

X-RAY GÖRÜNTÜLERİNDE DERİN ÖĞRENME YÖNTEMLERİ İLE TEHDİT TESPİTİ

THREAT DETECTION IN X-RAY IMAGES WITH DEEP LEARNING TECHNIQUES

ELİF ALTINDAĞ

DOÇ. DR. SENİHA ESEN YÜKSEL ERDEM

Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliği'nin
Elektrik ve Elektronik Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2022

ÖZET

X-RAY GÖRÜNTÜLERİNDE DERİN ÖĞRENME YÖNTEMLERİ İLE TEHDİT TESPİTİ

ELİF ALTINDAĞ

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı: Doç. Dr. Seniha Esen YÜKSEL ERDEM

Ocak 2022, 119 sayfa

Son yıllarda toplum güvenliğini sağlamak için havaalanları, okullar, alışveriş merkezleri gibi kalabalık ve tehlike arz edebilecek ortamlarda X-ray görüntüleme cihazları sıklıkla kullanılmaktadır. Ancak X-ray görüntüleme tekniklerinin kullanıldığı yerlerde insan gücüne hala gerek duyulmaktadır. Bunun nedeni, X-ray görüntülerinde tehdit unsuru olabilecek nesnelerin otomatik olarak tespit edilmesi için gereken ilerlemenin yeterli olmamasıdır. Bu amaçla, son yıllarda derin öğrenme yöntemlerini kullanarak X-ray bagaj görüntülerinde bulunan ve toplu alanlara sokulması yasaklı nesnelerin tespitine yönelik çalışmalar ivme kazanmıştır. Bu tez çalışmasında da, derin öğrenme tabanlı nesne tespiti yöntemleri kullanılarak X-ray görüntülerinde tehdit tespiti yapılmış ve bu alana katkı sunmak hedeflenmiştir.

Bu çalışmada biri açık kaynaklı ve biri ise çalışma kapsamında toplanmış iki farklı veri kümesi kullanılmıştır. Toplanan X-ray görüntüleri çift kanallı X-ray tarayıcı tarafından elde edilmiştir. Çift kanallı X-ray tarayıcılar hem düşük hem de yüksek enerji seviyelerini kullanmaktadır. İki enerji seviyesi kullanarak taranan maddenin atom numarası, yoğunluğu gibi ek bilgiler elde edilir ve bu yöntemle nesnelerin karakteristik özellikleri hakkında daha detaylı bilgiye sahip olunur. Böylece bir tarama sonucu elimizde düşük enerjili, yüksek enerjili ve bu görüntüler kullanılarak renklendirmiş X-ray görüntüsü olmak üzere toplamda üç adet görüntü bulunur. Açık kaynaklı veri kümesi, toplanılan veri kümesine göre daha kapsamlı olduğu için kullanılan nesne tespiti algoritmaları önce-

likle bu küme üzerinde eğitilmiştir. Bu kapsamda popüler nesne tespiti algoritmalarından YOLOv3, SSD ve Faster RCNN kullanılmış olup açık kaynaklı ve kapsamlı veri kümesi üzerinde elde edilen sonuçlar karşılaştırılmıştır. Karşılaştırma sonucunda performansı en yüksek olan yöntem ile elde edilen sonuçlar kullanılarak toplanılan veri kümesinin eğitimi tamamlanmaya çalışılmıştır. Bunun nedeni, toplanılan veri kümesinin az sayıda görüntü içermesidir. Toplanılan veri kümesine aktarım yapılırken ilk aşamada az vuruşlu öğrenme yöntemi kullanılmış ve mümkün olduğunca az görüntü ile toplanılan veri kümesindeki tehditlerin ne kadar başarılı tespit edileceği araştırılmıştır. Az vuruşlu öğrenmeden sonra yapılan diğer çalışma ise açık kaynaklı veri kümesinin eğitimleri sonucu elde edilen en iyi sonucun toplanılan veri kümesine ince ayar yöntemi ile aktarılmasıdır. Çalışma kapsamında yapılan son denemede ise toplanılan veri kümesine ait renklendirilmiş görüntüler ile düşük ve yüksek enerjili görüntüler farklı kombinasyonlar halinde uçtan uca eğitilmiştir. Buradaki amaç ise düşük ve yüksek enerjili görüntülerden elde edilen bilgi ile az sayıda görüntü içeren veri kümesindeki tehditlerin tespitinde ne kadar başarı sağlanacağını gözlemlemektir. Elde edilen sonuçlar incelendiğinde açık kaynaklı veri kümesi üzerinde Faster RCNN ile yapılan eğitimlerin en yüksek başarıya ulaştığı görülmüştür. Bu sonuçların toplanılan veri kümesine aktarıldığı durumda ise en iyi başarı ince ayar yöntemi ile alınmıştır. Toplanılan veri kümesine ait düşük ve yüksek enerjili görüntülerin farklı kombinasyonlar ile denendiği durumda ise en yüksek başarı düşük enerjili, yüksek enerjili ve renklendirilmiş X-ray verilerinin eğitime katıldığı durumda elde edilmiştir.

Anahtar Kelimeler: X-RAY Görüntüleme, Derin Öğrenme, Evrimsel Sinir Ağı, Nesne Tespiti

ABSTRACT

THREAT DETECTION IN X-RAY IMAGES WITH DEEP LEARNING TECHNIQUES

ELİF ALTINDAĞ

**Master of Science, Department of
Electrical and Electronics Engineering**

Supervisor: Doç. Dr. Seniha Esen YÜKSEL ERDEM

January 2022, 119 pages

In recent years, X-ray imaging equipment are frequently used in crowded and risky environments such as airports, schools and shopping malls to ensure public safety. However, human labor is still needed in places where X-ray imaging techniques are used. This is because the accuracy required to automatically detect potentially threatening objects in X-ray images is not sufficient. For this purpose, studies on the detection of prohibited items that are not allowed to be brought into public area using deep learning techniques have gained acceleration in recent years. In this thesis, threat detection from X-ray images was accomplished using deep learning-based object detection methods and it was aimed to contribute to this field.

In this study, two different datasets, one of which is open source and the other collected within the scope of the study, were used. The X-ray images collected were acquired by a dual-channel X-ray scanner. Dual-channel X-ray scanners use both low and high energy levels. By using two energy levels, additional information such as the atomic number and density of the scanned material is obtained, and with this method, more detailed information about the characteristics of the objects is obtained. Thus, as a result of a scan, we have three images in total which are low-energy, high-energy, and colored X-ray image. It is important to note that a colored X-ray image is obtained by using high-energy and low-energy X-ray images. Since the open-source dataset is more

comprehensive than the collected dataset, the object detection algorithms used are primarily trained on this set. In this context, YOLOv3, SSD, and Faster RCNN, which are popular object detection algorithms, were used and the results obtained on the open-source and comprehensive dataset were compared. As a result of the comparison, the training of the collected dataset was done by using the results obtained from the method with the highest performance. This is because the collected dataset contains a small number of images. While transferring to the collected dataset, the few-shot learning method was used and it was investigated how successfully the threats in the collected dataset would be detected with as few images as possible. Another study after the few-shot learning is transferring the best result obtained as a result of training the open-source dataset to the collected dataset with fine tuning. In the last experiment, within the scope of the study, colored, low and high energy images of the collected data set were trained end-to-end in different combinations. The aim here is to observe how successful the information obtained from low and high-energy images will be in detecting threats in a dataset containing a small number of images. When the results were examined, it was seen that the training made with Faster RCNN achieved the highest success on the open source dataset. In the case of transferring these results to the collected dataset, the best success was obtained with the fine-tuning method. In the case where the low and high-energy images of the collected dataset are applied with different combinations, the highest success is achieved when low-energy, high-energy and colored X-ray images are included in the training.

Keywords: X-RAY Imagery, Deep Learning, Convolutional Neural Network, Object Detection

TEŐEKKÜR

Bu alıőmanın gerekleőtirilmesinde yardım ve rehberliđini esirgemeyen, deđerli fikirleriyle katkıda bulunan, sabır ve anlayıőından dolayı tez danıőmanım Do. Dr. Seniha Esen Yüksel Erdem'e en iten teőekkürlerimi ve saygılarımı sunarım.

Hayatım boyunca hep yanımda olan, maddi ve manevi destekleriyle beni her zaman kucaklayan aileme ve sevgili eőime ok teőekkür ederim.

Öđretim hayatım boyunca desteklerini esirgemeyen tüm öđretmenlerime teőekkürlerimi ve saygılarımı sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ÇİZELGELER	ix
ŞEKİLLER	x
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ	1
1.1. X-ray Görüntülerinde Tehdit Tespiti İçin Mevcut Çalışmalar	1
1.2. Tezin Amacı	3
1.3. Tezin Kapsamı ve Akışı	6
2. NESNE TESPİTİ	8
2.1. Geleneksel Nesne Tespiti Yöntemleri	8
2.1.1. Viola Jones Yöntemi	9
2.1.1.1. Haar Öznitelikleri	9
2.1.1.2. İmge İntegrali	9
2.1.1.3. AdaBoost Yöntemi	10
2.1.1.4. Basamaklandırma Yöntemi	10
2.1.2. HOG Yöntemi	11
2.1.3. DPM Yöntemi	11
2.2. Derin Öğrenme Tabanlı Nesne Tespiti Yöntemleri	11
2.2.1. İki Aşamalı Nesne Tespiti Yöntemleri	12
2.2.1.1. RCNN	13
2.2.1.2. Fast RCNN	15
2.2.1.3. Faster RCNN	18

2.2.2. Tek Aşamalı Nesne Tespiti Yöntemleri	22
2.2.2.1. YOLO	22
2.2.2.2. SSD	28
2.2.3. Derin Öğrenmede Farklı Öğrenme Çeşitleri	33
2.2.3.1. Uçtan Uca Öğrenme	33
2.2.3.2. Aktarım Öğrenmesi	34
2.2.3.3. Az Vuruşlu Öğrenme	36
2.2.4. Öznitelik Piramit Ağı	39
3. LİTERATÜRDE SUNULAN FARKLI X-RAY VERİ KÜMELERİ VE DENENEN YÖNTEMLER	42
3.1. GDXray Veri Kümesi	42
3.2. Multi-view (Çoklu Görünüm) X-Ray Veri Kümesi	43
3.3. NS-100-L X-Ray Veri Kümesi	44
3.4. Rapiscan X-Ray Veri Kümesi	45
3.5. DBF3 ve DBF6 X-ray Veri Kümeleri	46
3.6. SIXray Veri Kümesi	48
3.7. Tez Kapsamında Toplanan HUMS X-ray Veri Kümesi	51
3.8. Literatürde X-ray Veri Kümeleri Üzerinde Uygulanan Yöntemler ve Alınan Sonuçlar	54
4. GERÇEK VERİLER ÜZERİNDE DENEYSEL SONUÇLAR	59
4.1. Deneyleerde Kullanılan Ölçüm Metriği	59
4.2. SIXray Veri Kümesi Üzerinde Gerçekleştirilen Deneyleer	61
4.2.1. YOLOv3 Yöntemi ile Elde Edilen Tehdit Tespiti Sonuçları	61
4.2.2. Faster RCNN Yöntemi ile Elde Edilen Tehdit Tespiti Sonuçları	64
4.2.2.1. RESNET-101 Omurga Ağı ile Eğitim	64
4.2.2.2. RESNET-101 + FPN Omurga Ağı ile Eğitim	65
4.2.3. SSD Yöntemi ile Elde Edilen Tehdit Tespiti Sonuçları	67
4.2.4. SIXray Veri Kümesi Üzerinde Alınan Sonuçların Karşılaştırılması	68

4.3. HUMS Veri Kümesi Üzerinde Yapılan Deneyler	79
4.3.1. Az Vuruşlu Öğrenmeye Ait Tehdit Tespiti Sonuçları	80
4.3.1.1. Modelin Temel Eğitimi	81
4.3.1.2. İnce Ayar ile Az Vuruşlu Öğrenme Aşaması	81
4.3.2. HUMS Veri Kümesindeki Yapay RGB Görüntülerin İnce Ayar ile Eğitimi	84
4.3.3. HUMS Veri Kümesine Ait Yüksek ve Düşük Enerjili Görüntüler ile Yapay RGB Görüntüler Kullanılarak Yapılan Eğitimin Sonuçları	86
4.3.3.1. HUMS Veri Kümesi Üzerinde Alınan Sonuçların Karşılaştırılması	91
5. SONUÇLAR	95
KAYNAKLAR	98
EKLER	104
ÖZGEÇMİŞ	106

ÇİZELGELER

	<u>Sayfa</u>
Çizelge 2.1. Fast RCNN ile RCNN yöntemlerinin karşılaştırılması [40].....	18
Çizelge 2.2. Faster RCNN ile RCNN yöntemlerinin karşılaştırılması [44].....	22
Çizelge 2.3. YOLO ile diğer modellerin karşılaştırılması [47]	27
Çizelge 2.4. YOLO omurga ağları ile diğer ağların karşılaştırılması [49].....	28
Çizelge 2.5. SSD ile diğer modellerin karşılaştırılması [50].....	33
Çizelge 3.1. GDXray veri kümesinde kategorilere göre görüntü sayısı dağılımı [22]	42
Çizelge 3.2. MV-XRAY veri kümesinde kategorilere göre görüntü sayısı dağılımı [8]	43
Çizelge 3.3. Rapiscan veri kümesinde kategorilere göre görüntü sayısı dağılımı [62]	46
Çizelge 3.4. Durham bagaj veri kümesinde kategorilere göre görüntü sayısı dağılımı [14].....	47
Çizelge 3.5. SIXray veri kümesinin farklı sınıflara göre dağılımı [67].....	49
Çizelge 3.6. Etiketleme sonucunda tehditlere ait elde edilen kesin referans sayıları	50
Çizelge 3.7. HUMS veri kümesine ait görüntülerin dağılımı ve kesin referans sayıları	52
Çizelge 3.8. Literatürdeki X-ray veri kümeleri ile alınan sonuçların karşılaştırması - 1.....	54
Çizelge 3.9. Literatürdeki X-ray veri kümeleri ile alınan sonuçların karşılaştırması - 2.....	55
Çizelge 4.1. SIXray veri kümesinde denenen modellere ait sınıf bazlı ve genel sonuçların karşılaştırılması (**Literatürde sunulan en iyi sonuç)	68
Çizelge 4.2. HUMS veri kümesi üzerinde yapılan test sonuçları	77
Çizelge 4.3. Az vuruşlu öğrenmede kullanılan temel ve yeni kategoriler	80
Çizelge 4.4. HUMS veri kümesindeki bıçakların tespitinde uygulanan yöntemlerin karşılaştırılması (Y. RGB, Yapay RGB için kısaltma olarak sunulmuştur.).....	93

ŞEKİLLER

	<u>Sayfa</u>
Şekil 2.1. Nesne tespiti alanında yayınlanan araştırmaların yıllara göre değişimi [23]	8
Şekil 2.2. Haar tipi öznelikler (üst) ve hesaplama yöntemi (alt) [27]	9
Şekil 2.3. İmge integrali hesaplama yöntemi [26]	10
Şekil 2.4. İki aşamalı nesne tespiti [32]	13
Şekil 2.5. Seçici arama yöntemi: (a) çok sayıda alt bölütün oluşturulması, (b) bölütlerin birleştirilmeye başlanması, (c) nesnelere ait son önerilerin çıkarılması [34]	14
Şekil 2.6. AlexNet mimarisi [38]	14
Şekil 2.7. RCNN mimarisi [39]	15
Şekil 2.8. Fast RCNN mimarisi [41]	17
Şekil 2.9. Faster RCNN mimarisi [43]	19
Şekil 2.10. Bölge önerisi ağı (RPN) [43]	20
Şekil 2.11. YOLO mimarisine ait genel özet [47]	22
Şekil 2.12. YOLO ile nesne tespiti süreci [47]	24
Şekil 2.13. YOLO CNN ağ yapısı [47]	25
Şekil 2.14. SSD varsayılan kutu yapısı	30
Şekil 2.15. SSD modeline ait genel bir özet	30
Şekil 2.16. Aktarım öğrenmesine ait farklı konfigürasyonlar [54]	36
Şekil 2.17. Az vuruşlu öğrenmeye ait ağ yapısı [56]	38
Şekil 2.18. Görüntü piramidi yapısı [59]	39
Şekil 2.19. Aşağıdan yukarı ve yukarıdan aşağı yollarda elde edilen öznelik haritalarının yanıl bağlantılar kullanılarak eklenmesine ait bir blok [59] .	41
Şekil 3.1. GDXray bagaj görüntülerine ait örnekler [22]	43
Şekil 3.2. MV-Xray veri kümesine ait bir örnek [8]	44
Şekil 3.3. NS-100-L veri kümesine ait bir örnek [61]	45

Şekil 3.4. Rapiscan veri kümesine ait örnek görüntüler [62].....	46
Şekil 3.5. DBF3 veri kümesine ait örnek bir görüntü [63]	48
Şekil 3.6. DBF6 veri kümesine ait örnek bir görüntü [63]	48
Şekil 3.7. SIXray veri kümesinden örnek görüntüler: sade görüntü (sol), karmaşık görüntü (sağ) [67]	50
Şekil 3.8. Çift kanallı X-ray görüntüleme adımları	51
Şekil 3.9. HUMS veri kümesinden örnek görüntüler: Tekli bıçak (sol üst), bıçak ve bavul (sağ üst), düzgün yerleştirilmiş bıçaklar (sol alt), karmaşık yerleştirilmiş bıçaklar (sağ alt)	52
Şekil 3.10. SIXray ve HUMS veri kümelerindeki bıçakların farklı kanallar için histogram grafikleri	53
Şekil 4.1. YOLOv3 eğitimi sonucu elde edilen mAP grafiği	63
Şekil 4.2. Faster RCNN (ResNet-101) eğitimi sonucu elde edilen mAP grafiği	65
Şekil 4.3. Faster RCNN (ResNet-101+FPN) eğitimi sonucu elde edilen mAP grafiği	66
Şekil 4.4. SSD eğitimi sonucu elde edilen mAP grafiği	67
Şekil 4.5. Sütunlarda sırayla YOLOv3, Faster RCNN (ResNet-101, ResNet-101+FPN) ve SSD modellerinden alınan farklı kategorilere ait tespitler verilmiştir. Satırlar sırayla silah, bıçak, makas, İngiliz anahtarı ve penseye ait görüntülerdir.	71
Şekil 4.6. Farklı sınıflara ait kesin referans içeren örnek görüntüler	74
Şekil 4.7. Dört satırda sırası ile YOLOv3, Faster RCNN (ResNet-101), Faster RCNN (ResNet-101+FPN) ve SSD modellerinden alınan ve farklı kategorilere ait tehdit tespitleri içeren görüntüler	75
Şekil 4.8. HUMS veri kümesine ait kesin referans içeren örnek görüntüler	78
Şekil 4.9. HUMS veri kümesine ait örnek tespitler dört satırda sırası ile YOLOv3, Faster RCNN (ResNet-101), Faster RCNN (ResNet-101+FPN) ve SDD modelleri için sunulmuştur	79
Şekil 4.10. HUMS veri kümesindeki bıçaklara ait $K = 1, 3, 5, 10$ ve 30 vuruş için elde edilen AP grafiği	83

Şekil 4.11. Az vuruşlu öğrenme sonucu en iyi sonuçlar kullanılarak (30 vuruş) alınan tespit örnekleri	84
Şekil 4.12. HUMS veri kümesindeki yapay RGB görüntülerin SIXray ile eğitimler sonucu elde edilen en iyi ağırlıklarla ince ayarına ait AP grafiği.....	85
Şekil 4.13. SIXray veri kümesi ile elde edilen en iyi ağırlıklar kullanılarak yapılan ince ayar sonucu HUMS veri kümesine ait örnek tespitler	86
Şekil 4.14. HUMS veri kümesindeki görüntülerin farklı kombinasyonlarla eğitilmesi sonucu alınan AP değerleri (Y. RGB, Yapay RGB kısaltması için sunulmuştur)	88
Şekil 4.15. Yapay RGB görüntüler ile birlikte düşük ve yüksek enerjili görüntülerin farklı kombinasyonlar ile eğitilmesi sonucu alınan örnek tespitler	90
Şekil 4.16. Sadece yapay RGB görüntüler ile sadece düşük ve yüksek enerjili görüntülerin eğitimi sonucu alınan tespitler	91

SİMGELER VE KISALTMALAR

Simgeler

L_{cls}	Sınıflandırma kaybı
L_{loc}	Lokalizasyon kaybı
L_{reg}	Regresyon kaybı
$\lambda_{koordinat}$	Sınırlandırıcı kutucuk tahminlerine ait kayıplarının kat sayısı
$\lambda_{objeyok}$	Nesne içermeyen durumlara ait güven skoru kayıplarının kat sayısı

Kısaltmalar

ANN	Artificial Neural Network	Yapay Sinir Ağı
CNN	Convolutional Neural Network	Evrişimsel Sinir Ağı
RCNN	Region Based CNN	Bölgeye Dayalı Evrişimli Sinir Ağı
mAP	Mean Average Precision	Ortalama Kesinlik Değeri
IoU	Intersection Over Union	Birleşim Üzerinden Kesişim
SGD	Stochastic Gradient Descent	Stokastik Gradyan İnişi
SVM	Support Vector Machine	Destek Vektör Makinesi
RoI	Region of Interest	İlgi Bölgesi
FPN	Feature Pyramid Network	Öznitelik Piramidi Ağı
YOLO	You Only Look Once	
SSD	Single Shot Multibox Detector	

1. GİRİŞ

Derin öğrenme ve evrişimsel sinir ağları son yıllarda gittikçe yaygınlaşan ve hızla gelişen alanlardır. Bu alandaki gelişmeler birçok yerde olduğu gibi bilgisayarla görünün farklı çalışma disiplinlerinde de kullanılmaya başlanmış ve başarılı sonuçlar alınmıştır. Bu disiplinlerden birisi de nesne tespiti olmuştur. Nesne tespiti, sınıflandırma görevinden farklı olarak görüntü içerisinde bulunan bir nesnenin hem kategorisini hem de konumlandırılmasını amaçlamaktadır.

Nesne tespiti, otonom sürüş, tarım, spor, endüstri ve sağlık gibi birçok farklı uygulama alanına sahiptir. Bunlardan biri de güvenlik için X-ray cihazlarından taranan bavul ve çanta gibi eşyaların içerisinde bulunan tehdit unsurlarını tespit etmektir. Bu amaçla birçok veri kümesi oluşturulmuş ve bunlardan bazıları başka araştırmacıların çalışması için açık kaynaklı olarak sunulmuştur. Benzer şekilde bu alanda hem geleneksel hem de derin öğrenme tabanlı nesne tespiti algoritmaları birçok araştırmacı tarafından farklı veri kümeleri üzerinde denenmiş ve sonuçları paylaşılmıştır.

1.1 X-ray Görüntülerinde Tehdit Tespiti İçin Mevcut Çalışmalar

Derin öğrenme ve evrişimsel sinir ağları yöntemlerinin birçok alanda başarılı sonuçlar vermesiyle birlikte bu konular X-ray görüntülerinde nesne tespitine de yansımıştır. Bu kapsamda yapılan ilk çalışmalardan biri olan ve Akçay ve Breckon tarafından toplanılan fakat paylaşımına açılmayan *DBF2/6* veri kümeleri üzerinde kayar pencere tabanlı evrişimsel sinir ağı (CNN), Faster RCNN [43] ve R-FCN [2] yöntemleri denenmiştir [1]. Çalışmada kullanılan *DBF2* veri kümesi iki sınıflı (silah vs. silah bulunmayan görüntüler) bir veri kümesi iken *DBF6* ise altı farklı sınıf içeren daha kapsamlı bir veri kümesidir. Çalışma sonucunda elde edilen veriler ışığında R-FCN modelinin ResNet-101 [3] omurga ağı ile kullanıldığı durumda *DBF2* veri kümesinde %96.3 ortalama kesinlik değeri (mAP) ile en iyi sonuca ulaştığı görülmüştür. *DBF6* veri kümesi için yapılan deneylerde ise VGG-16 [4] omurga ağı tabanlı Faster RCNN modelinin %88.3 ortalama kesinlik değeri ile en iyi sonucu verdiği görülmüştür.

Bir başka çalışmada ise Liang ve diğerleri X-ray görüntülerinde tekli ve çoklu bakış açılarından alınan tehdit tespitleri için Faster RCNN, R-FCN ve SSD [50] yöntemle-

rini denemiştir [5]. Farklı bakış açılarından elde edilen tespitlerin OR kapısı mantığı ile kaynaştırılması sonucu elde edilen performansın tekli bakış açılarından elde edilen sonuçlardan daha yüksek olduğu görülmüştür. Sırası ile farklı bakış açılarının ortalama kesinlik değeri %0.938 iken tekli bakış açılarından elde edilen sonuç ise %0.798 olmuştur. Benzer bir çalışmada ise dört farklı tehdit sınıfı içeren ve her sınıf için yaklaşık 3,400 adet görüntüye sahip bir veri kümesi üzerinde [62] Inception ResNet v2 [7] omurga ağını kullanan Faster RCNN modelinin tekli bakış açısına sahip görüntülerde %92.2 ve çoklu bakış açısına sahip görüntülerde ise %97.7 başarıya ulaştığı görülmüştür.

Çoklu ve tekli bakış açısına sahip X-ray görüntüleri için yapılan başka bir çalışmada ise Faster RCNN modeli modifiye edilmiştir. Bu kapsamda evrimsel katmanlardan elde edilen iki boyutlu öznitelikler havuzlama katmanında üç boyutlu hale getirilecek şekilde düzenlenmiştir. Bu aşamadan sonra sınıflandırma ve sınırlandırıcı kutucuk regresyon işlemleri yapılmıştır. Elde edilen sonuçlara göre çoklu bakış açısına sahip görüntülerde tekli bakış açısına sahip görüntülere göre performansta bir ilerleme kaydedilmiştir. Çoklu bakış açısına sahip görüntülerde %95.96 ortalama kesinlik değerine ulaşılırken tekli bakış açısı ile alınan görüntüler %91.23 ortalama kesinlik değerine ulaşmıştır [8]. İki aşamalı yaklaşım ile yapılan bir çalışmada ise 32,253 adet X-ray görüntüsü içeren bir veri kümesinde öncelikle ön plan görüntüleri çıkarılmış ve ardından Faster RCNN kullanılarak nesne tespitinde %77 ortalama kesinlik değerine ulaşılmıştır [9]. Makas ve aerosol görüntüleri içeren bir X-ray veri kümesinde ise nesne tespiti için YOLOv2 kullanılmıştır [48]. 6000 iterasyonluk eğitim sonucunda %94.5 ortalama kesinlik değerine ulaşılmıştır [10].

Başka bir çalışmada ise büyük bir yük konteynerinden taranan 5000 adet X-ray görüntüsü ve 544 adet tabanca görüntüsünün sentetik yollarla üretilmesi sonucu elde edilen 30,000 görüntünün RetinaNet [11] ve geleneksel kayar pencere yöntemleri ile eğitilmesinin sonuçları karşılaştırılmıştır. Sonuçlara göre RetinaNet eğitimi ile elde edilen performans ile kayar pencere eğitimi sonucu elde edilen performansların birbirine yakın çıktığı gözlemlenirken RetinaNet'in geleneksel kayar pencere yöntemine göre çok daha hızlı olduğu görülmüştür [12]. Şişe, saç kurutma makinesi, ütü, ekmek kızartıcı, cep telefonu ve laptop görüntüleri içeren altı sınıflı bir veri kümesi ise bu nesnelerin tespiti için

Faster RCNN, Mask RCNN [13] ve RetinaNet yöntemleri kullanılarak eğitilmiştir. Alınan sonuçlar karşılaştırıldığında ResNet-101 omurga ağı kullanıldığı durumda Mask RCNN ile %97.9 ile en yüksek ortalama kesinlik değerine ulaşıldığı gözlemlenmiştir. Bununla birlikte Faster RCNN %97.4 ve RetinaNet ise %95.9 ortalama kesinlik değerine ulaşmıştır [14]. Yapılan başka bir çalışmada ise hem *GDXray* veri kümesi hem de sentetik üretilmiş X-ray görüntüleri kullanılarak tehdit tespiti hedeflenmiştir. Çalışmada *GDXray* ve sentetik veriler birleştirilmiştir. Verilerde silah, bıçak, jilet ve ninja yıldızı olmak üzere toplamda dört farklı tehdit sınıfı bulunmaktadır. Tehdit tespiti için RetinaNet, YOLOv3 [49] ve SSD yöntemleri kullanılmıştır. YOLOv3 ortalamada %80.0 kesinlik değeri ile en başarılı sonucu vermiştir [15].

Farklı türde kesici aletlerin kullanıldığı veri kümesi ile yapılan bir çalışmada ise görüntülerdeki oklüzyonu gidermek için *DOAM* (De-occlusion Attention Module) adı verilen bir yöntem geliştirilmiştir. Bu yöntem çalışma kapsamında SSD, FCOS [16] ve YOLOv3 modellerine uygulanmış ve sonuçları sunulmuştur. Elde edilen sonuçlara göre SSD modelinin *DOAM* uygulanmamış hali ortalamada %70.89 kesinlik değerine ulaşırken *DOAM* ile birlikte bu sonuç %74.01'e yükselmiştir. FCOS modeli için ise sonuç %82.02'den %82.41'e yükselmiştir. Son olarak YOLOv3 ile yapılan denemelerde sonuç %78.21'den %79.25'e çıkmıştır [65]. Başka bir çalışma kapsamında ise [14]'de belirtilen çift kanallı X-ray veri kümesinde nesne tespiti ve bölütleme işlemleri çift kanaldan alınan bilgiler kullanılarak yapılmıştır. Çalışma kapsamında sırası ile sadece RGB görüntüler, yüksek enerjili X-ray görüntüleri, düşük enerjili X-ray görüntüleri, materyelin atom numarasını veren Z görüntüleri ve tüm görüntülerinin birleşimi Mask RCNN, YOLACT [18], CARAFE [19] ve Cascade Mask RCNN [20] modelleri ile eğitilmiştir. Elde edilen sonuçlar karşılaştırıldığında tüm görüntülerin CARAFE modeli ile eğitildiği durumda %70.0 ile en yüksek tespit ve bölütleme performansı gözlemlenmiştir [21].

1.2 Tezin Amacı

Bu tez çalışmasında X-ray verilerinde insan gücünden bağımsız bir şekilde tehdit tespiti yapılması için çok sınıflı ve tek sınıflı X-ray veri kümelerinde derin öğrenme tabanlı nesne tespiti algoritmaları kullanılmıştır. Bu kapsamda öncelikle çalışmaların başlangıç aşamasında Hacettepe Üniversitesi Teknokent yerleşkesinde bulunan MS Spektral Sa-

vunma Sanayi A.Ş. isimli firma aracılığı ile çift kanallı X-ray tarayıcı kullanılarak X-ray görüntüleri toplanmış ve tek sınıflı küçük bir veri kümesi oluşturulmuştur. Oluşturulan veri kümesine Hacettepe Üniversitesi ve MS Spektral Savunma Sanayi A.Ş. ortaklığını belirtmesi amacı ile HUMS ismi verilmiştir. Görüntüler toplanırken farklı tipte bıçaklar kullanılmıştır. Kullanılan bıçaklar veri kümesinde çeşitliliği sağlamak ve daha güvenilir sonuçlar alabilmek amacıyla farklı kombinasyonlar halinde taranmıştır. Bu kapsamda bıçaklar X-ray tarayıcıya hem tekli hem de farklı bıçaklar bir arada olacak şekilde yerleştirilmiştir. Bıçakların çoklu tarandığı durumda benzer şekilde çeşitliliği sağlamak için bıçakların yerleşimine önem verilmiştir. Bıçaklar hem nizami ve birbirlerini engellemecek şekilde hem de karmaşık ve birbirini engelleyecek şekilde farklı kombinasyonlarda taranmıştır. Bıçakların yanı sıra tehdit içermeyen görüntüler de toplanmış ve olası bir sınıflandırma işlemi için kaydedilmiştir. Veriler toplanırken kullanılan tarayıcı çift kanallı olduğundan bir tarama için hem düşük enerjili, hem yüksek enerjili hem de bu görüntülerin renklendirilmesi ile elde edilen yapay RGB görüntü olmak üzere toplamda 3 farklı görüntü elde edilmiştir. Ancak veri toplama işlemleri çeşitli kısıtlardan dolayı bir noktada kesilmiş olduğundan toplamda düşük enerjili, yüksek enerjili ve yapay RGB görüntüler dahil olmak üzere yaklaşık 300 adet görüntü alınmıştır.

HUMS veri kümesindeki görüntü sayısı az olduğu için kaynaklarda sunulan farklı X-ray veri kümeleri araştırılmıştır. Araştırmalar sonucunda bazı veri kümelerinin akademik çalışmalar için açık kaynaklı sunulduğu görülürken bazılarının ise gizli projelerde kullanıldığı görülmüştür. Bu nedenle açık kaynaklı veri kümeleri incelenmiş ve kapsamlı bir veri kümesi olan SIXray'in tez çalışmasında kullanılmasına karar verilmiştir [67]. SIXray veri kümesi hem sınıflandırma hem de nesne tespiti çalışmalarına uygun olacak şekilde toplanmıştır. Veri kümesinde hem tehdit içeren hem de tehdit içermeyen görüntüler mevcuttur. SIXray veri kümesi toplamda silah, bıçak, makas, İngiliz anahtarı, pense ve çekiç olmak üzere altı farklı tehdit sınıfına ait veri içermektedir. Ancak veri yetersizliğinden dolayı çekiç sınıfı kullanılmamıştır.

SIXray veri kümesi sırası ile YOLOv3, Faster RCNN ve SSD modelleri ile eğitilerek bu veri kümesinde bulunan beş farklı kategoriye ait tehditlerin tespiti amaçlanmıştır. Eğitim sonucunda elde edilen en iyi ağırlıklar kullanılarak örnek tespitler alınmış ve

tezde sunulmuştur. Daha sonra üç modele ait sonuçlar bir çizelgede karşılaştırılmıştır. Tez kapsamındaki bir amaç da SIXray veri kümesinden elde edilen eğitim sonuçlarının HUMS veri kümesinde de kullanılması olduğu için eğitimlerin sonucu ayrıca HUMS veri kümesi üzerinde de test edilmiş ve alınan sonuçlar karşılaştırılmıştır.

Sonraki adımda SIXray veri kümesinde en iyi başarıya sahip modelden elde edilen sonuçlar HUMS veri kümesine aktarılmaya çalışılmıştır. Sonuçlar aktarılırken ilk uygulamada az vuruşlu öğrenme tekniği kullanılmıştır. Bu tekniğin uygulanmasındaki amaç mümkün olduğunca az veri ile bıçakları doğru bir şekilde tespit edebilmek olmuştur. Az vuruşlu öğrenme tekniği kullanırken ince ayar (fine tune) aşamasında kullanılan görüntü sayısı gittikçe arttırılmış ve bıçak tespiti performansındaki değişim gözlemlenmiştir. Az vuruşlu öğrenmeden sonra ise HUMS veri kümesinde bulunan renklendirilmiş görüntülerin tamamı SIXray veri kümesi ile elde edilen en iyi sonuçlar kullanılarak ince ayar ile eğitilmiştir. Bu yöntemler ile elde edilen sonuçlar da tez çalışmasında paylaşılmıştır.

SIXray ile elde edilen sonuçların aktarım denemelerinin ardından HUMS veri kümesi üzerinde direkt eğitimler uygulanmaya başlanmıştır. Bu kapsamda SIXray veri kümesinin eğitimleri sonucunda, HUMS veri kümesinde en iyi test başarısını gösteren model kullanılmıştır. HUMS veri kümesi çift kanala sahip olduğu için bu bilgiler de farklı şekilde kullanılmıştır. Eğitimler yapılırken ilk denemede yapay RGB görüntüler ile yüksek enerjili görüntüler algoritmaya beslenmiştir. Sonraki adımda ise eğitimde yapay RGB görüntüler ve düşük enerjili görüntüler kullanılmıştır. Sonuçlar elde edildikten sonra başka bir kombinasyon olarak yapay RGB görüntüler ile yüksek enerjili ve düşük enerjili görüntülerin tümü kullanılarak eğitim yapılmıştır. Daha sonra sadece yapay RGB görüntüler eğitimde kullanılmıştır. Denenen başka bir kombinasyon ise sadece düşük ve yüksek enerjili görüntülerin eğitimde kullanıldığı durum olmuştur. Farklı kombinasyonların denemesindeki amaç ise veri kümesindeki farklılıkların eğitim sonucunu nasıl etkilediğini gözlemlemek ve bu denemelerde en az görüntü ile en yüksek başarıyı sağlayabilecek kombinasyonun belirlenmesini sağlamaktır.

1.3 Tezin Kapsamı ve Akışı

Tezin geri kalanı şu şekilde organize edilmiştir:

İkinci bölümde nesne tespiti yöntemlerine değinilmiştir. Yıllara göre nesne tespiti alanında yapılan çalışmalar sunulmuş ve nesne tespiti algortimalarının birçok alanda yaygın kullanıldığından bahsedilmiştir. Bu bölümün devamında ilk önce geleneksel nesne tespiti yöntemleri sunulmuştur. Daha sonra ise derin öğrenme yöntemlerinin son yıllarda yaygınlaşmasından ve bu alanda yapılan gelişmelerden bahsedilmiştir. Derin öğrenme yöntemlerinin gelişmesi ile birlikte nesne tespiti algoritmalarına yansması ve derin öğrenme tabanlı nesne tespiti algoritmalarının gelişimine değinilmiştir. Bu kapsamda çift ve tek aşamalı nesne tespiti yöntemleri anlatılmış ve tez süresince kullanılan algoritmalar ilgili başlıklar altında detaylı bir şekilde açıklanmıştır. İkinci bölümün son alt başlığında ise derin öğrenme problemlerinde kullanılan farklı öğrenme teknikleri sunulmuştur. Bu teknikler arasından uçtan uca eğitim, aktarım öğrenmesi ve az vuruşlu öğrenme yöntemlerine değinilmiştir.

Üçüncü bölümde ise literatürde kullanılan farklı X-ray veri kümeleri sunulmuştur. Sunulan veri kümelerinde bulunan farklı kategorideki tehditler veya nesnelere tanıtılmıştır. İlgili veri kümelerinden açık kaynaklı olarak paylaşılanlara ait örnek görüntüler verilmiştir. Üçüncü bölümün ilerleyen kısmında ise literatürdeki X-ray veri kümeleri kullanılarak yapılan nesne tespiti çalışmalarına ait sonuçlar ve kullanılan yöntemler çizelge şeklinde paylaşılmıştır. Son olarak tez çalışmasında kullanılan SIXray veri kümesi ve toplanan çift kanallı veri kümesi detaylı bir şekilde açıklanmış ve örnek görüntüler sunulmuştur.

Dördüncü bölümde seçilen nesne tespiti algortimalarının gerçek veriler üzerinde uygulanması ile elde edilen sonuçlar paylaşılmıştır. Bu kapsamda denenen her nesne tespiti yöntemi için eğitim süresince elde edilen başarı değerinin değişimi grafik halinde sunulmuştur. Eğitim tamamlandığında elde edilen en iyi sonuçlara ait farklı tehdit tespiti örnekleri hem SIXray hem de toplanan veri kümesi için sunulmuştur. Bu bölümde ayrıca iki veri kümesi için de denenen yöntemler sonucunda elde edilen başarılarla ait karşılaştırma çizelgeleri sunulmuştur.

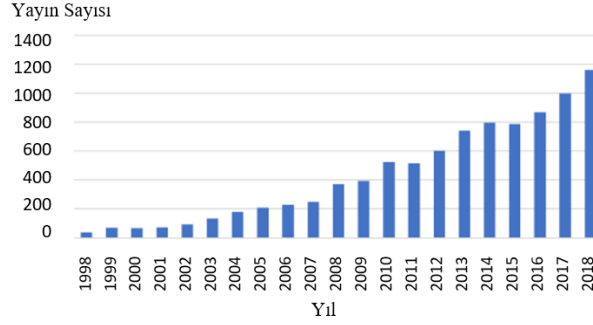
Son bölümde ise tez çalışması süreci genel kapsamı ile özetlenmiş olup tez çalışması

boyunca uygulanan nesne tespiti yöntemlerinin X-ray görüntülerinde tehdit tespitindeki etkilerine ait çıkarımlara yer verilmiştir.

2. NESNE TESPİTİ

Nesne tespiti, dijital bir görüntü veya video içerisinde bulunan belirli bir sınıfa ait nesnelerin tespit edilmesi ile ilgilenen bir bilgisayarlı görü tekniğidir. Nesne tespiti yöntemi ile dijital görüntü ve videolar için anlamlı semantik verilere ulaşılır. Bu yöntem, görüntü sınıflandırma, yüz tanıma, insan davranışı analizi, otonom sürüş gibi birçok alanda kullanılmaktadır.

Derin öğrenme yöntemlerinin son yıllarda hızla gelişmesi ile birlikte nesne tespiti yöntemleri de gittikçe gelişmiş ve yaygın bir araştırma konusu haline gelmiştir. Bu konuda yazılan makalelerin sayısının yıllara göre artışını gösteren grafik Şekil 2.1'de sunulmuştur.



