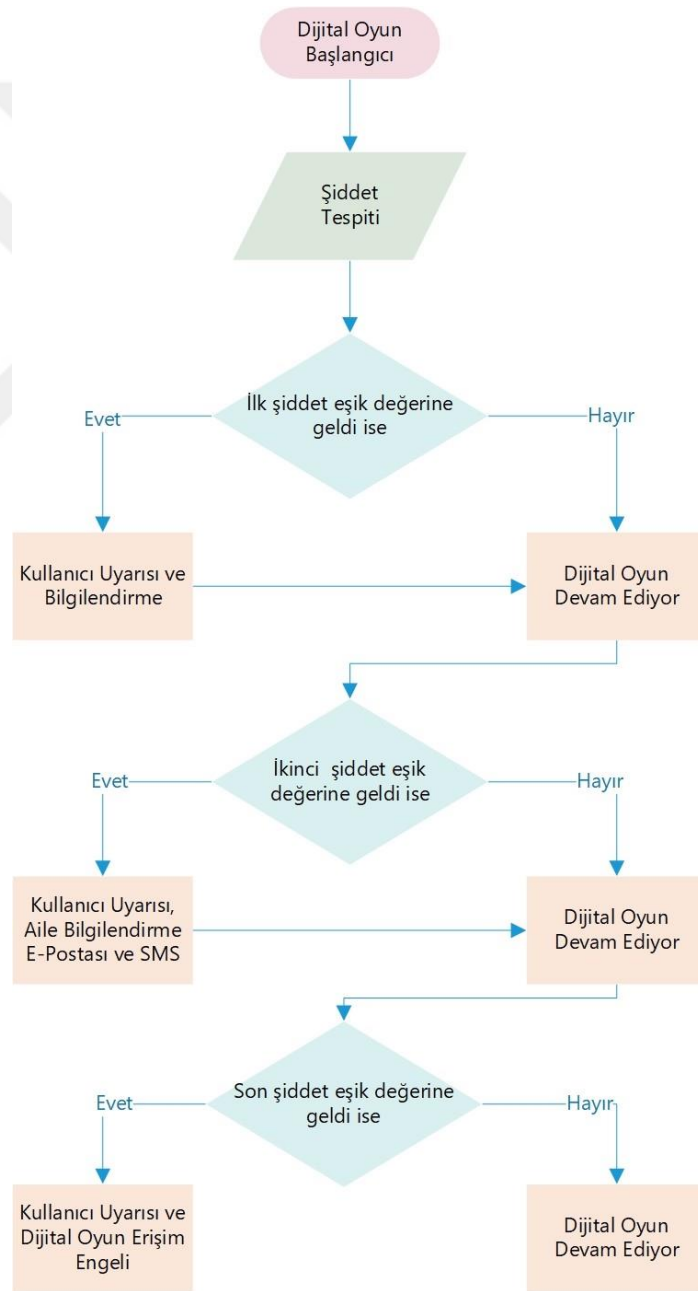
Fusing layers...
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
INDataTart.fuse
AppData\Local\Programs\Python\Python37\lib\site-packages\torch\functional.py:478: UserWarning:
h.meshgrid: in an upcoming release, it will be required to pass the indexing argument. (Triggered internally at C:\acti
ons-runner\work\pytorch\pytorch\builder\windows\pytorch\aten\src\ATen\native\TensorShape.cpp:2895.)
    return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Model Summary: 314 layers, 36508742 parameters, 6194944 gradients, 103.2 GFLOPS
test3.mp4 has knife: 17, gun: 418, bomb: 1, fire: 17, blood: 116, fight: 1
Done. (166.275s)
137.6375
Subject: Warning

Warning: So much violence detected!!!.
knife: 17
gun: 418
bomb: 1
fire: 17
blood: 116
fight: 1
Sent Email.....
