



**T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



DOKTORA TEZİ

**TEHLİKELİ NESNELERİN TESPİTİ İÇİN SÜPER ÇÖZÜNÜRLÜK TABANLI
HİBRİT MODELİN GELİŞTİRİLMESİ**

Ayşe Berika VAROL MALKOÇOĞLU

**DANIŞMAN
Prof. Dr. Rüya ŞAMLI**

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Doktora Programı

Haziran, 2024

TEZ KABUL VE ONAYI

Ayşe Berika VAROL MALKOÇOĞLU tarafından, Prof. Dr. Rüya ŞAMLI danışmanlığında hazırlanan "**Tehlikeli Nesnelerin Tespiti İçin Süper Çözünürlük Tabanlı Hibrit Modelin Geliştirilmesi**" başlıklı bu çalışma, jürimiz tarafından 25/06/2024 tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Doktora Tezi** olarak kabul edilmiştir.

Tez Jürisi

	İmza	Sonuç
DANIŞMAN	Prof. Dr. Rüya ŞAMLI	☒ Kabul
	İstanbul Üniversitesi-Cerrahpaşa Bilgisayar Yazılımı Anabilim Dalı	☐ Ret
ÜYE	Prof. Dr. Ahmet SERTBAŞ	☒ Kabul
	İstanbul Üniversitesi-Cerrahpaşa Bilgisayar Donanımı ve Gömülü Sistemler Anabilim Dalı	☐ Ret
ÜYE	Prof. Dr. Ümit GÜZ	☒ Kabul
	Işık Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı	☐ Ret
ÜYE	Doç. Dr. Selçuk SEVGİN	☒ Kabul
	İstanbul Üniversitesi-Cerrahpaşa Yazılım Mühendisliği Anabilim Dalı	☐ Ret
ÜYE	Dr. Öğr. Üyesi Şengül BAYRAK	☒ Kabul
	Sabahattin Zaim Üniversitesi Yazılım Mühendisliği Anabilim Dalı	☐ Ret



Aileme ithafen...

BÜTÇE DESTEKLERİ

TEHLİKELİ NESNELERİN TESPİTİ İÇİN SÜPER ÇÖZÜNÜRLÜK TABANLI HİBRİT MODELİN GELİŞTİRİLMESİ

Bu tez çalışması, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'nun (TÜBİTAK) 2211-C numaralı projesi ile desteklenmiştir.

TEŞEKKÜR

Akademik yolculuğum boyunca bilgisi ve tecrübeleriyle bana yol gösteren, sorunlarıma her zaman çözüm odaklı yaklaşarak içimi rahatlatan hem kişisel hem akademik konularda fikirleriyle ufkumu açan, her daim yanımda olduğunu ve beni desteklediğini hissettiğim çok değerli danışman hocam Prof. Dr. Rüya ŞAMLI'ya sonsuz teşekkür ederim.

Tüm hayatım boyunca her zaman yanımda olan, beni destekleyen ve bu süreçte beni motive eden canım annem Çiğdem VAROL'a, canım babam Mustafa Kemal VAROL'a ve en kıymetlim kardeşim Bersu VAROL'a çok teşekkür ederim. Ayrıca üniversite sıralarında tanıştığımız günden beri derdime de sevincime de ortak olan, her kararımı destekleyen, her anımda yanımda olan, meslektaşım, yol arkadaşım, canım eşim Şevki Utku MALKOÇOĞLU'na ve bu süreçte farkında olmasada stresimi azaltarak bana neşe veren canım köpeğim Bobi'ye sonsuz teşekkür ederim.

Fikirleri ve yönlendirmeleri ile tezimin gelişmesine katkı sağlayan değerli hocalarım Doç. Dr. Selçuk SEVGİN ve Dr. Öğr. Üyesi Şengül BAYRAK'a ayrıca teşekkürlerimi bir borç bilirim.

Son olarak süreçte beni destekleyen, çalışmama yön veren, yazdıklarıyla ilham kaynağı olan herkese çok teşekkürler.