Şekil 2.1. Nesne tespiti alanında yayınlanan araştırmaların yıllara göre değişimi [23]

Genel anlamda nesne tespiti uygulamalarının gelişimini *geleneksel nesne tespiti dönemi* ve *derin öğrenme tabanlı nesne tespiti dönemi* olarak iki ayrı ana başlıkta inceleyebiliriz [23].

2.1 Geleneksel Nesne Tespiti Yöntemleri

Geleneksel yöntemlerde algoritma düzeni ön işleme, görüntü pekiştirme, segmentasyon, öznitelik çıkarımı ve sınıflandırma aşamalarından oluşur. Ön işleme ve pekiştirme adımlarında girdi verisinde gürültü seviyesini azaltmak ve veriye ait genel kaliteyi arttırmak amaçlanır. Segmentasyon safhasında ise karmaşık görüntünün tümünden ilgili bölgenin kırılması hedeflenmektedir. Öznitelik çıkarımı aşamasında nesneye ait kenar ve şekiller elde edilir. Son olarak görüntüdeki nesneler çıkarılan öznitelikler kullanılarak sınıflandırılır. Geleneksel yöntemlerin en büyük dezavantajı el ile öznitelik çıkarımına

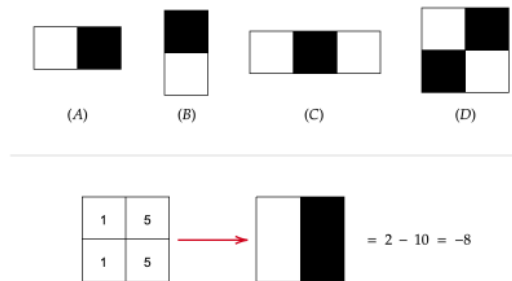
bağımlı olmasıdır [24]. Geleneksel nesne tespiti algoritmalarına *Viola Jones*, *HOG (Histogram of Gradients)* ve *DPM (Deformable Part-based Model)* yöntemleri örnek verilebilir [25]. Bu algoritmalara ait detaylı bilgiler alt başlıklarda sunulmuştur.

2.1.1 Viola Jones Yöntemi

Viola Jones [26], insan yüzü tanımda kullanılan hızlı ve doğruluğu yüksek bir nesne tespiti yöntemidir. Paul Viola ve Michael Jones tarafından 2001 yılında geliştirilmesine rağmen günümüzde popülerliğini koruyan bir algoritmadır. Viola Jones nesne tespiti yöntemi dört ana bileşenden oluşmaktadır. Bu bileşenler Haar öznitelikleri, imge integrali, AdaBoost ve basamaklandırma yöntemleridir.

2.1.1.1 Haar Öznitelikleri

Bir Haar tipi öznitelik [26], bir görüntüde belirlenen bitişik siyah ve beyaz dikdörtgensel bölgelerden beyaz bölgenin piksel değerlerinin toplamından siyah bölgenin piksel değerlerinin toplamının çıkarılması ile hesaplanır. Farklı şekildeki Haar tipi öznitelikler kullanılarak görüntüdeki kenar, düz ve yanal çizgiler hakkında bilgi elde edilir. Viola Jones algoritmasında Şekil 2.2'deki Haar tipi öznitelik kullanılmaktadır. Şeklin üst kısmında farklı Haar tipi öznitelikler sunulmuştur. Şeklin alt kısmında ise 2x2 boyutlu görüntünün üzerine bir Haar özniteliği evrişim uygulamasının sonucu sunulmuştur.

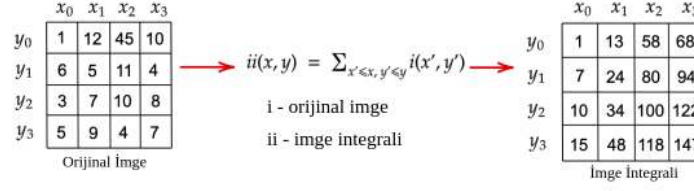


Şekil 2.2. Haar tipi öznitelikler (üst) ve hesaplama yöntemi (alt) [27]

2.1.1.2 İmge İntegrali

Bir imge integrali [26], görüntünün (x,y) koordinatları için üst-sol alanında kalan tüm piksel yoğunluklarının toplamı ile elde edilen ara bir gösterimdir. Bu gösterim, Haar

öznitelikleri çıkarılırken dikdörtgenel bölgelikler üzerinde hızlı işlem yapabilmek için gereklidir. Şekil 2.3'de imge integralinin hesaplaması sunulmuştur.



Şekil 2.3. İmge integrali hesaplama yöntemi [26]

2.1.1.3 AdaBoost Yöntemi

AdaBoost (Adaptive Boosting) yöntemi [26], tüm öznitelikler arasından en iyi özniteliklere sahip olan alt kümeleri seçmeyi amaçlar. Bu yöntemin çıktısı *Strong Classifier* (Güçlü Sınıflandırıcı) olarak adlandırılır. Bir güçlü sınıflandırıcı oluşturulurken ilk adımda özniteliklerin farklı alt kümelerinden biri seçilerek bir sınıflandırıcı elde edilir. Daha sonra, elde edilen sınıflandırıcının bir veriyi yeterince ayırtıramadığı durumda farklı bir öznitelik alt kümesi seçilir. Bu işlem, tüm öznitelikler içinden en iyi özniteliklerin (zayıf sınıflandırıcılar) lineer kombinasyonu seçilene kadar aşamalı bir şekilde devam etmektedir.

2.1.1.4 Basamaklandırma Yöntemi

Basamaklı sınıflandırıcı [26], çok aşamalı bir sınıflandırıcı türü olup hızlı ve doğru sonuç üretmeyi amaçlar. Her bir aşama AdaBoost yönteminin çıktısı olan güçlü sınıflandırıcıdan oluşur. Bir basamaktan diğerine doğru gittikçe güçlü sınıflandırıcı içerisinde bulunan zayıf sınıflandırıcıların (en iyi öznitelikler) sayısı artar. Her girdi aşama aşama işlem görmektedir. Herhangi bir basamakta sınıflandırıcı negatif bir çıktı üretirse bu veri hemen göz ardı edilir. Tam tersi şekilde bir girdi sınıflandırıcı sonucunda pozitif olarak algılanırsa bu girdi sonraki basamağa geçer. Yazarlara göre bu yaklaşım yüz tanıma algoritmasında yüz olmayan bir girdinin çabuk elenmesi ile gerçekten yüz olan veriye daha çok zaman harcanmasını sağlamaktadır.

2.1.2 HOG Yöntemi

HOG (Histogram of Oriented Gradients) yöntemi [29], bir görüntüde bulunan nesnenin lokal görüntü ve şekillerinin gradyan yönelim yoğunlukları ile tanımlanmasını baz almaktadır. Bu tanımlar görüntünün küçük bölgelere bölünüp her bir bölge için gradyan yönelmelerinin veya kenar yönelmelerinin bölge içerisindeki her piksel için çıkarılan histogramlarının kombinasyonları ile bulunur. HOG tanımlayıcılar lokal bölgeler içerisinde işlem yaptığından bu yöntemin temel avantajı geometrik ve fotometrik değişimlerden bağımsız olmasıdır. HOG yöntemi sonucunda elde edilen tanımlayıcılar daha sonra bir sınıflandırıcı kullanılarak elde edilen sınıf bilgisi ile birleştirilir ve nesne tespiti gerçekleştirilir.

2.1.3 DPM Yöntemi

DPM (deformable part-based model) [30], HOG yönteminden türemiş olup bunun üzerine R. Girshick tarafından birtakım geliştirmeler yapılarak VOC-07, VOC-08 ve VOC-09 yarışmalarında birinci gelen bir yöntemdir [23].

Tipik bir DPM algoritması, kök filtre ve bozunabilir filtrelerden oluşur. Görüntü üzerinde otomatik tespit gerçekleştirebilmek için zayıf bir öğretici ile öğrenme yöntemi geliştirilmiştir. Geliştirilen bu yöntem, R. Girshick tarafından özel bir çoklu obje öğrenme aşaması olarak tanımlanmış olup tespit doğruluğunu geliştirmek için zor olumsuz madencilik ve sınırlayıcı kutucuk regresyonu gibi yöntemlerden yararlanılmıştır. R. Girshick tespit hızını artırmak amacıyla çok daha hızlı tespit derleme yapabilen ardışık bir mimari [23] geliştirmiştir. Geliştirilen bu mimarinin modelin doğruluğu üzerinde olumsuz bir etkisi görülmemiştir.

2.2 Derin Öğrenme Tabanlı Nesne Tespiti Yöntemleri

Derin öğrenme, verilen bir veri kümesi ile sonuçları tahmin eden birden fazla katmandan oluşan bir makine öğrenme yöntemidir. Derin öğrenme, sinir ağları, hiyerarşik modeller, çeşitli gözetimli ve gözetimsiz algoritmaları kapsayan birçok yöntemi içeren çalışma alanıdır [31]. Sinir ağları ile ilgili ilk çalışmalar 1940 yıllarına kadar dayanmaktadır. Bu çalışmaların temel çıkış amacı öğrenme ile ilgili genel problemleri insan beyninin işleyiş

sistemini taklit ederek çözmektir. Ancak eğitim verilerindeki yetersizlik, eğitimde aşırı uyumluluk görülmesi, makine öğrenmesi yöntemlerine göre düşük başarımlar elde edilmesi ve sınırlı hesaplama gücü gibi nedenlerden dolayı sinir ağları yöntemleri uzun bir süre tercih edilmemiştir [23].

LeCun [28], 1990 yılında evrimsel sinir ağları (CNN) yöntemini kullanan ilk görüntü sınıflandırma modeli olan LeNet-5 ağını oluşturmuştur. Bu model, geri yayılma yolu ile stokastik gradyan inişi (SGD) kullanarak gözetimli bir şekilde eğitilmiştir [32]. 2012 yılında ise AlexNet [35] mimarisinin ortaya atılması ile birlikte birçok araştırmacı bilgisayarlı görü alanında derin öğrenmenin bir yöntemi olan evrimsel sinir ağlarını kullanarak görüntü sınıflandırma, nesne tespiti, segmentasyon, yer belirleme gibi farklı dallarda birçok yeni ve başarılı modeller üzerinde çalışmaya başlamıştır [25]. Nesne tespiti alanında CNN yöntemleri ile yapılan ilk çalışmalardan biri R. Girshick'in önermiş olduğu RCNN yöntemidir. Bu yöntem ile birlikte nesne tespiti alanında yapılan çalışmalar çok hızlı bir şekilde artmıştır [23].

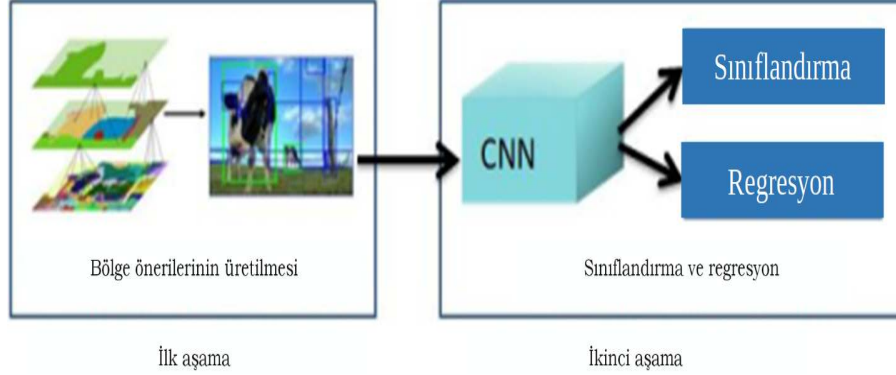
Derin öğrenme tabanlı nesne tespiti yöntemleri tek aşamalı dedektörler ve iki aşamalı dedektör olarak iki ana grupta incelenebilir. İki aşamalı nesne tespiti algoritmalarında ilk aşama görüntüde bulunan olası objelere ait bölge önerileri üretirken ikinci aşama ise önceki aşamaya ait önerilerin sınıflandırılmasını ve sınırlandırıcı kutucuklar kullanılarak tespit edilmesini sağlar. İki aşamalı nesne tespiti algoritmalarına RCNN [37], SPP-net [36], Fast RCNN [40], Faster RCNN [43], RFCN [2] ve Mask RCNN [13] örnek verilebilir.

Tek aşamalı nesne tespiti yöntemlerinde ise sınıflandırma ve regresyon işlemlerinin görüntüye ait konum, ölçek ve boy-en oranı gibi bilgilerin düzenli ve yoğun örneklendirme kullanılarak bir seferde tamamlanması amaçlanır. Tek aşamalı nesne tespiti algoritmalarına YOLO [47], SSD [50], YOLOv3 [49] ve RetinaNet [11] örnek verilebilir [32].

2.2.1 İki Aşamalı Nesne Tespiti Yöntemleri

İki aşamalı nesne tespiti algoritmaları, nesne konumlandırılmadan ve sınıflandırılmadan önce bölge önerilerinin üretilmesine ihtiyaç duymaktadır. Bu aşamaların özeti Şekil 2.4'te verilmiştir. İki aşamalı nesne tespiti algoritmalarının temel avantajı doğruluk paylarının yüksek olması iken bu tip algoritmaların dezavantajı ise karmaşık ve yavaş

olmalarıdır [33].



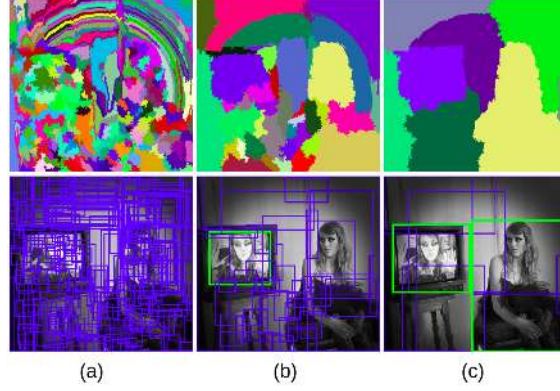
Şekil 2.4. İki aşamalı nesne tespiti [32]

2.2.1.1 RCNN

Bölgeye dayalı evrimsel sinir ağı (Region Based CNN - RCNN) yöntemi [37], iki aşamalı nesne tespiti kapsamında geliştirilen ilk CNN tabanlı modeldir. RCNN mimarisi temel olarak üç aşamadan oluşmaktadır:

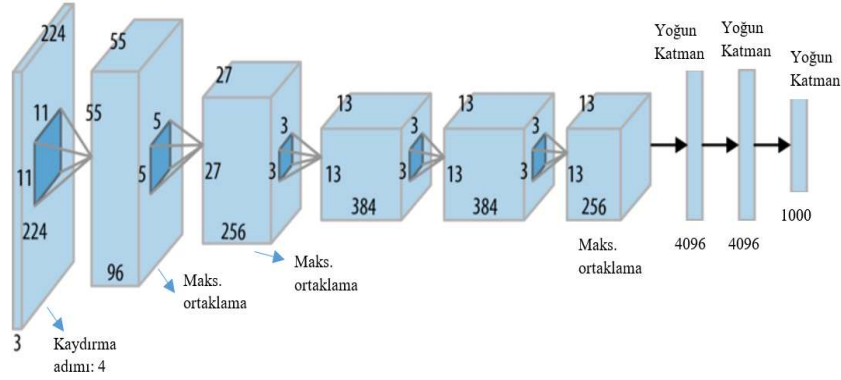
- Seçici arama yöntemi ile bölge önerilerinin oluşturulması,
- CNN yöntemi ile bölge önerilerine ait özniteliklerin çıkarılması,
- Bölge önerilerinin SVM yöntemi ile sınıflandırılması.

RCNN ile nesne tespitinde, bir görüntü üzerinde birçok kutucuk oluşturulur ve bu kutucukların bir obje içerip içermediği kontrol edilir. Bir görüntüye ait nesne önerileri oluşturulurken seçici arama yöntemi kullanılır. Tipik bir görüntüde bir objeyi oluşturan dört temel unsur bulunur. Bu unsurlar boyut, renk, desen ve şekil olarak sınıflandırılabilir. Seçici arama yöntemi ile bu unsurlar tespit edilerek farklı bölge önerileri sunulur. Seçici arama yönteminde ilk adımda algoritma girdisi olarak bir görüntü kabul edilir. Daha sonra, başlangıç için birçok alt bölütleme oluşturulur. İlk adımın sonunda çok sayıda bölge elde edilir. Bir sonraki aşamada ise oluşturulan alt bölütler renk, desen, boyut ve şekil gibi bir nesneye ait olabilecek benzer unsurlar göz önünde bulundurularak birleştirilir. Böylece son bölge önerileri oluşturulur. Şekil 2.5'te RCNN yönteminde kullanılan seçici arama yönteminin aşamaları verilmiştir.



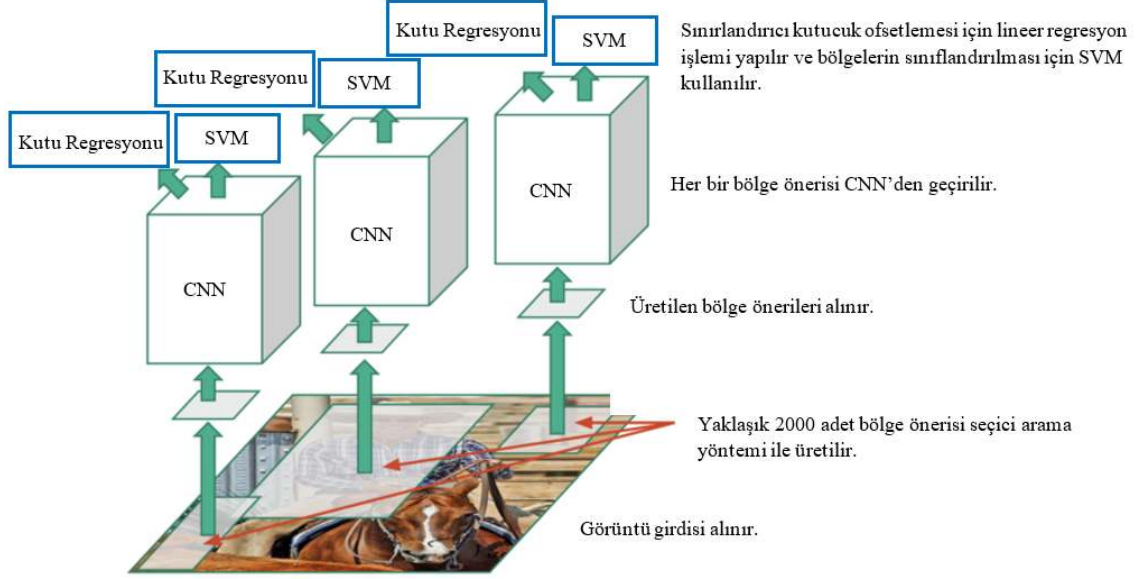
Şekil 2.5. Seçici arama yöntemi: (a) çok sayıda alt bölütün oluşturulması, (b) bölütlerin birleştirilmeye başlanması, (c) nesnelere ait son önerilerin çıkarılması [34]

Seçici arama yöntemi aşaması ile bir görüntü için yaklaşık iki bin adet farklı bölge önerisi üretilir. Daha sonra, üretilen her bölge önerisi için AlexNet CNN mimarisi kullanılarak öznetelikler üretilir. AlexNet mimarisi, beş adet evrimsel katman ve üç adet tam bağlantılı katman olmak üzere toplam sekiz adet katmandan oluşur. Aşağıdaki şekilde AlexNet mimarisi verilmiştir.



Şekil 2.6. AlexNet mimarisi [38]

Her bir bölge önerisinin öznetelikleri AlexNet ile elde edildikten sonra bu bölgeler için SVM kullanılarak sınıflandırma yapılır. Son adım olarak ise bölge önerileri için lineer regresyon ile sınırlandırıcı kutucuklar elde edilir. Böylece görüntüde bulunan nesnelere tespit edilmiş olur. Şekil 2.7'de RCNN mimarisine ait özet bir görsel sunulmaktadır.



Şekil 2.7. RCNN mimarisi [39]

RCNN yöntemi, derin öğrenme ile nesne tespiti algoritmalarının gelişmesinde önemli bir adım olmuştur. Ancak bu yöntem, her bir görüntü için iki bin bölge oluşturup bu bölgeleri yukarıda anlatıldığı gibi tek tek değerlendirip sınıflandırdığı için çok yavaş bir yöntemdir. Bir test görüntüsünün RCNN ile tespit edilme süresi 40 ile 50 saniye arasında değişmektedir. Bu nedenle RCNN kullanılarak büyük veri kümelerinde ve gerçek zamanlı uygulamalarda nesne tespiti işlemini gerçekleştirmek pratik olarak mümkün değildir.

2.2.1.2 Fast RCNN

Fast RCNN [40], RCNN algoritmasına ait hız sorununu gidermeyi amaçlarken aynı zamanda RCNN'e göre doğruluğu daha yüksek olan bir algoritmadır. Fast RCNN ağı, ilk adımda girdi görüntüsünün tümünü evrişimli ve maksimum ortaklama katmanlarından geçirerek öznitelik haritasını oluşturur. Daha sonra oluşturulan öznitelikler üzerinde seçici arama algoritması ile bölge önerileri üretilir. Her bir öneri, ilgi bölgesi (RoI) ortaklama katmanından geçirilir ve sabit boyutlu bir öznitelik vektörü oluşturulur. Öznitelik vektörleri son adımda iki dala ayrılan tam bağlantılı katmandan geçer. Bu aşamada K adet sınıf ve arka plana ait olasılıklar ile her K sınıfı için sınırlandırıcı kutucukların pozisyon bilgisini içeren dört adet koordinat değeri üretilir.

Rol ortaklama katmanı, maksimum ortaklama yöntemini kullanarak herhangi bir bölge önerisine ait öznitelikleri $H \times W$ boyutunda daha küçük bir öznitelik haritasına çevirir. H ve W değerleri bu katmana ait hiperparametreler olmakla beraber herhangi bir Rol'den bağımsızdırlar. Fast RCNN algoritmasında bir Rol, evrişimsel öznitelik haritalarında bulunan ve sol üst köşe nokta ile boyut bilgisini içeren pencerelerdir. Rol maksimum ortaklama yöntemi, bir $h \times w$ boyutlu Rol penceresini $H \times W$ boyutunda alt pencerelere bölerek elde edilen pencerelere maksimum ortaklama işlemini uygular.

Eğitim aşamasında kullanılan her ilgi bölgesi bir kesin referans barındırır. Bu referansta hem sınıf (u) hem de sınırlandırıcı kutucuk (v) bilgisi bulunur. Sınırlandırıcı kutucuk bilgisi sol üst köşe koordinatları ile kutucuk boyutu bilgisini içerir. Fast RCNN yöntemi iki ayrı tam bağlantılı katmanda çoklu görev kaybı (multi-task loss L) kullanarak sınıflandırma ve sınırlandırıcı kutucuk regresyon işlemini gerçekleştirir:

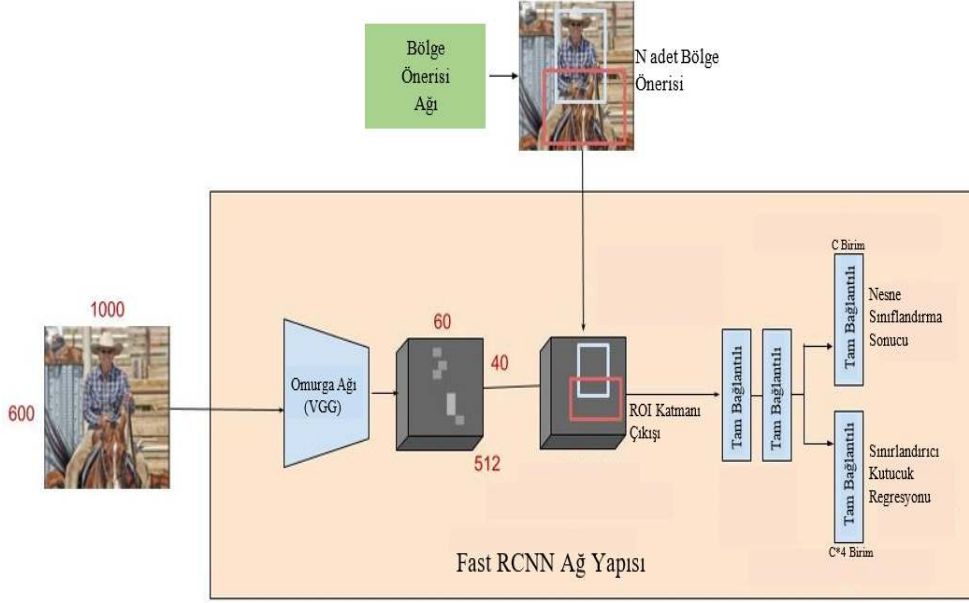
$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v). \quad (2.1)$$

Denklem 2.1'de $L_{cls}(p, u) = -\log p_u$, kesin referansa ait sınıf (u) için elde edilen logaritmik kayıp değeridir. İkinci göreve ait kayıp L_{loc} ise kesin referans değerinde sunulan doğru sınıfa (u) ait doğru sınırlandırıcı kutucuk değerleri ($v = v_x, v_y, v_w, v_h$) ile doğru sınıf için regresyon sonucu elde edilen sınırlandırıcı kutucuk değerleri ($t^u = t_x^u, t_y^u, t_w^u, t_h^u$) arasındaki farka ait kayıp değeridir. Denklemde $[u \geq 1]$ ifadesi $u \geq 1$ ise 1, değil ise 0 olarak hesaplanır. Yöntemde, arka plan sınıflar için u etiket değeri 0 olarak verilir. Arka plan ilgi bölgeleri için kesin referans sınırlandırıcı kutucuk bilgisi bulunmadığı için bu bölgelere ait L_{loc} değeri ihmal edilir. Sınırlandırıcı kutucuk regresyonu için elde edilen kayıp formülü 2.2'de verilmiştir:

$$L_{loc}(t^u, v) = \sum_{i \in x, y, w, h} \text{smooth}_{L_1}(t_i^u - v_i) \quad (2.2)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & x \leq 0 \\ |x| - 0.5 & \text{aksi takdirde} \end{cases} \quad (2.3)$$

Denklem 2.3 ile L_1 kayıp değeri uç değerlere karşı daha az hassas hale gelmiştir. Denklem 2.1’de kullanılan λ değeri iki kayıp değeri arasındaki dengeyi kontrol etmek için kullanılmış olup deney aşamasında 1 olarak belirlenmiştir. Ayrıca, kesin referansa ait regresyon hedef değerleri sıfır ortalama ve birim sapma değerine göre normalize edilmiştir. Aşağıdaki şekilde Fast RCNN mimarisi sunulmuştur.



Şekil 2.8. Fast RCNN mimarisi [41]

Girshick’in sunduğu deneysel çalışmalara göre Fast RCNN, RCNN yöntemine göre daha hızlı ve daha yüksek doğruluğa sahip sonuçlar vermiştir [40]. Çizelge 2.1’de Fast RCNN ile RCNN’e ait bir karşılaştırma sunulmuştur. Bu karşılaştırma küçük, orta ve büyük boyutlu omurga ağ yapıları için ayrı ayrı ele alınmıştır. Burada küçük boyutlu ağ olarak belirtilen yapı RCNN’de kullanılan AlexNet’tir [35]. Orta boyutlu olarak bildirilen ağ ise AlexNet’e kıyasla daha büyük bir boyuta sahip olan VGG-CNN-M-1024 [42] ağıdır. Son olarak büyük boyutlu omurga ağ ise AlexNet ve VGG-CNN-M-1024 ağlarına kıyasla daha derin bir ağ olan VGG-16 [4] ağıdır. Karşılaştırma yapılırken VOC07 [68] yarışmasında kullanılan veri kümesi kullanılmıştır. Karşılaştırma çizelgesine bakıldığında Fast RCNN yöntemi ile eğitim hızının küçük omurga ağda RCNN’e oranla 18.3 kat arttığı gözlemlenmektedir. Orta ve büyük boyutlu ağlarda bu oran küçük boyutlu ağa göre daha az olmakla birlikte RCNN’e kıyasla sırası ile 14 ve 8.8 kat artmıştır. Bir imgenin test süresini saniye cinsinden kıyasladığımızda ise Fast RCNN’in RCNN’e göre

test işlemini çok daha hızlı gerçekleştirdiği görülmektedir. Test hızı RCNN'e göre 146 kat ile en fazla büyük ağ yapısında artmıştır. Ayrıca, küçük omurga ağ için 98 kat hızlı test yapılırken orta boyutlu ağ yapısında ise test hızı 80 kat yükselmiştir. Son karşılaştırma metriği olan VOC07 verilerinden elde edilen ortalama kesinlik değerlerine (mAP) bakıldığında ise büyük omurga ağda tüm sınıfların ortalama başarımları %66.0'dan %66.6'ya yükselmiştir. Küçük ve orta büyüklükteki ağların başarımları karşılaştırıldığında ise RCNN yönteminin Fast RCNN'e göre sırası ile %1.4 ve %1.0 daha yüksek olduğu tespit edilmiştir.

Çizelge 2.1. Fast RCNN ile RCNN yöntemlerinin karşılaştırılması [40]

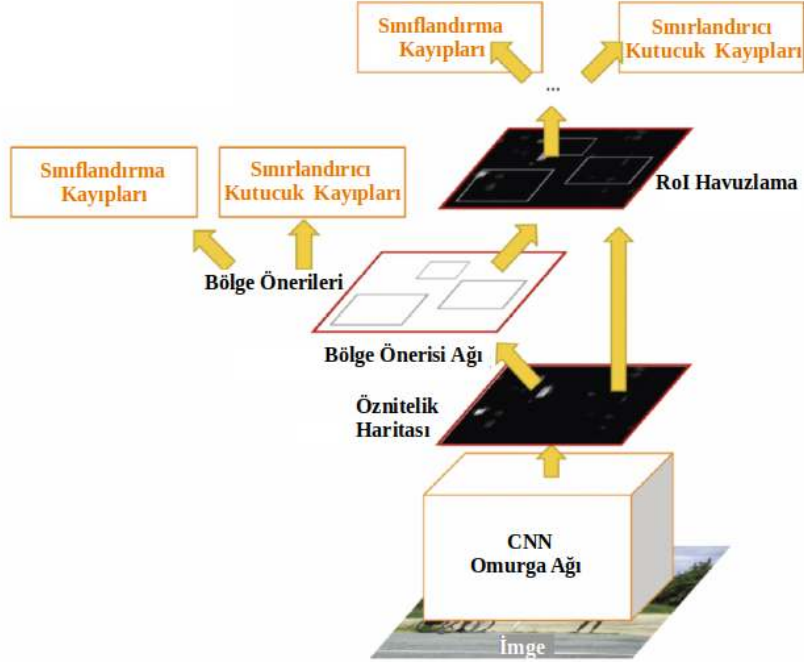
	Fast RCNN			RCNN		
	Küçük	Orta	Büyük	Küçük	Orta	Büyük
Eğitim süresi (saat)	1.2	2.0	9.5	22	28	84
Eğitim hızı	18.3x	14.0x	8.8x	1x	1x	1x
Test oranı (sn/imge)	0.10	0.15	0.32	9.8	12.1	47.0
Test hızı	98x	80x	146x	1x	1x	1x
VOC07 mAP	57.1	59.2	66.9	58.5	60.2	66.0

2.2.1.3 Faster RCNN

Faster RCNN [43], RCNN ailesinin diğer yöntemlerine göre hem hızı hem de doğruluğu arttırmak amacı ile oluşturulan bir nesne tespiti yöntemidir. Bu yöntemde, eğitim ve test hızını yükseltmek için önceki RCNN yöntemlerinden farklı olarak bölge önerisi ağı (RPN) sunulmuştur. Faster RCNN, RPN yöntemi ile bölge önerilerini elde ederken yöntemin geri kalan kısımları Fast RCNN ile aynıdır. Faster RCNN yönteminde ilk adımda görüntü, seçilen CNN omurga ağından geçer ve görüntüye ait öznelik haritası elde edilmiş olur. Daha sonra öznelik haritası üzerinde RPN yöntemi uygulanarak bölge önerileri alınır ve bu bölge önerileri kullanılarak sınıflandırma ile sınırlandırıcı kutucuk tahminleri yapılır. Faster RCNN'e ait genel mimari yapısı Şekil 2.9'da sunulmuştur.

Bölge önerisi ağı (RPN) yöntemini detaylı açıklayacak olursak bu yöntemde herhangi boyutta bir görüntü girdisi alınır ve çıktı olarak dikdörtgensel şekilde nesne önerileri bir nesnelik skoru ile birlikte sunulur. Bölge önerileri üretilirken öznelik haritası üzerinde

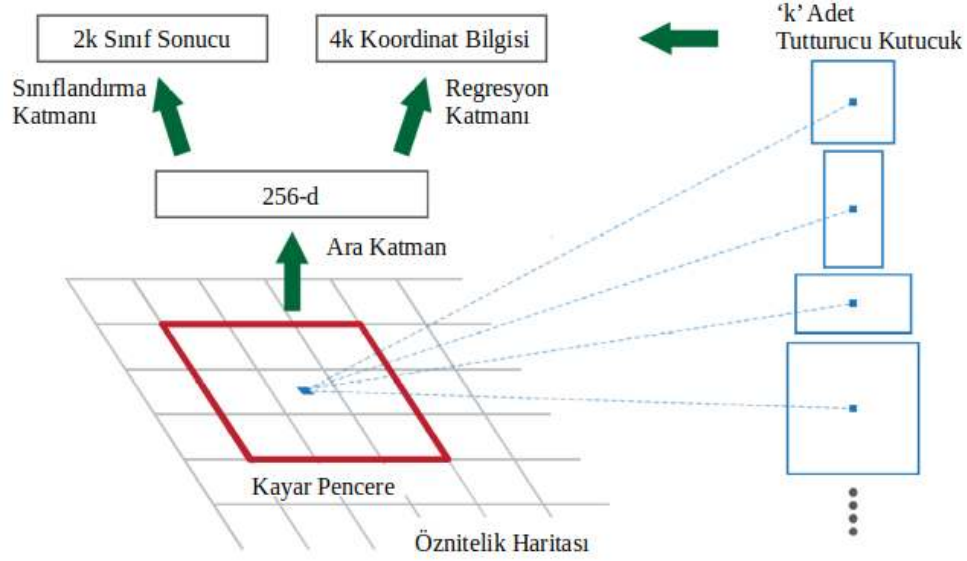
$n \times n$ boyutlu kayar pencereler gezdirilir. Orijinal kaynakta n sayısı 3 olarak seçilmiştir. Daha sonra her bir kayar pencere bir düşük boyutlu öznitelik ile eşleştirilir. Örneğin bu boyut VGG omurga ağı için 512'dir. Eşleştirilen öznitelik daha sonra aynı anda tam bağlantılı sınırlandırıcı kutucuk regresyon katmanı ve tam bağlantılı kutucuk sınıflandırıcı katmanlarına iletilir.



Şekil 2.9. Faster RCNN mimarisi [43]

Her bir kayar pencere konumu için maksimum k adet eş zamanlı bölge önerisi tahmini yapılır. Dolayısı ile regresyon katmanı çıktı olarak bölge önerilerine ait k adet kutucuk için $4k$ koordinat noktası üretirken sınıflandırıcı katmanı ise bölge önerilerinin bir nesneye ait olup olmama olasılığına dair $2k$ tahmin çıktısı üretir. Üretilen k adet bölge önerisi k adet referans kutucuğa bağlı olarak parametrize edilmektedir. Referans kutucukların her birine tutturucu (anchor) adı verilmektedir. Bir tutturucu kutucuk, kayar pencerenin merkez noktasına denk gelecek şekilde farklı en boy oranı ve ölçeklerde üretilir. Genel olarak kayar pencerenin her konumu için 3 farklı ölçek ve 3 farklı en boy oranında toplamda $k = 9$ adet tutturucu kutucuk kullanılır. Formülize edecek olursak $W \times H$ boyutlu bir evrişimsel öznitelik haritası için toplamda $W \times H \times k$ adet tutturucu kutucuk kullanılır. Şekil 2.10'da bölge önerisi ağına ait özet bir gösterim sunulmaktadır.

RPN yönteminin önemli bir özelliği hem tutturucu kutucukların hem de bu kutucuklara bağlı olarak önerileri hesaplayan fonksiyonların ötelemelerden bağımsız olmasıdır. Yani bu yöntemde bir görüntüdeki nesne ötelenirse öneri de aynı şekilde ötelenir ve aynı fonksiyon bu öneriyi yeni lokasyonda da tahmin edebilir.



Şekil 2.10. Bölge önerisi ağı (RPN) [43]

Bölge önerisi ağının eğitiminde tutturucu kutucukların her birine nesne içerip içermemesine göre ikili sınıflandırma etiketi atanır. Tutturuculardan kesin referans ile kesişimi (IoU) en yüksek olana veya herhangi bir kesin referans ile IoU değeri 0.7'den büyük veya eşit olana pozitif etiket atanır. Pozitif etiket atanması ilgili tutturucunun nesne içerdiği anlamına gelmektedir. Öte yandan eğer bir tutturucunun herhangi bir kesin referans ile IoU değeri 0.3'ü geçmiyorsa bu kutucuğa negatif etiket atanır. Negatif etiket ilgili bölgede bir nesne bulunmadığını belirtmektedir. Tutturucu kutucuklardan iki şartı da sağlamayanların eğitime katkısı bulunmamaktadır. Tüm bu tanımlamalar kullanıldığında Faster RCNN'de kullanılan kayıp fonksiyonları denklem 2.4'te ifade edilmiştir.

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (2.4)$$

Denklem 2.4'te i bir mini grup içerisindeki tutturucu kutucuk indeksini, p_i ise ilgili indeksdeki tutturucu kutucuğun bir objeye ait olup olmama olasılığını belirtir. Kesin referansa

ait p_i^* etiketi tutturucu kutucuk pozitif ise 1, negatif ise 0 değerini alır. t_i tahmin edilen sınırlandırıcı kutucuğa, t_i^* ise pozitif tutturucunun ilişkili olduğu kesin referansa ait 4 adet koordinat bilgisini içerir. Sınıflandırma kaybı L_{cls} , nesne varlığına ait logaritmik kaybı tanımlarken regresyon kaybı için $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ ifadesi kullanılır. Bu ifadede R , denklem 2.3'te tanımlanan gürbüz kayıp fonksiyonudur. $p_i^* L_{reg}$ terimi, regresyon kayıplarının sadece pozitif tutturucular için etkin olduğunu açıklar. Sınıflandırma ve regresyon katmanları sırası ile p_i ve t_i çıktıları üretir. Sınıflandırma kayıpları N_{cls} değeri ile normalize edilirken regresyon kayıpları N_{reg} değeri ile normalize edilip λ dengeleme parametresi ile ağırlıklandırılmıştır. Faster RCNN'in sunulan ilk versiyonunda N_{cls} mini grup sayısı 256 olarak seçilirken N_{reg} ise tutturucuların sayıları ~ 2400 olarak seçilmiştir. Sınıflandırma ve regresyon değerlerinin eşit ağırlıklandırılması için λ değeri 10 olarak varsayılmıştır. Sınırlandırıcı kutucuk regresyonu için 4 koordinat değeri denklem 2.5'te sunulan yöntemle parametrize edilmiştir.

$$\begin{aligned}
t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a, \\
t_w &= \log\left(\frac{w}{w_a}\right), t_h = \log\left(\frac{h}{h_a}\right), \\
t_x^* &= (x^* - x_a)/w_a, t_y^* = (y^* - y_a)/h_a, \\
t_w^* &= \log\left(\frac{w^*}{w_a}\right), t_h^* = \log\left(\frac{h^*}{h_a}\right)
\end{aligned} \tag{2.5}$$

Denklem 2.5'te x ve y kutucuğun merkez koordinatlarını belirtirken w ve h en ve boy değerlerini belirtir. x , x_a ve x^* değişkenleri sırası ile tahmin edilen kutucuk, tutturucu kutucuk ve kesin referans değerleridir. Bir tutturucu kutucuğun en yakın kesin referans ile regresyonu hedeflenmiştir.

Yapılan denemeler sonucunda [44] Faster RCNN ile diğer RCNN ailelerine göre daha hızlı ve yüksek doğrulukta sonuç alındığı gözlemlenmiştir. Denemeler yapılırken [44]'de belirtilen ve Korea Expressway Corporation tarafından toplanılan veri kümesi kullanılmıştır. İlgili veri kümesi hem sınıflandırma hem de tespit amacı ile toplanmış olup farklı tipte araç sınıfları içermektedir. Tüm RCNN yöntemlerinin karşılaştırılması Çizelge 2.2'de sunulmuştur.

Çizelge 2.2. Faster RCNN ile RCNN yöntemlerinin karşılaştırılması [44]

	RCNN	Fast RCNN	Faster RCNN
Bir imgenin test zamanı	50 saniye	2 saniye	0.2 saniye
Hız	1x	25x	250x
VOC07 mAP	66.0	66.9	66.9

2.2.2 Tek Aşamalı Nesne Tespiti Yöntemleri

Tek aşamalı nesne tespiti algoritmaları, iki aşamalılarından farklı olarak sinir ağı üzerinden tek geçiş ile sınırlandırıcı kutucuk tahmini yapabilmektedir. Tek aşamalı algoritmalar, sınıflandırma ve regresyon işlemlerini yoğun tutturucu kutucuklar kullanarak ayrı bir ağ ile bölge önerileri (RoI) üretmeden tamamlarlar [45]. Böylece sınıflandırma ve sınırlandırıcı kutucuk tahmini iki ayrı ağ yerine tek ağ üzerinde yapılmış olur. Bu yöntemlerin iki aşamalı yöntemlere kıyasla temel avantajı hızlı sonuç vermeleridir. Ancak iki aşamalı yöntemlere göre daha düşük doğruluk değerine sahip olabilmektedirler.