```

Şekil 36.1 Modelin analiz ekran görüntüsü

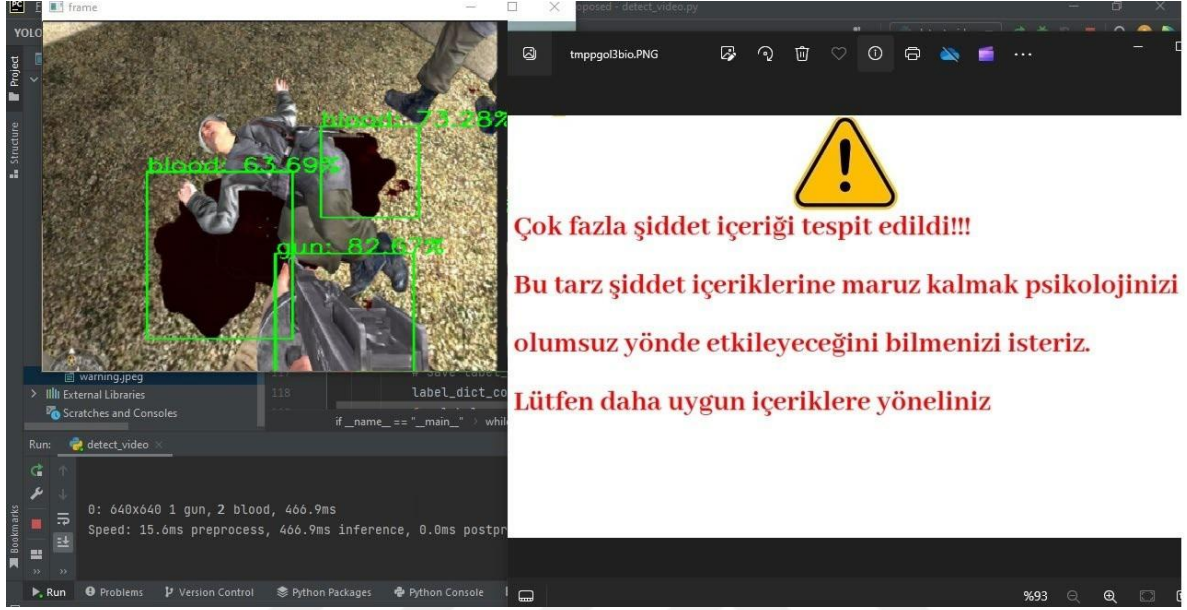
Şiddet Eşik Değeri: Uyarı verilmesine neden olacak minumum şiddet miktarı olarak tanımlayabiliriz. Belirlenen şiddet eşik değeri üstünde tespit edilen şiddet unsurları için e-posta ve mesaj sistemleri ile uyarı verilmektedir. Şiddet eşik değeri kullanıcının yaşına, cinsiyetine, ruh haline, ebeveyn kontrolüne, psikologların önerilerine ve birçok etkene bağlı

olarak değişiklik gösterebilir. Örneğin 6 yaş ve altı çocukların dijital oyun ortamında şiddet içeriklerine maruz kalma durumunun mümkün olmaması gereken bir hipotezde şiddet eşik değerinin 1 olarak ayarlanması gerekmektedir. Bu durumda yapay zekâ modeli tespit ettiği ilk şiddet unsuru için kullanıcı ve ebeveyn bilgilendirmesi yaptıktan sonra oynanan dijital oyunu kapatır. Farklı durumlar için farklı şiddet eşik değerleri tanımlanarak daha esnek durumlar oluşturulabilir. Çalışma kapsamında üç farklı şiddet eşik değeri tanımlanarak bu eşik değerleri kadar şiddet içeriğine maruz kalan kullanıcılar için yapılacak adımlar Şekil 5.2 de gösterilmiştir.



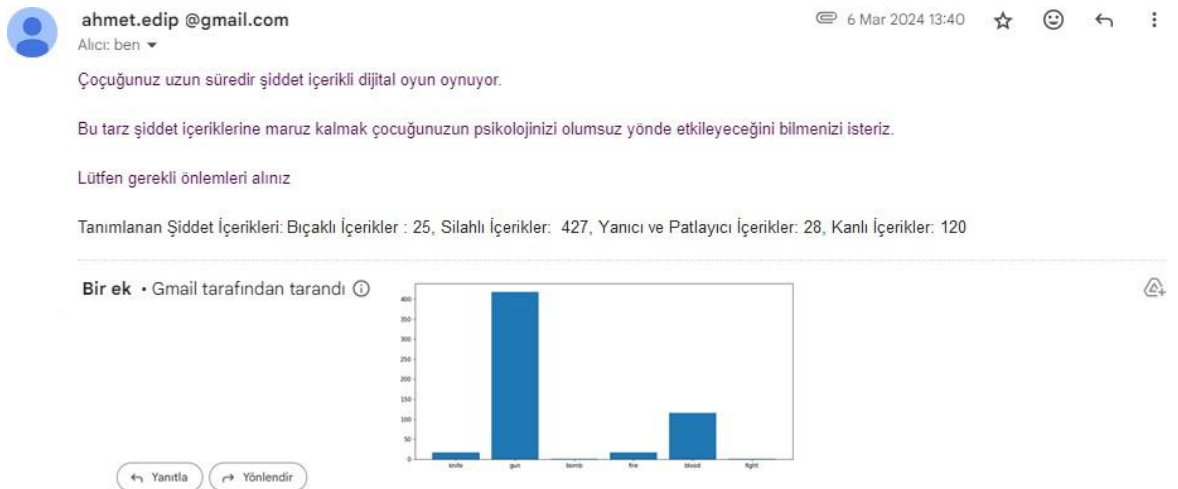
Şekil 37.2 Şiddet analizi aktivite diyagramı

İlk olarak kullanıcı dijital oyun içerisinde birinci eşik değeri kadar şiddet içeriğine maruz kaldığında bilgilendirme metin kutusu ile uyarılmaktadır. Şekil 5.3 de örnek uyarı mesajı gösterilmiştir.



Şekil 38.3 Birinci şiddet eşik değeri için uyarı mesajı

İlk uyarı sonrasında şiddet içerikli dijital oyunu oynamaya devam eden kullanıcılar için ikinci şiddet eşik değeri tanımlanarak bu eşik değeri kadar şiddet içerikleri tespit edildiğinde ikinci bir bilgilendirme uyarısı yapılmaktadır. Ayrıca bu durumun normal olmadığı ailesine bilgilendirme e-postası ile iletilmektedir. Şekil 5.4 de aile bilgilendirme e-posta örneği gösterilmiştir.