2.2.2.1 YOLO

YOLO [47], nesne tespiti işleminin imge girdisinden başlayıp sınırlandırıcı kutucuk koordinat ve sınıflandırma tahminlerine kadar tek bir regresyon problemi olarak ele alındığı bir yöntemdir. YOLO (You Only Look Once) ismi açılımı ile de amacını özetlemektedir. Görüntüde bulunan nesnenin sınıf ve konum tahmini tek bir aşama ile tamamlanır. Şekil 2.11'de YOLO mimarisine ait genel bir özet verilmiştir. Buna göre, tek bir evrişimsel



Şekil 2.11. YOLO mimarisine ait genel özet [47]

ağ modeli üzerinden hem çok sayıda sınırlandırıcı kutucuk tahmini hem de bu kutucuklara ait sınıf tahminleri yapılmaktadır. Eğitim işlemi tüm görüntü üzerinde yapıldığı için tespit performansı da optimize edilmiş olur. Bu yöntem ile YOLO, geleneksel nesne tespiti algoritmalarına göre birçok avantaj sağlamaktadır.

YOLO, tespit işlemine bir regresyon problemi gibi yaklaştığı için diğer yöntemlere göre daha basit bir yapısı vardır ve oldukça hızlıdır. YOLO algoritmasının asıl versiyonu saniyede 45 çerçeve ile işlem yapabilirken hızlı bir versiyonu ise saniyede 150 çerçeveden fazla işlem yapabilmektedir. Yani bir video akışı YOLO algoritması kullanılarak 25 milisaniye bir gecikme ile gerçek zamanlı işlenebilmektedir. YOLO'nun diğer gerçek zamanlı algoritmalara göre bir avantajı ise ortalama doğruluk değerinin daha iyi olmasıdır. YOLO, RCNN ailesinde uygulanan kayar pencere ve bölge önerisi tabanlı teknikler yerine eğitim ve test sırasında tüm görüntüyü kullandığı için dolaylı olarak sınıflara ait içerik bilgisini de kodlar. Böylece RCNN ailesinin bir dezavantajı olan görüntünün arka plan kısımlarına ait yapılan hatalar YOLO ile daha az gözlemlenmektedir.

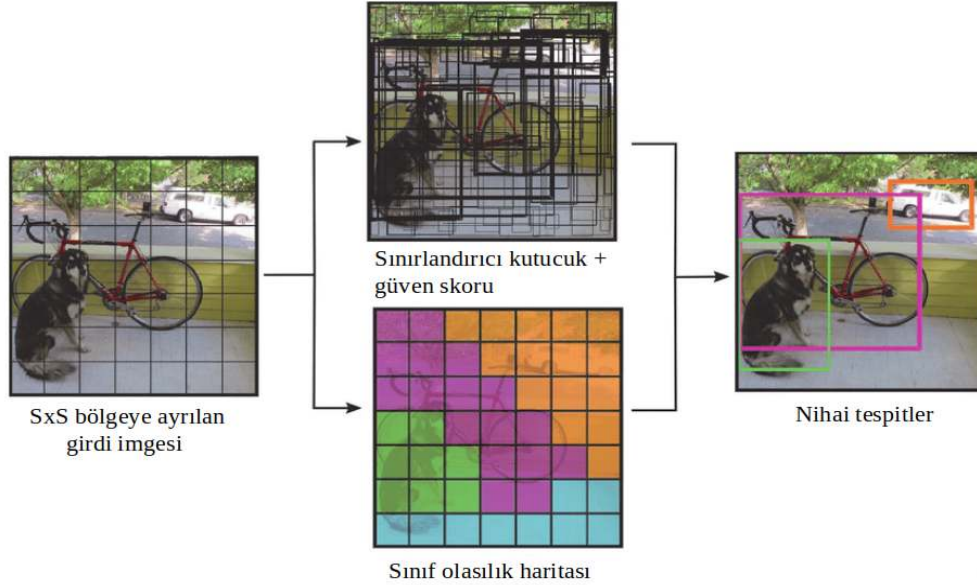
YOLO'nun bir başka avantajı ise nesnelerin genel görünümünü öğrenmesidir. Bu nedenle doğal görüntüler üzerinde eğitilip sanatsal eserler üzerinde test edildiğinde DPM ve RCNN yöntemlerine göre daha iyi sonuç vermektedir. Bu özelliği ile YOLO'nun farklı alanlarda kullanımında veya beklenmeyen girdi görüntüleri ile karşılaşıldığında performansının daha az etkilendiği düşünülmektedir. Ancak YOLO'nun başarı oranı Faster RCNN gibi gelişmiş yöntemlere kıyasla daha düşük çıkmaktadır. Nesne tespitinde hızlı olmasına rağmen özellikle küçük nesnelerin konumlandırılmasında zorlanmaktadır.

YOLO, girdi görüntüsünü $S \times S$ adet bölgeye bölerek işlem yapmaktadır. Eğer bir nesnenin merkez noktası herhangi bir bölgenin içine düşerse ilgili bölge bu nesneyi tespit etmekle yükümlüdür. Her bölgede B adet sınırlandırıcı kutucuk ve bu kutucuklara ait güven skoru tahmin edilir. Güven skoru, modelin bir kutucuk içerisinde nesne olup olmadığını ve bu tahmin edilen kutucuğun doğruluğunu ölçmektedir. Güven skoru $P_r(Nesne) * IoU_{tahmin}^{referans}$ ile ifade edilmektedir. Eğer bir bölgede nesne bulunmuyorsa güven skoru sıfır olarak hesaplanır. Bölge içerisinde nesne var ise güven skoru kesin doğruluk ve tahmin edilen kutucuğun kesişimi ile belirlenir.

Her sınırlandırıcı kutucuk x, y, w, h ve güven skoru olmak üzere 5 adet tahminden oluşmaktadır. (x, y) , bölge sınırlarına göreceli olarak alınan merkezi koordinatları temsil ederken nesnenin eni ve boyu tüm görüntüye göre göreceli şekilde tahmin edilir. Güven skoru ise bir üst paragrafta açıklandığı gibi tahmin edilen kutucuk ve kesin değer arasındaki kesişim ile belirlenir. Her bölgede ek olarak ilgili bölgenin nesne içermesine bağlı olarak C adet koşullu sınıf olasılıkları $P_r(Kategori_i|Nesne)$ tahmin edilir. Bir bölgede B adet sınırlandırıcı kutucuktan bağımsız olarak sadece bir adet sınıf tahmini yapılır. Test aşamasında ise koşullu sınıf olasılıkları ile her bir kutucuğa ait güven skoru denklem 2.6'da verildiği gibi çarpılır. Böylece her kutucuk için sınıf özelinde güven skoru elde edilir. Bu skor ile bir kutucuğa ait sınıf olasılığı ve bu kutucuğun nesneye ne kadar iyi denk geldiği belirlenmiş olur.

$$\text{Güven skoru} = P_r(Kategori_i|Nesne) * IoU_{\text{tahmin}}^{\text{referans}} \quad (2.6)$$

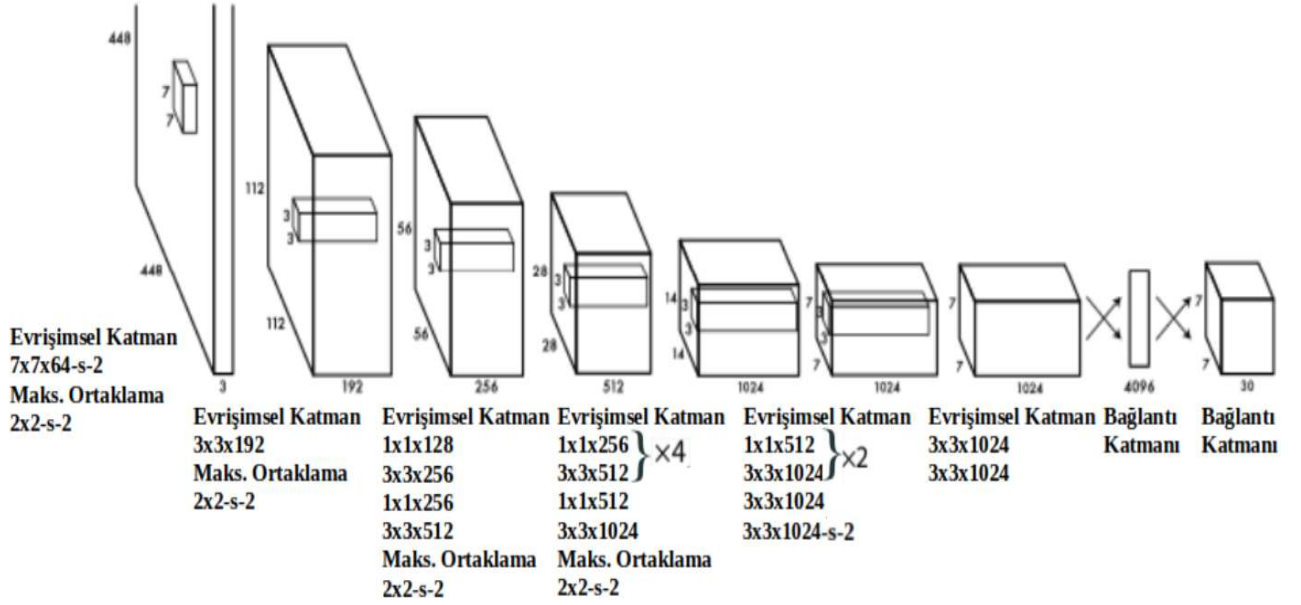
Şekil 2.12'de girdi görüntüsünün SxS bölgeye ayrılmasından başlayıp nihai tespitlerin elde edilme sürecini özetlemektedir.



Şekil 2.12. YOLO ile nesne tespiti süreci [47]

YOLO, omurga ağ olarak GoogLeNet [46] mimarisinden esinlenmiş olup 24 adet evrişimsel katman ve bunları takiben iki adet tam bağlantılı katman içermektedir. GoogLe-

Net mimarisinde kullanılan başlangıç modülleri yerine 1x1 boyutlu indirgeme katmanı ile devamında 3x3 boyutlu evrişimsel katmanlar kullanılmıştır. Şekil 2.13'te YOLO'ya ait omurga ağın özet bir görseli sunulmuştur. Şekil 2.11'de ikinci adımda belirtilen ve yeniden boyutlandırılan görüntü üzerinde işletilen evrişimsel ağ burada bahsedilen ağdır.



Şekil 2.13. YOLO CNN ağ yapısı [47]

YOLO'nun ortalama kesinlik değerini arttırmak için artık kareler toplamı (sum squared error) kullanılarak optimizasyon yapılmıştır. Artık kareler toplamı ile optimizasyon hızlı ve kolay iken bu yöntem kullanıldığında ortalama kesinlik değerinin istenilen seviyeye ulaşmadığı gözlemlenmiştir. Bunun nedeni, konumlandırma hataları ile sınıflandırma hatalarının eşit ağırlıklandırılmasıdır. Ancak nesne tespiti görevlerinde iki etmen her zaman aynı görülmemektedir ve genel olarak bir görüntüde nesne içermeyen hücrelerin sayısı nesne içerenlere göre oldukça fazladır. Nesne içermeyen hücrelerin güven skoru sıfıra yakınsayacağı için bu hücrelerin nesne içerenler ile eşit ağırlıkta hesaba katılması modelin erken ıraksamasına ve kararsızlığın artmasına neden olacaktır. Bunu önlemek için sınırlandırıcı kutucuk tahminlerine ait kayıpların kat sayısı arttırılırken obje içermeyen hücrelere ait güven skoru kayıplarının kat sayısı azaltılmıştır. Kat sayıları düzenlemek için sırasıyla $\lambda_{koordinat}$ ve $\lambda_{objeyok}$ olmak üzere iki adet parametre kullanılır.

mıştır. $\lambda_{koordinat}$ değeri 5 ve $\lambda_{objeyok}$ değeri 0.5 olarak seçilmiştir. Artık kareler toplamı yöntemi, büyük ve küçük kutucuklara eşit ağırlık vermektedir. Ancak büyük kutular hesaplanırken yapılan küçük sapmalar küçük kutulardaki sapmalara göre daha önemsiz olacağı için hata hesaplamalarında bu durum da göz önünde bulundurulmalıdır. Bunu önlemek için YOLO'da sınırlandırıcı kutucukların boyut bilgileri direkt hesaba katılmak yerine karekökleri alınarak revize edilmiştir.

YOLO, her hücre içerisinde birçok sınırlandırıcı kutucuk tahmini yapmaktadır. Ancak eğitim aşamasında her bir nesne için bir sınırlandırıcı kutucuk olması istendiğinden kesin referans değeri ile kesişimi en yüksek olan kutucuk dikkate alınmaktadır. Böylece sınırlandırıcı kutucuk tahminleri belli sınıf, boyut ve en boy oranlarına yoğunlaşmış ve daha doğru sonuçlar verecek ve modelin genel geri çekme değeri yükselecektir. Eğitim aşamasında yukarıda bahsedilen optimizasyonlar denklem 2.7'de verildiği gibi çok terimli kayıp fonksiyonu ile uygulanmıştır.

$$\begin{aligned}
& \lambda_{koordinat} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{nesne} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \lambda_{koordinat} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{nesne} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{nesne} (K_i - \hat{K}_i) \\
& + \lambda_{objeyok} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{nesne} (K_i - \hat{K}_i) \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{nesne} \sum_{k \in kategori} (p_i(k) - \hat{p}_i(k))^2
\end{aligned} \tag{2.7}$$

Denklem 2.7'de $\mathbb{1}_{ij}^{nesne}$ bir hücredeki nesnenin varlığını belirtmek için kullanılırken $\mathbb{1}_i^{nesne}$ ise i numaralı hücredeki j 'inci sınırlandırıcı kutucuk tahmininden sorumludur. Kayıp fonksiyonu sınıflandırma hatalarını sadece ilgili hücrede obje bulunursa değerlendirir. Sınırlandırıcı kutucuk koordinatlarına ait hatalar ise kesin referans ile kesişimi en yüksek olan kutucuk için ele alınmaktadır.

Çizelge 2.3'te YOLO ile RCNN ailesine ait başarı ve saniyedeki çerçeve sayısı bilgileri (frame per second - FPS) karşılaştırılmıştır [47]. Tüm modellerde eğitim ve test PASCAL VOC 2007 ve PASCAL VOC 2012 verileri ile yapılmıştır. Çizelgeye göre YOLO'nun hızı diğer algoritmalara göre çok daha fazla iken modelin başarısı iki aşamalı yöntemlere göre daha düşük çıkmıştır.

Model	Eğitim	Başarı (%)	FPS
YOLO	2007+2012	63.4	45
Fast RCNN	2007+2012	70	0.5
Faster RCNN	2007+2012	73.2	7

Çizelge 2.3. YOLO ile diğer modellerin karşılaştırılması [47]

YOLO'nun daha hızlı veya daha yüksek doğrulağa sahip olabilmesi için farklı versiyonları çıkarılmıştır. Bunlardan ilki olan Fast YOLO (Hızlı YOLO) [47], 24 yerine 9 adet evrimsel katman kullandığı için YOLO'ya kıyasla daha hızlı sonuç vermekle birlikte daha düşük performansa sahiptir. Bir başka versiyon olan YOLOv2 [48], YOLO'nun dezavantajı olan lokalizasyon hatalarını azaltmak ve görece düşük geri çağırma değerini yükseltmek için YOLO'daki evrimsel katmanlara toplu normalleştirme ekleyerek ortalama kesinlik değerini %2 arttırmıştır.

YOLOv2, Faster RCNN'de kullanılan tutturucu kutucuk yöntemini de kullanarak daha doğru tespitler yapmayı hedeflemiştir. Ayrıca, modelin performansını arttırmak için Darknet-19 adında yeni bir omurga ağ tasarlanmıştır. Darknet-19, 19 adet evrimsel ağıdan ve 5 adet maksimum ortaklama katmanlarından oluşmaktadır. %91.2 kesinlik değeri ile %88 kesinliğe ulaşan YOLO'dan daha iyi sonuç vermiştir [48]. YOLOv3 [49] ise öznetlik çıkarımı için yeni bir ağ sunmuştur. Sunulan ağ, YOLOv2'de kullanılan Darknet-19 ile artık ağların kaynaştırılması ile oluşturulmuştur. Bu ağın toplamda 53 adet katmanı olduğu için Darknet-53 ismi verilmiştir. Darknet-53 mimarisinde 3x3 ve 1x1 boyutlu filtreleri atlama bağlantıları ile birleştirilerek ResNet ağındaki artık ağa benzer bir yapı kurulmuştur.

Çizelge 2.4'te YOLO'nun farklı versiyonlarında kullanılan omurga ağlar ile yüksek başarı oranına sahip olan Resnet-101 ve Resnet-152 ağlarının karşılaştırılması sunulmuştur.

Omurga Ağ	ilk-1 (%)	İlk-5 (%)	FPS
Darknet-19	74.1	91.8	171
Resnet-101	77.1	93.7	53
Resnet-152	77.6	93.8	37
Darknet-53	77.2	93.8	78

Çizelge 2.4. YOLO omurga ağları ile diğer ağların karşılaştırılması [49]

Karşılaştırma yapılırken ilk 1 ve ilk 5 hata metrikleri göz önünde bulundurulmuştur. İlk 1 hata, en yüksek tahmin ile etiketin aynı olup olmadığını kontrol ederken ilk 5 hata ise en yüksek olasılıklara sahip beş tahminden birinin etiket ile aynı olup olmadığını kontrol etmektedir. Buna göre YOLOv3'de kullanılan Darknet-53 omurga ağının ilk 1 ve ilk 5 hata oranları yüksek başarıya sahip Resnet-152 ağı ile benzer iken Darknet-53 bu ağa göre neredeyse iki kat daha hızlıdır. Benzer şekilde karşılaştırma yapılan tüm ağlar içinde en hızlısı saniyede 171 çerçeve sayısı (FPS) ile YOLOv2'de kullanılan Darknet-19 ağıdır. Ancak bu ağın ilk 1 ve ilk 5 hata değerleri karşılaştırma yapılan ağlara göre daha düşüktür.

2.2.2.2 SSD

SSD (single shot multibox detector) [50], nesne tespiti için tek bir derin öğrenme ağı kullanan bir yöntemdir. SSD, öznetelik haritasındaki her konum için sınırlandırıcı kutuların çıktılarını farklı en boy oranları ve boyutlara sahip varsayılan kutulara ayırır. Model, tahmin yaparken varsayılan her kutu için nesne varlığını puanlar ve nesnenin şekli ile daha doğru eşleşmesi için kutular üzerinde düzeltme işlemleri uygular. Modelin önemli bir özelliği ise farklı öznetelik haritaları ve çözünürlük değerlerine ait birçok tahmini birleştirmesidir. Böylece farklı boyuttaki nesnelere doğru bir şekilde tespit edebilmeyi amaçlamaktadır.

Detaylı inceleyecek olursak SSD modeli ileri beslemeli bir evrimsel ağ yöntemine dayanmaktadır. Bu yöntem ile sabit boyutlardaki sınırlandırıcı kutular ve bu kutuların nesne içerme puanları üretilir. Daha sonra maksimum olmayan bastırma işlemi ile son tespitler elde edilir. Modelin ilk katmanları görüntü sınıflandırma görevlerinde kullanılan

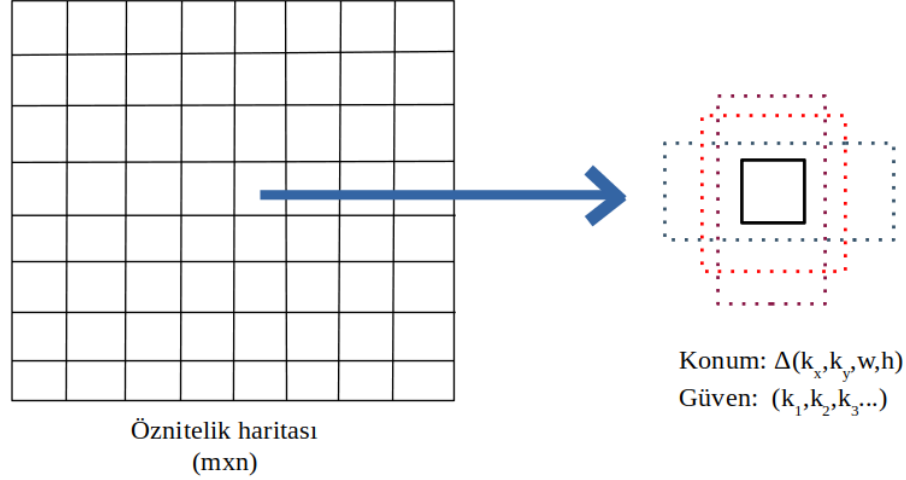
standart CNN mimarilerinden birine aittir ve temel ağ olarak adlandırılır. Daha sonraki katmanlarda ise tespit için yardımcı yapılar eklenir. Yardımcı yapıların eklenmesindeki amaç aşağıda belirtilen özelliklere sahip tespitler üretebilmektir:

Tespit için çok ölçekli öznitelik haritaları: Seçilen temel ağın belli bir kısmı kullanılır ve kullanılan kısmın ardına evrimsel öznitelik çıkarılması için ekstra katmanlar eklenir. Bu katmanlar boyutun gitgide düşürülmesini sağlayarak tespitler için çok ölçekli tahminler yapabilir.

Tespit için evrimsel tahminler: Eklenen her öznitelik katmanı, evrimsel filtreleri kullanarak tespitlere ait sabit tahminler oluşturabilmektedir. Örneğin $m \times n$ boyutlu ve p adet kanala sahip bir öznitelik haritasında olası bir tespite ait parametre tahmini $3 \times 3 \times p$ boyutlu çekirdek ile yapılır. Bu çekirdek tespite ait kategori skoru veya varsayılan kutu koordinatları ile tahmin edilen arasındaki ofseti hesaplar ve uygulandığı her $m \times n$ konumu için bir çıktı üretir.

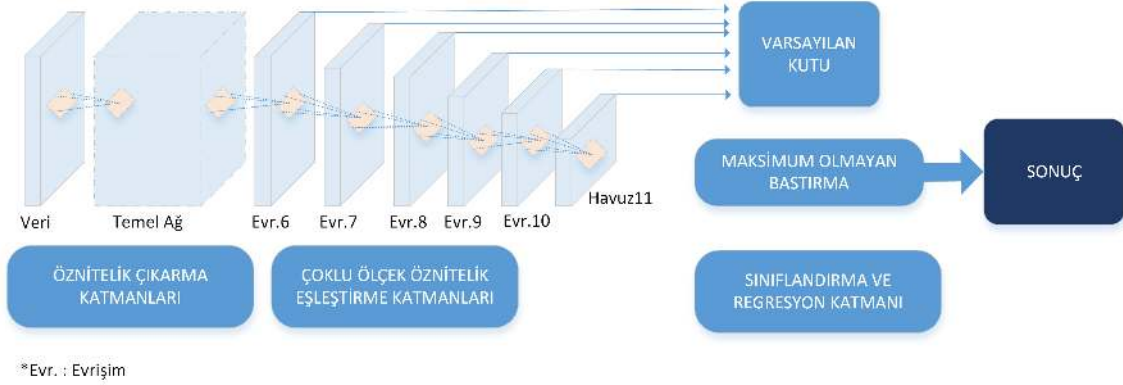
Varsayılan kutular ve en/boy oranları: Modeldeki çoklu öznitelik haritalarının her öznitelik harita hücresi için varsayılan sınırlandırıcı kutular bağdaştırılır. Bağdaştırılmış varsayılan kutular öznitelik haritasını evrimsel bir biçimde döşeyerek her kutunun pozisyonunun ilgili hücreye göre sabit kalmasını sağlamaktadır. Öznitelik haritasındaki her hücre için varsayılan kutu şekli ile göreceli olarak ofsetler ve bu kutuların içinde her sınıfa ait bir nesne bulunup bulunmama tahminleri üretilir. Bir konumdaki k tane kutunun her biri için c adet sınıf ve dört adet ofset hesaplaması yapılır. Böylece bir konum için toplamda $(c + 4)k$ adet filtre uygulanmaktadır. $m \times n$ boyutlu bir öznitelik haritasının tamamı düşünüldüğünde ise bu sayı $(c + 4)kmn$ olacaktır. Şekil 2.14'te varsayılan kutulara ait bir görsel sunulmuştur. Varsayılan kutular Faster RCNN'de bahsi geçen tutturucu kutucuk yöntemine benzer olsa da farklı çözünürlükteki birçok öznitelik haritasına uygulanır. Böylece olası birçok kutunun ayırt edilmesi amaçlanır.

Şekil 2.15'te SSD mimarisi sunulmuştur. Buna göre, girdi görüntüsü ilk başta temel ağdan geçirilir. Temel ağın öznitelik haritası çıkarımı işlemlerinin bir kısmından sonrası kullanılmamakla birlikte devamına yardımcı yapılar eklenir. Böylece görüntüde bulunan farklı boyut ve ölçekli nesnelerin tespiti sağlanır. SSD ile diğer modellerin eğitimi ara-



Şekil 2.14. SSD varsayılan kutu yapısı

sındaki temel fark SSD'nin kesin referans bilgisini modelin belli aşamalarında belirli çıktılara atmasıdır. Daha sonra kayıp fonksiyonları ve geri yayılım işlemi uçtan uca uygulanır. Eğitim aşamasında nesne tespiti için varsayılan kutular ve farklı boyutlara ait seçim yapılırken ek olarak veri arttırma tekniklerinden de yararlanır.



Şekil 2.15. SSD modeline ait genel bir özet

Eğitim aşamasında varsayılan kutuların hangi kesin referans ile eşleştirilmesi gerektiğinin karar verilmesi gerekmektedir. Bu nedenle her kesin referans kutusu için farklı konum, en boy oranı ve ölçeklerdeki varsayılan kutular arasında eşleştirme yapılmaktadır. Bu seçim yapılırken kesin referans ile varsayılan kutuların IoU değerinin belirli bir eşik değerini sağlamasına bakılmaktadır. Bu değer modelin orijinal versiyonunda 0.5 olarak seçilmiştir. Bu eşleştirme yöntemi ile kesişen birçok varsayılan kutu için yüksek

başarılı tahmin yapılarak modelin öğrenme süreci kolaylaştırılır.

Eğitim aşamasında eşleştirme yöntemi çoklu kategori için yapılmaktadır. Buna göre, $x_{ij}^p = \{1, 0\}$ ifadesi p kategorisi için i 'nci varsayılan kutu ile j 'nci kesin referans kutusunun eşleştirilmesine ait bir gösterimdir. Bu eşleştirme yönteminin sonucunda $\sum_i x_{ij}^p \geq 1$ ifadesine ulaşılır. Modelin kayıp fonksiyonu L , denklem 2.8'de sunulduğu gibi konumlandırma kayıpları (*loc*) ve güven kayıplarının (*conf*) ağırlıklandırılmış toplamı ile hesaplanmaktadır.

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \quad (2.8)$$

Denklem 2.8'de N eşleştirilmiş varsayılan kutuların sayısını belirtmektedir ve eğer $N = 0$ ise kayıp da 0'a eşitlenmektedir. Konumlandırma hatası ise tahmin edilen kutu (l) ile kesin referansa ait kutu (g) arasında yapılmakta olup denklem 2.3'te belirtilen $smooth_{L1}$ kaybı kullanılmaktadır. Varsayılan sınırlandırıcı kutuya (d) ait ofset düzenlenmeleri kutunun merkez koordinatları (cx, cy) ve en (w), boy (h) değerleri için yapılmaktadır. Konumlandırma hatası denklem 2.9'da verilmiştir. Denklemde, \hat{g}_j^{cx} ve \hat{g}_j^{cy} terimleri varsayılan sınırlandırıcı kutunun merkez noktalarının kesin referansa göre düzenlenmesini temsil ederken \hat{g}_j^w ve \hat{g}_j^h ise varsayılan sınırlandırıcı kutunun boyutlarının kesin referansa göre düzeltilmesini ifade etmektedir.

$$L_{loc}(x, l, g) = \sum_{i \in Konum} \sum_{m \in cx, cy, w, h} x_{ij}^k smooth_{L1}(l_i^m - \hat{g}_j^m) \quad (2.9)$$

$$\hat{g}_j^{cx} = \frac{(g_j^{cx} - d_i^{cx})}{d_i^w} \quad \hat{g}_j^{cy} = \frac{(g_j^{cy} - d_i^{cy})}{d_i^h}$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \quad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

Güven kaybı, softmax kaybının çoklu sınıflara ait güven (g) değerlerine uygulanması ile hesaplanmaktadır. Denklem 2.10'da güven kaybına ait formülizasyon sunulmuştur. Denklemde \hat{c}_i^p ifadesi softmax fonksiyonunu belirtirken c_i^p ifadesi ise i 'nci indekste p

kategorisine ait sınıf bazlı güven skorunu belirtir.

$$L_{conf}(X, c) = - \sum_{i \in konum} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0), \quad \hat{c}_i^p = \frac{\exp c_i^p}{\sum_p \exp c_i^p} \quad (2.10)$$

SSD'de yararlanılan ek yöntemlerden biri olan negatif madencilik (hard negative mining), eşleştirme adımından sonra uygulanmaktadır. Genellikle bu adımdan sonra varsayılan kutuların çoğu eşleştirmeden geçemeyeceği için negatif olacaktır. Bu nedenle pozitif ve negatif kutular arasında dengesizlik meydana gelecektir. Oluşan dengesizliği önlemek için negatif örneklerin hepsini kullanmak yerine bunları varsayılan her kutu için en yüksek güven kaybına göre sıralayıp içlerinden en yükseği seçerek negatif ve pozitiflerin oranının en fazla 3 : 1 olması sağlanır. Bu yöntemin hızlı bir optimizasyon ve eğitimde stabilite sağladığı gözlemlenmiştir. Kullanılan diğer yöntem olan veri artırma ile algoritmanın çeşitli nesne boyutlarına ve şekillerine karşı daha gürbüz olması hedeflenmiştir. Bunun için her eğitim görüntüsü aşağıdaki yöntemlere göre rastgele örneklenmektedir:

- 1- Asıl girdi görüntüsünün hepsi kullanılır.
- 2- Nesnelere ile IoU değeri 0.1, 0.3, 0.5, 0.7 veya 0.9 olacak şekilde bölgeler örneklenir.
- 3- Rastgele bir bölge örneklenir.

Buna göre görüntü üzerinden örneklenen her bir bölgenin boyutu görüntünün 0.1 ile 1 katı, en boy oranı ise 0.5 ile 2 arasında olacaktır. Kesin referansın merkez noktası örneklenen bölge içinde ise bölge ile kesişen kısım alınır. Bu adımlardan sonra her bir örneklenen bölge sabit bir boyuta ulaşacak şekilde tekrar boyutlandırılır ve yatay düzlemde 0.5 olasılık ile ters çevrilir. Ek olarak çeşitliliği arttırmak için görüntü üzerinde fotometrik bozunumlar uygulanır.

Çizelge 2.5'te SSD ile YOLO ve RCNN ailelerinin karşılaştırılması gösterilmiştir [50]. Karşılaştırmada Faster RCNN ve YOLO modellerinde VGG16 omurga ağı kullanılmıştır. SSD modeli ise SSD300 ve SSD512 olmak üzere iki versiyonda incelenmiştir. SSD300 ile SSD512'nin farkı girdi görüntülerinin çözünürlük değerleridir. Karşılaştırma yapılırken Pascal VOC2007 veri kümesi kullanılmıştır. Karşılaştırma sonucunda SSD512'nin

ortalama kesinlik değerinin diğer modellerden daha yüksek olduğu gözlemlenmiştir. Ancak saniyedeki çerçeve sayısına (FPS) bakıldığında gerçek zamanlı olma özelliğinden uzaklaşmıştır. Karşılaştırılan modellerin içinden en hızlısı Fast YOLO olmakla beraber ortalama kesinlik değeri en düşük olanı da odur. Genel sonuç olarak ise SSD300 hem gerçek zamanlı hem de %74.3 ortalama kesinlik değeri ile başarılı bir modeldir. Karşılaştırmada tüm modellerde küme boyutu (batch size) 1 olarak seçilmiştir.

Model	mAP	FPS	Girdi Çözünürlüğü
Faster RCNN (VGG16)	73.2	7	1000 × 600
Fast YOLO	52.7	155	448 × 448
YOLO (VGG16)	66.4	21	448 × 448
SSD300	74.3	46	300 × 300
SSD512	76.8	19	512 × 512

Çizelge 2.5. SSD ile diğer modellerin karşılaştırılması [50]

2.2.3 Derin Öğrenmede Farklı Öğrenme Çeşitleri

Derin öğrenme tabanlı eğitim yapılırken veri kümesinin özelliklerine, eğitim için harcanan zaman kriterine ve performans beklentisine göre farklı öğrenme çeşitleri uygulanmaktadır. Farklı öğrenme tiplerine örnek olarak uçtan uca öğrenme, aktarım öğrenmesi ve az vuruşlu öğrenme verilebilir. Bu yöntemlere aşağıdaki alt başlıklarda değinilmiştir.

2.2.3.1 Uçtan Uca Öğrenme

Uçtan uca öğrenme [51], işlenmemiş verinin bir öğrenme algoritmasına el ile herhangi bir öznitelik çıkarımı yapılmadan beslenerek eğitilmesidir. Evrimsel sinir ağlarına verinin direkt girdi olarak verilmesi ve öznitelik çıkarımının omurga ağ tarafından yapılarak tüm öğrenme sürecinin tek seferde yapılması işlemi uçtan uca öğrenmenin bir örneğidir. Geri yayılım aşaması da seçilen ağın ağırlıkları öğrenip ilgili girdiler ile çıktılarını eşleştirilmesinde önemli rol oynamaktadır.

Sinir ağlarında uçtan uca öğrenmenin optimum sonucu vermediği koşullar olabilmektedir. Bu koşullardan bir tanesi eğitim verisinin sayısıdır. Uçtan uca eğitimde modelin özniteliklerinin çıkarabilmesi için yeterli sayıda veri olmalıdır. Bu nedenle çok sayıda

veriye gerek duyulmaktadır. Eğer veri kümesindeki veri sayısı gerekenden az ise uçtan uca eğitimle iyi bir sonuç alınmayabilir. Başka bir örnek ise çok sayıda katman içeren sinir ağı modelleridir. Bu tarz modellerin tek bir yapıda eğitilmesi için büyük bir hafıza gerekebilir. Böyle bir durumda ise ağın bütünü küçük parçalara bölünerek bu parçalar bağımsız bir şekilde eğitilebilir. Ancak bu yöntem her bir ağ kendi içinde optimize edildiği ve çıktılar sıradaki ağın girdisi olduğu için ideal bir çözüm olmayabilir.

2.2.3.2 Aktarım Öğrenmesi

Makine öğrenmesi ve derin öğrenme görevleri son yıllarda büyük gelişme gösterse de öğrenme süreci için çok sayıda veri gerekmektedir. Ancak uygulamalarda çok sayıda veri kullanmak maaliyetli olabilmektedir. Bu nedenle öğrenme işleminde modelin doğruluğunun çok sayıda veriye bağlı olduğu durumu ortadan kaldırmak ve elde olan az sayıda veriyi kullanarak iyi sonuç alabilmek için aktarım öğrenmesi kullanılmaktadır.

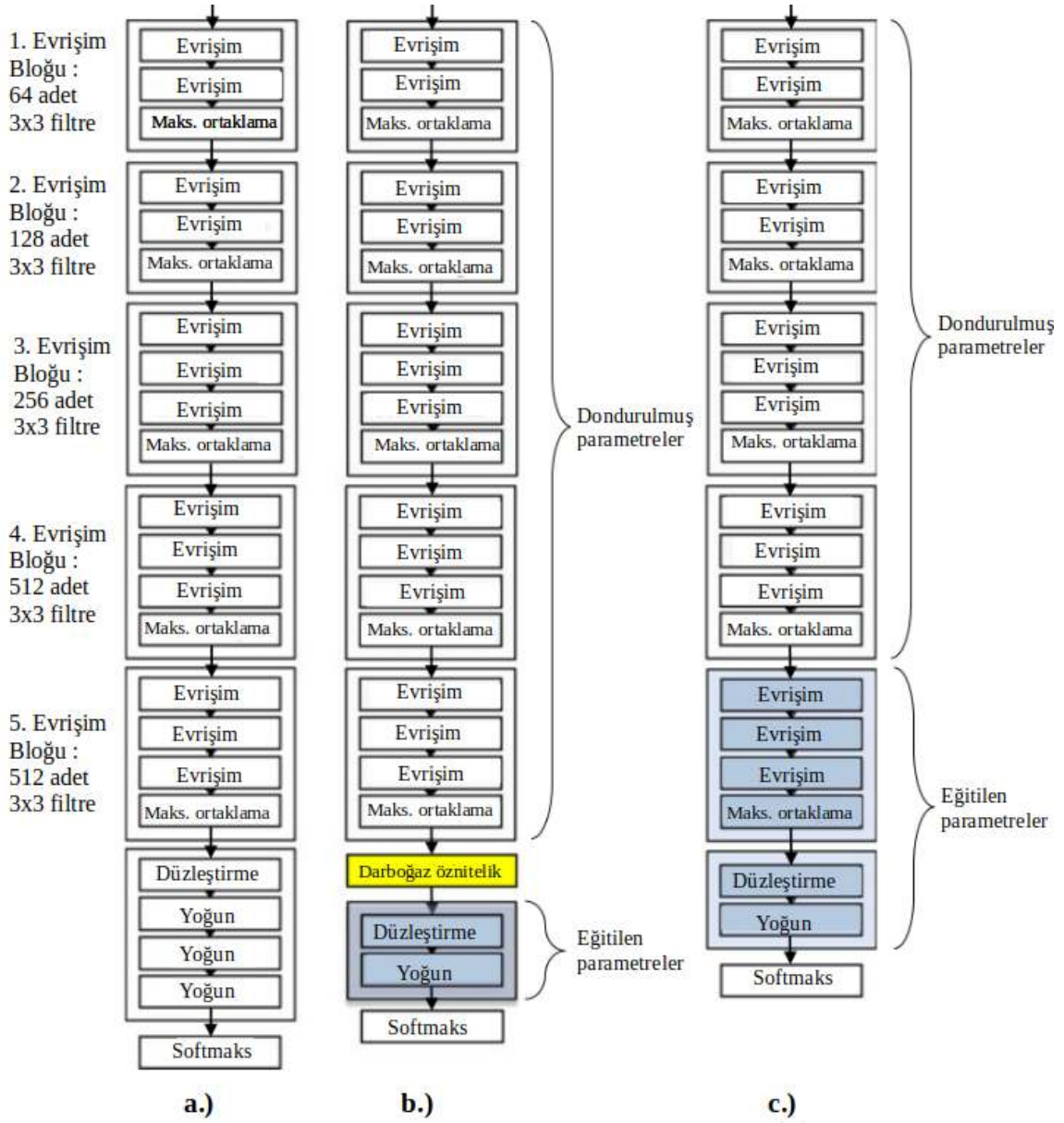
Genel tanımda aktarım öğrenmesi [52], verilen bir kaynak alanda öğrenilen görevin başka bir hedef alandaki öğrenme görevi için kullanılarak öğrenilen bilgilerle hedef alandaki başarıyı arttırmaktır. Son yıllarda derin evrimsel sinir ağlarında da aktarım öğrenmesi çokça kullanılmaktadır. Bunun başlıca nedeni, modelin derinliğinin arttıkça karmaşıklığının da artması ve buna bağlı olarak eğitim için çok zaman ve hesaplama kaynağı gerekmesidir. Ek olarak, veri eksikliğinden kaynaklı aşırı öğrenme ve eğitimde meydana gelen doğruluk değerindeki satürasyonlar da karmaşık modellerde açığa çıkan problemlerdendir. Bu gibi durumlarda aktarım öğrenmesi daha önce de bahsedildiği gibi eğitimdeki verimliliği oldukça arttırmaktadır. Tipik bir evrimsel sinir ağına ilk katmanlar renk, doku ve kenar gibi alt seviye öznitelikleri temsil ederken katmanlar derinleştikçe daha karmaşık ve yüksek seviye öznitelikler temsil edilmektedir. Eğer iki veri kümesi de benzer özellikler gösteriyorsa, evrimsel sinir ağları tarafından elde edilen alt seviye özniteliklerinin de benzer olması beklenmektedir. Bu benzerlik sayesinde bu katmanlara ait ağırlıklar paylaşılabilir ve üst seviye öznitelikler aktarım öğrenmesi ile öğrenilebilir. Bu nedenle aktarım öğrenmesi, evrimsel sinir ağlarında öznitelik betimlemelerinin aktarımı ve parametre aktarımına benzer bir göreve sahiptir.

Aktarım öğrenmesinde genellikle kaynak ve hedef alanlara ait görüntülerde bulunan

nesnelerin tipinden, farklı bakış açılarından vb. meydana gelen farklardan dolayı dengeleme işlemi yapılmaktadır. En yaygın yöntem, kaynak alana ait tam bağlantılı katmanın çıkarılıp yerine hedef kaynağa ait tam bağlantılı katmanın uyarlanmasıdır. Yeni uyarlanan katmanların boyutu hedef alandaki görüntülere ait özniteliklerin karmaşıklığına bağlı olmakla birlikte deneysel çalışmalara bağlı olarak değişebilmektedir [53].