Şekil 39.4 Aile bilgilendirme e-postası

Son tanımlanan eşik değeri ise kullanıcının ısrarla oynamaya devam ettiği şiddet içerikli dijital oyunu sonlandırmak için kullanılmaktadır. Son eşik değerine gelen kullanıcılar bilgilendirme mesajının yanı sıra dijital oyun sistemi tarafından sonlandırıldıktan sonra bilgisayar kilitlenerek kullanıcının korunması hedeflenmektedir.



6. SONUÇ VE ÖNERİLER

Dijital oyunların bilinçsiz ve aşırı şekilde oynanması, yaş grubuna bakılmaksızın herkese açık şekilde yaygınlaşması şiddet, bağımlılık, saldırganlık, zorbalık, depresyon gibi birçok problemi de beraberinde getirmektedir. Bu problemlerin her biri ayrı ayrı değerlendirilip üzerinde çalışılması gereken önemli kavramlardır. Literatürde bu problemler ortaya konulmuş olsa da tespit ve müdahale konusunda çalışmalar eksik kalmıştır. Ülkemiz ve dünyada ki duruma bakıldığında ise dijital oyunlarla ilgili genel bir kısıtlamanın olmadığı her yaş grubundan insanların herhangi bir oyunu oynayabildiği görülmektedir. Bu çalışmada derin öğrenme yöntemleri kullanılarak dijital oyunlarda çocukları ve gençleri etkileyen şiddet unsurlarından silah, bıçak, bomba, yangın, kan ve kavga tespiti yapabilen model geliştirilmiştir. İlk olarak mevcut bir veri seti olmadığı için farklı dijital oyunlardan elde edilen video görüntülerini kullanarak 6256 görüntüden oluşan veri seti oluşturulmuştur. Eğitim aşamasında veri setinde bulunan 6256 görüntünün 4379 tanesi eğitim, 626 tanesi test ve 1251 tanesi doğrulama için kullanılmıştır. Veri setindeki görüntülerin tamamının etiketlenmesi sonucu 18704 etiketli görüntü elde edilmiştir. Bu etiketli görüntüler 9922 silah, 2543 bıçak, 1191 bomba, 1010 ateş, 2934 kan ve 1104 kavga sınıfına ait nesneleri içermektedir.

Günümüzde yapay zekâ tabanlı nesne tanıma yöntemlerinin başarılı sonuçlar verdiğinden şiddet içeriklerinin tespit ve analizi derin öğrenme yöntemleri ile gerçekleştirilmiştir. Dijital oyunlar ve yapay zekâ modellerinin yüksek donanım gereksiniminden dolayı gerçek zamanlı oyunlarda birlikte çalıştıkları zaman donanım yetersizliği problemleri ile karşılaşmaktadır. Bu problemi en az seviyeye indirmek için daha az katman ve parametre değerine sahip yeni bir yöntem geliştirilmiştir. Geliştirilen modelin ağ yapısı daha hafif hale getirildiğinden modelin tespit yeteneğinde telafi edilebilir düşüşler gözlemlenmiştir. Fakat gerçek zamanlı dijital oyunlarda kullanıldığında modelin bir görüntü karesinde tespit edemediği şiddet unsurunu bir sonraki görüntü karesinde de analiz edileceğinden saniyeler içerisinde bu şiddet unsurlarının tespit edilmesi mümkün hale gelmektedir. Buda modelin gerçek zamanlı oyunlarda daha yüksek tahmin yeteneğine ulaşacağını göstermektedir. Geliştirilen model ile 6 farklı nesne tanıma modelinin performansları karşılaştırılmıştır. Performans sonuçları test

edilen modeller içerisinde geliştirilen modelin en hızlı nesne tanıma hızına sahip olduğu gösterilmiştir. Gerçek zamanlı oyun ortamı, dijital oyun video ve görüntüleri ile testler gerçekleştirilerek geliştirilen yöntemin hız ve doğruluk değerleri ile gerçek zamanlı oyunlar için kullanılabileceğini gösterilmiştir.

Çalışmamız özellikle dijital oyunlarda şiddet eylemlerinin çok fazla artması ve bu durumun toplumlara yansması sonucu şiddet eğilimlerinden etkilenen toplumlar için bir farkındalık oluşturması açısından önemlidir. Aileler dijital oyunlar içerikleri hakkında yeterli bilgi ve deneyime sahip olmadıkları için çocuklarının ne tür oyunlar oynadıkları ve nasıl bir şiddete maruz kaldıkları hakkında da bilgi sahibi değillerdir. Çalışmamızda tespit edilen her şiddet içeriği kaydedilerek sınıflandırılmaktadır. Belirli bir şiddet eşik değerinin üstünde şiddete maruz kalan gençler öncelikle kendileri daha sonra aileleri dijital oyun içerikleri ve tespit edilen görüntülerin analizi gönderilerek bilgilendirme yapılmaktadır. Son eşik değerine geldiği durumlarda kullanıcı bilgilendirilerek şiddet içerikli oyun kapatılmaktadır.

Çalışmamız ayrıca gerçek zamanlı ekran analizi yaparak nesne tespiti yapabildiğinden kullanıcının bilgisayar ortamında kayıtlı bir dijital oyun, görüntü, video vb. içeriklerden veya youtube, instagram, facebook vb. sosyal medya içeriklerini izlerken devreye girip benzer şekilde kullanıcı ve aileleri ile etkileşime geçebilmektedir.

Planlanan çalışmalar: Şiddet içerikleri görüntü içerisinde barınan kullanıcıları olumsuz yönde etkileyebildikleri gibi ses içerisinde de bulunarak benzer olumsuz etkilere neden olmaktadır. Ses içerisinde bulunan şiddet içeriklerinin tespiti de önemlidir. İlerleyen çalışmalarda ses içerisinde bulunan bu şiddet içerikleri tespit edilerek analiz edilmesi hedeflenmektedir.

KAYNAKLAR

1. Esposito, N. *A Short and Simple Definition of What a Videogame Is.* in *DiGRA Conference*. 2005.
2. Bergonse, R., *Fifty years on, what exactly is a videogame? An essentialistic definitional approach.* *The Computer Games Journal*, 2017. 6(4): p. 239-255.
3. Yengin.D, *Dijital oyunlarda şiddet kavramı: Yeni şiddet*, in *Marmara Üniversitesi, Sosyal Bilimler Enstitüsü*. 2010.
4. Clement, J., *Statista:"Number of digital video game users worldwide from 2017 to 2027 "*. 2023.
5. Bogost, I., *Persuasive games: The expressive power of videogames*. 2010: mit Press.
6. Prensky, M., *Digital game-based learning.* *Computers in entertainment (CIE)*, 2003. 1(1): p. 21-21.
7. Li, M.-C. and C.-C. Tsai, *Game-based learning in science education: A review of relevant research.* *Journal of Science Education and Technology*, 2013. 22: p. 877-898.
8. Liao, C.-W., C.-H. Chen, and S.-J. Shih, *The interactivity of video and collaboration for learning achievement, intrinsic motivation, cognitive load, and behavior patterns in a digital game-based learning environment.* *Computers & Education*, 2019. 133: p. 43-55.
9. Hussein, M.H., et al., *A digital game-based learning method to improve students' critical thinking skills in elementary science.* *Ieee Access*, 2019. 7: p. 96309-96318.
10. Gros, B., *Digital games in education: The design of games-based learning environments.* *Journal of research on technology in education*, 2007. 40(1): p. 23-38.
11. Griffiths, M.D., *The educational benefits of videogames.* *Education and health*, 2002. 20(3): p. 47-51.
12. Moisala, M., et al., *Media multitasking is associated with distractibility and increased prefrontal activity in adolescents and young adults.* *NeuroImage*, 2016. 134: p. 113-121.
13. Prot, S., et al., *The positive and negative effects of video game play.* *Media and the well-being of children and adolescents*, 2014. 109: p. 2010-2014.
14. Nakamura, L., *Don't hate the player, hate the game: The racialization of labor in World of Warcraft*, in *Digital Labor*. 2012, Routledge. p. 187-204.
15. Shaw, A., *Gaming at the edge: Sexuality and gender at the margins of gamer culture.* 2015: U of Minnesota Press.

16. Woltermann, L., *A Fatwa Against Gaming? How Indonesian Muslims deal with online games and Islamophobia*. Indonesia and the Malay World, 2023. 51(149): p. 27-49.
17. Mayar, M., *Feasts of Indifference: Racialization, Affect, and Necropolitics in 1X War Games*. European journal of American studies, 2021. 16(16-3).
18. Self-Reflexivity, A.-P., *The Biopolitics of Gaming*. The Play Versus Story Divide in Game Studies: Critical Essays, 2015: p. 110.
19. Elson, M. and C.J. Ferguson, *Twenty-five years of research on violence in digital games and aggression*. European Psychologist, 2014.
20. Chiu, S.-I., J.-Z. Lee, and D.-H. Huang, *Video game addiction in children and teenagers in Taiwan*. CyberPsychology & behavior, 2004. 7(5): p. 571-581.
21. Ferguson, C.J., et al., *Violent video games and aggression: Causal relationship or byproduct of family violence and intrinsic violence motivation?* Criminal Justice and Behavior, 2008. 35(3): p. 311-332.
22. Cunningham, S., B. Engelstätter, and M.R. Ward, *Violent video games and violent crime*. Southern Economic Journal, 2016. 82(4): p. 1247-1265.
23. Gonzalez, J.M. and T. Greitemeyer, *The relationship between everyday sadism, violent video game play, and fascination with weapons*. Personality and Individual Differences, 2018. 124: p. 51-53.
24. Peper, E. and R. Harvey, *Digital addiction: Increased loneliness, anxiety, and depression*. NeuroRegulation, 2018. 5(1): p. 3-3.
25. Rehbein, F., et al., *Prevalence and risk factors of video game dependency in adolescence: results of a German nationwide survey*. Cyberpsychology, behavior, and social networking, 2010. 13(3): p. 269-277.
26. Messias, E., et al., *Sadness, suicide, and their association with video game and internet overuse among teens: results from the youth risk behavior survey 2007 and 2009*. Suicide and Life-Threatening Behavior, 2011. 41(3): p. 307-315.
27. Mukhra, R., et al., *'Blue Whale Challenge': A game or crime?* Science and engineering ethics, 2019. 25: p. 285-291.
28. İnternet, *TBMM Araştırma Komisyonu Raporu 2020* , <https://www.tbmm.gov.tr/sirasayi/donem27/yil01/ss214.pdf> Son Erişim Tarihi : 03.03.2024.