Derin evrimsel ağlarda aktarım öğrenmesi yapılırken öznitelik çıkarıcı ve ince ayar olmak üzere iki popüler yöntem kullanılmaktadır [54]. Şekil 2.16'da sırası ile tüm evrimsel ağın eğitimi, öznitelik çıkarımı ile aktarım öğrenmesi ve ince ayar aktarım öğrenmesi sunulmuştur. Öznitelik çıkarıcı, öznitelik çıkarımı işlemi yapar. Bu kapsamda evrimsel işlemler sadece tek seferde ve ileri besleme yöntemi ile tamamlanır. Tam bağlantılı katmandan önceki katmanın çıktısı, eğitim verisinin darboğaz özniteliği olarak alınır. Daha sonra çeşitli sınıflandırıcılar sınıflandırma problemlerine uygun olacak şekilde uygulanır. Aktarım öğrenmesinde öznitelik çıkarıcının uygulanması eğitim zamanını ve hali hazırda çıkarılan öznitelikler olacağı için epok sayısını azaltacaktır. Çok sayıda evrimsel katmanı baştan aşağı birçok kez eğitmek yerine, öznitelikler tam bağlantılı katmana veya SVM gibi lineer sınıflandırıcılara beslenir. Her ne kadar evrim işlemi pahalı olsa da ve girdi görüntünün önceden eğitilmiş parametreler ve ağırlıklar kullanılarak öznitelik çıkarımının yapılması uzun zaman olsa da bu işlem sadece bir defa yapılmaktadır. Böylece, tüm öznitelik çıkarımı işlemi diğer yöntemlere göre daha az vakit almaktadır.

İnce ayar yönteminde ise Şekil 2.16 (c)'de gösterildiği şekilde evrimsel katmanların bir kısmı da eğitilir. Daha önce de bahsedildiği gibi bir evrimsel sinir ağında ilk evrim katmanları alt seviye öznitelikleri temsil ettiği için bu seviyedeki öznitelikler kaynak ve hedef alanlar için benzerlik gösterebilmektedir. Bu nedenle ince ayarda daha ileri seviyedeki katmanlar eğitilerek hedef alana ait ayırt edici özniteliklerin adaptasyonu amaçlanmıştır. Bu yöntemde ilk katmanlarda sabit parametreler kullanıldığı ve sadece eğitilen katmanlarda ileri ve geri yayılım işlemleri yapıldığı için uçtan uca eğitime göre daha az vakit alır.



Şekil 2.16. Aktarım öğrenmesine ait farklı konfigürasyonlar [54]

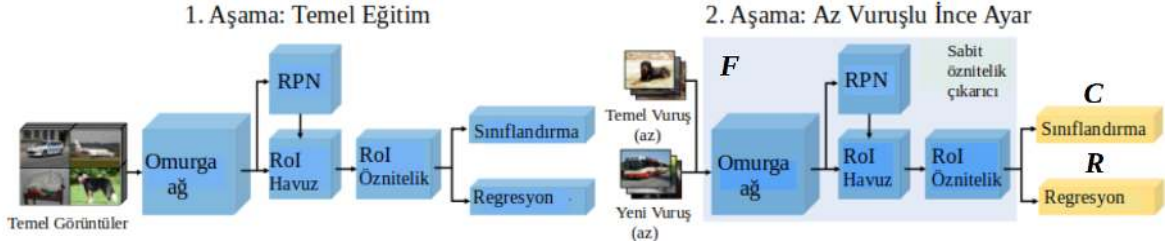
2.2.3.3 Az Vuruşlu Öğrenme

Az vuruşlu öğrenme [55], limitli sayıda örnek kullanılarak tahmin yapabilmeyi hedefleyen bir problemdir. Az vuruşlu öğrenme bilinen gözetimli öğrenmeden farklı olarak eğitim veri kümesinden öğrenileni test kümesi ile genellemek yerine direkt öğrenmeyi hedeflemektedir. Az vuruşlu öğrenme meta öğrenmenin bir çeşidi olduğu için öncelikle meta öğrenme açıklanmıştır.

Meta öğrenmenin temel hedefi, modelin çok az sayıda etiketli örnek veri kullanarak yeni görevlere ve alanlara hızla uyum sağlamasına yardımcı olabilecek görev seviyesinde meta bilgi elde etmektir. Bu nedenle birçok öğrenme bölümü uygulayarak bir öğrenme algoritmasını geliştirmeyi hedefler. Temel öğrenme aşamasında bir iç (temel) öğrenme modeli, tanımlanan bir veri kümesi ve objektif için görüntü sınıflandırma gibi bir görevi çözümler. Meta öğrenme aşamasında ise bir dış (meta) algoritma, iç öğrenme algoritmasını öğrenilen modelin bir dış hedefi iyileştirebileceği şekilde günceller. Örneğin bu hedef, bir performans genellemesi veya iç algoritmanın öğrenme hızı olabilir. Temel görevin öğrenme aşamaları, dış algoritmanın temel öğrenme algoritmasını öğrenmek için ihtiyaç duyduğu örnekleri sağlamaktır.

Meta öğrenme tanımı kapsamına geleneksel yöntemlerden çapraz doğrulama ile hiper parametrelerin rastgele aranması gibi birçok algoritma girebilir. Güncel sinir ağı meta öğreniminin belirgin özelliği, açıkça tanımlanmış bir meta seviye hedefi ve iç algoritmanın bu hedefe yönelik uçtan uca optimizasyonudur. Genellikle, meta öğrenme bir görev ailesinden örneklenen öğrenme bölümleri üzerinde uygulanır ve bu aileden örneklenen yeni görevler üzerinde iyi başarı gösteren bir temel öğrenme algoritması sağlar. Ancak, sınırlayıcı bir durumda, tüm eğitim bölümleri tek bir görevden örneklenir. Tez çalışmasında kullanılan az vuruşlu öğrenme modelinde iki aşamalı ince ayar yaklaşımı (two-stage fine-tuning (TFA)) kullanılmıştır [56]. Bu kapsamda temel tespit yöntemi olarak Faster RCNN kullanılmıştır.

Şekil 2.17 tez kapsamında kullanılan az vuruşlu öğrenme algoritmasına ait ağın genel bir şemasını çizmektedir. Şekil 2.17'de temel eğitimde kullanılan Faster RCNN modele ait öznitelik öğrenmeye ait bileşenler F ile belirtilmiş olup omurga ağını, bölge öneri ağını ve iki katmanlı tam bağlantılı alt modeli içerir. Bunların yanında kutu sınıflandırma C ve koordinat tahmini için kutu regresör R işlemlerinden oluşan kutu tahmincisi bulunmaktadır. Omurga ağ sonucu ve bölge önerisi ağı sonucu elde edilen öznitelikler sınıf bilgisine sahip değildir. Bu nedenle, temel sınıflardan öğrenilen öznitelikler ek bir parametre güncellemesi olmadan yeni sınıflara aktarılabilmektedir. Kullanılan az vuruşlu öğrenme yönteminde de bu nedenle öznitelik temsiline ait öğrenme ve kutu tahminine ait öğrenme işlemleri iki aşamaya ayrılmıştır.



Şekil 2.17. Az vuruşlu öğrenmeye ait ağ yapısı [56]

Temel Model Eğitimi

Modelin ilk aşamasında, öznitelik çıkarımı ve kutu tahmincisi sadece temel sınıflar C_b için eğitilmiştir. Eğitim sırasında kayıp fonksiyonları Faster RCNN'in orijinal modelinde belirtilen hali ile kullanılmıştır [43]. Birleştirilmiş toplam kayıp fonksiyonu aşağıdaki denklemde belirtilmiştir.

$$L = L_{rpn} + L_{cls} + L_{loc} \quad (2.11)$$

Denklem 2.11'de L_{rpn} bölge önerisi ağının çıktısına ilgili bölgenin arkaplan veya bir nesneye ait olup olmadığını anlamak için uygulanırken L_{cls} ise kutu sınıflandırıcı C için uygulanmaktadır. Son olarak L_{loc} ise kutu regresörü R 'ye ait kayıptır.

Az Vuruşlu İnce Ayar

İkinci aşamada ise hem temel hem de yeni sınıflardan oluşan ve her sınıf için K vuruşu sahip dengelenmiş küçük veri kümeleri oluşturulur. Rastgele ilklendirilen ağırlıklar kutu tahmin ağına beslenir ve tespit ağının son katmanları olan kutu sınıflandırıcı ve regresyon ağları ince ayar işlemine sokulur. Bu aşamada tüm öznitelik çıkarıcı F ağ sabit tutulur. Denklem 2.11'de belirtilen kayıp fonksiyonu bu aşamada da kullanılmıştır.

Kutu Sınıflandırıcı için Kosinüs Benzerliği

İkinci ince ayar aşamasında, kosinüs benzerliğine dayalı sınıflandırıcı kullanılmıştır [57]. Kutu sınıflandırıcı C 'ye ait ağırlık matrisi $W \in \mathbb{R}^{d \times c}$, $[w_1, w_2, \dots, w_c]$ şeklinde yazılabilir. Bu ifadede $w_c \in \mathbb{R}^d$ her bir sınıfa ait ağırlık vektörünü tanımlamaktır. C 'nin çıktısı, girdi özneliği $F(x)$ 'in ölçeklendirilmiş benzerlik skoru S ve farklı sınıflara ait ağırlık vektörle-

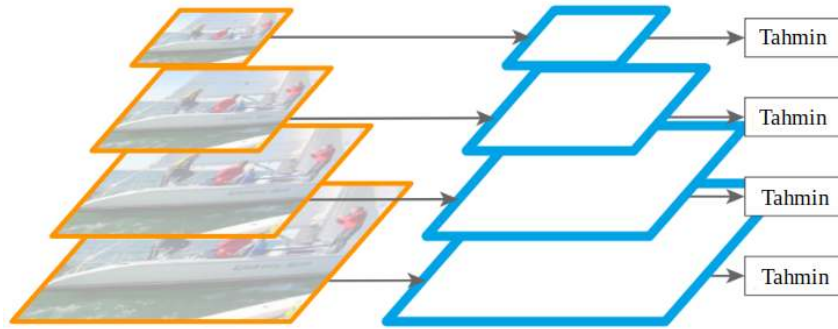
ridir. S 'nin girdileri ise şöyle tanımlanmıştır:

$$s_{i,j} = \frac{\alpha F(x)_i^T w_j}{\|F(x)_i\| \|w_j\|}, \quad (2.12)$$

Denklemden $s_{i,j}$ x girdisinin i 'nci nesne önerisi ile j sınıfına ait ağırlık vektörü arasındaki benzerlik skorunu belirtir. α ise ölçeklendirme faktörüdür. Kullanılan az vuruşlu öğrenme modelinde α sabit olarak 20 alınmıştır. Yapılan çalışmalarda kosinüs benzerliğinde kullanılan nesne seviyesinde öznitelik normalizasyon işleminin, tam bağlantılı sınıflandırıcıya kıyasla sınıflar arası varyansı düşürdüğünü ve yeni sınıflara ait tespit doğruluğunu arttırdığı gözlemlenmiştir. Bu fark özellikle az sayıda eğitim örneği olduğunda daha belirgin olmaktadır.

2.2.4 Öznitelik Piramit Ağı

Farklı boyuttaki nesnelerin tespiti bilgisayarlı görüde temel ve zor bir görevdir. Görüntü piramitlerinin üzerine kurulan öznitelik piramitleri (öznitelik görüntü piramitleri), bu probleme ait genel bir çözüm sunmaktadır [58]. Şekil 2.18'de görüntü piramidi yapısı sunulmuştur. Şekilden de görüldüğü üzere girdi görüntünün boyutları piramidin her seviyesinde değiştiği için piramit yapısı boyuttan bağımsız hale gelmektedir. Böylece, model farklı boyut skalasındaki nesneleri farklı piramit seviyelerini kullanarak tespit edebilmektedir.



Şekil 2.18. Görüntü piramidi yapısı [59]

Öznitelik piramit ağı da görüntü piramidi yapısına benzer şekilde, farklı seviyelerde öznitelik haritaları ile çalışır [59]. Farklı seviyelerde öznitelik haritaları çıkarılırken girdi

görüntüsü evrişimsel yaklaşım kullanılarak orantılı boyutta çıktılar verir. Bu işlem evrişimsel omurga ağ mimarilerinden bağımsız bir şekilde kullanılır. Orijinal çalışmada ResNet omurga ağ kullanılmıştır. Piramit ağı yapısı temel olarak aşağıdan yukarıya yolu, yukarıdan aşağıya yolu ve yanal bağlantıları içermektedir.

Aşağıdan yukarıya yol

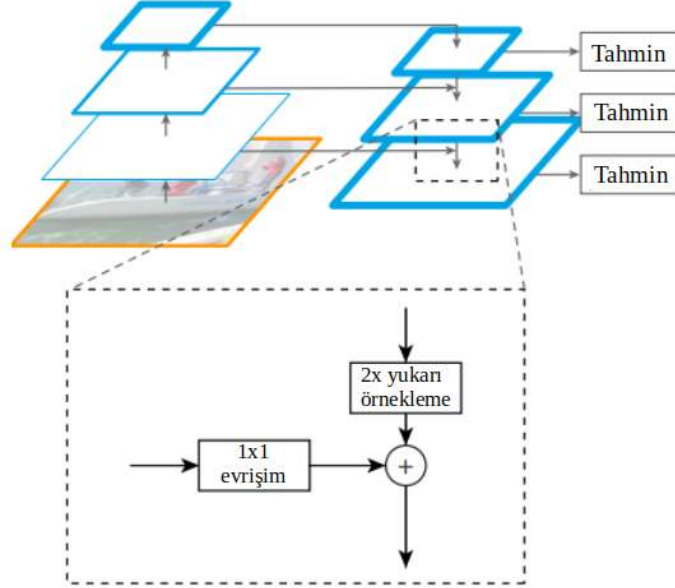
Aşağıdan yukarıya yol, omurga ağa beslenen girdi görüntünün evrişimsel katmanlarda ileri beslendiği aşamada farklı ölçeklerde öznitelik haritalarından oluşan bir öznitelik hiyerarşisini hesaplamakla sorumludur. Farklı ölçeklerde öznitelik haritaları oluşturulurken ölçekleme adımı olarak 2 kullanılır. Çoğu zaman aynı boyutta çıktı haritaları birçok katman tarafından üretilmektedir. Ancak, öznitelik piramit ağı kapsamında her aşama için bir adet piramit seviyesi tanımlanır. Piramit oluşturmak için geliştirilen referans öznitelik haritaları, her aşama için son katmanın çıktısı olacak şekilde seçilir. Bu seçimin nedeni, her aşamada en derin katmanın en güçlü özniteliklere sahip olmasıdır.

Yukarıdan aşağıya yol ve yanal bağlantılar

Yukarıdan aşağıya yol, uzamsal olarak yüzeysel olan ancak anlamsal olarak daha güçlü öznitelik haritalarını daha yüksek piramit seviyelerinden yukarı örnekleme yöntemi ile örnekleyerek daha yüksek çözünürlüklü öznitelikleri gerçekler. Daha sonra elde edilen öznitelikler aşağıdan yukarı yolda üretilen öznitelikler ile yanal bağlantılar kullanılarak zenginleştirilir. Her bir yanal bağlantı, aynı uzamsal boyuta ait aşağıdan yukarıya yoldan ve yukarıdan aşağıya yoldan alınan öznitelik haritalarını birleştirir. Aşağıdan yukarı yola ait öznitelik haritası daha alt seviye anlam bilgisini içerirken bu haritalara ait aktivasyonlar daha doğru bir şekilde konumlandırılır. Bunun nedeni, aşağıdan yukarı yolda öznitelik haritaları oluşturulurken birkaç sefer alt örnekleme uygulanmış olmasıdır.

Şekil 2.19'da yukarıdan aşağı öznitelik haritalarının yapısını anlatan bir blok sunulmuştur. Aşağıdan yukarıya yolda elde edilen düşük çözünürlüklü ancak anlamsal olarak daha çok bilgiye sahip öznitelik haritalarının çözünürlüğü 2 faktör ile yukarı örneklenir. Yukarı örneklenen öznitelik haritası daha sonra ilgili aşağıdan yukarı yolda elde edilen harita ile birleştirilir. Bu aşamada aşağıdan yukarı harita 1x1 evrişimsel katmandan geçirilir. Bunun nedeni kanal sayısını düşürmektir. Bu işlemden sonra yukarı örneklenen

Öznitelik haritası ile evrişimsel katmandan geçirilen harita element bazında toplanır. Bu işlem en iyi çözünürlük elde edilene kadar tekrarlanır.



Şekil 2.19. Aşağıdan yukarı ve yukarıdan aşağı yollarda elde edilen öznitelik haritalarının yanıl bağlantılar kullanılarak eklenmesine ait bir blok [59]

3. LİTERATÜRDE SUNULAN FARKLI X-RAY VERİ KÜMELERİ VE DENENEN YÖNTEMLER

X-ray görüntülerinde tehdit tespiti görevi için farklı tipte zararlı nesnelere bulunduğu ve farklı X-ray tarayıcılardan alınan çeşitli veri kümeleri literatüre kazandırılmıştır. Bu veri kümelerinden bazıları başka araştırmalar için ulaşılabilir olsa da kimi veri kümeleri gizli tutulmuştur. Tez çalışmasının bu bölümünde literatürde bulunan veri kümelerinden bir kısmı tanıtılmıştır. Tez çalışmasında kullanılan iki farklı veri kümesi de bu kısımda açıklanmıştır. Ayrıca, farklı X-ray veri kümeleri üzerinde uygulanan nesne tespiti sonuçlarına ait karşılaştırma çizelgesi sunulmuştur.

3.1 GDxray Veri Kümesi

GDxray veri kümesi [22], Şili Papa Katolik Üniversitesi'nin Bilgisayar Bilimleri bölümüne ait Yapay Zeka Grubu tarafından toplanmış ve akademik çalışmalar için herkesin kullanımına sunulmuştur. GDxray veri kümesi bilindiği kadarı ile tehdit içeren X-ray görüntülerinin bilgisayarlı görü alanında kullanılması için toplanan geniş çaplı ilk veri kümesidir. GDxray, toplamda 5 ana kategoriden oluşur ve 19,407 adet görüntü içermektedir. Bu kategoriler metal döküm, lehim, bagaj, doğal nesnelere ve ayarlama görüntüleridir. GDxray veri kümesindeki görüntülerin kategorilere göre dağılımı aşağıdaki çizelgede verilmiştir.

Grup	Görüntü sayısı
Metal döküm	2727
Lehim	88
Bagaj	8150
Doğal nesnelere	8290
Ayarlama	152

Çizelge 3.1. GDxray veri kümesinde kategorilere göre görüntü sayısı dağılımı [22]

Genel olarak literatürde sunulan X-ray görüntülerinde tehdit tespiti çalışmaları kapsamında bu kategorilerden bagaj görüntüleri kullanılmıştır. Bagaj görüntüleri de kendi içinde silah, bıçak, jilet ve yıldız olmak üzere dört gruba ayrılmıştır.



Şekil 3.1. GDXray bagaj görüntülerine ait örnekler [22]

Şekil 3.1’de GDXray veri kümesine ait farklı bagaj görüntüleri örnek olarak verilmiştir. Bu veri kümesinin bir dezavantajı alınan görüntülerin yeterince karmaşık olmamasıdır. Bu nedenle gerçekçi bir senaryo için uygun olmayabilir [24].

3.2 Multi-view (Çoklu Görünüm) X-Ray Veri Kümesi

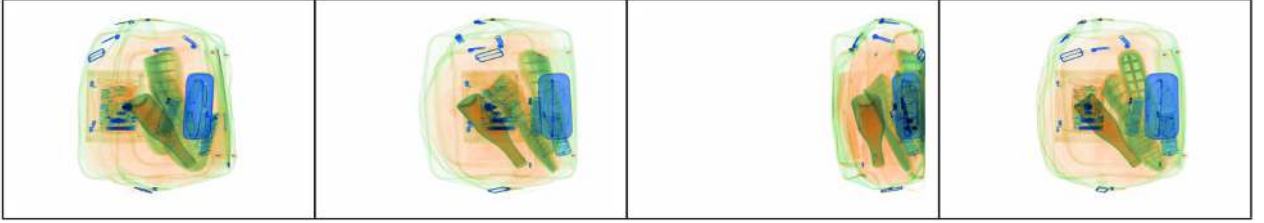
Multi-view X-ray veri kümesi [8], çoklu görünüme sahip X-ray veri kümelerinin eksikliğinden dolayı orijinal çalışma kapsamında çift kanallı X-ray tarayıcılar kullanılarak toplanmıştır. Her tarama kaydı dört farklı bakıştan oluşmuştur. Dört bakıştan üçü, tarayıcı bandın alt kısmından biri ise kenardan elde edilmiştir. Veri kümesinde dört farklı sınıf bulunmaktadır. Bunlar cam şişe, TIP silah, gerçek silah ve negatif görüntülerdir. TIP silah sınıfı, önceden taranan bir silahın bagaj görüntüsü üzerine TIP (threat image projection) yöntemi ile rastgele iz düşümleri alınarak oluşturulan sentetik görüntülerdir [60]. Çizelge 3.2’de multi-view X-ray veri kümesindeki farklı sınıflara ait görüntü sayıları ve-

Tehdit Tipi	Görüntü sayısı
Cam şişe	607
TIP silah	2160
Gerçek silah	464
Negatif	950

Çizelge 3.2. MV-XRAY veri kümesinde kategorilere göre görüntü sayısı dağılımı [8]

rilmiştir. Silah görüntüleri içerisinde sentetik TIP görüntüleri eğitimde kullanılırken sadece gerçek silahlar testte kullanılmıştır.

Şekil 3.2’de multi-view X-ray veri kümesinden alınan ve cam şişe içeren bir bagaj görüntüsü sunulmuştur. Şekilde, soldan başlayacak şekilde ilk iki ve sonuncu kutucuktaki bagaj görüntüleri tarayıcı bandın alt kısmından alınan üç görüntü, üçüncü kutucuktaki ise kenar kısımdan alınan görüntüdür. Multi-view X-ray veri kümesi sadece [8]’de kullanılmış olup diğer çalışmalar için kullanıma açık değildir.



Şekil 3.2. MV-Xray veri kümesine ait bir örnek [8]

3.3 NS-100-L X-Ray Veri Kümesi

NS-100-L X-ray veri kümesi [61], NS-100-L isimli X-ray cihazı kullanarak toplandı için tez çalışmasında bu veri kümesi NS-100-L olarak isimlendirilmiştir. NS-100-L, hızlı X-ray görüntüsü alabilmek için düzenlenen taşınabilir bir sistemdir. Tek bir X-ray görüntüsü bu sistem ile hızlı bir şekilde alınabilmektedir.

NS-100-L kullanılarak toplanılan veri kümesinde makas, bıçak, şişe ve tehdit unsuru olmayan başka nesnelere kullanılmıştır. Toplamda 1104 adet ham X-ray görüntüsü toplanmıştır. Görüntüler toplanırken nesnelere farklı kombinasyonlar elde edebilmek için kırılmış, döndürülmüş, birbirlerini engelleyecek şekilde konumlandırılmış ve açıları değiştirilmiştir. Alınan görüntüler iki farklı çeşide bölünmüştür. Bunlardan biri, tek veya çok nesne içeren basit görüntüler, diğeri ise hem karmaşık içeren hem de tehdit ve tehdit içermeyen nesnelere bir arada bulunduğu görüntülerdir.

Şekil 3.3’te NS-100-L X-ray veri kümesinden örnek bir görüntü sunulmuştur. Şekilde soldan başlamak üzere sırası ile ilk ve ikinci görüntüler basit görüntülerdir. İlk görüntüde bir makas tek başına taranmıştır. İkinci görüntüde ise makas ve bıçak nesnelere

birbirini tamamen engellemeyecek şekilde konulmuş ve başka hiçbir obje taranmamıştır. Son görüntü ise karmaşık bir örnektir. Bu örnekte farklı tehditler bir arada olduğu gibi bunlarla birlikte tehdit içermeyen nesnelere de rastgele bir şekilde konulup taratılmıştır. NS-100-L X-ray veri kümesi, sadece [61]'de belirtilen çalışma kapsamında kullanılmıştır.



Şekil 3.3. NS-100-L veri kümesine ait bir örnek [61]

3.4 Rapiscan X-Ray Veri Kümesi

Rapiscan X-ray veri kümesi [62], Rapiscan 620DV X-ray görüntüleme sistemi kullanılarak toplanmıştır. Rapiscan 620DV X-ray görüntüleme sistemi, havacılık ve yüksek güvenlik içermesi gereken uygulamalar için tasarlanmıştır. Bu nedenle tez çalışması kapsamında bu veri kümesi Rapiscan ismi ile tanıtılmıştır.

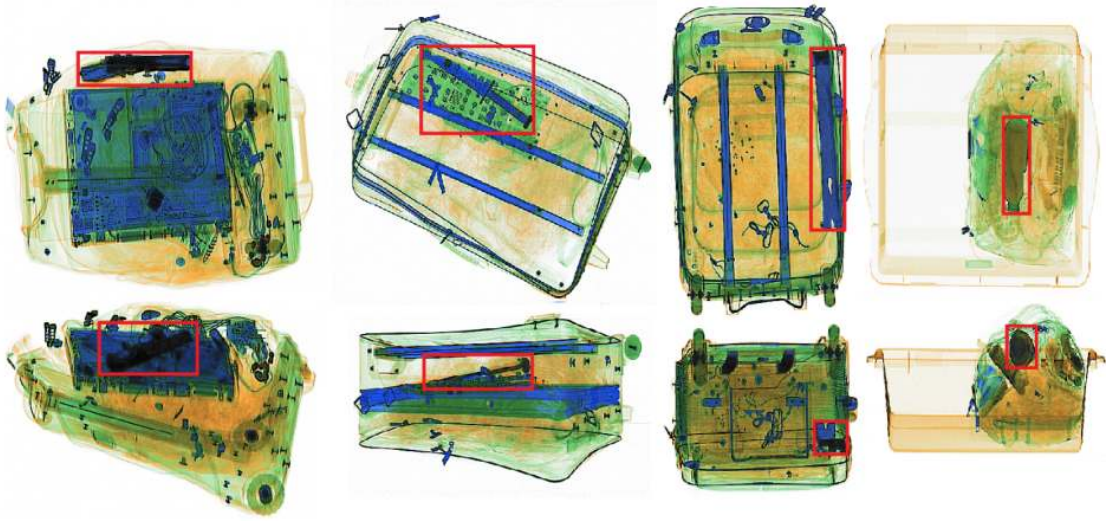
Rapiscan veri kümesi tabanca, kesici aletler, kör bıçaklar, LAG yani sıvı (liquid), sprey (aerosol) ve jel (gel) içeren ürünleri kapsamaktadır. Her tehdit türü için çeşitliliği sağlamak amacıyla veriler toplanırken her sınıftan farklı nesnelere seçilmiştir. Veri toplama işlemi yapılırken gerçek dünyadaki senaryoya uygun olması açısından tekerlekli çanta, laptop çantası gibi farklı tipte çantalar kullanılmıştır. Bu çantaların içi hem tehdit içermeyen nesnelere hem de tehditlere doldurulmuştur. Tehditler yerleştirilirken farklı konumlar ve açılar kullanılmıştır. Ayrıca, genel olarak tehdit nesnelere yaygın olmasından dolayı çoğu çanta içerisinde sadece bir adet tehdit konulmuştur.

Veri toplanırken zaman almaması açısından tek bir çanta içerisine farklı tehdit öğeleri yerleştirilmiş ve bu çanta birçok taramada kullanılmıştır. Taramalar yapılırken tehdit olmayan nesnelere de kargaşa yaratılması için küçük değişikliklerle çantaya yerleştirilmiştir. Ayrıca, her çanta çeşitli rotasyonlarda ve pozlarda da taranmıştır.

Çizelge 3.3'te Rapiscan X-ray veri kümesinde bulunan farklı tehdit sınıflarına ait görüntü dağılımı verilmiştir. Şekil 3.4'te ise farklı tipte tehditler içeren Rapiscan X-ray veri kümesine ait görüntü örnekleri verilmiştir. Daha önce de bahsedildiği ve resimlerde de görüldüğü gibi genellikle görüntülerde bir adet tehdit kullanılmıştır. Şekilde her bir kolda bir tehdit sınıfına ait iki adet örnek sunulmuştur. En soldan başlayacak şekilde sırası ile tabanca, kesici aletler, kör bıçaklar ve sıvı içeren nesnelere ait görüntüler verilmiştir.

Tehdit Tipi	Görüntü sayısı
Tabanca	3480
Kesici aletler	3484
Kör bıçak	3366
LAG	3456

Çizelge 3.3. Rapiscan veri kümesinde kategorilere göre görüntü sayısı dağılımı [62]



Şekil 3.4. Rapiscan veri kümesine ait örnek görüntüler [62]

3.5 DBF3 ve DBF6 X-ray Veri Kümeleri

DBF3 ve DBF6 X-ray veri kümeleri [24], Durham bagaj yama/tam körüntü kümesinin (Durham Baggage Patch/Full Image Dataset) alt kombinasyonlarıdır. Durham bagaj veri kümesi, İngiltere'de bulunan Durham Üniversitesi kapsamında X-ray görüntülerinden sınıflandırma, nesne tespiti, anomali tespiti vb. gibi çalışmalarda kullanılmak üzere

toplanmıştır. Çift kanallı X-ray tarayıcılardan toplanan görüntülerin farklı sınıflara göre dağılımı aşağıdaki çizelgede verilmiştir.

Kategori	Görüntü sayısı
Kamera	494
Seramik bıçak	1596
Bıçak	3208
Silah	3192
Silah parçası	1203
Laptop	2390
Zararsız nesne	3366

Çizelge 3.4. Durham bagaj veri kümesinde kategorilere göre görüntü sayısı dağılımı [14]

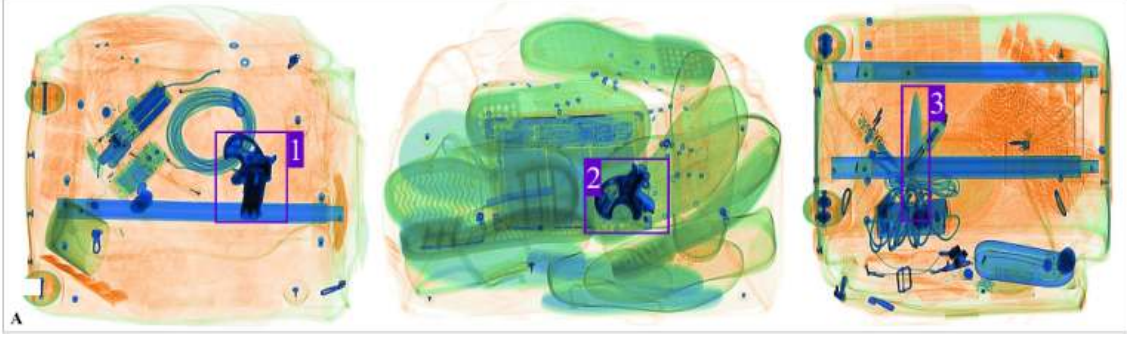
Aşağıda, Durham bagaj veri kümesinden farklı amaçlar için oluşturulan DBF3 ve DBF6 veri kümeleri açıklanmıştır. DBF3 ve DBF6 veri kümeleri nesne tespiti çalışmalarında kullanılmak için oluşturulmuştur. Bu verilerin çalışma dışında paylaşılması yasaktır.

DBF3 Veri Kümesi

DBF3 (Durham Dataset Full Three-class) veri kümesi [14], Durham bagaj veri kümesinin üç farklı kategorisinin birleştirilmesi ile oluşturulmuştur. Bu kategoriler metal parça içeren tehdit sınıflarıdır. DBF3 veri kümesinde 3192 adet silah, 1203 adet silah parçası ve 3208 adet bıçak bulunmaktadır. Şekil 3.5'te DBF3 veri kümesinden örnek bir görüntü verilmiştir. Şekilde, 1 numaralı nesne silah, 2 numaralı nesne silaha ait bir parça ve 3 numaralı nesne ise bıçak görüntüsüdür.

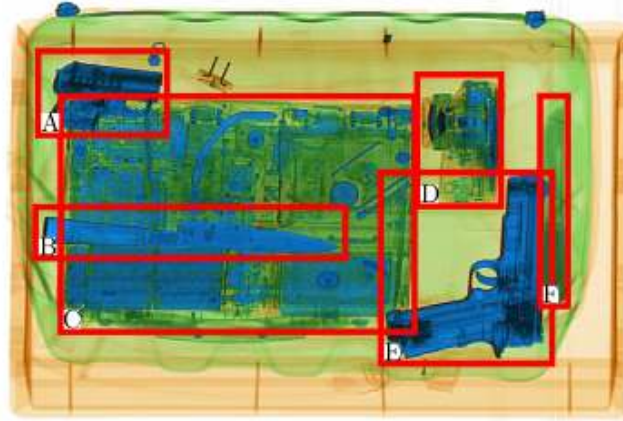
DBF6 Veri Kümesi

DBF6 (Durham Dataset Full Six-class) veri kümesinde [63] ise Durham bagaj veri kümesine ait altı farklı sınıf kullanılmıştır. Bu sınıflar kamera, laptop, silah, silah parçası, bıçak ve seramik bıçaktır. DBF6 veri kümesinin oluşturulmasındaki temel amaç, çok sınıflı ve nesne tespiti modellerini zorlayıcı bir küme oluşturmaktır.



Şekil 3.5. DBF3 veri kümesine ait örnek bir görüntü [63]

Şekil 3.6'da tüm farklı kategorilere ait bir nesne içeren örnek görüntü sunulmuştur. Veri kümesinin gizli olması nedeni ile çeşitli örneklere ulaşılamamıştır. Şekilde verilen A bölgesinde silahın bir bileşeni, B bölgesinde seramik bıçak, C bölgesinde laptop, D bölgesinde kamera, E bölgesinde silah ve F bölgesinde bıçak olmak üzere toplam altı farklı sınıfa ait obje örnekleri bulunmaktadır.



Şekil 3.6. DBF6 veri kümesine ait örnek bir görüntü [63]

3.6 SIXray Veri Kümesi

SIXray veri kümesi [67], Çin Bilimler Akademisi bünyesinde bulunan Örüntü Tanıma ve Akıllı Sistemler Laboratuvarı tarafından toplanmıştır. Tez çalışması kapsamında kullanıldığı için bu kısımda detaylı bir şekilde anlatılmıştır. SIXray veri kümesi 1.059.231 adet X-ray görüntüsü içermektedir. Görüntüler alınırken farklı metro istasyonları kullanılmıştır. Elde edilen X-ray görüntüleri içerisinde hem taşınması yasak olan eşyalar hem de

normal eşyalar bulunmaktadır. Yasak olan eşyalar tehdit unsuru olabilecek silah, bıçak, İngiliz anahtarı, pense, makas ve çekiç olarak altı kategoride sınırlandırılmıştır. Ancak çekiç sınıfına ait veri sayısı sadece 60 adet olduğu için çalışmalarda bu sınıfa ait verilerin kullanılmamasına karar verilmiştir. SIXray veri kümesinde bahsedilen tehdit unsurları gerçekçi bir senaryo dağılımına sahiptir. Yani, negatif (tehdit unsuru içermeyen) görüntülerin sayısı pozitif (tehdit unsuru içeren) görüntülere kıyasla çok daha fazladır. Çizelge 3.5'te SIXray veri kümesindeki görüntülerin silah, bıçak, makas, İngiliz anahtarı, pense ve tehdit içermeyen sınıflara göre dağılımı sunulmuştur.

Çizelge 3.5. SIXray veri kümesinin farklı sınıflara göre dağılımı [67]

Silah	Bıçak	Makas	İngiliz Anahtarı	Pense	Negatif
3131	1943	2199	3961	983	1.050.302

SIXray veri kümesindeki tüm görüntüler çift kanallı X-ray cihazları tarafından taranmıştır. Çift kanallı X-ray cihazlarında yüksek ve düşük enerjili olmak üzere iki adet kanal bulunmaktadır. Bu kanallardan elde edilen gri ölçekli görüntülerdeki objelere belli piksel aralıkları için farklı değerler atanarak renklendirme işlemleri uygulanmıştır. Çift kanallı X-ray görüntüleme tekniğine ait daha detaylı bilgi sonraki başlıkta verilmiştir.

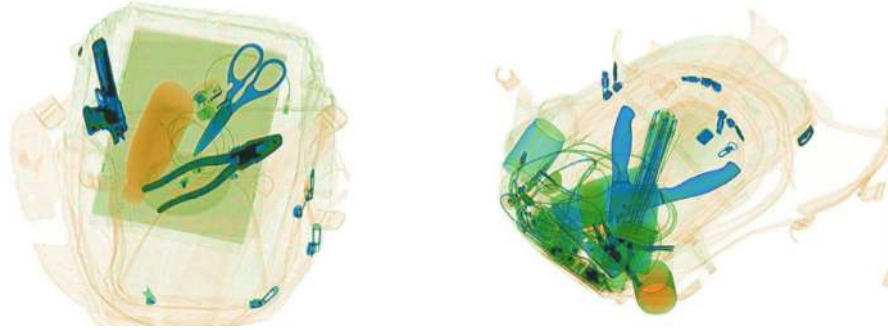
SIXray veri kümesi görüntü tanımayı zorlaştıracak birçok özellik içermektedir. Bunlardan ilki görüntülerin kişisel bavul, çanta gibi eşyalardan toplanmasıdır. Bu nedenle elde edilen görüntülerde bulunan nesnelere karmaşık bir halde konumlandırılmıştır. Bu eşyalar X-ray cihazında taratıldığında rastgele yerleştirilen ve diğer eşyalar tarafından engellenen nesnelere de X-ray cihazının ışınları tarafından görüntülenebileceği için örtüşen bir veri kümesi elde edilmiştir. Veri kümesinin görüntü tanımayı zorlaştıran ikinci özelliği ise tehditlerin farklı boyut, bakış açısı ve değişik tarzlarda alınmış olmasıdır. Bir başka özellik ise bazı görüntülerin çokça karmaşa içermesidir. Bu görüntülerde bulunan nesnelere gürültüden dolayı sınıf ataması yapmak zor olabilmektedir. Son olarak pozitif görüntüler veri kümesinin küçük bir kısmını oluşturduğu için sınıflandırma görevlerinde negatif sınıfa doğru bir sapma gözlemlenecektir. Ancak nesne tanıma kapsamında kesin referansta sınırlandırıcı kutucuk ile belirlenen nesnelere dışında kalan alanlar arkaplan olarak algılandığı için negatif görüntüler kullanılmamıştır.

SIXray veri kümesi ile yapılan orijinal çalışmada [67] nesne sınıflandırma işlemi yapıldığı için yazarlar sadece yaklaşık 1400 adet pozitif görüntüye ait sınırlandırıcı kutucuk içeren etiket verisi paylaşmıştır. Bu nedenle çalışma kapsamında etiketleme işlemi de ayrıca yapılmıştır. Etiketleme işlemi yapılırken açık kaynaklı kodlar kullanılmıştır. SIXray veri kümesinde bulunan yaklaşık 5500 görüntünün etiketi oluşturulmuştur. Etiketler oluşturulurken Pascal VOC [68] veri kümesinin formatı baz alınmıştır. Bu formata göre bir görüntünün boyutları, içerdiği objeler ve objelere ait sınırlandırıcı kutucuk bilgileri *xml* formatı ile oluşturulmaktadır. Çizelge 3.6'da her bir sınıf için etiketleme sonucu elde edilen kesin referans kutucuklarının sayısı sunulmuştur. Buna göre, en çok silah verisi bulunurken makasa ait kesin referansların sayısı 496 adettir. Bu nedenle seçilen modelin makas sınıfını öğrenmesi diğer sınıflara göre daha zor olacaktır.

Silah	Bıçak	Makas	İngiliz Anahtarı	Pense
4636	2482	496	1516	2248

Çizelge 3.6. Etiketleme sonucunda tehditlere ait elde edilen kesin referans sayıları

Şekil 3.7'de SIXray veri kümesine ait örnek görüntüler bulunmaktadır. Sunulan örnek görüntülerde farklı tehditler bir arada bulunmaktadır. Şekil 3.7'ye detaylı bakıldığında sol görüntüde bulunan tehditlerin birbirini engellemediği gözlemlenirken sağ görüntüde tehditlerin birbirini engellediği ve görüntüde karmaşa olduğu görülmektedir.