29. Chang, J.H. and B.J. Bushman, *Effect of exposure to gun violence in video games on children's dangerous behavior with real guns: a randomized clinical trial*. JAMA network open, 2019. 2(5): p. e194319-e194319.

30. Alotaibi, N.B. and M. Mukred, *Factors affecting the cyber violence behavior among Saudi youth and its relation with the suiciding: A descriptive study on university students in Riyadh city of KSA*. Technology in Society, 2022. 68: p. 101863.
31. Rostad, W.L., K.C. Basile, and H.B. Clayton, *Association among television and computer/video game use, victimization, and suicide risk among US high school students*. Journal of interpersonal violence, 2021. 36(5-6): p. 2282-2305.
32. Mitchell, S.M., et al., *The relationship between video game play and the acquired capability for suicide: An examination of differences by category of video game and gender*. Cyberpsychology, Behavior, and Social Networking, 2015. 18(12): p. 757-762.
33. DeCamp, W. and C.J. Ferguson, *The impact of degree of exposure to violent video games, family background, and other factors on youth violence*. Journal of youth and adolescence, 2017. 46: p. 388-400.
34. Smith, L.J., M. Gradisar, and D.L. King, *Parental influences on adolescent video game play: A study of accessibility, rules, limit setting, monitoring, and cybersafety*. Cyberpsychology, Behavior, and Social Networking, 2015. 18(5): p. 273-279.
35. Kneer, J., M. Elson, and F. Knapp, *Fight fire with rainbows: The effects of displayed violence, difficulty, and performance in digital games on affect, aggression, and physiological arousal*. Computers in human behavior, 2016. 54: p. 142-148.
36. Min, D., H. Lee, and I.H. Song, *Effect of internet game overflow on suicidal ideation among adolescents: Analysis of the mediating effect of depressive mood*. Journal of Digital Convergence, 2014. 12(9): p. 445-454.
37. Makashvili, M., et al., *Violent video games cause aggression in young females*. Journal of Law and Criminal Justice, 2014. 2(2): p. 151-160.
38. Yee, H.K., et al., *Dying in cyberworld: Violent video games extinguished children's death concept and attitude*. Southeast Asia Psychology Journal, 2019. 7: p. 58-69.
39. Funk, J.B., et al., *Violence exposure in real-life, video games, television, movies, and the internet: is there desensitization?* Journal of adolescence, 2004. 27(1): p. 23-39.
40. Anderson, C.A. and B.J. Bushman, *Effects of violent video games on aggressive behavior, aggressive cognition, aggressive affect, physiological arousal, and prosocial behavior: A meta-analytic review of the scientific literature*. Psychological science, 2001. 12(5): p. 353-359.
41. Jeong, E.J., et al. *A study of digital game addiction from aggression, loneliness and depression perspectives*. in 2016 49Th Hawaii International Conference On System Sciences (HICSS). 2016. IEEE.
42. Kalthoff, L., *Player Agency in Game Depictions of Suicide*. 2022.

43. Rowlands, T., S. Ratnabalasuriar, and K. Noel, *Video gaming, crime, and popular culture*, in *Oxford Research Encyclopedia of Criminology and Criminal Justice*. 2016.
44. Kocurek, C.A., *Who hearkens to the monster's scream? Death, violence and the veil of the monstrous in video games*. Visual Studies, 2015. 30(1): p. 79-89.
45. PEGI, in *PEGI (Pan European Game Information)*, <https://pegi.info/> , Son Erişim Tarihi :03.04.2024.
46. İnternet: Türk Dil Kurumu, T.D.K, *Şiddetin tanımı* , <https://sozluk.gov.tr/>, Son Erişim Tarihi :03.03.2024.
47. Yengin, D., *Dijital oyunlarda şiddet kavramı: Yeni şiddet*. 2010, Marmara Üniversitesi (Turkey).
48. Elbert, T., et al., *The influence of organized violence and terror on brain and mind: A co-constructive perspective*. 2006.
49. Peterson, J. and J. Densley, *Cyber violence: What do we know and where do we go from here?* Aggression and violent behavior, 2017. 34: p. 193-200.
50. Wall, D., *Crime and the Internet*, ISBN 9780415244299. 2001.
51. Hürriyet, *Counter-Strike kötü bitti.*, <https://www.hurriyet.com.tr/teknoloji/counter-strike-kotu-bitti-14897841>, 2010, Son Erişim Tarihi :03.04.2024.
52. Denning, D.E., *Cyberterrorism: Testimony before the special oversight panel on terrorism committee on armed services US House of Representatives*. Focus on terrorism, 2000. 9(1): p. 71-76.
53. Luijff, E., *Definitions of cyber terrorism*, in *Cyber Crime and Cyber Terrorism Investigator's Handbook*. 2014, Elsevier. p. 11-17.
54. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning*. nature, 2015. 521(7553): p. 436-444.
55. Jiao, L. and J. Zhao, *A survey on the new generation of deep learning in image processing*. Ieee Access, 2019. 7: p. 172231-172263.
56. Xin, Y., et al., *Machine learning and deep learning methods for cybersecurity*. Ieee access, 2018. 6: p. 35365-35381.
57. Lauriola, I., A. Lavelli, and F. Aiolli, *An introduction to deep learning in natural language processing: Models, techniques, and tools*. Neurocomputing, 2022. 470: p. 443-456.
58. Huang, J., J. Chai, and S. Cho, *Deep learning in finance and banking: A literature review and classification*. Frontiers of Business Research in China, 2020. 14(1): p. 13.

59. Nogales, A., et al., *A survey of deep learning models in medical therapeutic areas*. Artificial Intelligence in Medicine, 2021. 112: p. 102020.
60. Yong, B., et al., *Online practical deep learning education*. Educational Technology & Society, 2022. 25(1): p. 193-204.
61. Gharibshah, Z., et al., *Deep learning for user interest and response prediction in online display advertising*. Data Science and Engineering, 2020. 5(1): p. 12-26.
62. Zhang, L. and B. Zhang, *A geometrical representation of McCulloch-Pitts neural model and its applications*. IEEE transactions on neural networks, 1999. 10(4): p. 925-929.
63. Abraham, T.H., *(Physio) logical circuits: The intellectual origins of the McCulloch–Pitts neural networks*. Journal of the History of the Behavioral Sciences, 2002. 38(1): p. 3-25.
64. McCarthy, J., et al., *A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955*. AI magazine, 2006. 27(4): p. 12-12.
65. Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological review, 1958. 65(6): p. 386.
66. Rosenblatt, F., *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Vol. 55. 1962: Spartan books Washington, DC.
67. Minsky, M. and S.A. Papert, *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry*. 2017: MIT press.
68. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*. nature, 1986. 323(6088): p. 533-536.
69. Cybenko, G., *Approximation by superpositions of a sigmoidal function*. Mathematics of control, signals and systems, 1989. 2(4): p. 303-314.
70. Campbell, M., A.J. Hoane Jr, and F.-h. Hsu, *Deep blue*. Artificial intelligence, 2002. 134(1-2): p. 57-83.
71. LeCun, Y., et al., *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998. 86(11): p. 2278-2324.
72. Deng, J., et al. *Imagenet: A large-scale hierarchical image database*. in *2009 IEEE conference on computer vision and pattern recognition*. 2009. Ieee.
73. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks*. Advances in neural information processing systems, 2012. 25.

74. Lin, M., Q. Chen, and S. Yan, *Network in network*. arXiv preprint arXiv:1312.4400, 2013.
75. Szegedy, C., et al. *Going deeper with convolutions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
76. Goodfellow, I., et al., *Generative adversarial nets*. Advances in neural information processing systems, 2014. 27.
77. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556, 2014.
78. Wang, F.-Y., et al., *Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond*. IEEE/CAA Journal of Automatica Sinica, 2016. 3(2): p. 113-120.
79. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
80. Sabour, S., N. Frosst, and G.E. Hinton, *Dynamic routing between capsules*. Advances in neural information processing systems, 2017. 30.
81. İnternet: ChatGPT, *Optimizing Language Models for Dialogue*. OpenAI. 2022 , Son Erişim Tarihi :03.03.2024.
82. Deng, L. and D. Yu, *Deep learning: methods and applications*. Foundations and trends® in signal processing, 2014. 7(3–4): p. 197-387.
83. Alom, M.Z., et al., *A state-of-the-art survey on deep learning theory and architectures*. electronics, 2019. 8(3): p. 292.
84. Goodfellow, I., Y. Bengio, and A. Courville, *Deep learning*. 2016: MIT press.
85. Montesinos López, O.A., A. Montesinos López, and J. Crossa, *Fundamentals of artificial neural networks and deep learning*, in *Multivariate statistical machine learning methods for genomic prediction*. 2022, Springer. p. 379-425.
86. Anderson, J.A., E. Rosenfeld, and A. Pellionisz, *Neurocomputing*. Vol. 2. 1988: MIT press.
87. Graupe, D., *Deep learning neural networks: design and case studies*. 2016: World Scientific Publishing Company.
88. Ciresan, D.C., et al. *Flexible, high performance convolutional neural networks for image classification*. in *Twenty-second international joint conference on artificial intelligence*. 2011. Citeseer.
89. Bishop, C.M., *Neural networks for pattern recognition*. 1995: Oxford university press.

90. Ding, B., H. Qian, and J. Zhou. *Activation functions and their characteristics in deep neural networks*. in *2018 Chinese control and decision conference (CCDC)*. 2018. IEEE.
91. Wang, Y., et al., *The influence of the activation function in a convolution neural network model of facial expression recognition*. *Applied Sciences*, 2020. 10(5): p. 1897.
92. Dubey, S.R., S.K. Singh, and B.B. Chaudhuri, *Activation functions in deep learning: A comprehensive survey and benchmark*. *Neurocomputing*, 2022.
93. David, K.C.L. and W. Yu, *Deep Learning*. World Scientific Book Chapters, 2022: p. 109-143.
94. Chieng, H.H., et al., *Flatten-T Swish: a thresholded ReLU-Swish-like activation function for deep learning*. arXiv preprint arXiv:1812.06247, 2018.
95. Kouretas, I. and V. Paliouras. *Simplified hardware implementation of the softmax activation function*. in *2019 8th international conference on modern circuits and systems technologies (MOCAS)*. 2019. IEEE.