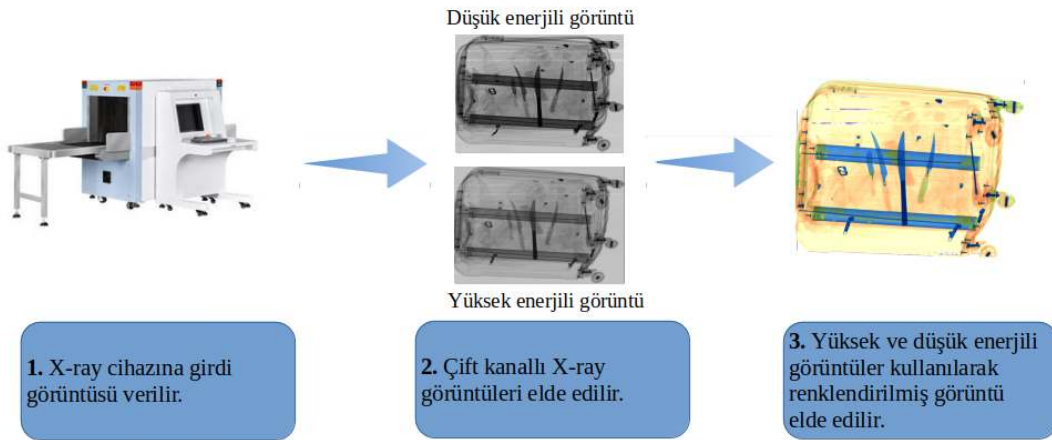


Şekil 3.7. SIXray veri kümesinden örnek görüntüler: sade görüntü (sol), karmaşık görüntü (sağ) [67]

3.7 Tez Kapsamında Toplanan HUMS X-ray Veri Kümesi

Tez çalışması kapsamında açık kaynaklı X-ray verileri dışında MS Spektral Savunma Sanayi A.Ş. bünyesinde bulunan çift kanallı X-ray cihazı kullanılarak tehdit verileri toplanmıştır. Veri toplanırken farklı tipteki bıçaklar kullanılmıştır. Olası bir sınıflandırma çalışmasında toplanılan veri kümesinin kullanımı için ayrıca tehdit içermeyen negatif veriler de toplanmıştır. Veriler toplanırken SIXray'de olduğu gibi hem karmaşa olmayan hem de karmaşık ve birbirini engelleyen tehditlerin bulunduğu farklı görüntüler alınmıştır. Toplanılan veri kümesine HUMS ismi verilmiştir.

Çift kanallı X-ray cihazlarında hem yüksek hem de düşük frekansta X-ray ışınları gönderilerek görüntüde bulunan maddelerin hem yoğunluğu hem de atom numarası hakkında bilgi edinilmesini sağlar. Böylece taranan maddelerin yapısı hakkında daha ayrıntılı bilgi edinilir. Tarama sonucunda düşük ve yüksek frekanslı ışınlarla ait iki adet gri ölçekli görüntü elde edilir. Elde edilen gri ölçekli görüntüler üzerinde renklendirme teknikleri uygulanarak üç boyutlu yapay RGB görüntüler hazırlanır. Şekil 3.8'de çift kanallı X-ray görüntüleme tekniğine ait bir akış verilmiştir. Veri toplama aşamasından sonra tehdit içeren 100 ve tehdit içermeyen 100 ayrı veri toplanmıştır. Ancak daha önce SIXray veri kümesinin anlatımında bahsedildiği gibi nesne tespiti işlemlerinde ayrıca negatif görüntülere ihtiyaç duyulmadığı için tehdit içermeyen görüntüler kullanılmamıştır.



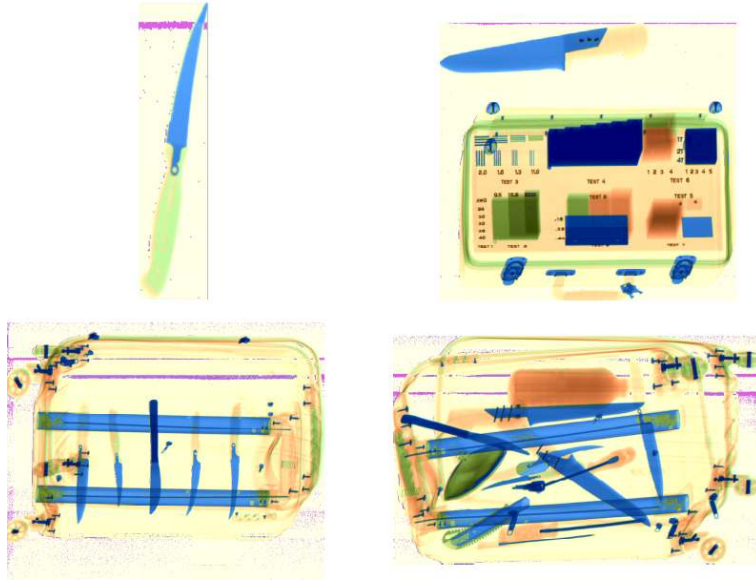
Şekil 3.8. Çift kanallı X-ray görüntüleme adımları

Çizelge 3.7'de HUMS veri kümesinde bulunan görüntülerin dağılımı sunulmuştur. Çizel-

gedeki sayılara yapay RGB görüntüler, düşük enerjili görüntüler ve yüksek enerjili görüntülerin tamamı dahil edilmiştir. Veri arttırımı ise sadece yapay RGB görüntüler üzerinde uygulanmıştır. Kesin referans sayıları ise arttırılmış görüntüler ile düşük ve yüksek enerjili görüntülerde bulunan nesnelere hepsi dahil edilerek alınmıştır. Şekil 3.9'da HUMS verilerine ait örnek görüntüler sunulmuştur. Veriler toplanırken şekilde de görüldüğü gibi farklı tip bıçaklar kullanılmıştır. Bu bıçaklara ait sol üst ve sağ üst görüntülere benzer şekilde hem tekli hem de sol alt ve sağ alt görüntülerdeki gibi çoklu taramalar yapılmıştır. Özellikle çoklu taramalar yapılırken sol ve sağ alttaki görüntülerde açıklandığı gibi hem bıçakların birbirini etkilemediği temiz veriler, hem de bıçakların üst üste konularak birbirlerini engellediği karmaşık örnekler alınmıştır. SIXray verilerinde olduğu gibi toplanılan veriler için de ayrıca PASCAL VOC [68] formatında etiketleme işlemleri yapılmıştır.

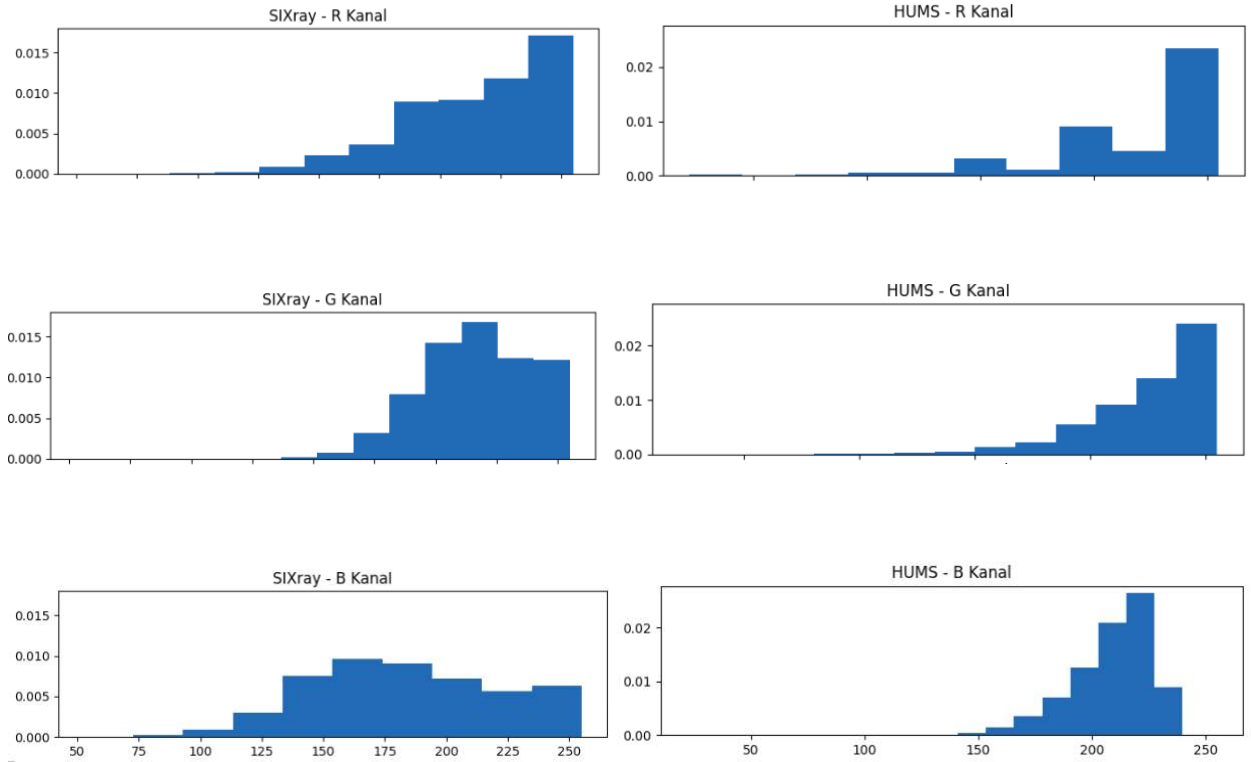
Tüm Tehdit Verileri	Ayıklama Sonucu Kalanlar	Tehdit İçermeyenler	Veri Arttırımı Sonucu	Kesin Referans
300	243	300	312	987

Çizelge 3.7. HUMS veri kümesine ait görüntülerin dağılımı ve kesin referans sayıları



Şekil 3.9. HUMS veri kümesinden örnek görüntüler: Tekli bıçak (sol üst), bıçak ve bavul (sağ üst), düzgün yerleştirilmiş bıçaklar (sol alt), karmaşık yerleştirilmiş bıçaklar (sağ alt)

Şekil 3.10'da SIXray ve HUMS veri kümelerinde bulunan bıçak tehditlerinin kırmızı (red - R), yeşil (green - G) ve mavi (blue - B) kanallar için histogramları karşılaştırılmıştır. SIXray ve HUMS veri kümelerindeki ortak tehdit sınıfı bıçak olduğu için histogram grafikleri bu sınıf için alınmıştır. SIXray ve HUMS veri kümelerinin ayrı kanallar için yoğunluk dağılımlarına bakıldığında iki veri kümesi de çift kanallı X-ray cihazı kullanılarak toplanmasına rağmen farklı yoğunluk dağılımlarına sahip oldukları görülmektedir. Farklılığın sebepleri, kullanılan X-ray cihazlarının farklı olması, kullanılan bıçak tiplerinin farklı olması, bıçakların yanında farklı tip nesnelere birlikte taranması gibi birçok kaynağa dayanmaktadır. İki veri kümesindeki yoğunluk dağılımlarının farklılığı, veri kümeleri arasındaki alan farkının (domain difference) açılmasına neden olmaktadır. Oluşan alan farkı, bir veri kümesinden elde edilen eğitim sonuçlarının alan farkı olan veri kümesi üzerinde test edildiğine performans düşüşüne neden olmaktadır.



Şekil 3.10. SIXray ve HUMS veri kümelerindeki bıçakların farklı kanallar için histogram grafikleri

3.8 Literatürde X-ray Veri Kümeleri Üzerinde Uygulanan Yöntemler ve Alınan Sonuçlar

Bu bölümde, literatürde bulunan ve bir kısmı yukarıdaki bölümlerde tanıtılan veri kümeleri üzerinde denenen yöntemlere ait karşılaştırma çizelgeleri sunulmuştur. Çizelgelerde Faster RCNN yöntemi F. RCNN olarak kısaltılmıştır.

GDXRAY [22]	Silah	Bıçak	Jilet	Yıldız	-	-	mAP
YOLOv2 [15]	0.780	0.054	0.607	0.880	-	-	0.580
YOLOv3 [15]	0.963	0.762	0.869	0.937	-	-	0.883
SSD300 [15]	0.901	0.011	0.696	0.904	-	-	0.630
RetinaNet [15]	0.981	0.005	0.819	0.956	-	-	0.690
TinyYolo [64]	-	-	-	-	-	-	0.89
YOLOv2 [64]	-	-	-	-	-	-	0.97
F. RCNN [64]	-	-	-	-	-	-	0.98
YOLOv2 [65]	0.906	0.902	0.900	0.882	-	-	0.898
SSD300 [65]	0.905	0.891	0.904	0.887	-	-	0.897
F. RCNN [65]	0.915	0.912	0.911	0.913	-	-	0.913
MV-XRAY [8]	Silah	Cam Şişe	-	-	-	-	mAP
F. RCNN [8]	0.85	0.96	-	-	-	-	0.91
MX-RCNN [8]	0.92	0.98	-	-	-	-	0.95
Liu et al. [9]	Laptop	Cep Tlf.	Şarj Aleti	Şemsiye	Şişe	Anahtar	mAP
Retina Net [9]	-	-	-	-	-	-	0.70
F. RCNN [9]	0.86	0.84	0.69	0.86	0.67	0.53	0.74
Ek Yöntem [9]	0.87	0.86	0.71	0.88	0.69	0.58	0.77
Xu et al. [66]	Cihaz	Bıçak	Basınç Tankı	Teneke	Makas	Metal	mAP
YOLOv3 [66]	0.94	0.66	0.91	0.98	0.95	0.42	0.75
NS-100-L [61]	Şişe	Bıçak	Makas	-	-	-	mAP
YOLOv2 [61]	0.88	0.78	0.85	-	-	-	0.84

Çizelge 3.8. Literatürdeki X-ray veri kümeleri ile alınan sonuçların karşılaştırması - 1

Rapiscan [62]	Kesiciler	Kör Bıçak	Tabanca	LAG	-	-	mAP
SSD [62]	0.408	0.918	0.757	0.907	-	-	0.7523
F. RCNN (1) [62]	0.766	0.976	0.944	0.973	-	-	0.9166
F. RCNN (2) [62]	0.786	0.980	0.947	0.976	-	-	0.9244
F. RCNN (3) [62]	0.818	0.983	0.962	0.985	-	-	0.9410
Dbf3 [14]	Bıçak	Silah	Silah Parçası	-	-	-	mAP
F. RCNN [14]	0.85	0.91	0.88	-	-	-	0.88
Mask RCNN [14]	0.80	0.89	0.86	-	-	-	0.85
RetinaNet [14]	0.73	0.89	0.86	-	-	-	0.83
SIXRAY [67]	Bıçak	Silah	Makas	İngiliz Anahtarı	Pense	-	mAP
F. RCNN [14]	0.81	0.91	-	-	-	-	0.86
Mask RCNN [14]	0.79	0.89	-	-	-	-	0.84
RetinaNet [14]	0.79	0.92	-	-	-	-	0.86
YOLOv3 *	0.722	0.949	0.661	0.830	0.776	-	0.787
F. RCNN (1) *	0.824	0.955	0.682	0.832	0.877	-	0.834
F. RCNN (2) *	0.823	0.907	0.878	0.905	0.903	-	0.883
SSD *	0.709	0.918	0.651	0.686	0.791	-	0.751
Dbf6 [63]	Bıçak	Silah	Silah Parçası	Kamera	Laptop	Seramik Bıçak	mAP
SWCNN [63]	0.392	0.831	0.848	0.803	0.881	0.902	0.776
RCNN [63]	0.304	0.876	0.832	0.888	0.954	0.819	0.779
F. RCNN [63]	0.732	0.931	0.911	0.857	0.904	0.907	0.874
R-FCN [63]	0.556	0.942	0.925	0.887	0.906	0.920	0.856
YOLOv2 [63]	0.728	0.943	0.933	0.896	0.894	0.913	0.885
HUMS X-ray	Bıçak	-	-	-	-	-	mAP
F. RCNN *	0.916	-	-	-	-	-	0.916

Çizelge 3.9. Literatürdeki X-ray veri kümeleri ile alınan sonuçların karşılaştırması - 2

Çizelge 3.8'de, GDXray, Multi-view X-ray (MV-XRAY), ismi verilmeyen iki adet veri kümesi (Liu et al., Xu et al. olarak sunulmuştur) ve NS-100-L veri kümeleri üzerinde yapılan çalışmaların sonuçları sunulmuştur. Çizelgeye göre, GDXray veri kümesi üzerinde yapılan üç farklı çalışmadan ilkinde [15] YOLOv2, YOLOv3, SSD300 ve RetinaNet denenmiştir. Bu çalışmaların sonuçları karşılaştırıldığında silah ve yıldız sınıfında en iyi performansı RetinaNet verirken bıçak, jilet ve modelin genel ortalama sonucunda en iyi performans ise YOLOv3 ile gözlemlenmiştir. GDXray veri kümesinde yapılan diğer çalışmada [64] ise TinyYolo, YOLOv2 ve Faster RCNN modelleri kullanılmıştır. Ancak sınıf bazlı sonuçlar yerine modellerin ortalama kesinlik değerleri (mAP) sunulmuştur. Sunulan sonuçlara göre Faster RCNN diğer modellere kıyasla daha başarılı bir sonuç vermiştir. Sunulan son GDXray veri kümesinde nesne tespiti çalışmasında [65] ise YOLOv2, SSD300 ve Faster RCNN yöntemleri kullanılmıştır. Bu çalışmanın karşılaştırmasında ise en iyi sonuç hem tüm sınıflar hem de genel ortalama için Faster RCNN ile elde edilmiştir.

MV-XRAY veri kümesinde ise hem Faster RCNN hem de bu veri kümesinde üç boyutlu nesne tespitini gerçekleştirebilmek için geliştirilen Multi-view X-ray RCNN (MX-RCNN) modelleri denenmiştir [8]. MX-RCNN kısaca omurga ağlardan elde edilen iki boyutlu öz-niteliklerinin farklı izdüşümleri birleştirilerek üç boyutlu hale getirilmesi ile oluşturulmuştur. Elde edilen üç boyutlu öz-nitelikler, bölge öneri ağının girdisi olur ve böylece tespite ait konum bilgisi üç boyutlu çıkarılır. Sıradan Faster RCNN ile karşılaştırıldığında MX-RCNN, çoklu görünüme sahip MV-XRAY veri kümesinde silah ve cam şişe için daha başarılı bir tespit sonucu vermiştir. Ortalama kesinlik değeri karşılaştırıldığında ise Faster RCNN'e göre %4 daha başarılı olmuştur.

Bir başka çalışmada [9] ise kullanılan X-ray veri kümesinin ismi veya hangi taracıyından alındığına dair bir bilgi verilmediği için çizelgede *Liu et al.* olarak sunulan veriler üzerinde uygulanan yöntemler belirtilmiştir. Bu çalışmada RetinaNet, Faster RCNN ve Faster RCNN ile kullanılan ek bir algoritma olan ön plan/arka plan bölütleme yöntemleri karşılaştırılmıştır. Ön plan / arka plan bölütleme yöntemi kısaca X-ray görüntülerinde bulunan farklı tip nesnelere yoğunluğu baz alınarak belli piksel değerlerinin eşik değeri seçilmesi ile ön plan nesnelere ile arka planın birbirinden ayrılması işlemidir. Sonuçlar

karşılaştırılırken RetinaNet için sadece ortalama kesinlik değeri sunulmuş olup Faster RCNN ve ön plan / arka plan bölütleme sonucu tüm sınıflar için ayrı ayrı sunulmuştur. Karşılaştırmalar sonucunda ek yöntemin tüm sınıflar ve mAP bazında en başarılı sonucu verdiği gözlemlenmiştir. Bir başka ismi belirtilmeyen X-ray veri kümesi [66], karşılaştırma çizelgesinde *Xu et al.* olarak sunulmuştur. Bu çalışmada, çok sınıflı veri kümesi üzerinde YOLOv3 algoritması denenmiş ve sonuçlar sunulmuştur. Alınan sonuçlara göre en iyi tespit %98 ile teneke sınıfı için olurken metalik nesnelerin tespitinde ise %42 başarı görülmüş. Ortalama kesinlik değeri ise %75 olarak elde edilmiştir. NS-100-L veri kümesi ile yapılan çalışmada [61] ise şişe, bıçak ve makas tespiti YOLOv2 kullanılarak yapılmıştır. Sonuçlara göre şişe %88, bıçak %78, makas %85 başarıya ulaşırken modelin ortalama kesinlik değeri ise %84 olarak hesaplanmıştır.

Çizelge 3.9'da ise diğer X-ray veri kümelerinde yapılan çalışmaların devamı sunulmuştur. Çizelgede tez çalışması kapsamında yapılan deneylere ait sonuçlar * ile belirtilmiştir. Rapiscan veri kümesi ile yapılan bir çalışmada [62] SSD modeli Inception V2 omurga ağı ile, Faster RCNN modeli ResNet-101 (Çizelge 3.9'da Faster RCNN (1) olarak belirtilmiştir.), ResNet-152 (Çizelge 3.9'da Faster RCNN (2) olarak belirtilmiştir.) ve Inception ResNetV2 (Çizelge 3.9'da Faster RCNN (3) olarak belirtilmiştir.) omurga ağları ile denenmiştir. Alınan sonuçlara göre Faster RCNN'in Inception ResNetV2 omurga ağı ile her kategoride dolayısı ile tüm model için en iyi performansa sahip olduğu görülmüştür.

DBF3 veri kümesi ile yapılan çalışmada [14] ise Faster RCNN, Mask RCNN ve RetinaNet modelleri kullanılmış olup Faster RCNN'in silah, silah parçası ve bıçak sınıflarının hepsinde daha başarılı sonuç verdiği görülmüştür. Aynı çalışmada SIXray veri kümesine ait silah ve bıçak sınıfları da kullanılmış olup Faster RCNN, Mask RCNN ve RetinaNet bu sınıflar üzerinde de denenmiştir. SIXray'de denenilen sonuçlarda RetinaNet silah için %92 başarı ile en iyi sonucu verirken genel sonuca bakıldığında Faster RCNN daha başarılı çıkmıştır. SIXray veri kümesi üzerinde yapılan başka bir karşılaştırma ise tez çalışması kapsamında alınan sonuçlardır. Tez çalışmasında sırası ile YOLOv3, Faster RCNN ResNet-101 (Çizelge 3.9'da Faster RCNN (1) olarak belirtilmiştir.), Faster RCNN ResNet-101+FPN (Çizelge 3.9'da Faster RCNN (2) olarak belirtilmiştir.) ve SSD modelleri kullanılmış olup [14]'ten farklı olarak deneyler SIXray'de bulunan tüm sınıflar

üzerinde uygulanmıştır. Tez kapsamında alınan sonuçlar ile [14]'de sunulan sonuçlara ait karşılaştırma yorumları tezin ilerleyen bölümlerinde verilmiştir.

DBF6 veri kümesi ile yapılan çalışmada [63] SWCNN, RCNN, Faster RCNN, R-FCN ve YOLOv2 modelleri denenmiştir. Silah, silah parçası ve kamera için en iyi sonucu YOLOv2 modeli verirken bıçak sınıfı için en iyi sonuç Faster RCNN'den alınmıştır. Laptop sınıfı için en başarılı sonucu ise RCNN modeli vermiştir. R-FCN ise seramik bıçaklar üzerinde en iyi performansa sahiptir. Ortalama kesinlik değerleri karşılaştırıldığında ise en iyi sonuç %88.5 ile YOLOv2 kullanılarak alınmıştır. Son olarak çalışma kapsamında toplanılan HUMS veri kümesindeki bıçaklar için yapılan denemelerde Faster RCNN modeli kullanılmış ve %91.6 kesinlik değerine ulaşılabildiği görülmüştür. HUMS veri kümesinde yapılan farklı denemelere ait detaylı bilgiler ise tezin ilerleyen bölümlerinde sunulmuştur.

4. GERÇEK VERİLER ÜZERİNDE DENEYSEL SONUÇLAR

Çalışmanın bu kısmında, tez süresince kullanılan iki adet veri kümesi üzerinde ikinci bölümde açıklanan nesne tespiti yöntemleri uygulanmıştır. Elde edilen sonuçlar karşılaştırılmış ve ilgili veri kümelerine ait örnek tespitler sunulmuştur. İlgili veri kümeleri SIXray ve HUMS bıçak veri kümesi olup çalışmanın üçüncü bölümünde tanıtılmıştır.

4.1 Deneylede Kullanılan Ölçüm Metriği

Gerçek veriler üzerinde tehdit tespiti yapılırken öncelikle SIXray veri kümesi üzerinde eğitimler yapılmıştır. Bunun nedeni SIXray veri kümesinde daha fazla veri bulunması ve bu kümenin HUMS veri kümesine göre daha kapsamlı olmasıdır. Tehdit tespiti yapabilmek için ikinci kısımda açıklanan iki aşamalı nesne tespiti algoritmalarından Faster RCNN, tek aşamalı nesne tespiti algoritmalarından ise YOLOv3 ve SSD kullanılmıştır. SIXray veri kümesi sırası ile YOLOv3, Faster RCNN ve SSD modelleri ile eğitilip sonuçlar test edilmiştir. Elde edilen sonuçlardan en yüksek doğruluğa sahip olan algoritma ile HUMS veri kümesinde bulunan bıçaklar tespit edilmeye çalışılmıştır. Bu aşamada aktarım öğrenimi için az vuruşlu öğrenme, ince ayar gibi yöntemler kullanılmıştır. Bu adımlardan sonra toplanılan düşük ve yüksek enerjili X-ray görüntüleri ile renklendirilmiş X-ray görüntüleri çeşitli kombinasyonlar şeklinde (görüntülerin tamamı, yüksek enerjili ve renklendirilmiş görüntüler, düşük enerjili ve renklendirilmiş görüntüler vb.) kullanılarak HUMS veri kümesi Faster RCNN ile eğitilmiştir. Buradaki amaç düşük ve yüksek enerjili X-ray görüntülerinin eğitimdeki etkisini araştırmaktır. Tüm modeller eğitilirken GPU kullanmak için Google'ın sunmuş olduğu Colab uygulamasından yararlanılmıştır. Böylece Colab üzerinden Tesla T4 ve Tesla PCIE P100 GPU'ları kullanılabilmiştir.

Sonuçlar alınırken PASCAL VOC yarışmasında kullanılan hesaplama kriteri göz önünde bulundurulmuştur. Bu hesaplama yapılırken kullanılan önemli terimler ve hesaplama ait formüller bu bölümde açıklanmıştır. Doğru pozitif (TP), görüntüde bulunan bir nesnenin model tarafından doğru kategorize edilmesidir. Yanlış pozitif (FP) ise bir görüntüde nesne olmamasına rağmen modelin ilgili görüntüde nesne tespit etmesidir. Eğer bir görüntüde nesne yoksa ve model de herhangi bir tespitte bulunmamışsa bu durum doğru negatif (TN) olarak adlandırılır. Son olarak seçilen model görüntüde bir nesne olmasına

rağmen bunu tespit edemiyorsa yanlış negatif (FN) sonucu ortaya çıkar. Bu dört terim kullanılarak kesinlik (precision) ve geri çağırma (recall) değerleri hesaplanır. Kesinlik ve geri çağırma terimleri modelin performansını hesaplarken kullanılan önemli metriklerdendir. Sırası ile denklem 4.1 ve denklem 4.2'de gösterildiği gibi doğru pozitiflerin yanlış pozitif ve doğru pozitiflerin toplamına oranı kesinliği verirken doğru pozitiflerin yanlış negatif ve doğru pozitiflerin toplamına oranı ile geri çağırma değeri elde edilmektedir [69].

$$\text{Kesinlik} = \frac{TP}{TP + FP} \quad (4.1)$$

$$\text{GeriÇağırma} = \frac{TP}{TP + FN} \quad (4.2)$$

Bu terimler, nesne tespiti görevlerinde modelin performansını ölçmek için gereken güven skoru, birleşim üzerinden kesişme (IoU) gibi metriklerin hesaplanmasında kullanılmaktadır. İkinci kısımda da belirtildiği gibi bir tutturucu kutucuğun nesne içerme olasılığı güven skoru olarak tanımlanmaktadır. Birleşim üzerinden kesişme (IoU) ise tahmin edilen sınırlandırıcı kutucuk ile kesin referansın kesişiminin iki kutucuğun birleşim alanına oranı olarak tanımlanır. IoU'ya ait formülizasyon denklem 4.3'te verilmiştir. Denklem 4.3'de B_p tahmin edilen kutucuğu, B_{gt} ise kesin referansa ait kutucuğu gösterir.

$$\text{IoU} = \frac{\text{Alan}(B_p \cap B_{gt})}{\text{Alan}(B_p \cup B_{gt})} \quad (4.3)$$

Nesne tespitinde doğru pozitif terimi, IoU değerinin seçilen eşik değerinden büyük olduğu durumlar için kullanılır. 0 ile 1 arasında farklı eşik değerleri seçilerek farklı kesinlik ve geri çağırma değerleri elde edilir. Elde edilen geri çağırma değerlerinin x ekseninde, kesinlik değerlerinin ise y ekseninde gösterilmesi ile kesinlik-geri çağırma eğrisi çizilir. Daha sonra ortalama kesinlik (AP) değeri kesinlik-geri çağırma eğrisinin altında kalan alanın hesaplanması ile elde edilir. Ancak, kesinlik-geri çağırma eğrisinin altında kalan alan hesaplanmadan önce monoton azalan bir eğri elde etmek için kesinlik değerleri denklem 4.4'te verildiği gibi interpolate edilir. Denklem 4.4'te p kesinliği, r ise geri çağırma'yı ifade etmektedir [68].

$$p_{\text{interpole}}(r) = \max_{r' \geq r} p(r') \quad (4.4)$$

İnterpole edilmiş kesinlik değerleri hesaplandıktan sonra ortalama kesinlik değeri (AP) denklem 4.5'teki gibi hesaplanır. Denklem 4.5'te r_1, r_2, \dots, r_n interpolate edilmiş kesinlik değerlerine karşılık gelen geri çağırma değerlerine karşılık gelmektedir.

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interpole}(r_{i+1}) \quad (4.5)$$

Tüm bu hesaplamaların sonucunda C adet sınıf için bulunan ortalama kesinlik değerlerinin ortalaması alınarak modele ait kesinlik değerlerinin ortalaması (mAP) denklem 4.6'da verilen şekilde hesaplanır.

$$mAP = \frac{\sum_{i=1}^C AP_i}{C} \quad (4.6)$$

PASCAL VOC yarışmasına göre, yukarıda bahsedilen hesaplamalar IoU için 0.5 eşik değeri baz alınarak yapılmaktadır [68].

4.2 SIXray Veri Kümesi Üzerinde Gerçekleştirilen Deneyler

Gerçek veriler üzerinde deneysel sonuçlar alınırken öncelikle SIXray veri kümesi üzerinde sırası ile YOLOv3, Faster RCNN ve SSD modelleri denenmiştir. YOLOv3 kendi omurga ağı olan ve ikinci kısımda açıklanan Darknet-53 omurga ağını kullanmıştır. Faster RCNN ile deneyler yapılırken ilk önce ResNet-101 omurga ağı kullanılmıştır. Daha sonra ise ResNet-101 omurga ağı öznitelik piramit ağı (FPN) ile birlikte kullanılmıştır. Öznitelik piramit ağı ile alınan sonuçların ortalama kesinlik değeri daha yüksek çıktığı için SSD modelinde de ResNet-101 omurga ağı öznitelik piramit ağı ile birlikte kullanılmıştır.

SIXray veri kümesinin eğitiminde üç model için de eğitim ve test görüntüleri aynı şekilde kullanılmıştır. Toplamda etiketli 5417 X-ray veri kümesinden 4875 tanesi eğitimde, 542 tanesi ise testte kullanılmıştır.

4.2.1 YOLOv3 Yöntemi ile Elde Edilen Tehdit Tespiti Sonuçları

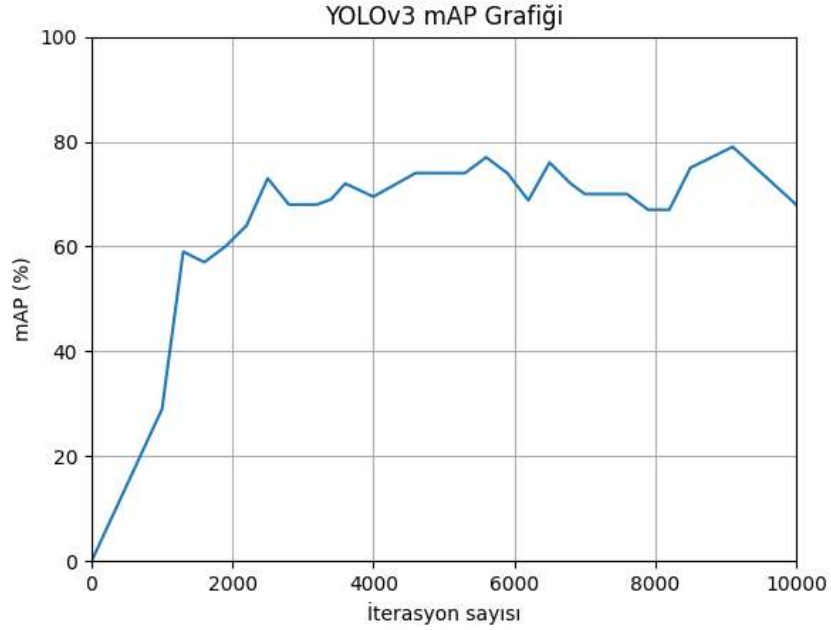
SIXray veri kümesi ilk adımda YOLOv3 ile eğitilmiştir. Eğitimden önce YOLOv3'ün SIXray veri kümesini doğru işleyebilmesi için bazı ön işlemler yapılmıştır. Bunun için ilk

önce YOLOv3'ün eğitim konfigürasyon dosyası güncellenmiştir. Konfigürasyon dosyasında sınıf sayısı silah, bıçak, makas, İngiliz anahtarı ve pense için toplam 5 olacak şekilde düzenlenmiştir. Modelin tam bağlantılı katmanı için filtre sayısı da beş sınıfa uygun olacak şekilde hesaplanıp değiştirilmiştir. Bunun nedeni YOLOv3'ün sunulan orijinal çerçevesinde eğitimlerin 80 sınıflık MS COCO veri kümesi üzerinden yapılmış olmasıdır. Ayrıca modelin toplam iterasyon sayısı da YOLOv3 için önerilen formülizasyona uygun olarak 5 sınıflı S1Xray veri kümesi için 10000 olacak şekilde güncellenmiştir. Bu düzenlemelerden sonra küme boyutu (batch size) güncellenmiştir. Küme boyutu, derin öğrenmede önemli bir hiperparametredir. Küme boyutu, öğrenmenin her iterasyonunda işlenen veri adedini belirler. Bu sayı 1 ile veri kümesindeki toplam görüntü sayısı kadar olabilir. Ancak küme boyutu arttıkça işlem yükü de artar. Bunun nedeni, geriye yayılım (back propagation) işleminde modelin ağırlık değerlerinin gradyan inişi (gradient descent) hesaplaması ile güncellenirken kullanılan veri sayısının artmasıdır. Veri sayısı arttığında hesaplamaların süresi de artacaktır. Bunun için veri kümesinin tamamının bir anda işlenmesi yerine veri kümesi küçük gruplar halinde işleme sokulur. Bu nedenle genelde küme boyutu veri kümesinin tamamı olacak şekilde girilmemektedir. Küme boyutunun 1 olduğu durumlarda ise stokastik gradyan inişi ile aynı şey yapılmış olur. Model her bir öğrenme sürecini tek veri üzerinden yapar. Bu işlem ise öğrenme sürecinin uzamasına neden olmaktadır. Tüm bu parametreler göz önüne alındığında küme boyutu 1 veya 2'nin katları (2,4,8,16,32..) olacak şekilde seçilir. Bu sayının 2'nin katları olacak şekilde seçilmesinin nedeni ise GPU belleği ile uyumlu olmasıdır. Ancak küçük bir bellekte çalışılıyorsa büyük küme boyutu sorun yaratabilir. Tüm bu etmenler göz önüne alındığında YOLOv3 için küme boyutu 64 olarak belirlenmiştir.

Öğrenme oranı (learning rate) ise başka bir önemli hiperparametredir. Öğrenme oranı özet olarak geriye yayılım aşamasında modelin ağırlıklarının güncellenmesi ile oluşan hata tahminine göre modelin ne kadar değiştirileceğini kontrol eder. Çok düşük öğrenme oranı modelin öğrenme süresinin çok uzamasına veya modelin bir lokal optimum noktasında takılmasına neden olabilmektedir. Yüksek öğrenme oranı ise öğrenim süresini hızlandırırken modelin yakınsamamasına neden olabilmektedir. Genel olarak öğrenme hızı seçilirken ilk önce belli bir aşamaya kadar görece büyük bir oran seçilirken daha sonra bu oran düşürülür. Bazı uygulamalarda ise adaptif bir şekilde güncellene-

bilmektedir. YOLOv3 konfigürasyonunda öğrenme oranı 0.001 olarak belirlenmiştir.

Konfigürasyon dosyası güncellendikten sonra verilere ait etiketler ve kesin referans kutucukların koordinat değerleri YOLOv3'e uygun olacak şekilde hem eğitim hem de test için ayrı bir dosyaya aktarılmıştır. Tüm bu işlemler tamamlandıktan sonra modelin eğitimine başlanmıştır. Şekil 4.1'de YOLOv3 modelinin eğitim sonucunda elde edilen ortalama kesinlik değerine (mAP) ait grafik sunulmuştur. Kısım 4.2'de de belirtildiği gibi ortalama kesinlik değerleri PASCAL VOC kriterine uygun olarak 0.5 IoU eşik değeri baz alınarak hesaplanmıştır.



Şekil 4.1. YOLOv3 eğitimi sonucu elde edilen mAP grafiği

Grafiğe bakıldığında ortalama kesinlik değerinin 2000. adımdan sonra %70'e ulaştığı ve bu adımdan sonra 10000. adıma kadar yaklaşık %70 ile %80 aralığında salındığı gözlemlenmiştir. Maksimum iterasyon sayısına kadar ulaşılabilen en yüksek ortalama kesinlik değeri ise **%78.7** olmuştur. Sınıf bazlı kesinlik değerleri ve örnek tespitler 4.2.4'de sunulmuştur.

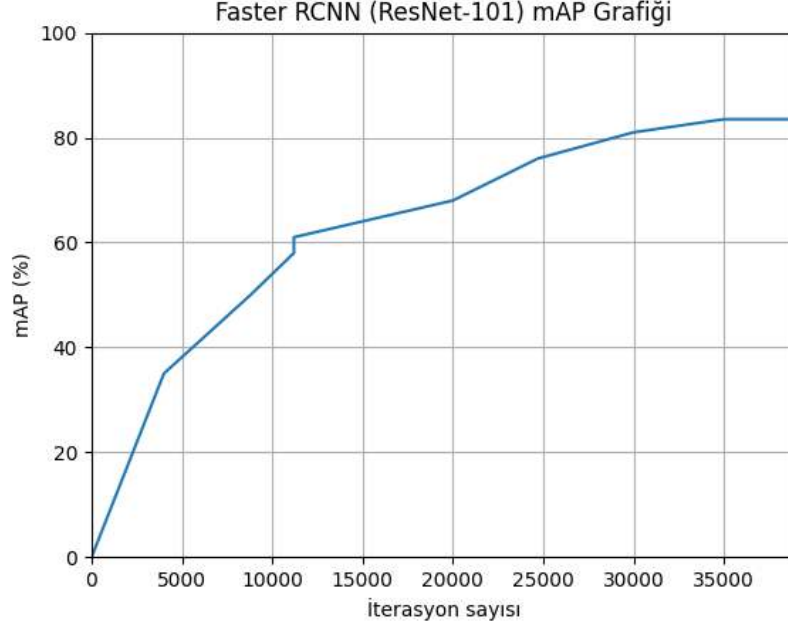
4.2.2 Faster RCNN Yöntemi ile Elde Edilen Tehdit Tespiti Sonuçları

4.2.2.1 RESNET-101 Omurga Ağı ile Eğitim

Bu adımda SIXray veri kümesi Faster RCNN ile eğitilmiştir. Faster RCNN ile eğitilirken ilk aşamada omurga ağ olarak ResNet-101 kullanılmıştır. Faster RCNN için gerekli olan kodlamalar ve ağ yapısı için Tensorflow nesne tespiti uygulamasından (Tensorflow Object Detection API) yararlanılmıştır. SIXray veri kümesi eğitimden önce bu uygulamaya uygun olacak şekilde düzenlenmiştir. Bunun için öncelikle eğitim ve test verileri farklı dosyalara ayrılmış ve her kesin referans için sınıf ve koordinat bilgileri csv uzantılı dosyalara kaydedilmiştir. Daha sonra eğitim ve test verileri Tensorflow nesne tespiti uygulamasına uygun olacak şekilde kayıt edilmiştir. Bu işlemlerden sonra SIXray'de bulunan tehdit sınıfları uygulamaya tanıtmak için ayrı bir dosya oluşturulmuştur. Eğitime başlamadan önce Faster RCNN ResNet-101 omurga ağı için konfigürasyon dosyası düzenlenmiştir. Bu düzenlemede küme boyutu 2 olarak seçilmiştir. Bunun nedeni daha yüksek küme boyutu denemelerinde hafıza kaynaklı hatalar alınmasıdır. Konfigürasyon dosyasından yapılan başka güncellemeler ise adım sayısı ve öğrenme oranı hiperparametrelerinin düzenlenmesi olmuştur. Faster RCNN için temel öğrenme oranı 0.01 olarak seçilmiştir. Toplam adım sayısı ise 50000 olarak belirlenmiştir.