96. Misra, D., *Mish: A self regularized non-monotonic activation function*. arXiv preprint arXiv:1908.08681, 2019.
97. Wu, H. and X. Gu. *Max-pooling dropout for regularization of convolutional neural networks*. in *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I* 22. 2015. Springer.
98. Shyam, R., *Convolutional neural network and its architectures*. *Journal of Computer Technology & Applications*, 2021. 12(2): p. 6-14p.
99. Kim, P. and P. Kim, *Convolutional neural network*. *MATLAB deep learning: with machine learning, neural networks and artificial intelligence*, 2017: p. 121-147.
100. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *ImageNet classification with deep convolutional neural networks*. *Communications of the ACM*, 2017. 60(6): p. 84-90.
101. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting*. *The journal of machine learning research*, 2014. 15(1): p. 1929-1958.
102. Ba, J. and B. Frey, *Adaptive dropout for training deep neural networks*. *Advances in neural information processing systems*, 2013. 26.
103. Wan, L., et al. *Regularization of neural networks using dropconnect*. in *International conference on machine learning*. 2013. PMLR.
104. Ho, Y. and S. Wookey, *The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling*. *IEEE access*, 2019. 8: p. 4806-4813.

105. Chai, L., et al., *Using generalized Gaussian distributions to improve regression error modeling for deep learning-based speech enhancement*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2019. 27(12): p. 1919-1931.
106. Zheng, Z., et al. *Distance-IoU loss: Faster and better learning for bounding box regression*. in *Proceedings of the AAAI conference on artificial intelligence*. 2020.
107. Li, X., et al., *Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection*. Advances in Neural Information Processing Systems, 2020. 33: p. 21002-21012.
108. Bosman, A.S., A. Engelbrecht, and M. Helbig, *Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions*. Neurocomputing, 2020. 400: p. 113-136.
109. Zhou, Y., et al., *MPCE: a maximum probability based cross entropy loss function for neural network classification*. IEEE Access, 2019. 7: p. 146331-146341.
110. Liu, G., et al., *Partial convolution based padding*. arXiv preprint arXiv:1811.11718, 2018.
111. Alrasheedi, F., X. Zhong, and P.-C. Huang, *Padding Module: Learning the Padding in Deep Neural Networks*. IEEE Access, 2023. 11: p. 7348-7357.
112. Zhao, Z.-Q., et al., *Object detection with deep learning: A review*. IEEE transactions on neural networks and learning systems, 2019. 30(11): p. 3212-3232.
113. Xie, S., et al. *Aggregated residual transformations for deep neural networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
114. Redmon, J. and A. Farhadi, *Yolov3: An incremental improvement*. arXiv preprint arXiv:1804.02767, 2018.
115. Lin, T.-Y., et al. *Feature pyramid networks for object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
116. Liu, S., et al. *Path aggregation network for instance segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
117. Tan, M., R. Pang, and Q.V. Le. *Efficientdet: Scalable and efficient object detection*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
118. Ren, S., et al., *Faster r-cnn: Towards real-time object detection with region proposal networks*. Advances in neural information processing systems, 2015. 28.
119. Redmon, J., et al. *You only look once: Unified, real-time object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

120. Redmon, J. and A. Farhadi. *YOLO9000: better, faster, stronger*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
121. Liu, W., et al. *Ssd: Single shot multibox detector*. in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14. 2016. Springer.
122. Lin, T.-Y., et al. *Focal loss for dense object detection*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
123. Tian, Z., et al., *Fully convolutional one-stage 3d object detection on lidar range images*. *Advances in Neural Information Processing Systems*, 2022. 35: p. 34899-34911.
124. Dai, J., et al., *R-fcn: Object detection via region-based fully convolutional networks*. *Advances in neural information processing systems*, 2016. 29.
125. Bochkovskiy, A., C.-Y. Wang, and H.-Y.M. Liao, *Yolov4: Optimal speed and accuracy of object detection*. *arXiv preprint arXiv:2004.10934*, 2020.
126. Wang, C.-Y., et al. *CSPNet: A new backbone that can enhance learning capability of CNN*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020.
127. He, K., et al., *Spatial pyramid pooling in deep convolutional networks for visual recognition*. *IEEE transactions on pattern analysis and machine intelligence*, 2015. 37(9): p. 1904-1916.
128. Jiang, Z., et al., *Real-time object detection method based on improved YOLOv4-tiny*. *arXiv preprint arXiv:2011.04244*, 2020.
129. Thuan, D., *Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detection algorithm*. 2021.
130. Jiang, P., et al., *A Review of Yolo algorithm developments*. *Procedia Computer Science*, 2022. 199: p. 1066-1073.
131. Wang, C.-Y., A. Bochkovskiy, and H.-Y.M. Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
132. Zhang, Z., et al., *Bag of freebies for training object detection neural networks*. *arXiv preprint arXiv:1902.04103*, 2019.
133. Hussain, M., *YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection*. *Machines*, 2023. 11(7): p. 677.

134. Reis, D., et al., *Real-Time Flying Object Detection with YOLOv8*. arXiv preprint arXiv:2305.09972, 2023.
135. Zhang, S., et al. *Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
136. Zhou, Y., *IYOLO-NL: An improved you only look once and none left object detector for real-time face mask detection*. Heliyon, 2023. 9(8).