Konfigürasyon dosyasında modelin eğitimi için önemli olan başka özellikler de vardır. Bunlar seyreltme (dropout) ve veri arttırma (data augmentation) parametreleridir. Kısaca açıklayacak olursak seyreltme, eğitim aşamasında aşırı uyumu (overfitting) engellemek için uygulanan bir yöntemdir. Bu yöntem ile sinir ağlarındaki nöronların rastgele seçilen bir kısmı ileri ve geri çağırma aşamalarında hesaba katılmaz. Böylece algoritmanın aşırı öğrenme riski azaltılmış olur. Bunun dışında görüntülerin rastgele karesel bir formatta kırılması ile veri arttırımı da sağlanmaktadır. Bu yöntem sayesinde her zaman görüntünün tümü eğitime katılmayacağı için modelin daha genel bir sonuç alması sağlanabilmektedir. Konfigürasyon dosyası SIXray verilerine uygun bir şekilde güncellendikten ve gerekli olan ekstra yazılımlar yüklenip çalıştırıldıktan sonra eğitim başlamıştır. Şekil 4.2'de modelin eğitim sonucu elde edilen ortalama kesinlik değeri (mAP) sunulmaktadır. Grafik incelendiğinde 35000. adımdan sonra ortalama kesinlik değerinin değişmediği görülmüştür. Bu nedenle bu adımda eğitim durdurulmuştur. Elde edilen en

iyi ağırlıklı modelin ortalama kesinlik değeri (mAP) 0.5 IoU eşik değeri baz alındığında %83.4 olmuştur. Sınıf bazlı kesinlik değerleri ve örnek tespitler 4.2.4'te sunulmuştur.

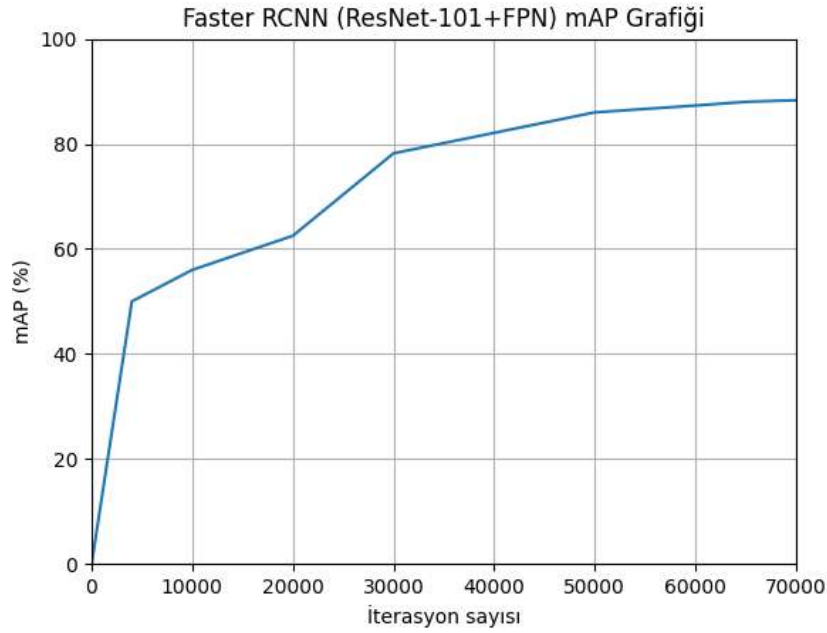


Şekil 4.2. Faster RCNN (ResNet-101) eğitimi sonucu elde edilen mAP grafiği

4.2.2.2 RESNET-101 + FPN Omurga Ağı ile Eğitim

Tez çalışmasının bu bölümünde SIXray veri kümesi Faster RCNN modeli ResNet-101+FPN omurga ağı kullanılarak eğitilmiştir. Öznitelik piramit ağı (FPN), ResNet-101 omurga ağı ile kullanırken ResNet-101 omurga ağından geçen görüntüler kısım 2.2.4'de açıklanan aşağıdan yukarıya yol aşamasında kullanılır. ResNet-101 omurga ağında işlenmeye başlayan girdi görüntüsünün her evrimsel operasyonda 2 katsayısı ile aşağı örnekleme alınacağı için görüntünün boyutu bir önceki evrimsel katmana göre yarıya düşmektedir. Bu nedenle görüntünün çözünürlüğü azalırken aynı zamanda görüntü semantik açıdan daha anlamlı hale gelmektedir. Öznitelik piramit ağı, ResNet-101 omurga ağı çıktıları yukarıdan aşağıya yol aşamasında 2 katsayısı ile yukarı örnekler ve böylece çözünürlüğü azalan görüntülerin boyutu iki katına çıkacağı için çözünürlüğün artması beklenir. Aşağıdan yukarıya ve yukarıdan aşağıya aşamalardan elde edilen aynı boyutlu görüntüler birleştirilerek orijinal ResNet-101 omurga ağının çıktısına göre daha anlamlı bir öznitelik haritası elde etmek amaçlanır. ResNet-101 omurga ağı ile öznitelik

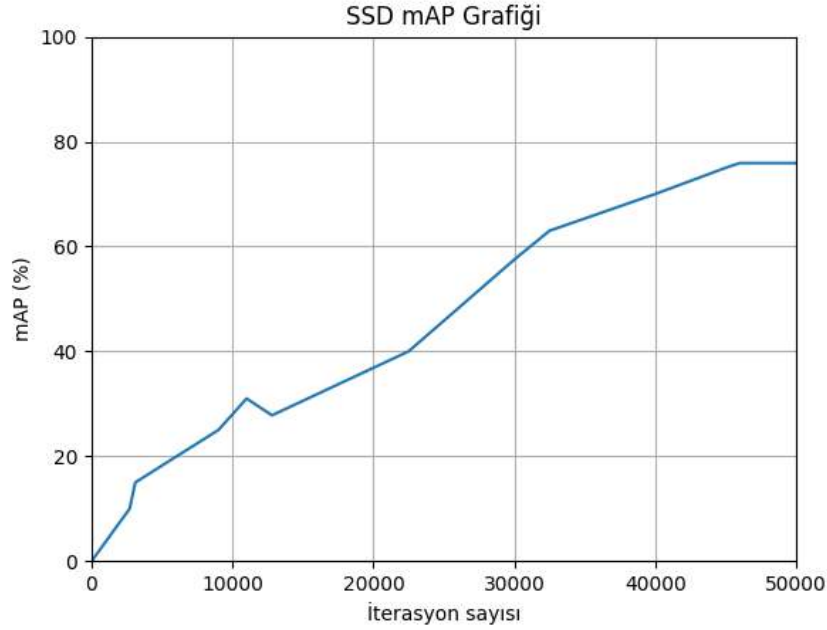
piramit ađını birlikte kullanırken PyTorch kütüphanesinden yararlanılmıştır. SIXray veri kümesinin bu kütüphaneye uygun hale getirilmesi için eğitim ve test görüntüleri PASCAL VOC veri kümesinin yapısına uygun bir biçimde düzenlenmiştir. Bunun için eğitim ve test görüntüleri ayrı ayrı *txt* dosyasında sadece isimleri olacak şekilde kaydedilmiştir. Daha sonra SIXray veri kümesi ayrıca kütüphaneye tanıtılmıştır. Bu kütüphanede Faster RCNN ResNet-101+FPN omurga ađı ile eğitim için gerekli konfigürasyonlar yapılırken küme boyutu 2 olarak seçilmiştir. Eğitim için öğrenme oranı ise 0.01 olarak seçilirken toplam adım sayısı ise 70000 olarak seçilmiştir. Şekil 4.3'te, SIXray veri kümesinin ResNet-101+FPN omurga ađı kullanan Faster RCNN ile eğitimi sonucu elde edilen ortalama kesinlik değeri (mAP), 0.5 IoU eşik değeri için verilmiştir. Grafik incelendiğinde modelinin öğrenme sürecinin 60000. adımdan sonra %88 seviyesinde fazla deđişmediđi saptanmıştır. Son adım sayısına kadar devam ettirilen eğitimde elde edilen en iyi ortalama kesinlik değeri **%88.3** olmuştur. Sınıf bazlı kesinlik değeri ve örnek tespitler diđer modellerle karşılaştırma amacı ile 4.2.4'te sunulmuştur.



Şekil 4.3. Faster RCNN (ResNet-101+FPN) eğitimi sonucu elde edilen mAP grafiđi

4.2.3 SSD Yöntemi ile Elde Edilen Tehdit Tespiti Sonuçları

Bu aşamada ise SIXray veri kümesi SSD ile eğitilmiştir. SSD eğitimi için de omurga ağ olarak ResNet-101 + FPN (feature pyramid network) seçilmiş ve modelin eğitimi için gerekli olan ağ yapısına ait kodlamalar için Tensorflow nesne tespiti uygulaması kullanılmıştır. Faster RCNN ile benzer şekilde konfigürasyon düzenlemeleri yapılmıştır. Düzenlemeler yapılırken küme boyutu Faster RCNN'de olduğu gibi hafıza kaynaklı hataları önlemek için 2 olarak seçilmiştir. Öğrenme oranı ve toplam adım sayısı ise sırasıyla 0.01 ve 50000 olacak şekilde düzenlenmiştir. Faster RCNN ile aynı uygulama kullanıldığı için eğitim ve test verileri de Faster RCNN'de olduğu gibi kullanılmıştır. Benzer şekilde veri arttırımı için rastgele karesel kırpma yöntemi uygulanmıştır. Şekil 4.4'te SSD eğitimi sonucu alınan ortalama kesinlik değerine (mAP) ait grafik sunulmuştur.



Şekil 4.4. SSD eğitimi sonucu elde edilen mAP grafiği

Grafiğe bakıldığında ilk 10000 adımda modelin başarımının %30 mAP değerine kadar ulaşabildiği görülmüştür. Adım sayısı ilerledikçe başarımlar artmaya devam etmiştir. Yaklaşık 45000. adımdan sonra ise modelin mAP değeri yaklaşık olarak %75 değerinde sabit kalmaya başlamıştır ve belirlenen maksimum adım sayısı olan 50000. adımda modelin mAP değeri **%75.1**'e ulaşmıştır. Sınıf bazlı kesinlik değerleri ve örnek tespitler 4.2.4'de

sunulmuştur.

4.2.4 SIXray Veri Kümesi Üzerinde Alınan Sonuçların Karşılaştırılması

Tezin bu bölümünde SIXray veri kümesinin eğitildiği YOLOv3, Faster RCNN ve SSD modellerinin performansları karşılaştırılmıştır. Farklı modellere ait örnek tespitler hem bir kategoriye ait tehdit içeren hem de karışık sınıflardan tehditler içeren görüntüler sunulmaktadır. Ayrıca, HUMS veri kümesinde bulunan yapay RGB görüntülerden bir test alt kümesi oluşturulmuş ve her modelden alınan en iyi ağırlıklarla test edilmiştir. Elde edilen sonuçlar da bu kısımda karşılaştırılmıştır. Çizelge 4.1’de modellere ait sınıf bazlı ve ortalama kesinlik değerleri sunulmuştur.

Model	Omurga Ağ	Silah	Bıçak	Makas	İngiliz Anahtarı	Pense	mAP (%)
YOLOv3	Darknet-53	94.9	72.2	66.1	83.0	77.6	78.7
Faster RCNN	ResNet-101	95.5	82.4	68.2	83.2	87.7	83.4
Faster RCNN	ResNet-101+FPN	90.7	82.3	87.8	90.5	90.3	88.3
SSD	ResNet-101+FPN	91.8	70.9	65.1	68.6	79.1	75.1
Faster RCNN **	ResNet-101	91	81	-	-	-	86

Çizelge 4.1. SIXray veri kümesinde denenen modellere ait sınıf bazlı ve genel sonuçların karşılaştırılması (**Literatürde sunulan en iyi sonuç)

Sonuçlar incelendiğinde tüm modeller için en yüksek başarımın silah sınıfında alındığı görülmektedir. Bunun bir nedeni silah sınıfına ait verilerin sayıca daha fazla olması ve diğer objeler tarafından engellenmemiş olmasıdır. Bir başka nedeni ise X-ray görüntülerinde silahların karakteristik özelliklerinin (metalik parçaları, şekli vb.) diğer sınıflara göre daha belirgin olmasıdır. Tüm modeller karşılaştırıldığında ise silah sınıfında en yüksek başarımın %95.5 ile Faster RCNN modelinin ResNet-101 omurga ağı ile kullanıldığı durumda alındığı görülmüştür. Buna en yakın sonuç ise YOLOv3 ile elde edilmiştir. Bıçak sınıfına bakıldığında ise en yüksek sonucun yine Faster RCNN ile alındığı gözlemlenmektedir. Faster RCNN’in hem ResNet-101 (%82.4) hem de ResNet-101 + FPN (%82.3) omurga ağları ile eğitimi sonucu bıçak sınıfında hemen hemen aynı başarıya ulaşılmıştır. YOLOv3 ve SSD modellerinde ise bıçak sınıfının kesinlik değeri sı-

rası ile %72.2 ve %70.9'a ulaşabilmiştir. Bıçak sınıfında başarının düşmesinin başlıca nedenleri kullanılan bıçakların aynı tipte olmaması ve görüntülerde bulunan bıçakların karmaşık ve diğer nesnelere tarafından oklüziona uğramış bir biçimde konumlandırılmış olmasıdır. Makas sınıfı için ise en iyi sonuç %87.8 ile Faster RCNN'in ResNet-101 + FPN omurga ağı ile kullanıldığı durumda elde edilmiştir. Genel çerçevede bakıldığında ise modellerin makas sınıfında yeterince başarılı olmadığı görülmüştür. Makas sınıfında alınan en iyi sonuç Faster RCNN ResNet-101 + FPN omurga ağı ile %87.8 olmuştur. Makas sınıfındaki başarısızlığın başlıca nedenlerinin ise bu sınıf için etiketli örnek sayısının az olması ve bazı durumlarda modellerin makas ile bıçak görüntülerini bu tehditlerin şekil benzerliğinden dolayı karıştırmaları olarak yorumlanmaktadır. İngiliz anahtarı için incelendiğinde ise en iyi sonucun yine ResNet-101 + FPN omurga ağı kullanıldığı durumda Faster RCNN ile %90.5 olarak alındığı saptanmıştır. Bu sınıf için YOLOv3 ve ResNet-101 omurga ağı ile kullanılan Faster RCNN modellerin sırası ile %83.0 ve %83.2 olmak üzere benzer sonuçlara ulaştığı görülmüştür. Son kategori olan pense için en yüksek başarı %90.3 olmuştur. Bu sonuç ResNet-101 + FPN omurga ağı kullanılan Faster RCNN ile alınmıştır. Buna en yakın başarıyı ResNet-101 omurga ağı kullanılan Faster RCNN (%87.7) elde etmiştir. YOLOv3 ise bu kategoride en düşük başarıyı göstermiştir. Modellerin ortalama kesinlik değerlerine (mAP) bakıldığında ise en yüksek sonucun Faster RCNN ResNet-101 + FPN omurga ağı ile alındığı ve bu değer %88.3 olduğu görülmüştür. SIXray veri kümesinde en düşük skoru gösteren algoritma ise %75.1 ortalama kesinlik değeri ile SSD olmuştur. Literatürde SIXray veri kümesi üzerinde alınan en iyi sonuçlar incelendiğinde ise silah ve bıçak sınıfları için tez çalışmasında literatüre kıyasla Faster RCNN ResNet-101 modeli için kesinlik değerleri sırasıyla %91'den %95.5'e ve %81'den %82.4'e yükselmiştir. Ortalama kesinlik değerleri incelendiğinde ise Faster RCNN ResNet-101+FPN modeli için sonucun %86'dan %88.3'e yükseldiği görülmüştür.

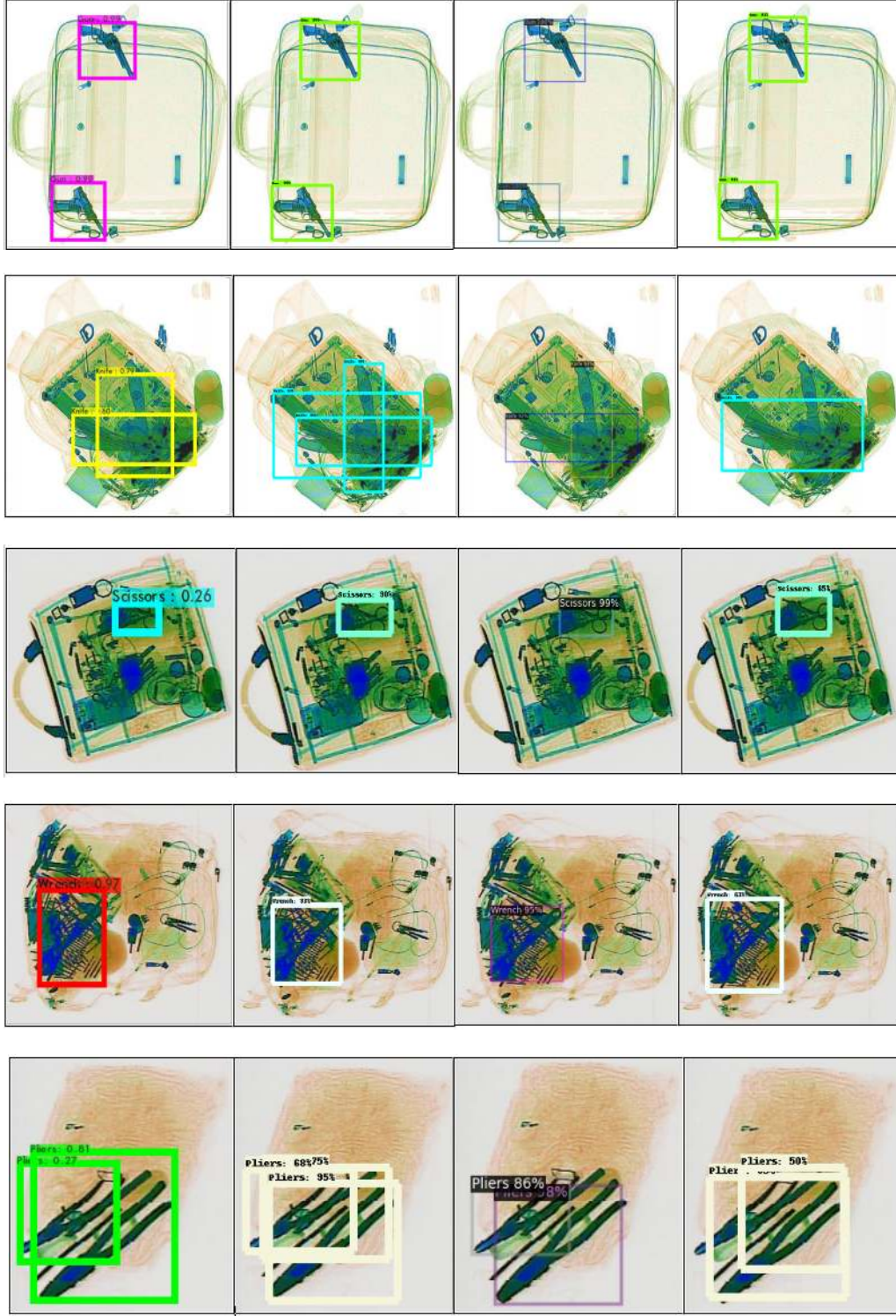
Şekil 4.5'te sadece bir adet kategoriye ait tehdit içeren örnek görüntülerin farklı modellere ait tespit sonuçları sunulmuş ve bu sonuçlar karşılaştırılmıştır. Şekilde sütun bazlı düşünüldüğünde sırası ile YOLOv3, Faster RCNN (ResNet-101), Faster RCNN (ResNet-101 + FPN) ve SSD modellerinden elde edilen örnek tespitler sunulmuştur. Satır bazlı düşünüldüğünde ise sırasıyla silah, bıçak, makas, İngiliz anahtarı ve pense

sınıflarına ait tespitler sunulmuştur. Modellerin tehdit tespitindeki başarılarını karşılaştırmak amacı ile tüm modellerde aynı sınıf için aynı X-ray görüntüsü kullanılmıştır.

Şekil 4.5'te ilk satırda bulunan silah tespitleri incelendiğinde ilk sütunda bulunan YOLOv3'e ait tespitlerde iki adet silah tehdidinin de %99 güven skoru ile başarılı bir şekilde tespit edildiği görülmüştür. İkinci sütunda sunulan ve Faster RCNN (ResNet-101) ile test edilen görüntüde iki adet silah nesnesi YOLOv3 ile benzer şekilde %99 güven skoru ile tespit edilmiştir. Üçüncü sütunda bulunan Faster RCNN (ResNet-101 + FPN) ile alınan silah tespitlerinin ikisinin de %100 güven skoru ile tespit edildiği görülmüştür. Son sütunda ise SSD ile alınan tespitler sunulmuş ve görüntüde bulunan iki silahın da %91 güven skoru ile tespit edildiği saptanmıştır. Silah için sunulan örnekteki görüntü incelendiğinde silahların birbirinden bağımsız bir şekilde konumlandığı ve bu nesnelere engelleyen başka bir nesne olmadığı görülmektedir. Bu nedenle bu görüntü için yapılan testte kullanılan tüm modellerde silahların yüksek başarı ile tespit edilebildiği gözlemlenmiştir. Silahlara ait en düşük tespit skoru ise SSD ile alınmıştır.

Şekil 4.5'in ikinci satırında ise bıçak sınıfına ait tespitler verilmiştir. Görüntüye bakıldığında bıçakların üst üste konumlandığı ve çanta içerisinde bulunan diğer nesnelere tarafından engellendikleri görülmektedir. Bu etmenler görüntüdeki bıçakların tespitlerini zorlaştırmaktadır. YOLOv3 ile test edildiğinde görüntüde bulunan büyük bıçak %79 güven skoru ile tespit edilirken küçük bıçak ise %60 güven skoru ile tespit edilebilmiştir. İkinci sütuna bakıldığında Faster RCNN ResNet-101 omurga ağı kullanılarak alınan sonuçlar ile test edilen iki adet bıçaktan dikey olanın %96, yatay olanın ise %86 güven skoru ile tespit edildiği görülmüştür. Ancak iki tespit dışında bir de %53 güven skoru ile yanlış pozitif bir bıçak tespiti yapıldığı saptanmıştır. Üçüncü sütunda bulunan Faster RCNN ResNet-101 + FPN kullanılarak alınan bıçak tespit skorlarına bakıldığında görüntüde bulunan dikey bıçağın %93, yatay bıçağın ise %92 güven skoru ile tespit edildiği görülmüştür. Dördüncü sütunda bulunan ve SSD ile test edilen bıçaklara ait tespitlere bakıldığında görüntüde bulunan iki adet bıçaktan sadece birinin tespit edilebildiği görülmüştür. Tespit edilen bıçak ise %50 güven skoru ile tespit edilebilmiştir. Tespit edilemeyen bıçak model için bir yanlış negatiftir.

Şekil 4.5'in üçüncü satırında ise makas sınıfı için örnek bir tespit sunulmuştur. Gö-



Şekil 4.5. Sütunlarda sırayla YOLOv3, Faster RCNN (ResNet-101, ResNet-101+FPN) ve SSD modellerinden alınan farklı kategorilere ait tespitler verilmiştir. Satırlar sırayla silah, bıçak, makas, İngiliz anahtarı ve penseye ait görüntülerdir.

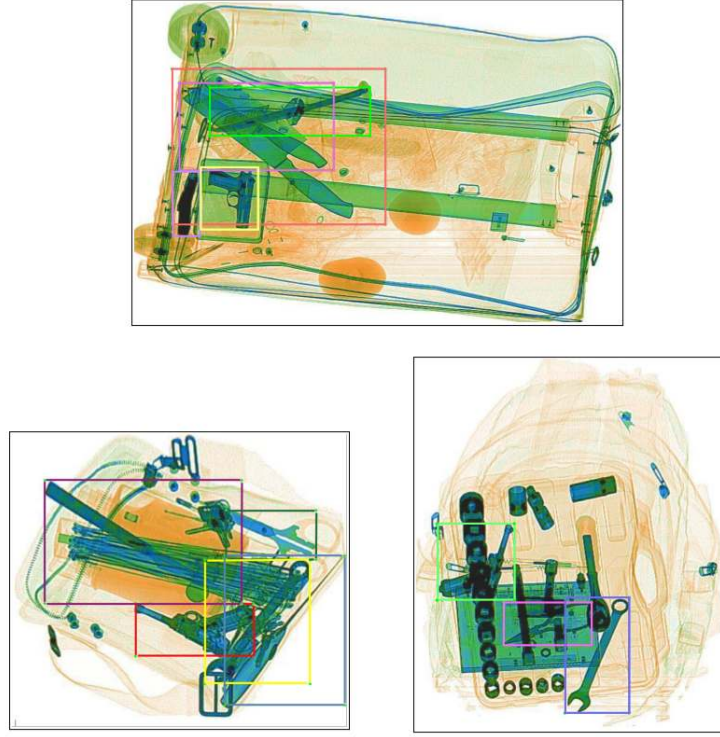
rüntüye bakıldığında makas nesnesinin diğer nesnelere göre belirginliğinin az olduğu ve keskin kısmının başka bir nesne ile çakıştığı görülmektedir. İlk sütunda bulunan ve YOLOv3 ile test edilen görüntüdeki makasın %26 güven skoru ile tespit edilebilmiştir. Genel anlamda modelin makas sınıfı için kesinlik değerinin de diğer sınıflara kıyasla daha düşük olduğu düşünüldüğünde makasların tespitinin daha zor olacağı tahmin edilmektedir. Aynı makas görüntüsü için ikinci sütunda bulunan Faster RCNN ResNet-101 modelinden alınan tespit incelendiğinde ise bu makasın %90 güven skoru ile doğru bir şekilde tespit edilebildiği görülmüştür. Dördüncü kolonda bulunan Faster RCNN ResNet-101+FPN ile alınan örnekte ise makas %99 güven skoru ile tespit edilmiştir. Son olarak aynı makas görüntüsü SSD ile test edildiğinde %65 güven skoru ile tespit edilebilmiştir. Buna göre tek makas içeren bir görüntüde en başarılı sonucu ResNet-101+FPN omurga ağı kullanılan Faster RCNN modeli vermiştir. Modelin eğitimi sonucunda çizelge 4.1'de de verildiği gibi makas sınıfı diğer modellere kıyasla daha başarılı bir şekilde eğitilebilmiştir.

Şekil 4.5'in dördüncü satırında ise İngiliz anahtarı sınıfı için farklı modellere ait tespitler sunulmuştur. YOLOv3 ile alınan tespitte modelin görüntüdeki İngiliz anahtarını %97 güven skoru ile tespit edebildiği görülmüştür. İkinci sütunda bulunan ve Faster RCNN ResNet-101 modelinden alınan tespit incelendiğinde ise İngiliz anahtarının %93 güven skoru ile tespit edildiği görülmüştür. Dördüncü kolonda bulunan Faster RCNN ResNet-101+FPN ile alınan örnekte aynı İngiliz anahtarı %100 güven skoru ile tespit edilmiştir. Son olarak İngiliz anahtarı içeren görüntü SSD ile test edildiğinde %63 güven skoru ile tespit edilebilmiştir. Son satırda bulunan pense tespitleri incelendiğinde ise YOLOv3 ile test edilen görüntüde öndeki pensenin model tarafından %81 güven skoru ile tespit edilirken arkada kalan pensenin ise %27 güven skoru ile tespit edildiği görülmüştür. İkinci kolonda bulunan ve Faster RCNN (ResNet-101) ile test edilen durumda görüntüde bulunan iki adet penseden önde olanı %95 güven skoru, arkada olanı ise %75 güven skoru ile tespit edilmiştir. Ancak ekstra bir tespit olan %68 güven skorlu pense tespiti ise bir yanlış pozitifdir. Üçüncü kolondaki Faster RCNN (ResNet-101+FPN) tespitinde ise önde duran pense %98 güven skoru ile tespit edilirken arkada duran pense ise %86 güven skoru ile tespit edilmiştir. Son denemede ise aynı görüntü SSD ile test edilmiş ve iki adet penseden önde olanı %65, arkada olanı ise %50 güven skoru ile tespit edilmiştir.

Ancak görüntüde tespitlere ait sınırlandırıcı kutucuklar incelendiğinde özellikle arkada bulunan pense için konumlandırmanın yeterince iyi yapılmadığı görülmüştür.

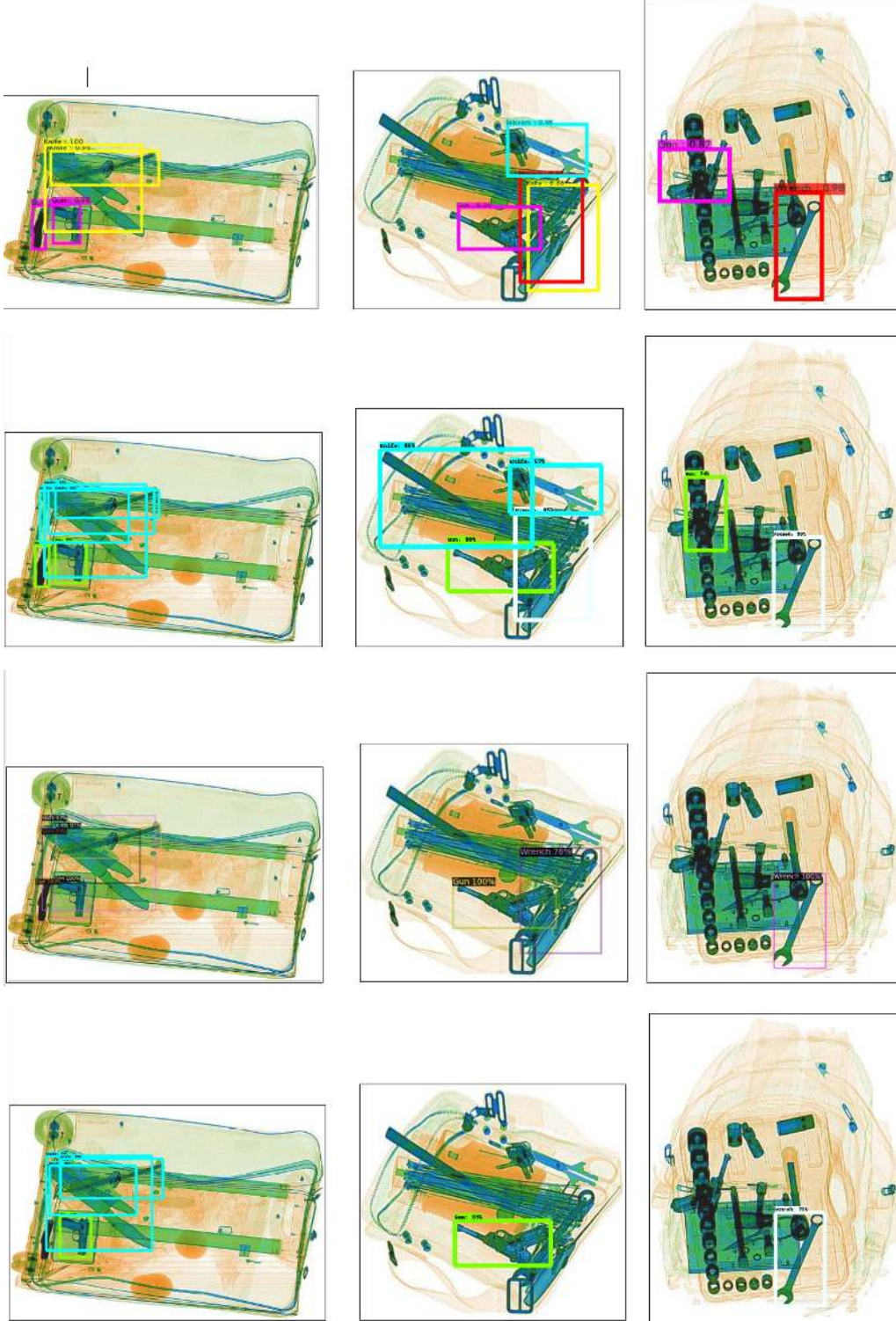
Şekil 4.6'da farklı sınıflara ait tehdit içeren örnek görüntülere ait kesin referanslar, Şekil 4.7'de ise Şekil 4.6'da verilen görüntülere ait tespitler verilmiştir. Şeklin ilk satırında YOLOv3, ikinci satırında Faster RCNN (ResNet-101), üçüncü satırında Faster RCNN (ResNet-101+FPN) ve son satırında SSD modeline ait tehdit tespitleri sunulmuş ve sonuçlar karşılaştırılmıştır. Verilen şekilde ilk sütundaki görüntüde bıçak ve silahlara ait tehditler bulunmaktadır. İlgili görüntünün YOLOv3 ile testi sonucunda görüntüde bulunan silahlardan önde olanı model tarafından %98, arka tarafında yaklaşık 90 derece dönük halde bulunan silah ise %97 güven skoru ile tespit edilmiştir. Bıçak örneklerinde ise büyük bıçak %100 güven skoru ile tespit edilirken üst kısımda bulunan ve farklı tipteki bıçak örneği ise %98 güven skoru ile tespit edilmiştir. Şeklin ilk sütununun ikinci satırında bulunan görüntünün ise ResNet-101 omurga ağı ile eğitilen Faster RCNN ile test edilmesi sonucu görüntüde bulunan iki adet silahtan 90 derece dönük olarak yerleştirilene %97 doğruluk ile tespit edilirken düz ve belirgin olarak duran silah ise %99 doğruluk ile tespit edilmiştir. Görüntüde bulunan büyük bıçak %82 güven skoru ile tespit edilirken ince bıçak ise %66 güven skoru ile tespit edilmiştir. İki bıçağın ortasında bulunan bir başka bıçak ise %52 güven skoru ile tespit edilirken aynı bıçak YOLOv3 tarafından tespit edilememiştir. Ayrıca, dördüncü bir bıçak tespiti daha yapılmıştır ancak görüntüde üç adet bıçak bulunduğu için bu tespit yanlış pozitif olarak sınıflandırılmıştır. Aynı görüntü, ResNet-101+FPN omurga ağı ile eğitilen Faster RCNN modeli ile test edildiğinde ise büyük bıçak ile ince ve farklı tipteki bıçak %97, ortada bulunan bıçak ise %99 güven skoruyla tespit edilmiştir. Görüntüde bulunan iki silahın da ResNet-101+FPN omurga ağı ile %100 güven skoru ile tespit edildiği görülmüştür. Son satırda ise aynı görüntünün SSD ile test edilmesi sonucu alınan tehdit tespitleri sunulmuştur. Görüntüde bulunan üç adet bıçaktan büyük olan bıçak %84 güven skoru ile tespit edilmiştir. İki bıçağın ortasındaki bıçak ise %86 güven skoru ile tespit edilirken çeşit olarak diğerlerinden farklı olan ince bıçak ise %78 güven skoru ile tespit edilmiştir. Görüntüde bulunan iki adet silahtan ise düz bir biçimde bulunan silah %94 güven skoru ile tespit edilirken açılı bir biçimde duran silah ise model tarafından tespit edilememiştir. Dört modelden elde edilen sonuçlar karşılaştırıldığında hem bıçak hem de silah tehditlerinin

öznitelik piramit ağı kullanılan durumda tespitlerde daha başarılı olduğu görülmektedir.



Şekil 4.6. Farklı sınıflara ait kesin referans içeren örnek görüntüler

Şekil 4.7'nin ikinci kolonunda bulunan görüntü ise makas, silah, bıçak ve İngiliz anahtarına ait tespitler içermektedir. Görüntüde bulunan makas model tarafından %96 doğruluk ile tespit edilirken silah %94, bıçak %68 ve İngiliz anahtarı %93 doğrulukla tespit edilmiştir. Ancak görüntünün solunda bulunan büyük bıçak ise model tarafından tespit edilememiştir. ResNet-101 omurga ağı ile eğitilen Faster RCNN ile test edildiğinde iki adet bıçak içerisinde büyük olan bıçak %66 başarı ile tespit edilirken diğer bıçak tespit edilememiştir. Görüntüde bulunan makas ise %67 başarı ile bıçak olarak yanlış tespit edilmiştir. Silaha ait tespit ise %99 doğrulukla gerçekleşmiştir. İngiliz anahtarı ise %85 güven skoru ile tespit edilmiştir. Aynı kolonun üçüncü satırında ise ResNet-101+FPN omurga ağı ile eğitilen Faster RCNN modeli kullanılarak test edilen görüntü incelendiğinde sadece silah ve İngiliz anahtarı tehditlerinin tespit edilebildiği, makas ve bıçaklara ait tespitlerin ise tespit edilemediği görülmüştür. Tespit edilen silah %100 ve İngiliz anahtarı ise %76 güven skoru ile tespit edilmiştir. Şeklin SSD ile test edilen alt görüntüsüne bakıldığında ise model tarafından sadece silahın %89 güven skoru ile



Şekil 4.7. Dört satırda sırası ile YOLOv3, Faster RCNN (ResNet-101), Faster RCNN (ResNet-101+FPN) ve SSD modellerinden alınan ve farklı kategorilere ait tehdit tespitleri içeren görüntüler

tespit edilebildiği ve diğer tehditlerin tespit edilemediği görülmüştür. Son olarak Şekil 4.7'nin üçüncü kolonunda bulunan X-ray test görüntüsü silah ve İngiliz anahtarı tehditleri içermektedir. YOLOv3 ile test edildiğinde görüntüde silahla çakışan bir nesnenin varlığından dolayı silaha ait tehdit %82 güven skoru ile tespit edilmiştir. İngiliz anahtarı ise %98 doğrulukla tespit edilebilmiştir. Aynı görüntü Faster RCNN ResNet-101 omurga ağı kullanılan versiyonu ile test edildiğinde ise başka bir nesne tarafından engellenen silah %74 güven skoru ile tespit edilirken İngiliz anahtarı ise %99 güven skoru ile tespit edilmiştir. Üçüncü satırda bulunan ve Faster RCNN ResNet-101+FPN omurga ağı kullanılan model ile test edilen durumda ise görüntüde bulunan tehditlerden sadece İngiliz anahtarı %100 güven skoru ile tespit edilirken silah ise model tarafından tespit edilememiştir. SSD ile yapılan test sonucunda da Faster RCNN (ResNet-101+FPN) ile benzer şekilde sadece İngiliz anahtarı model tarafından %73 güven skoru ile tespit edilirken silah ise tespit edilememiştir.

SIXray ile yapılan eğitim ve testlerin ardından HUMS veri kümesinde bulunan yapay RGB görüntüler de SIXray ile eğitimin sonucunda her model için elde edilen en iyi ağırlıklarla test edilmiştir. RGB görüntüler test edilirken tüm modellerde 4.3.1 kısmında belirtilen test kümesi kullanılmıştır. Buradaki amaç, büyük ve kapsamlı bir veri kümesi ile elde edilen sonuçların daha küçük bir veri kümesinde test edildiğinde ne kadar başarı gösterdiğini ölçmektir. Özetle veri kümeleri arasındaki alan farkının (domain difference) etkilerini gözlemlemektir.

Çizelge 4.2'de SIXray ile elde edilen en iyi sonuçların tüm modeller için HUMS veri kümesi üzerinde test edilmesine ait karşılaştırma sunulmuştur. Çizelgeye göre, YOLOv3 ile test edilen bıçakların tespitinde ortalama %13.6 kesinlik değerine ulaşılabilmektedir. Faster RCNN ResNet-101 omurga ağı ile elde edilen sonuçlar testte kullanıldığında ise %29.1 kesinlik değerine ulaşılırken Faster RCNN ResNet-101+FPN omurga ağı ile test sonucu %37.4'e yükselmiştir. SSD ile alınan sonuçlar HUMS veri kümesindeki bıçakların testinde kullanıldığında ise %19.0 kesinlik değeri elde edilmiştir. Tüm sonuçlar kıyaslandığında HUMS veri kümesinin testinde en iyi sonucu ResNet-101'in öznitelik piramit ağı ile kullanıldığı durumda eğitilen Faster RCNN modeli verse de SIXray test sonuçlarına göre performansta kayıp yaşanmaktadır. Farklı veri kümelerinde benzer sı-

nıflar olmasına rağmen performansın düşmesinin belli başlı nedenleri bulunmaktadır. İki veri kümesinde kullanılan bıçakların farklı tiplerde olması, HUMS veri kümesindeki bıçakların çantaların içerisinde birbirini engelleyecek şekilde konumlandırılması ve bu nedenlerden dolayı iki veri kümesi arasındaki farklılığın gittikçe açılmasıdır.

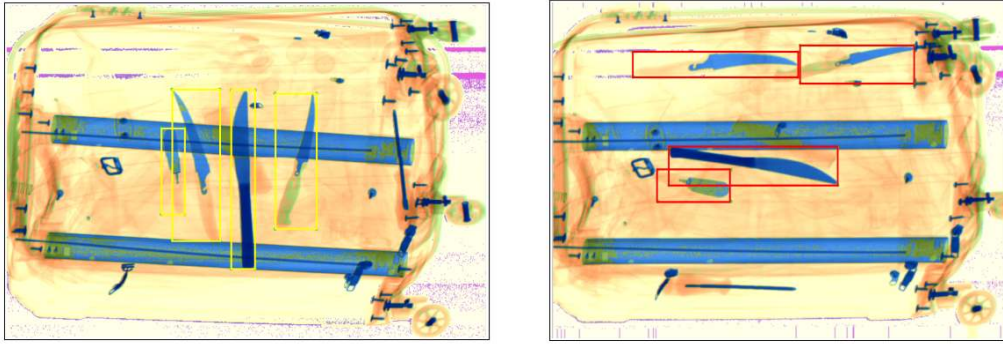
Model	Omurga Ağ	HUMS'teki bıçakların kesinlik değeri (AP)
YOLOv3	Darknet-53	13.6
Faster RCNN	ResNet-101	29.1
Faster RCNN	ResNet-101 + FPN	37.4
SSD	ResNet-101 + FPN	19.0

Çizelge 4.2. HUMS veri kümesi üzerinde yapılan test sonuçları

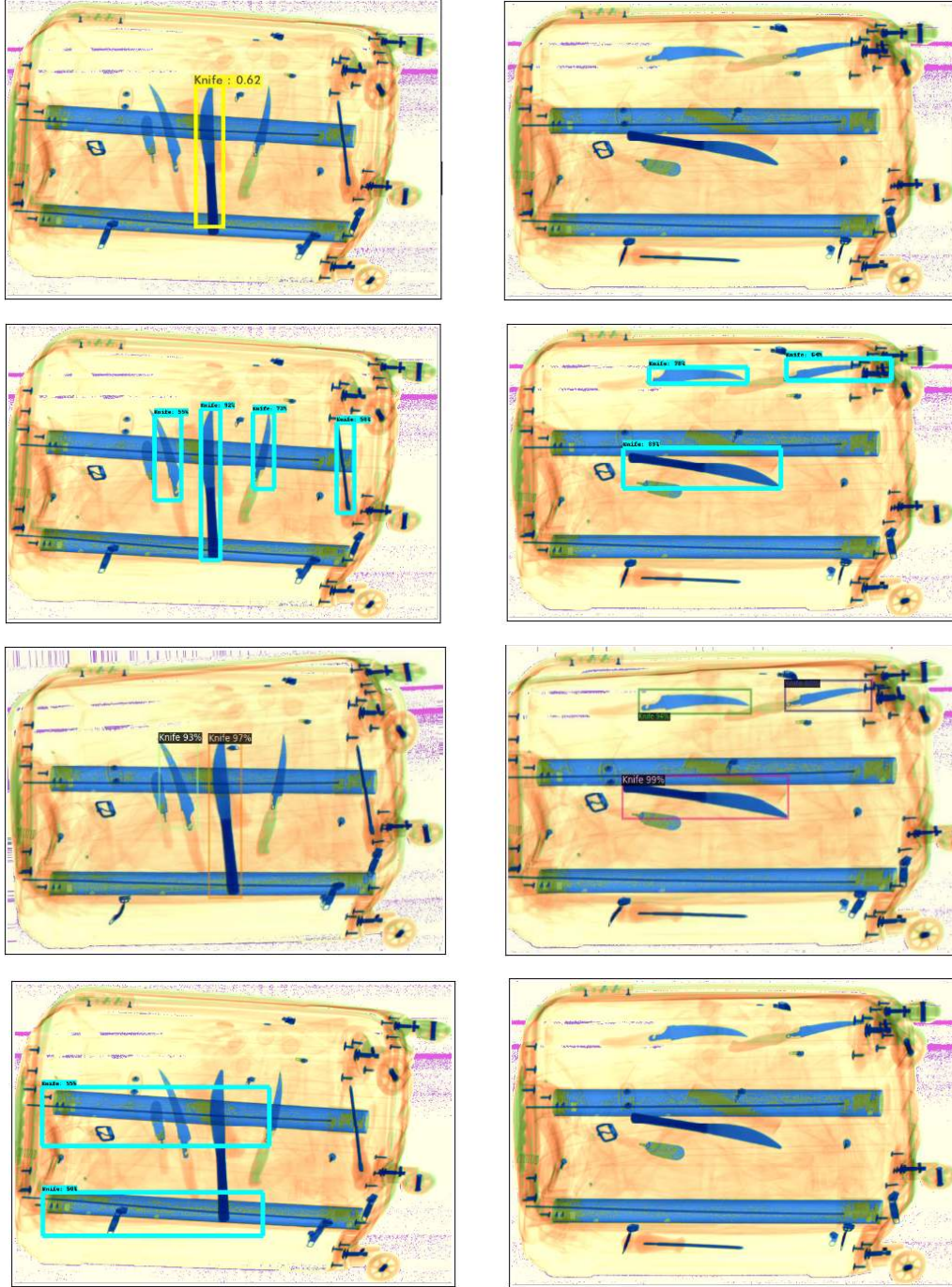
Şekil 4.8'de HUMS veri kümesine ait örneklerin kesin referansları, Şekil 4.9'da ise farklı modellerden elde edilen test sonuçları sunulmuştur. Şekilde sırası ile ilk satırda YOLOv3, ikinci satırda Faster RCNN (ResNet-101), üçüncü satırda Faster RCNN (ResNet-101+FPN) ve son satırda SSD modellerine ait test sonuçları gösterilmektedir. Buna göre ilk kolondaki HUMS veri kümesine ait örnek X-ray görüntüsü incelendiğinde bıçakların düz bir şekilde yerleştirildiği görülmektedir. Bu görüntü YOLOv3 ile test edildiğinde yerleştirilen bıçaklar içerisinde sadece ortadaki bıçağın %62 doğruluk ile tespit edilebilmiştir. İlk kolonun ikinci satırında bulunan ve ResNet-101 omurga ağı kullanılan Faster RCNN modeli ile yapılan testte ise bavul içerisine nizami bir şekilde yerleştirilen dört adet bıçaktan üçü tespit edilirken bavulun en sağ kısmında bulunan tornavida da %58 güven skoru ile bıçak olarak tespit edilmiştir. Doğru bir şekilde tespit edilen bıçaklar sırası ile %55, %92 ve %73 güven skorları ile tespit edilirken görüntünün en solunda bulunan küçük bıçak ise model tarafından tespit edilememiştir. Üçüncü satırın ilk kolonunda bulunan görüntüde ise ResNet-101+FPN omurga ağı kullanılan Faster RCNN modelinden alınan test sonucu sunulmuştur. Bu görüntüde ortada bulunan iki adet bıçaktan soldaki bıçak %93, sağdaki bıçak ise %98 güven skoru ile tespit edilirken iki adet bıçak ise model tarafından tespit edilememiştir. Son satırda bulunan ve SSD ile alınan tespitlerde ise bıçakların hiçbiri tespit edilemezken bavulun metalik parçaların-

dan üstteki kısım %55, alttaki kısım ise %50 güven skoru ile bıçak olarak algılanmıştır.

Şekil 4.9'un ikinci kolonunda bulunan tespitler incelendiğinde ilk satırda sunulan ve YOLOv3 ile test edilen görüntüdeki bıçakların hiçbirinin tespit edilemediği görülmüştür. İkinci satırdaki örnekte ise ResNet-101 omurga ağı kullanılan Faster RCNN modelinin görüntüde bulunan bıçaklardan üçünü tespit edebilirken küçük bıçağı tespit edemediği görülmüştür. Tespit edilen bıçaklardan sol üstte bulunan bıçak %78, sağ üstte bulunan bıçak %64 ve bavulun ortasında bulunan bıçak ise %89 güven skoru ile tespit edilmiştir. Benzer şekilde, üçüncü satırda bulunan ve ResNet-101+FPN omurga ağı kullanılan Faster RCNN tespitleri incelendiğinde görüntüdeki üç bıçağın tespit edilip küçük bıçağın tespit edilemediği görülmüştür. Tespit edilen bıçaklardan sol üstteki bıçak %94, sağ üstteki bıçak %80 ve bavulun ortasındaki bıçak ise %99 güven skoru ile tespit edilmiştir. Son satırda bulunan ve SSD ile alınan test örneğinde ise modelin hiçbir bıçağın tespitinde başarılı olamadığı görülmüştür. Özet olarak SIXray ile eğitimlerin sonucunda elde edilen sonuçların HUMS veri kümesi üzerindeki örnek test tespitleri karşılaştırıldığında ResNet-101+FPN omurga ağı kullanılan Faster RCNN modelinin diğer modellere göre en başarılı sonuçları verdiği görülmüştür.



Şekil 4.8. HUMS veri kümesine ait kesin referans içeren örnek görüntüler



Şekil 4.9. HUMS veri kümesine ait örnek tespitler dört satırda sırası ile YOLOv3, Faster RCNN (ResNet-101), Faster RCNN (ResNet-101+FPN) ve SDD modelleri için sunulmuştur

4.3 HUMS Veri Kümesi Üzerinde Yapılan Deneyler

Tez çalışmasının bir sonraki aşamasında ise hem SIXray veri kümesi üzerinde yapılan eğitimler sonucunda elde edilen ağırlıkların HUMS veri kümesine uygulanarak hem de

HUMS veri kümesindeki görüntülerin direkt eğitilmesi ile tehdit tespiti yapılması amaçlanmıştır. Bu kapsamda yapılan çalışmalar alt başlıklar şeklinde ilerleyen bölümlerde sunulmuştur.

4.3.1 Az Vuruşlu Öğrenmeye Ait Tehdit Tespiti Sonuçları

Az vuruşlu öğrenme, ikinci kısımda da açıklandığı gibi limitli sayıda görüntü ile öğrenmeyi hedefler. Çalışmanın bu kısmında da, SIXray veri kümesinin eğitimi sonucunda elde edilen en iyi ağırlıklar kullanılarak az sayıda yapay RGB görüntü içeren veri kümesindeki bıçak tespitlerinin öğrenilmesi hedeflenmiştir. Az vuruşlu öğrenme uygulanırken farklı sınıflar modele temel ve yeni kategoriler olmak üzere tanıtılır. Temel kategoriler, çok sayıda etiketli veriye sahip olan görüntüleri kapsayan kategorilerdir. Yeni kategoriler ise modelin temel kategoriler ile baştan eğitilmesi sonucu elde edilen ağırlıklarla az sayıda etiketli görüntüler kullanılarak öğrenilmesi amaçlanan sınıflardır. Çalışma kapsamında düşünüldüğünde, temel kategoriler SIXray veri kümesine ait kategoriler olmakla birlikte HUMS veri kümesine ait veriler ise farklı bir kategori ile modele tanıtılmıştır. Çizelge 4.3'te az vuruşlu öğrenmede kullanılan kategorilerin temel ve yeni kategoriler olarak dağılımı sunulmuştur. Temel kategoriler SIXray veri kümesine ait kategoriler olup bunlar *silah (gun)*, *bıçak (knife)*, *makas (scissors)*, *İngiliz anahtarı (wrench)* ve *pense (pliers)* sınıflarıdır. HUMS veri kümesinde bulunan bıçaklar ise modele tanıtılmak için *özel bıçak (KnifeCustom)* olacak şekilde etiketlenmiştir.

Temel kategori	Yeni kategori
Gun	KnifeCustom
Knife	-
Scissors	-
Wrench	-
Pliers	-

Çizelge 4.3. Az vuruşlu öğrenmede kullanılan temel ve yeni kategoriler

HUMS veri kümesinde bulunan ve tehdit içeren 100 adet yapay RGB görüntüden bir kısmında bıçak tehdidi olmayan görüntüler olduğu için bu görüntüler alınmamıştır. Ayık-

lama yapıldığında 80 adet görüntü kalmış ve geleneksel veri arttırımı yöntemi ile bu sayı 150'ye çıkarılmıştır. Veri arttırımı uygulanırken yapay RGB görüntülerin rotasyonunun değiştirilmesi, yakınlaştırma ayarlarıyla oynanması, yatay düzlemde ters çevrilmesi gibi yöntemler kullanılmıştır. Elde edilen 150 adet yapay RGB görüntüden 30 tanesi test için ayrılırken kalan 120 tanesi ise az vuruşlu öğrenmede farklı şekilde kullanılmak üzere eğitim için ayrılmıştır.

HUMS veri kümesindeki görüntülerin az vuruşlu öğrenme aşamasında limitli sayıda görüntü ile öğrenebilmesi için 120 adet görüntü K adet etiketli tehdit içerecek şekilde alt kümelere ayrılmıştır. Burada K , vuruş sayısını belirtmektedir. K vuruş için alt kümeler oluşturulurken hem temel hem de yeni sınıflara ait veriler kullanılır. Hem temel hem de yeni sınıflara ait verilerin kullanılmasının nedeni ise ince ayar aşamasında kullanılan kosinüs kutu benzerliğinin temel ve yeni kategoriler arasında bir benzerlik skoru elde ederek performansı arttırmayı amaçlamasıdır. Çalışmada alt kümeler sırası ile $K = 1, 3, 5, 10, 30$ olacak şekilde seçilmiştir. Örneğin, $K = 1$ durumunda her sınıf için sadece 1 adet tehdit nesnesi içeren görüntü ince ayar aşamasında kullanılmak üzere ayrılmıştır. Benzer şekilde diğer K vuruşları için de K adet tehdidi sağlayabilecek alt kümeler oluşturulmuştur. Veri kümesi temel ve yeni sınıflar olarak modele tanıtılıp K vuruş alt kümeleri de tanıtıldıktan sonra alt başlıklarda belirlenen adımlar uygulanarak az vuruşlu öğrenme gerçekleştirilmiştir.

4.3.1.1 Modelin Temel Eğitimi

Modelin temel eğitimi için 4.2.2.2 kısmında açıklanan ve SIXray veri kümesinin ResNet-101'in öznitelik piramit ağı (FPN) ile birlikte kullanıldığı omurga ağ (ResNet-101+FPN) tabanlı Faster RCNN ile eğitilmesi sonucu elde edilen en iyi ağırlıklar kullanılmıştır.

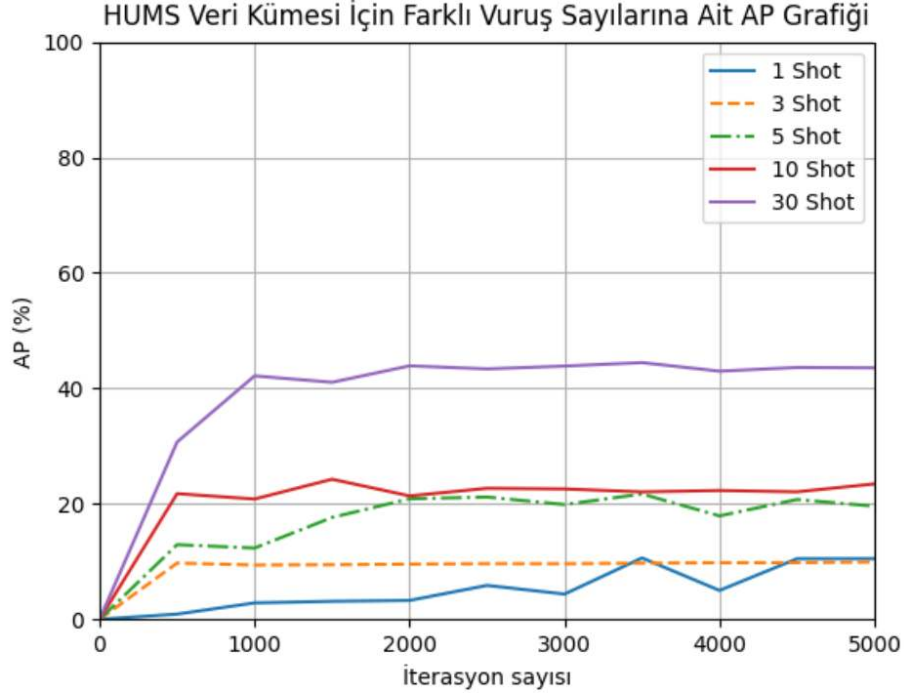
4.3.1.2 İnce Ayar ile Az Vuruşlu Öğrenme Aşaması

İnce ayar aşamasında ise temel eğitimde kullanılan ağırlıklar belirlenen K vuruş alt kümelerinin eğitimi için ince ayar aşamasında kullanılmıştır. Bu aşamada tüm K vuruş aşamaları için konfigürasyonda SIXray ile eğitilen en iyi ağırlıklar tanıtılmış ve omurga ağ kısmı tamamen dondurulmuştur. İnce ayar bölge öneri ağı (RPN) ve tam bağlantılı katmanlarda uygulanmıştır. Ancak tez çalışmasında kullanılan az vuruşlu öğrenme

aşamasının orijinal yayınında [56] sadece tam bağlantılı katman dondurulmuştur. Tez çalışmasında bölge öneri ağının da ince ayara katılmasının nedeni modelin yeni kategoriye daha iyi öğrenmesini amaçlamaktır.

Farklı vuruş sayıları için ince ayar yapılırken öğrenme oranı tüm vuruşlar için başlangıçta 0.01 olarak seçilmiştir. Benzer şekilde tüm vuruşlar için küme boyutu 16 olarak seçilmiştir. Tüm vuruşlar için ise maksimum adım sayısının 5000 olmasına karar verilmiştir. Şekil 4.10'da farklı vuruşlar için yapılan ince ayar sonucunda HUMS veri kümesindeki bıçaklara yani modele tanıtılan yeni sınıfa ait elde edilen kesinlik değeri (AP) grafiği sunulmuştur. Grafiğe bakıldığında $K = 1$ vuruş için eğitim süresince elde edilen maksimum AP değerinin %10.63 olarak hesaplandığı görülmüştür. Aynı grafikte $K = 3$ vuruş için yapılan ince ayar sonucu elde edilen eğri incelendiğinde ise her sınıf için 3 adet etiketli veri kullanıldığında sonucun 1 adet kullanılan duruma göre ince ayarın ilk adımlarında 2 katından fazla yükseldiği, ancak 3 vuruş için elde edilen kesinlik değerlerinin adım sayısı arttıkça stabil gitmesi nedeni ile son iterasyonlarda 1 vuruş ile benzer sonuçların alındığı gözlemlenmiştir. 5000 adım içerisinde $K = 3$ vuruş için alınan en yüksek AP değerinin son adımda %9.93 olarak kaydedildiği tespit edilmiştir. $K = 5$ vuruş için incelendiğinde ise ilgili eğrinin 5000 adım boyunca 4500. adımda en çok %21.75'e ulaştığı görülmüştür. $K = 10$ vuruş için incelendiğinde ise HUMS veri kümesindeki bıçakların en yüksek %24.26 kesinlik değeri ile tespit edilebildiği görülmüştür. Bu değere ince ayarın 1500. adımında ulaşılmıştır. Son denemede ise $K = 30$ vuruş denenmiştir. İnce ayarda kullanılan tüm sınıflara ait tehdit nesnelere göre en az üç kat arttığı için modelin sonucu da pozitif yönde etkilenmiştir. Grafik incelendiğinde $K = 30$ vuruşta, HUMS veri kümesindeki bıçaklar için elde edilen maksimum kesinlik değerinin %44.46 olduğu görülmüştür. Şekil 4.11'de az vuruşlu öğrenme denemeleri sonucunda en iyi sonuç veren 30 vuruş sayısı ile alınan tespit sonuçları sunulmuştur.

Şekil incelendiğinde sol üst köşedeki şekilde nizami bir şekilde dizilen bıçaklardan ortadaki iki bıçağın tespit edildiği görülmüştür. Ancak, HUMS veri kümesindeki tehditler *KnifeCustom* etiketi ile tanıtılmasına rağmen tespit örneğinde bıçaklardan tamamen mavi olanı %58 güven skoru ile *Knife* (bıçak) olarak tespit edilirken bu bıçağın yanındaki ise

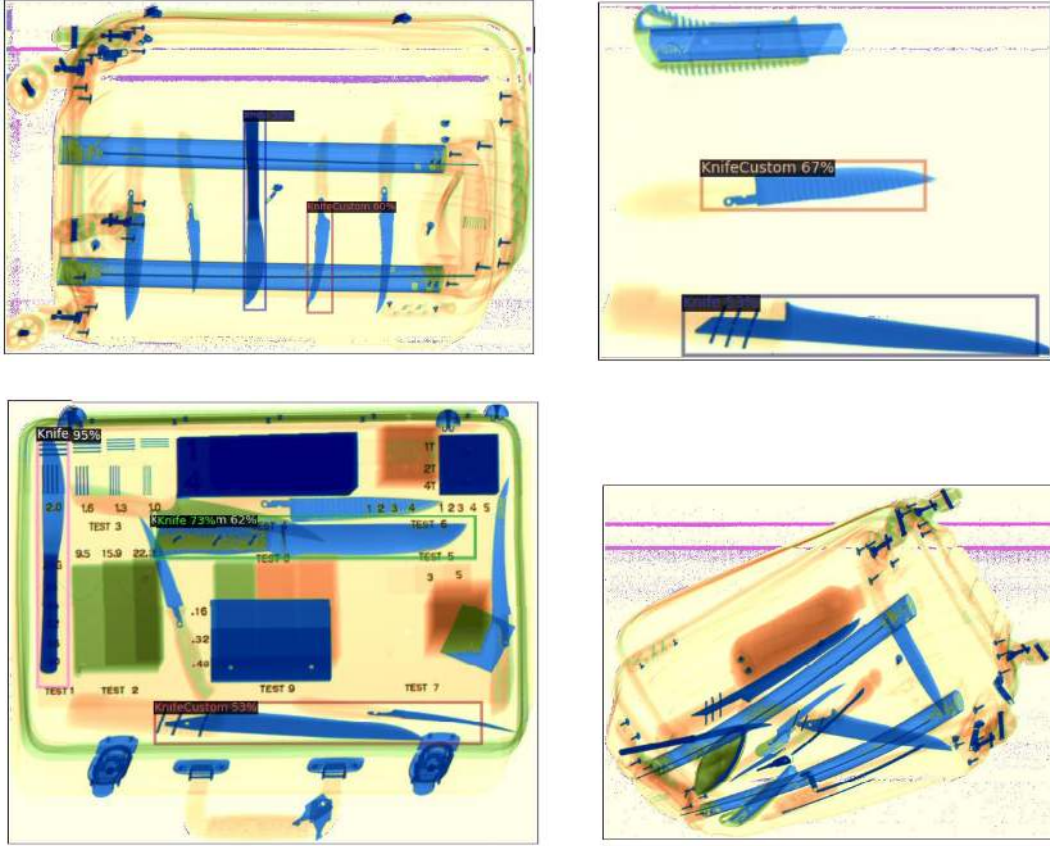


Şekil 4.10. HUMS veri kümesindeki bıçaklara ait $K = 1, 3, 5, 10$ ve 30 vuruş için elde edilen AP grafiği

%60 güven skoru ile *KnifeCustom* (özel bıçak) olarak tespit edilmiştir. Görüntüdeki diğer bıçaklar ise tespit edilememiştir. Şeklin sağ üst köşesinde ise bir adet modele tanıtılmayan maket bıçağı ve modele *KnifeCustom* etiketi ile tanıtılan iki adet bıçağa ait tespit sonucu verilmiştir. Bu sonuca göre ortadaki bıçak %67 güven skoru ile *KnifeCustom* olarak tespit edilirken alttaki bıçak ise %53 güven skoru ile *Knife* etiketi ile tespit edilmiştir. Sol alt köşedeki görüntü incelendiğinde ise bavulun en solunda dikey bir şekilde bulunan bıçak %95 güven skoru ile *Knife* olarak tespit edilirken yatay bir şekilde duran büyük bıçak ise %73 *Knife* ve %62 *KnifeCustom* olarak tespit edilmiştir. Görüntünün en altında bulunan bıçak ise %53 güven skoru ile *KnifeCustom* olarak tespit edilmiştir. Son olarak görüntünün sağ alt köşesinde bulunan bavuldaki bıçaklardan hiçbiri model tarafından tespit edilememiştir.

Alınan sonuçlar ve tespit örnekleri incelendiğinde az vuruşlu öğrenmede modelin *Knife* ve *KnifeCustom* sınıflarını karıştırdığı görülmektedir. Bunun nedeninin, HUMS veri kümesindeki bıçakların az vuruşlu öğrenme ile öğrenilebilmesi için modele yeni bir sınıf adı ile tanıtılması gerektiği olup HUMS veri kümesindeki bıçakların SIXray veri küme-

sindeki bıçaklarla benzerlik gösteren bir küme olduğu için model tarafından az vuruşlu öğrenme sonucunda yeterince ayırt edilememesi olarak düşünülmektedir.

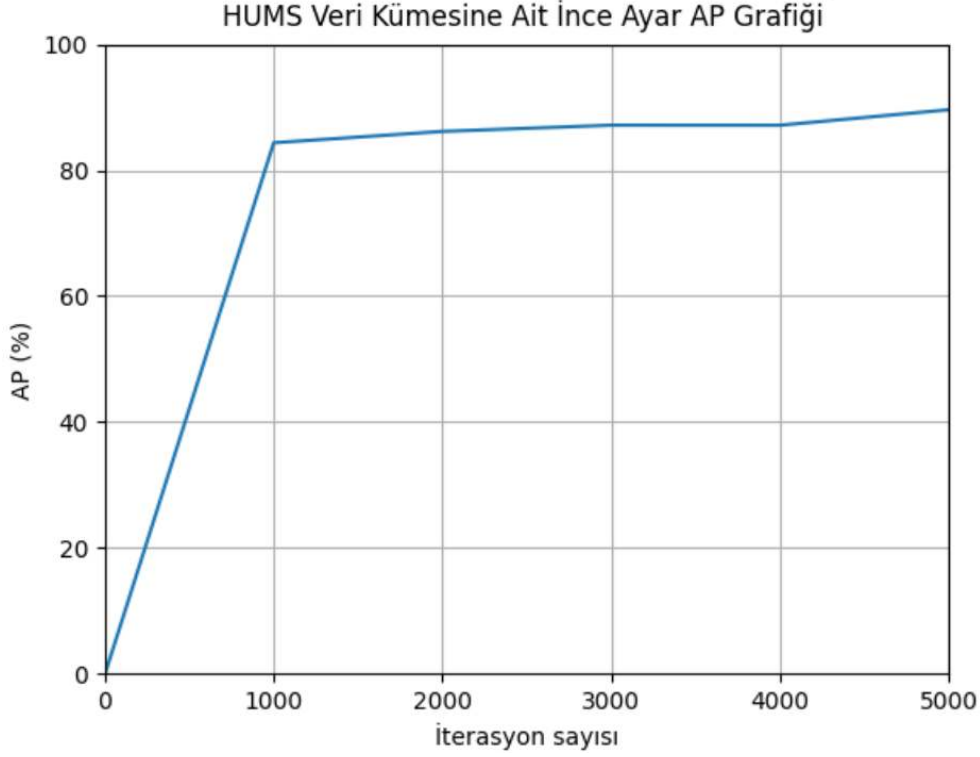


Şekil 4.11. Az vuruşlu öğrenme sonucu en iyi sonuçlar kullanılarak (30 vuruş) alınan tespit örnekleri

4.3.2 HUMS Veri Kümesindeki Yapay RGB Görüntülerin İnce Ayar ile Eğitimi

Tez çalışmasının bu adımında HUMS veri kümesindeki yapay RGB görüntüler üzerinde, SIXray veri kümesinin Faster RCNN (ResNet-101+FPN) ile eğitimi sonucu elde edilen en iyi ağırlıklar kullanılarak ince ayar yapılmıştır. Bu aşamada az vuruşlu öğrenme bölümünde açıklanan 150 adet yapay RGB görüntüden 120 adedi ince ayar için eğitimde kullanılırken kalan 30 tanesi ise test için ayrılmıştır. İnce ayar için konfigürasyon hazırlanırken az vuruşlu öğrenme ile benzer şekilde öğrenme oranı 0.01 ve maksimum adım sayısı ise 5000 olacak şekilde düzenlenmiştir. Küme boyutu ise 2 olarak seçilmiştir.

Şekil 4.12'de HUMS veri kümesindeki yapay RGB görüntülerde bulunan bıçak tehditlerinin SIXray veri kümesi ile elde edilen en iyi ağırlıklar kullanılarak ince ayar ile öğrenme

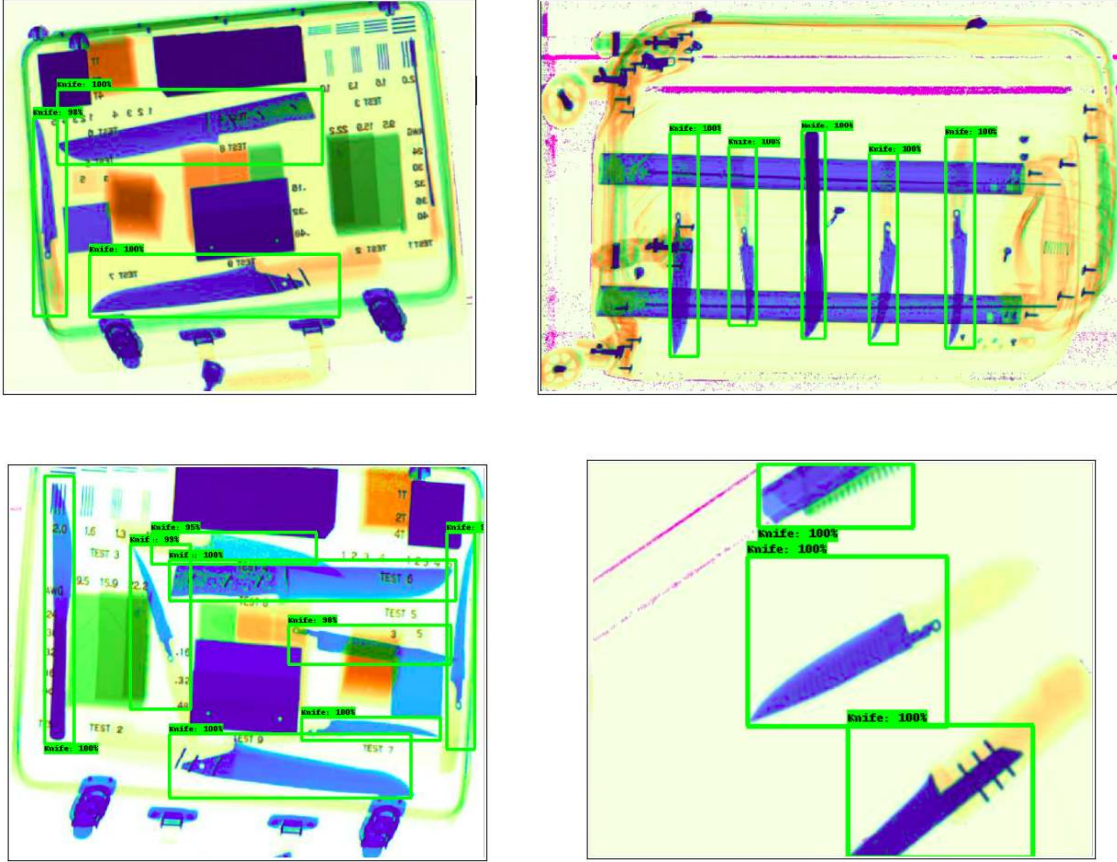


Şekil 4.12. HUMS veri kümesindeki yapay RGB görüntülerin SIXray ile eğitimler sonucu elde edilen en iyi ağırlıklarla ince ayarına ait AP grafiđi

aşamasında elde edilen kesinlik değeri (AP) ait grafik sunulmuştur. Grafiđe bakıldığında modelin ilk adımlarda %84.36 kesinlik değeri ile ince ayar işlemini yeterince tamamladığı ve bu adımdan sonra sonuçların belli bir miktar daha arttığı görülmektedir. İnce ayarın son adımında ise HUMS veri kümesindeki bıçaklar için %89.6 kesinlik değeri ulaşılmıştır.

Şekil 4.13'te ince ayar sonucu alınan en iyi ağırlıklarla test edilen örnek görüntüler sunulmuştur. Şekle bakıldığında sol üst köşede bulunan görüntüdeki bıçaklardan dikey olarak konumlandırılan bıçak %98 güven skoru ile tespit edilirken yatay şekilde duran bıçakların ikisinin de %100 güven skoru ile tespit edildiđi görülmüştür. Görüntünün sağ üst köşesinde bulunan ve nizami bir şekilde yerleştirilen bıçakların tümü %100 güven skoru ile tespit edilmiştir. Sol alt köşede bulunan görüntüde ise en solda dikey bir şekilde bulunan bıçağın %100 güven skoru ile tespit edildiđi görülürken bu bıçağın sağında yine dikey bir şekilde bulunan bıçak ise %98 güven skoru ile tespit edilmiştir. Yatay konumda bulunan bıçaklardan ise en üstte bulunan bıçak %95 ile tespit edilirken görüntünün

en sađında dikey bir Őekilde bulunan bıçak ise %98 gven skoru ile tespit edilmiŐtir. Grntde bulunan diđer bıçakların tm de %100 gven skoru ile tespit edilebilmiŐtir. Grntnn sađ alt kŐesinde bulunan alt grntde ise ortadaki ve alttaki bıçakların ikisinin de %100 gven skoru ile dođru bir Őekilde tespit edildiđi grlrken model en stte bulunan ve modele tanıtılmayan maket bıçađını da %100 gven skoru ile bıçak olarak algılamıŐtır.



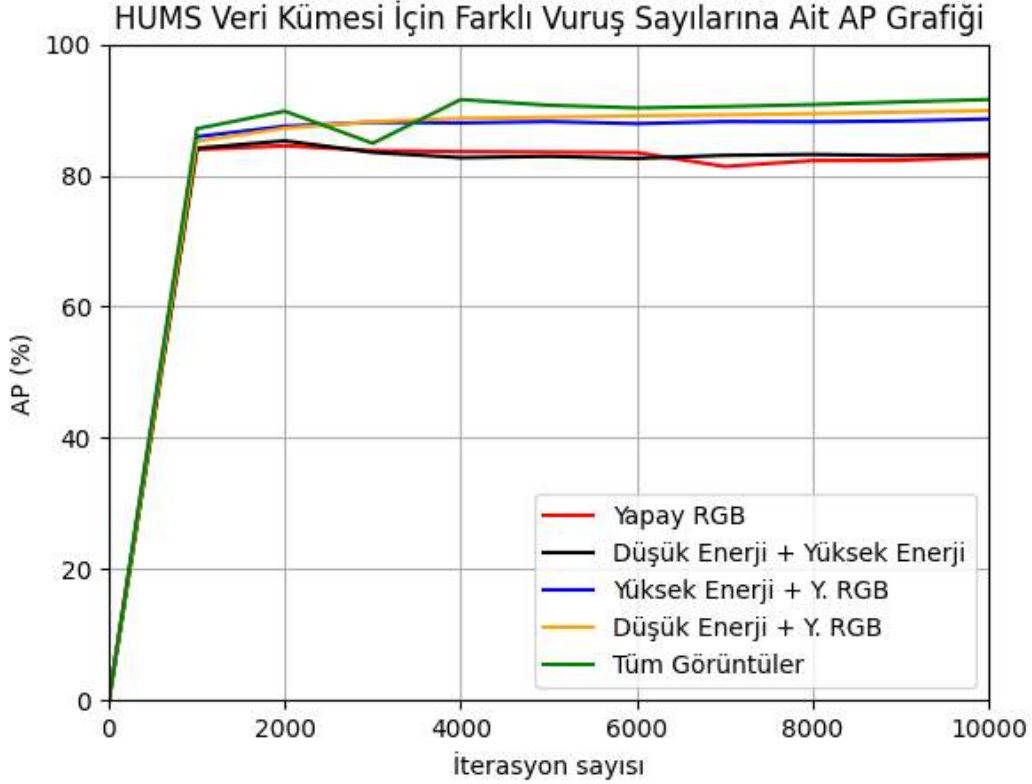
Őekil 4.13. SIXray veri kmesi ile elde edilen en iyi ađrılıklar kullanılarak yapılan ince ayar sonucu HUMS veri kmesine ait rnek tespitler

4.3.3 HUMS Veri Kmesine Ait Yksek ve DŐk Enerjili Grntler ile Yapay RGB Grntler Kullanılarak Yapılan Eđitimin Sonuçları

Tez alıŐmasının bu kısmında HUMS veri kmesine ait yapay RGB grntlerin yanında ift kanallı X-ray cihazından alınan dŐk ve yksek enerjili grntler de kullanılmıŐtır. nceki kısımda belirtildiđi gibi geleneksel veri arttırımı yntemi ile yapay RGB grntlerin sayısı toplamda 150 olacak Őekilde arttırılmıŐtır. DŐk ve yksek enerjili

görüntüler ise ayıklama sonucu her bir enerji seviyesi için 80 adet olmak üzere toplamda 160 adettir. HUMS veri kümesine ait yapay RGB görüntülerin yanında düşük ve yüksek enerjili görüntüler de kullanılarak yapılan çalışmalarda başlıca kombinasyonlar denenmiştir. Bunlar sırası ile yapay RGB görüntülerin yanında yüksek enerjili kanaldan alınan görüntülerin kullanılması ile yapılan eğitim, yapay RGB görüntülerin yanında düşük enerjili kanaldan alınan görüntülerin kullanılması ile yapılan eğitim, yapay RGB görüntüler ile düşük ve yüksek enerjili görüntülerin tamamının kullanılması ile yapılan eğitim, sadece yapay RGB görüntüler ile yapılan eğitim ve sadece düşük ve yüksek enerjili görüntüler ile yapılan eğitimidir. Tüm kombinasyonlar denenirken SIXray veri kümesinde en başarılı sonucu veren Faster RCNN (ResNet-101+FPN) modeli kullanılmış olup model her kombinasyon için baştan eğitilmiştir. Eğitimler yapılırken test görüntüsü için tüm kombinasyonlarda 30 adet yapay RGB görüntü kullanılmıştır. Kombinasyonların tümünde eğitim için başlangıç öğrenme oranı 0.01 olarak seçilmiştir. Küme boyutu ise 2 olarak seçilirken tüm eğitimler için maksimum iterasyon sayısı 10000 olarak seçilmiştir.

Şekil 4.14'te HUMS veri kümesindeki düşük ve yüksek enerjili X-ray görüntülerinin yapay RGB görüntüler ile birlikte kullanılarak Faster RCNN (ResNet-101+FPN) ile eğitilmesi sonucu bıçaklar için elde edilen kesinlik değerleri (AP) sunulmuştur. Grafiğe bakıldığında eğitimlerde 10000'inci iterasyon sayısı sonunda en iyi sonucun yaklaşık %91.6 ile yapay RGB görüntülerin yanında hem düşük hem de yüksek enerjili X-ray görüntülerinin kullanıldığı durumda alındığı görülürken yapay RGB görüntülerle birlikte sadece düşük enerjili veya sadece yüksek enerjili görüntülerin kullanıldığı durumda da elde edilen kesinlik değerinin çok değişmediği görülmüştür. Sırası ile yapay RGB görüntüler ile düşük enerjili görüntülerin kullanıldığı eğitimde maksimum %89.88 kesinlik değeri elde edilirken yapay RGB görüntüler ile yüksek enerjili görüntülerin eğitildiği durumda ise maksimum %88.6 kesinlik değerine ulaşılmıştır. HUMS veri kümesinde bulunan görüntülerden sadece yapay RGB görüntülerin eğitildiği durumda 10000 adım boyunca maksimum %84.51 kesinlik değerine ulaşılabilirdiği görülmüştür. Benzer şekilde sadece düşük ve yüksek enerjili görüntülerin tamamı kullanılıp yapay RGB görüntülerin eğitime katılmadığı durumda ise ortalama kesinlik değerinin en yüksek durumda %85.27'ye ulaşabildiği görülmektedir.