137. İnternet: *Brief summary of YOLOv8 model structure*. 2023 , Son Erişim Tarihi :03.03.2024.
138. İnternet: Jocher, G., *YOLOv8 Overview*. 2023, Son Erişim Tarihi :03.03.2024.
139. Chao, Y.-W., et al. *Rethinking the faster r-cnn architecture for temporal action localization*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
140. Cheng, D., et al. *Person re-identification by multi-channel parts-based cnn with improved triplet loss function*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
141. Shorten, C. and T.M. Khoshgoftaar, *A survey on image data augmentation for deep learning*. Journal of big data, 2019. 6(1): p. 1-48.
142. Yun, S., et al. *Cutmix: Regularization strategy to train strong classifiers with localizable features*. in *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
143. Nwankpa, C., et al., *Activation functions: Comparison of trends in practice and research for deep learning*. arXiv preprint arXiv:1811.03378, 2018.
144. Bingham, G., W. Macke, and R. Miikkulainen. *Evolutionary optimization of deep learning activation functions*. in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 2020.
145. Sun, R.-Y., *Optimization for deep learning: An overview*. Journal of the Operations Research Society of China, 2020. 8(2): p. 249-294.
146. Ranzato, M.A., et al. *Unsupervised learning of invariant feature hierarchies with applications to object recognition*. in *2007 IEEE conference on computer vision and pattern recognition*. 2007. IEEE.
147. Camgözlü, Y. and Y. Kutlu, *Analysis of filter size effect in deep learning*. arXiv preprint arXiv:2101.01115, 2020.
148. Goutte, C. and E. Gaussier. *A probabilistic interpretation of precision, recall and F-score, with implication for evaluation*. in *European conference on information retrieval*. 2005. Springer.

149. Heydarian, M., T.E. Doyle, and R. Samavi, *MLCM: Multi-label confusion matrix*. IEEE Access, 2022. 10: p. 19083-19095.
150. Howard, A.G., et al., *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. arXiv preprint arXiv:1704.04861, 2017.
151. Smahel, D., M.F. Wright, and M. Cernikova, *The impact of digital media on health: children's perspectives*. International journal of public health, 2015. 60: p. 131-137.
152. Koga, Y., K. Kawano, and D. Kawashima, *Does Video Game Play Elevate Suicide Risk? A Cross-sectional Study of Japanese Young Adults 1, 2*. Japanese Psychological Research, 2022.
153. Medeiros, B.G.d., et al., *"Brutal Kill!" Violent video games as a predictor of aggression*. Psico-USF, 2020. 25: p. 261-271.
154. Zhang, Q., D.L. Espelage, and D.-J. Zhang, *RETRACTED: The priming effect of violent game play on aggression among adolescents*. Youth & Society, 2018: p. 0044118X18770309.
155. Coyne, S.M., et al., *Who is most at risk for developing physical aggression after playing violent video games? An individual differences perspective from early adolescence to emerging adulthood*. Journal of youth and adolescence, 2023. 52(4): p. 719-733.
156. Greitemeyer, T., *The spreading impact of playing violent video games on aggression*. Computers in human behavior, 2018. 80: p. 216-219.
157. Evcin, S. and S. Erzi, *The Effect of Violent Computer Games on Adolescent Aggression Tendency/Şiddet İçerikli Bilgisayar Oyunlarının Ergenlerin Saldırganlık Eğilimine Etkisi*. Humanitas, 2019. 7(14): p. 181.
158. Kavlak, M., A.İ. Sarilir, And B. Tönbol, *Dijital Oyun Bağımlılığı Yalnızlığı Tetikler Mi? Necmettin Erbakan Üniversitesi Ereğli Eğitim Fakültesi Dergisi*, 2022. 4(1): p. 1-13.
159. D Griffiths, M., D. J Kuss, and D. L King, *Video game addiction: Past, present and future*. Current Psychiatry Reviews, 2012. 8(4): p. 308-318.
160. Mustafaoğlu, R. and Z. Yasacı, *The negative effects of digital game playing on children's mental and physical health*. Journal of Dependence, 2018. 19(3): p. 51-58.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÇAPANOĞLU, Ahmet Edip
Uyruğu : T.C.
Doğum tarihi ve yeri : -
Medeni hali : -
E-posta : -

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Adli Bilişim	2024
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	2016
Lisans	Uluslararası Kıbrıs Üniversitesi / Bilgisayar Mühendisliği	2009
Lise	Yozgat Anadolu Lisesi	2002

İş Deneyimi

Yıl	Yer	Görev
2015-Halen	Gazi Üniversitesi Mühendislik Fakültesi	Öğretim Görevlisi
2011-2015	Gazi Üniversitesi Bilgi İşlem Dairesi Başkanlığı	Uzman

Yabancı Dil

İngilizce

Yayınlar

Yalçın, N., & Çapanoğlu, A. E. (2023). Detection of Violent Elements in Digital Games Using Deep Learning. *SN Computer Science*, 4(5), 639.

Kitap Bölümü

Yalçın, N., Çapanoğlu A. E., (2023)"Dijital Oyunlarda Güvenlik", Bilişim Teknolojileri Güvenliği, Nobel Yayın Dağıtım, Ankara, ss.401-431.



GAZİLİ OLMAK AYRICALIKTIR.