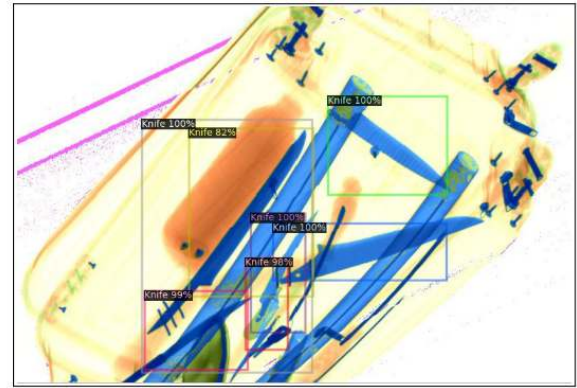
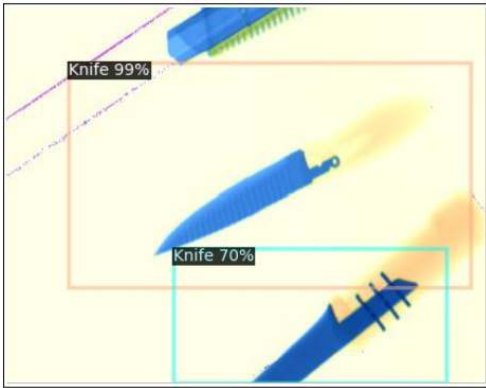
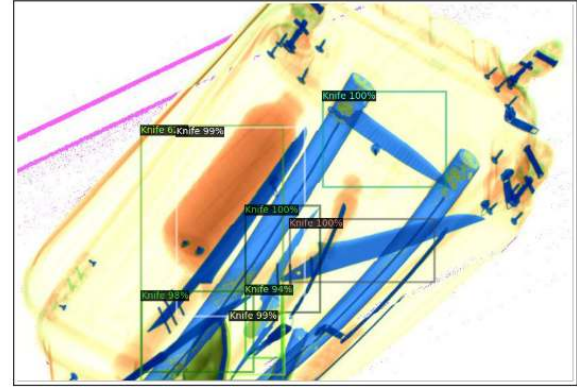
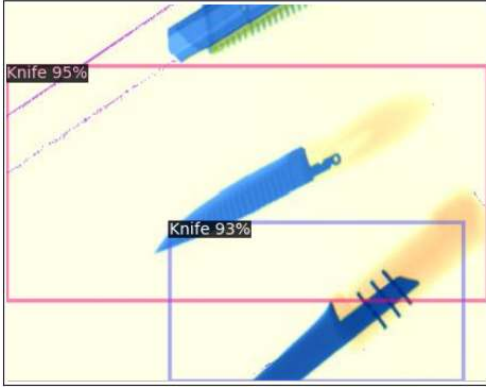
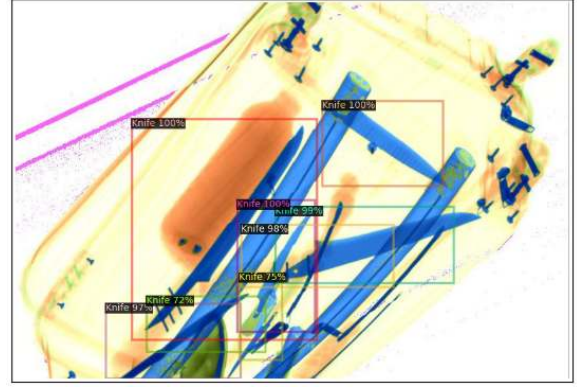
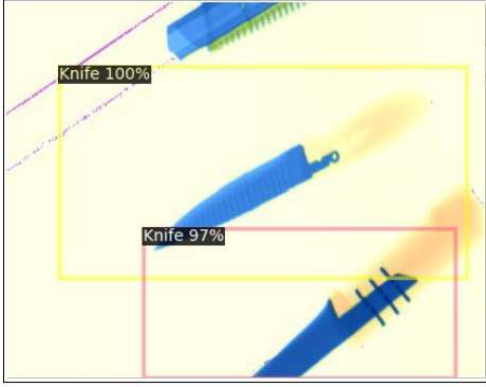
Şekil 4.14. HUMS veri kümesindeki görüntülerin farklı kombinasyonlarla eğitilmesi sonucu alınan AP değerleri (Y. RGB, Yapay RGB kısaltması için sunulmuştur)

Şekil 4.15'te veri kümesindeki görüntülerin tamamı kullanılarak uygulanan farklı kombinasyonlara ait tespit örnekleri verilmiştir. Şekilde hem tekli bıçaklara hem de bavul içerisinde karmaşık bir şekilde yerleştirilen bıçaklara ait tespitler sunulmuştur. Sırası ile şeklin sağ ve sol üst köşelerinde yapay RGB görüntüler ile yüksek enerjili görüntülerin birlikte eğitildiği duruma ait örnekler, şeklin sağ ve sol ortasında yapay RGB görüntüler ile düşük enerjili görüntülerin birlikte eğitildiği duruma ait örnekler, şeklin sol ve sağ alt köşesinde ise veri kümesindeki tüm görüntülerin eğitimi sonucu alınan en iyi ağırlıklara ait tespit örnekleri sunulmuştur. Şekil incelendiğinde ilk kolondaki görüntüde bulunan iki adet bıçağın yapay RGB ve yüksek enerjili görüntülerin eğitimi sonucu elde edilen ağırlıklar kullanılarak test edildiğinde üstteki bıçağın %100, alttaki bıçağın ise %97 güven skoru ile tespit edildiği görülmektedir. Aynı bıçaklar yapay RGB ve düşük enerjili görüntülerin eğitimi ile elde edilen ağırlıklar kullanılarak test edildiğinde üstteki bıçak %95 ve alttaki bıçak ise %93 güven skoru ile tespit edilmiştir. Son testte ise gö-

rüntü, tüm verilerin eğitildiği durumda elde edilen ağırlıklar ile test edilmiş olup üstteki bıçak %99 ve alttaki bıçak ise %70 güven skoru ile tespit edilmiştir. Karmaşık bıçakların bulunduğu ikinci kolondaki görüntülerde ise yapay RGB görüntülerin yüksek enerjili görüntüler ile birlikte eğitildiği duruma ait ağırlıklar kullanılarak alınan örnekte şişenin yanında bulunan büyük bıçak örneği ile bavulun metalik parçalarına dik bir şekilde yerleştirilen bıçağın %100 güven skoru ile tespit edildiği görülmüştür. Bavulun ortasında karmaşık bir şekilde bulunan dört bıçak ise %100, %99, %97 ve %75 güven skorları ile tespit edilmiştir. Ancak %99 güven skoru ile tespit edilen bıçak için ayrıca %98 güven skoru ile başka bir tespit daha yapılmıştır. Bu tespit, model için bir yanlış pozitif olarak yorumlanmaktadır. Benzer şekilde, %97 güven skoru ile tespit edilen bıçak örneği için ayrıca %72 güven skorlu yanlış pozitif bir tespit daha yapılmıştır. Yapay RGB görüntüler ile düşük enerjili görüntülerin birlikte eğitildiği durumda ise karmaşık örnekteki şişenin yanında bulunan büyük bıçak %99 güven skoru ile tespit edilirken aynı bıçak için %62 güven skoruna sahip yanlış pozitif bir tespit daha yapılmıştır. Bavulun metalik parçalarına dik yerleştirilen bıçak ise bu durumda da %100 güven skoru ile tespit edilebilmiştir. Bavulun ortasında bulunan dört bıçaktan ikisi %100, diğerleri ise %98 ve %94 güven skorları ile tespit edilmiştir. Son tespit örneğinde ise tüm veriler ile yapılan eğitim sonucunda alınan ağırlıklar kullanıldığında şişenin yanında bulunan bıçağın %100 güven skoru ile tespit edildiği görülürken modelin aynı bıçağa ait %82 güven skorlu bir yanlış pozitif tespit daha yaptığı görülmüştür. Bavulun metalik parçasına dik bir şekilde konumlandırılan bıçağın da %100 güven skoru ile tespit edildiği gözlemlenmiştir. Bavulun ortasında karmaşık bir şekilde bulunan dört bıçaktan ikisinin %100 güven skoru ile tespit edildiği görülürken kalan bıçakların ise %99 ve %98 güven skoru ile tespit edilebildiği görülmektedir.

Şekil 4.16'da sağ ve sol üst kısımda sadece düşük ve yüksek enerjili görüntülerin eğitimi sonucu alınan örnek tespitler, sağ ve sol alt kısımda ise sadece yapay RGB görüntülerin eğitimi sonucu alınan örnek tespitler verilmiştir. Tespitler incelendiğinde sol üst köşede bulunan görüntüdeki bıçaklardan ikisinin de %99 güven skoru ile tespit edildiği görülürken alttaki bıçak için fazladan bir tespit daha yapılmıştır. Bu tespit model için bir yanlış pozitif olarak değerlendirilmektedir. Aynı görüntünün yapay RGB görüntülerin eğitimi sonucu alınan tespitinde ise üstteki bıçağın %99 güven skoru ile tespit edildiği

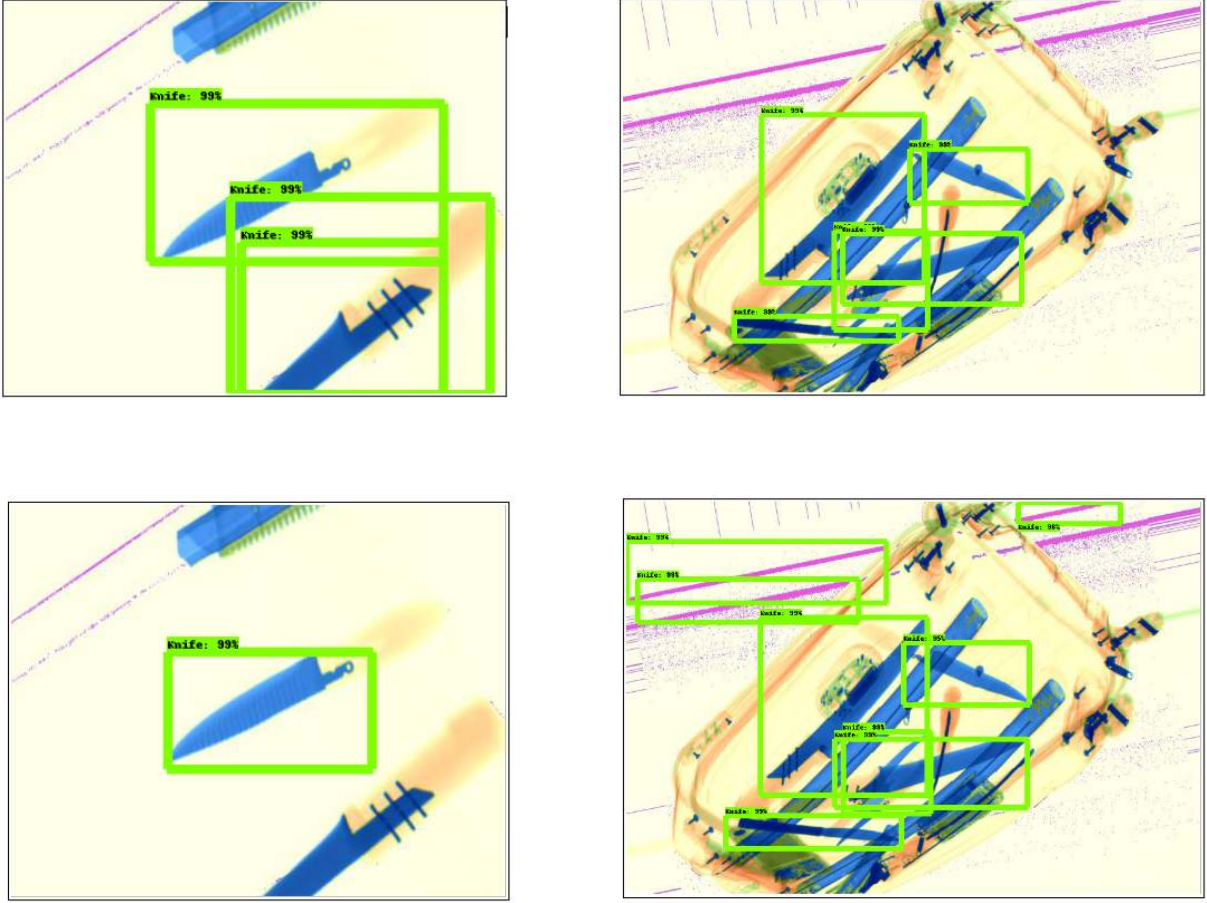
görülürken alttaki bıçak model tarafından tespit edilememiştir.



Şekil 4.15. Yapay RGB görüntüler ile birlikte düşük ve yüksek enerjili görüntülerin farklı kombinasyonlar ile eğitilmesi sonucu alınan örnek tespitler

Sağ kolondaki karmaşık görüntü incelendiğinde ise sadece düşük ve yüksek enerjili görüntüler kullanılarak yapılan eğitim sonucu ile alınan tespitte görüntüde bulunan altı

bıçaktan beşinin de %99 güven skoru ile tespit edildiği görülürken sadece yapay RGB görüntüler ile yapılan eğitim sonucu kullanılarak alınan tespitte ise beş bıçaktan dördünün %99, beşinci ve bavulun metalik kısımlarına dik bir şekilde yerleştirilen bıçağın ise %95 güven skoru ile tespit edildiği görülmüştür. Ancak bu örnekte bıçaklar dışında görüntüde bulunan çizgilerin de bıçak gibi algılandığı görülmüştür. Bu tespitlerin her biri model için birer yanlış pozitif olarak değerlendirilmektedir.



Şekil 4.16. Sadece yapay RGB görüntüler ile sadece düşük ve yüksek enerjili görüntülerin eğitimi sonucu alınan tespitler

4.3.3.1 HUMS Veri Kümesi Üzerinde Alınan Sonuçların Karşılaştırılması

Tez çalışmasının bu bölümünde HUMS veri kümesi üzerinde uygulanan yöntemler bir çizelgede sunulmuş ve sonuçlar karşılaştırılmıştır. Çizelge 4.4'te, HUMS bıçak sınıfı için uygulanan tüm yöntemlerden elde edilen kesinlik değerleri (AP) verilmiştir. Yapılan

denemelerde HUMS veri kümesinde az sayıda görüntü bulunduğu için bu görüntülerin 30 tanesi test verisi olarak kullanılmıştır. Çizelge incelendiğinde az vuruşlu öğrenme sonucu elde edilen kesinlik değerlerinin diğer yöntemlere kıyasla daha düşük olduğu görülmektedir. Bunun nedeni, az vuruşlu öğrenme kapsamında sadece çizelgede belirtilen sayıda bıçak nesnesi ile ince ayar yapılmasıdır. Az vuruşlu öğrenmenin farklı vuruşlar için denemelerinden elde edilen sonuçlar karşılaştırıldığında en çok vuruş olan 30 vuruş durumunda en iyi sonucun %44.46 kesinlik değeri olarak alındığı gözlemlenmektedir.

Az vuruşlu öğrenme dışında uygulanan yöntemler incelendiğinde ise SIXray veri kümesinden elde edilen sonuç ile yapılan ince ayar ve HUMS veri kümesine ait yapay RGB görüntüler ile düşük ve yüksek enerjili görüntülerin birlikte kullanılma durumlarından elde edilen sonuçların birbirlerine çok yakın olduğu görülmüştür. Buna göre, en iyi sonucun yapay RGB görüntüler ile düşük ve yüksek enerjili görüntülerin tamamının eğitildiği durumda %91.57 ile alınmıştır. Bu sonucu %89.88 kesinlik değeri ile yapay RGB görüntülerin düşük enerjili görüntüler ile birlikte eğitilmesiyle alınan sonuçlar takip etmiştir. Yapay RGB görüntüler ile yüksek enerjili görüntülerin birlikte eğitilmesi ile alınan sonuçlar ise %88.60 kesinlik değerine ulaşmıştır. Benzer şekilde yapılan deneylerden biri olan sadece düşük ve yüksek enerjili görüntülerin eğitildiği durumda ise %85.27 kesinlik değerine ulaşılırken başka bir deney olan sadece yapay RGB görüntülerin eğitildiği durumda ise kesinlik değeri %84.51'e ulaşmıştır. SIXray ile alınan sonuçlar kullanılarak yapılan ince ayar çalışmasında ise yapay RGB görüntüler ile düşük görüntülerin eğitimi sonucu alınan performans ile neredeyse aynı olan %89.60 kesinlik değerine ulaşılmıştır. HUMS veri kümesinden elde edilen sonuçlar genelleştirildiğinde, çift kanallı X-ray taraması ile elde edilen farklı enerji seviyesindeki görüntüler ile renklendirilmiş görüntülerin birlikte kullanıldığı durumda görüntü sayısı kullanılan farklı enerji seviyesindeki görüntüler ile birlikte artacağından eğitim sonucunda tatmin edici sonuçlar elde edilebilmektedir. Bunun dışında, eğer elimizde büyük bir X-ray veri kümesi mevcutsa bu kümenin eğitimi ile elde edilen sonuçlar az sayıda veri içeren başka bir X-ray veri kümesine aktarılarak X-ray görüntülerinde tehdit tespiti daha başarılı bir şekilde yapılabilmektedir.

Çalışma kapsamında HUMS veri kümesi üzerinde yapılan deneyler incelendiğinde lite-

Yöntem	Eğitimde Kullanılan Veri Kümeleri	HUMS Bıçak Verileri Test Sonuçları AP (%)
Az Vuruşlu Öğrenme (1 Vuruş)	Temel Eğitim: SIXray İnce Ayar: HUMS	10.63
Az Vuruşlu Öğrenme (3 Vuruş)	Temel Eğitim: SIXray İnce Ayar: HUMS	9.93
Az Vuruşlu Öğrenme (5 Vuruş)	Temel Eğitim: SIXray İnce Ayar: HUMS	21.75
Az Vuruşlu Öğrenme (10 Vuruş)	Temel Eğitim: SIXray İnce Ayar: HUMS	24.26
Az Vuruşlu Öğrenme (30 Vuruş)	Temel Eğitim: SIXray İnce Ayar: HUMS	44.46
İnce Ayar	Temel Eğitim: SIXray İnce Ayar: HUMS	89.60
Y. RGB Görüntülerle Eğitim	HUMS	84.51
Yüksek+Düşük Enerjili Görüntülerle Eğitim	HUMS	85.27
Y. RGB+Yüksek Enerjili Görüntülerle Eğitim	HUMS	88.60
Y. RGB+Düşük Enerjili Görüntülerle Eğitim	HUMS	89.88
Eğitim Görüntülerinin Tümü ile Eğitim	HUMS	91.57

Çizelge 4.4. HUMS veri kümesindeki bıçakların tespitinde uygulanan yöntemlerin karşılaştırılması (Y. RGB, Yapay RGB için kısaltma olarak sunulmuştur.)

ratürde sunulan X-ray veri kümelerinde yapılan çalışmalardan elde edilen bıçak başarısına kıyasla tatmin edici sonuçlar alındığı görülmüştür. Örneğin, Çizelge 3.8'de sunulan GDXray [22] veri kümesindeki bıçaklar Faster RCNN yöntemi ile en yüksek %76.2 kesinlik değeri ile tespit edilmiştir. Ancak, HUMS veri kümesindeki bıçakların yerleşimi

GDXray veri kümesine kıyasla daha karmaşıktır. HUMS veri kümesinde bıçaklar birbirini engelleyecek şekilde yerleştirilmiştir. Bıçak içeren başka bir veri kümesi olan Xu et al.'de [66] ise YOLOv3 ile deneme yapılmış ve bıçaklar maksimum %66 kesinlik değeri ile tespit edilmiştir. NS-100-L veri kümesindeki [61] bıçaklar ise %78 kesinlik değeri ile tespit edilebilmiştir. DBF3 [14] veri kümesindeki bıçaklar ise en yüksek %85 kesinlik değeri ile tespit edilirken SIXray [67] için bu değer %81 ve DBF6 [63] için ise %73.2 olarak sunulmuştur. Tüm bu sonuçlar ile kıyaslandığında HUMS veri kümesindeki bıçakların maksimum %91.57 ortalama kesinlik değeri ile tespit edilerek diğer bıçak içeren veri kümelerine kıyasla başarılı bir sonuca ulaştığı görülmüştür. HUMS veri kümesi üzerinde elde edilen en iyi sonucun detaylı bir analizinde ise test verilerinde bulunan toplam 81 adet bıçak tehdidi içerisinde 79 tane bıçağın model tarafından tespit edilirken 2 tane bıçağın ise tespit edilemediği gözlemlenmiştir. Tespit edilen 79 bıçak model için doğru pozitifken tespit edilemeyen 2 bıçak ise yanlış negatif olarak değerlendirilmektedir. Ek olarak modelin test sonucunda 9 adet bıçağı yanlış pozitif olarak tespit ettiği görülmüştür. Bu analiz sonucunda ise modelin bıçak tehditlerini kaçırma oranının düşük olduğu görülmüştür. Bu yorumun, HUMS veri kümesi ile yapılan bıçak tespit çalışmaları ile literatürde sunulan X-ray veri kümelerindeki bıçak tespiti çalışmalarının karşılaştırmalarına katkı sağlayabileceği düşünüldüğü için bu kısımda paylaşılmıştır.

5. SONUÇLAR

Bu tezde, son zamanlarda önemli bir çalışma alanı olan ve aynı zamanda bilgisayarlı görü çalışma alanının alt dalı olan nesne tespiti uygulamaları X-ray görüntülerinde tehdit tespiti için uygulanmıştır. Çalışmada öncelikle Hacettepe Teknokent'te bulunan MS Spektral Savunma Sanayi A.Ş. bünyesinde bulunan çift kanallı X-ray tarayıcıdan toplamda 300 adet bıçak tehdidi içeren görüntüler alınmıştır. Görüntüler alınırken hem düşük enerjili, hem yüksek enerjili hem de farklı enerji seviyelerine ait görüntüler kullanılarak renklendirilmiş X-ray görüntüleri toplanmıştır. Toplanılan veri kümesi HUMS veri kümesi olarak adlandırılmıştır. Daha sonra literatürdeki popüler nesne tespiti algoritmaları araştırılmış ve bu algoritmaların X-ray görüntülerinde uygulamaları da incelenmiştir. HUMS veri kümesindeki görüntülerin az sayıda olması nedeni ile ayrıca literatürde sunulan X-ray veri kümeleri araştırılmış olup açık kaynaklı sunulan ve içerisinde silah, bıçak, makas, ingiliz anahtarı ve pense tehdit sınıflarına ait çok sayıda örnek bulunduran geniş bir X-ray veri kümesi olan SIXray veri kümesi kullanılarak çoklu sınıf X-ray tehdit tespiti yapılmış ve bu sonuçlar HUMS veri kümesine de aktarılarak az sayıda görüntü barındıran X-ray veri kümesinde bıçak tespiti yapılmıştır. Ayrıca, HUMS veri kümesindeki farklı enerji seviyesine ait görüntüler kullanılarak bıçak tespiti için çeşitli denemeler yapılmıştır. Yapılan tüm denemeler hem SIXray hem de HUMS veri kümeleri için karşılaştırılmıştır.

Çalışmanın ilk aşamasında YOLOv3, Faster RCNN ve SSD nesne tespiti algoritmaları SIXray veri kümesi üzerinde uygulanmıştır. Algoritmalarından YOLOv3 kendi omurga ağı olan Darknet-53'ü kullanırken Faster RCNN için hem ResNet-101 hem de ResNet-101 ile birlikte kullanılan FPN (ResNet-101+FPN) omurga ağları kullanılmış ve iki ağ için de sonuçlar alınmıştır. SSD için de ResNet-101+FPN omurga ağı kullanılmıştır. SIXray veri kümesi üzerinde eğitilen algoritmalar karşılaştırıldığında en iyi sonucun %88.3 ortalama kesinlik değeri ile Faster RCNN (ResNet-101+FPN) modeliyle alındığı tespit edilmiştir. SIXray ile yapılan eğitimler sonucunda alınan en iyi sonuçlar HUMS X-ray veri kümesi üzerinde de test edilmiştir. Buradaki amaç ise iki veri kümesi arasındaki alan farkının test sonucunu nasıl etkilediğini araştırmak olmuştur. HUMS veri kümesi üzerinde yapılan testler karşılaştırıldığında da en iyi performansın Faster RCNN (ResNet-101+FPN)

ile elde edildiđi gözlemlenmiştir.

Çalışmanın ikinci aşamasında ise HUMS veri kümesindeki bıçakların tespiti için çalışmalar yapılmıştır. Bu kapsamda ilk aşamada SIXray veri kümesinde hem bıçaklar hem de modelin genelinde en başarılı sonucu veren Faster RCNN algoritması uygulanmıştır. Uygulamalara az vuruşlu öğrenme yöntemi ile başlanmıştır. Az vuruşlu öğrenmede SIXray veri kümesinden elde edilen en iyi sonuçlar kullanılmış ve bu sonuçlar ince ayar yapılarak HUMS veri kümesine aktarılmıştır. Az vuruşlu öğrenmede sırası ince ayar aşamasında sırası ile 1, 3, 5, 10 ve 30 vuruş denenmiş olup en iyi sonuç %44.46 ile 30 vuruşta alınmıştır. HUMS veri kümesinde az vuruşlu öğrenmenin denenmesindeki amaç ise az sayıda görüntü ile HUMS veri kümesinde bıçak tespiti yapabilmektir. Az vuruşlu öğrenmeden sonra ise SIXray veri kümesinden elde edilen en iyi sonuçlar ince ayar ile HUMS veri kümesine aktarılmış olup bu aşamada HUMS veri kümesindeki bütün RGB görüntüler kullanılmıştır. İnce ayarda HUMS veri kümesindeki renkli görüntülerin tümü kullanıldığında sonucun %89.60 kesinlik değerine kadar yükseldiđi görülmüştür. Bu aşamada büyük bir X-ray veri kümesi ile elde edilen sonuçlar kullanılarak az sayıda görüntü içeren X-ray veri kümesindeki bıçakların başarılı bir şekilde tespit edilebildiđi saptanmıştır.

HUMS veri kümesi üzerinde daha sonra RGB görüntüler ile düşük ve yüksek enerjili görüntüler farklı kombinasyonlar şeklinde bir arada kullanılmış ve denenilen her kombinasyon Faster RCNN (ResNet-101+FPN) ile baştan eğitilmiştir. Bu aşamada ilk kombinasyonda öncelikle RGB görüntüler ile yüksek enerjili görüntüler eğitilmiş ve tespit örnekleri alınmıştır. Daha sonra RGB görüntüler ile düşük enerjili görüntülerin eğitim sonucu elde edilmiştir. Bir başka denemede ise görüntülerin tümü algoritmaya beslenmiş ve sonuçlar alınmıştır. Bu denemelerin dışında HUMS veri kümesinde bulunan RGB görüntüler direkt eğitilmiştir. Son denemede ise sadece düşük ve yüksek enerjili görüntüler kullanılmıştır. Denenen beş farklı kombinasyondan en iyi sonucu %91.57 ortalama kesinlik değeri ile görüntülerin tümü kullanılarak yapılan eğitim vermiş olup diğer kombinasyonlar ise yüksek enerjili görüntülerin kullanıldığı durumda %88.60, düşük enerjili görüntülerin kullanıldığı durumda %89.88, sadece RGB görüntülerin kullanıldığı durumda %84.51 ve sadece düşük ve yüksek enerjili görüntülerin kullanıldığı durumda

ise %85.21 ortalama kesinlik deęerine ulařmıřtır. Bylece, ift kanallı X-ray tarayıcıdan elde edilen farklı enerji seviyelerine ait bilgiler kullanılarak HUMS veri kmesindeki bıçakların başarılı bir řekilde tespit edilebildięi grlmřtr.

KAYNAKLAR

- [1] Akcay, S., Breckon, T., P., *An evaluation of region based object detection strategies within X-ray baggage security imagery*, 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 1337-1341, doi: 10.1109/ICIP.2017.8296499.
- [2] Dai, J., Li, Y., He, K., Sun, J., *R-fcn: Object detection via region-based fully convolutional networks*, In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), 2016, pp. 379–387.
- [3] He, K., Zhang, X., Ren, S., Sun, J., *Deep Residual Learning for Image Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [4] Liu, S., Deng, W., *Very deep convolutional neural network based image classification using small training sample size*, 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 730-734, doi: 10.1109/ACPR.2015.7486599.
- [5] Liang, K., J., Heilmann, G., Gregory, C., Diallo, S., O., Carlson, D., Spell, G., P., Sigman, J., B., Roe, K., Carin, L., *Automatic threat recognition of prohibited items at aviation checkpoint with x-ray imaging: a deep learning approach*, Proc. SPIE 10632, Anomaly Detection and Imaging with X-Rays (ADIX) III, 1063203 (27 April 2018), doi: 10.1117/12.2309484
- [6] Liang, K., J., Sigman, J., B., Spell, G., P., Strellis, D., Chang, W., Liu, F., Mehta, T., Carin, L., *Toward Automatic Threat Recognition for Airport X-ray Baggage Screening with Deep Convolutional Object Detection*, ArXiv, 2019, abs/1912.06329.
- [7] Szegedy, C., Ioffe., S., Vanhoucke, V., Alemi, A., *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17)., AAAI Press, 2017, pp. 4278–4284.
- [8] Steitz, J., O., Saeedan, F., B., Roth, S., *Multi-view X-Ray R-CNN*, In German Conference on Pattern Recognition (GCPR), 2018, pp 153-168.
- [9] Liu, J., Leng, X., Liu, Y., *Deep Convolutional Neural Network Based Object Detector for X-Ray Baggage Security Imagery*, 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 1757-1761, doi: 10.1109/ICTAI.2019.00262.
- [10] Liu, Z., Li, J., Shu, Y., Zhang, D., *Detection and Recognition of Security Detection Object Based on Yolo9000*, 2018 5th International Conference on Systems and Informatics (ICSAI), 2018, pp. 278-282, doi: 10.1109/ICSAI.2018.8599420.
- [11] Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P., *Focal Loss for Dense Object Detection*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, 1 Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

- [12] Cui, Y., Oztan, B., *Automated firearms detection in cargo x-ray images using RetinaNet*, Anomaly Detection and Imaging with X-Rays (ADIX) IV, SPIE, 2019, pp. 105-115, doi: 10.1117/12.2517817.
- [13] He, K., Gkioxari, G., Dollár, P., Girshick, P., *Mask R-CNN*, 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.
- [14] Gaus, Y. F. A., Bhowmik, N., Akcay, S., Garcia, P., M., Barker, J., W., Breckon, T., *Evaluating the Transferability and Adversarial Discrimination of Convolutional Neural Networks for Threat Object Detection and Classification within X-Ray Security Imagery*, 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 420-425, doi: 10.1109/IJCNN.2019.8851829.
- [15] Saavedra, D., Banerjee, S., Mery, D., *Detection of threat objects in baggage inspection with X-ray images using deep learning*, Neural Computing and Applications, vol. 33, pp. 1-17, 2021, doi: 10.1007/s00521-020-05521-2.
- [16] Tian, Z., Shen, C., Chen, H., Tong, H., *FCOS: Fully Convolutional One-Stage Object Detection*, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9626-9635, doi: 10.1109/ICCV.2019.00972.
- [17] Wei, Y., Tao, R., Wu, Z., Ma, Y., Zhang, L., Liu, X., *Occluded Prohibited Items Detection: An X-ray Security Inspection Benchmark and De-occlusion Attention Module*, Proceedings of the 28th ACM International Conference on Multimedia , 2020, pp. 138-146, doi: 10.1145/3394171.3413828.
- [18] Bolya, D., Zhou, C., Xiao, F., Lee, Y., J., *Yolact: Real-time instance segmentation*, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9156-9165, doi: 10.1109/ICCV.2019.00925.
- [19] Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C., C., Lin, D., *CARAFE: Content-Aware ReAssembly of FEatures*, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 3007-3016, doi: 10.1109/ICCV.2019.00310.
- [20] Cai, Z., Vasconcelos, N., *Cascade r-cnn: High quality object detection and instance segmentation*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 5, pp. 1483-1498, 1 May 2021, doi: 10.1109/TPAMI.2019.2956516.
- [21] Bhowmik, N., Gaus, Y., F., A., Breckon, T., *On the impact of using X-ray energy response imagery for object detection via Convolutional Neural Networks*, 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 1224-1228, doi: 10.1109/ICIP42928.2021.9506608.
- [22] Mery, D., Rizzo, V., Zscherpel, U., Mondragon, G., Lillo, I., Zuccar, I., Lobel, H., Carrasco, M., *GDXray: The database of X-ray images for nondestructive testing*, Journal of Nondestructive Evaluation, vol. 34, no. 4, 2015.

- [23] Zou, Z., Shi, Z., Guo, Y., Ye, J., *Object Detection in 20 Years: A Survey*, ArXiv, 2019, abs/1905.05055.
- [24] Akcay, S., Breckon, T. P., *Towards Automatic Threat Detection: A Survey of Advances of Deep Learning within X-ray Security Imaging*, Pattern Recognition, Elsevier, vol. 122, no. 108245, p. 108245, 2022, doi: 10.1016/j.patcog.2021.108245.
- [25] Zhao, Z., Xu, S., *Object Detection with Deep Learning: A Review*, in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [26] Viola, P., Jones, M., *Rapid object detection using a boosted cascade of simple features*, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.
- [27] Lee, S., *Understanding face detection with the viola-jones object detection framework*, Medium, [Online]. Available: <https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>. [Accessed: 07-Jan-2022].
- [28] LeCun, Y., et al, *Backpropagation Applied to Handwritten Zip Code Recognition*, in Neural Computation, vol. 1, no. 4, pp. 541-551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.
- [29] Kachouane, M., Sahki, S., Lakrouf, M., Ouadah, N., *HOG based fast human detection*, 2012 24th International Conference on Microelectronics (ICM), 2012, pp. 1-4, doi: 10.1109/ICM.2012.6471380.
- [30] Felzenszwalb, P. F., Girshick, R., McAllester, D., Ramanan, D., *Object Detection with Discriminatively Trained Part-Based Models*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, Sept. 2010, doi: 10.1109/TPAMI.2009.167.
- [31] Voulodimos, F., Doulamis, N., Doulamis, A., Protopapadakis, E., *Deep Learning for Computer Vision: A Brief Review*, Computational Intelligence and Neuroscience, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.
- [32] Sultana, F., Sufian, A., Dutta, P., *A Review of Object Detection Models based on Convolutional Neural Network*, in Advances in Intelligent Systems and Computing, Singapore: Springer Singapore, 2020, pp. 1–16, doi: 10.1007/978-981-15-4288-6.
- [33] Du, L., et al., *Overview of two-stage object detection algorithms*, Journal of Physics: Conference Series, vol. 1544, no. 1, pp. 12-33, 2020, doi: 10.1088/1742-6596/1544/1/012033.
- [34] Uijlings, J., Sande, K., Gevers, T., Smeulders, A.W.M., *Selective Search for Object Recognition*, International Journal of Computer Vision, vol. 104, no. 2, pp. 154–171, 2013, doi: 10.1007/s11263-013-0620-5.

- [35] Krizhevsky, A., Sutskever, I., Hinton, G., E., *ImageNet classification with deep convolutional Neural Networks*, In Proceedings of the 25th International Conference on Neural Information Processing Systems, vol. 1, pp. 1097–1105, 2012, doi: 10.5555/2999134.2999257.
- [36] He, K., Zhang, X., Ren, S., Sun., J., *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015, doi: 10.1109/TPAMI.2015.2389824.
- [37] Girshick, R., Donahue, J., Darrell, T., Malik, T., *Rich feature hierarchies for accurate object detection and semantic segmentation*, 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587, doi: 10.1109/CVPR.2014.81.
- [38] Duchesnay, E., *Convolutional neural network — Statistics and Machine Learning in Python 0.5 documentation*, Github.io. [Online]. Available: <https://duchesnay.github.io/pystatsml/deeplearning/dlcnncifar10pytorch.html>. [Accessed: 08-Jan-2022].
- [39] Gandhi, R., *R-CNN, fast R-CNN, faster R-CNN, YOLO — object detection algorithms*, Towards Data Science, [Online]. Available: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>. [Accessed: 08-Jan-2022].
- [40] Girshick, R., *Fast R-CNN*, 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.
- [41] Ananth, S., *Fast R-CNN for object detection*, Towards Data Science, [Online]. Available: <https://towardsdatascience.com/fast-r-cnn-for-object-detection-a-technical-summary-a0ff94faa022>. [Accessed: 08-Jan-2022].
- [42] Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A., *Return of the devil in the details: Delving deep into convolutional nets*, in Proceedings of the British Machine Vision Conference 2014, 2014, doi: 10.5244/C.28.6.
- [43] Ren, S., Kaiming, H., Girshick, R., Sun, J., *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.
- [44] Kim, J., A., Sung, J., Y., Park, S., H., *Comparison of faster-RCNN, YOLO, and SSD for real-time vehicle type recognition*, in 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020, pp. 1-4, doi: 10.1109/ICCE-Asia49877.2020.9277040.
- [45] Lu, X., Li, Q., Li, B., Yan, J., *MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection*, in Computer Vision – ECCV 2020, Cham: Springer International Publishing, 2020, pp. 541–557, .

- [46] Szegedy, C., et al., *Going deeper with convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.
- [47] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., *You Only Look Once: Unified, Real-Time Object Detection*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- [48] Redmon, J., Farhadi, A., *YOLO9000: Better, faster, stronger*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.
- [49] Redmon, J., Farhadi, A. *YOLOv3: An Incremental Improvement*, 2018, ArXiv, abs/1804.02767.
- [50] Liu, W., et al., *SSD: Single Shot MultiBox Detector*, in *Computer Vision – ECCV 2016*, Cham: Springer International Publishing, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0.
- [51] Bojarski, M., et al., *End to End Learning for Self-Driving Cars*, 2016, ArXiv, abs/1604.07316
- [52] Shao, L., Zhu, F., Li, X., *Transfer learning for visual categorization: a survey*, in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019-1034, May 2015, doi: 10.1109/TNNLS.2014.2330900.
- [53] Oquab, M., Bottou, L., Laptev, I., Sivic, J., *Learning and transferring mid-level image representations using convolutional neural networks*, 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1717-1724, doi: 10.1109/CVPR.2014.222.
- [54] Gao, Y., Mosalam, K., *Deep Transfer Learning for Image-Based Structural Damage Recognition*, *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 748–768, 2018, doi: 10.1111/mice.12363.
- [55] Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A., *Meta-Learning in Neural Networks: A Survey*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2021.3079209.
- [56] Wang, X., Huang, T., E., Darrell, T., Gonzalez, J., Yu, F., *Frustratingly Simple Few-Shot Object Detection*, 2020, ArXiv, abs/2003.06957.
- [57] Gidaris, S., Komodakis, N., *Dynamic few-shot visual learning without forgetting*, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367-4375, doi: 10.1109/CVPR.2018.00459.
- [58] Adelson, E. H., Anderson, C., H., Bergen, J., R., Burt, P., J., Ogden, J., M., *Pyramid methods in image processing*, *RCA Engineer*, vol. 29, 1984.

- [59] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., *Feature Pyramid Networks for Object Detection*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106.
- [60] Brudy, T., Schilhab, S., *Projection of hazardous items into X-ray images of inspection objects*, Patent WO 2016/001282 AI, 2016.
- [61] Zou, L., Yusuke, T., Hitoshi, I., *Dangerous Objects Detection of X-Ray Images Using Convolution Neural Network*, in Security with Intelligent Computing and Big-data Services, Cham: Springer International Publishing, 2020, pp. 714–728, doi: 10.1007/978-3-030-16946-6.
- [62] Liang, K., J., Sigman, J., B., Spell, G., Strellis, D., A., Chang, W., Liu, F., Mehta, T., Carin, L., *Toward Automatic Threat Recognition for Airport X-ray Baggage Screening with Deep Convolutional Object Detection*, 2019, ArXiv, abs/1912.06329.
- [63] Akcay, S., Kundegorski, M., Willcocks, C., Breckon, T., *Using Deep Convolutional Neural Network Architectures for Object Classification and Detection Within X-Ray Baggage Security Imagery*, in IEEE Transactions on Information Forensics and Security, vol. 13, no. 9, pp. 2203-2215, Sept. 2018, doi: 10.1109/TIFS.2018.2812196.
- [64] Sangwan, D., Deepak, J., *An Evaluation of Deep Learning based Object Detection Strategies for Threat Object Detection in Baggage Security Imagery*, Pattern Recognition Letters, vol. 120, pp. 112–119, 2019, doi: 10.1016/j.patrec.2019.01.014.
- [65] Yuanxi, W., Liu, X., *A Dangerous goods detection based on transfer learning in X-ray images*, Neural Computing and Applications, vol. 32, no. 12, pp. 8711–8724, 2020, doi: 10.1007/s00521-019-04360-0.
- [66] Xu, C., Han, N., Li, H., *A Dangerous Goods Detection Approach Based on YOLOv3*, in Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence - CSAI '18, 2018, pp. 600-603, doi: 10.1145/3297156.3297199.
- [67] Miao, C., Xie, L., Wan, F., Su, C., Liu, H., Jiao, J., Ye, Y., *SIXray: A Large-scale Security Inspection X-ray Benchmark for Prohibited Item Discovery in Overlapping Images*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 2114-2123, doi: 10.1109/CVPR.2019.00222.
- [68] Everingham, M., Gool, L., V., Williams, C., K., I., Winn, J., Zisserman, A., *The Pascal Visual Object Classes (VOC) challenge*, International Journal of Computer Vision, vol. 88, no. 2, pp. 303-338, 2010, doi: 10.1007/s11263-009-0275-4.
- [69] Davis, J., Goadrich, M., *The relationship between precision-recall and ROC curves*, Proceedings of the 23rd international conference on Machine learning - ICML '06, 2006, pp. 233–240, doi: 10.1145/1143844.1143874.

EKLER

EK 1: SÖZLÜK DİZİNİ

Aktarım Öğrenmesi	: Transfer Learning
Alan Farkı	: Domain Difference
Aşağı Örnekleme	: Downsampling
Aşağıdan Yukarıya Yol	: Bottom Up Pathway
Az Vuruşlu Öğrenme	: Few-shot Learning
Bölge Öneri Ağı	: Region Proposal Network
Çift Kanal	: Dual Channel
Çoklu Görev Kaybı	: Multi-task Loss
Çoklu Görünüm	: Multi-view
Derin Öğrenme	: Deep Learning
Doğru Negatif	: True Negative
Doğru Pozitif	: True Positive
Dondurulmuş Katman	: Freezing Layer
Düşük Enerji	: Low Energy
Düzleştirme	: Flattening
Eşik Değeri	: Threshold
Evrişim	: Convolution
Geri Çağırma	: Recall
Güven Skoru	: Confidence Score
İnce Ayar	: Fine Tuning
Kayıp Fonksiyonu	: Loss Function
Kesin Referans	: Ground Truth
Kesinlik Değeri	: Average Precision
Küme Boyutu	: Batch Size
Maksimum Ortaklama	: Maximum Pooling
Nesne Tespiti	: Object Detection
Omurga Ağı	: Backbone Network
Ortalama Kesinlik Değeri	: Mean Average Precision

Öğrenme Oranı	: Learning Rate
Ön İşleme	: Preprocessing
Örnekleme	: Sampling
Öznitelik	: Feature
Öznitelik Piramit Ağı	: Feature Pyramid Network
Saniyedeki Kare Sayısı	: Frame Per Second
Sınırlandırıcı Kutucuk	: Bounding Box
Tam Bağlantılı Katman	: Fully Connected Layer
Tutturucu Kutucuk	: Anchor Box
Veri Kümesi	: Dataset
Yanlış Negatif	: False Negative
Yanlış Pozitif	: False Positive
Yoğun Katman	: Dense Layer
Yukarı Örnekleme	: Upsampling
Yukarıdan Aşağıya Yol	: Top Down Pathway
Yüksek Enerji	: High Energy