Haziran, 2024

Ayşe Berika VAROL MALKOÇOĞLU

İÇİNDEKİLER

Sayfa No

TEZ KABUL VE ONAYI	ii
BÜTÇE DESTEKLERİ.....	iv
TEŞEKKÜR.....	v
İÇİNDEKİLER.....	vi
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ	x
SİMGE VE KISALTMA LİSTESİ	xi
DENKLEM LİSTESİ	xiv
ÖZET	xv
SUMMARY	xvi
1. GİRİŞ	1
2. KAVRAMSAL ÇERÇEVE	7
2.1. GELENEKSEL NESNE TESPİT YÖNTEMLERİ.....	10
2.1.1. Bölge Seçimi	10
2.1.2. Özellik Çıkarımı.....	11
2.1.3. Sınıflandırma	11
2.2. DERİN ÖĞRENME TABANLI MODERN YÖNTEMLER	11
2.2.1. İki Aşamalı Nesne Tespiti	12
2.2.1.1. RCNN.....	12
2.2.1.2. Fast-RCNN.....	14
2.2.1.3. Faster-RCNN	17
2.2.1.4. SPP-Net	19
2.2.1.5. RFCN	19
2.2.1.6. Mask-RCNN	21
2.2.2. Tek Aşamalı Nesne Tespiti	23
2.2.2.1. YOLOv1	23
2.2.2.2. YOLOv2	26

2.2.2.3. YOLOv3	28
2.2.2.4. YOLOv4	30
2.2.2.5. YOLOv5	31
2.2.2.6. YOLOv6	33
2.2.2.7. YOLOv7	35
2.2.2.8. YOLOv8	37
2.3. SÜPER ÇÖZÜNÜRLÜK TEKNİKLERİ	39
2.3.1. Hızlı Süper Çözünürlüklü Evrişimsel Sinir Ağı	43
2.3.2. Etkin Alt Piksel Evrişimli Sinir Ağı	45
2.3.3. Geliştirilmiş Derin Süper Çözünürlük Ağı	46
2.3.4. Laplacian Piramidi Süper Çözünürlük Ağı	46
2.4. LİTERATÜR TARAMASI	48
2.4.1. Geleneksel Yöntemler ile Gerçekleştirilen Çalışmalar	48
2.4.2. Derin Öğrenme Tabanlı Yöntemler ile Gerçekleştirilen Çalışmalar	50
2.4.3. SR algoritmaları ile Gerçekleştirilen Çalışmalar	56
3. YÖNTEM	59
3.1. VERİ SETİ	59
3.2. DERİN KANAL DİKKAT SÜPER ÇÖZÜNÜRLÜĞÜ	63
3.3. İKİLİ FİLTRE	67
3.4. ÖNERİLEN MODEL	69
3.5. DEĞERLENDİRME METRİKLERİ	70
3.5.1. Nesne Tespitinin Değerlendirilmesi	70
3.5.2. Görüntü Kalitesinin Değerlendirilmesi	71
3.6. UYGULAMA PLATFORMU VE KÜTÜPHANELERİ	72
4. BULGULAR	76
4.1. YOLO MODELLERİNİN PERFORMANS DEĞERLENDİRMESİ	76
4.2. DCASR MODELİNİN PERFORMANS DEĞERLENDİRMESİ	78
4.3. YOLO+SR MODELLERİNİN PERFORMANS DEĞERLENDİRMESİ	81
5. TARTIŞMA	90
6. SONUÇ VE ÖNERİLER	92
KAYNAK	95
İNTİHAL RAPORU İLK SAYFASI	105

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1: Nesne tanıma işleminin görevleri.	2
Şekil 1.2: Görüntüdeki nesnenin sınıflandırılması.	2
Şekil 1.3: Görüntüdeki nesnenin sınıflandırılması ve yerleştirilmesi	3
Şekil 1.4: Çeşitli nesnelerin belirli sınıflara göre tespit edilmesi.	3
Şekil 1.5: Nesnelerin örnek bölütlemeğe göre maskelenmesi.	4
Şekil 1.6: Nesnelerin semantik bölütlemeğe göre maskelenmesi.	4
Şekil 2.1: Geleneksel ve modern nesne tespit algoritmalarının zaman çizelgesi.	10
Şekil 2.2: RCNN mimarîsi.	12
Şekil 2.3: SS algoritması ile bölge indirgeme.	13
Şekil 2.4: Fast-RCNN mimarîsi.	14
Şekil 2.5: Özellik haritasında roi'lerin izdüşümü.	15
Şekil 2.6: $\sim 3 \times 6$ boyutundaki RoI'nin havuzlama katmanından sonraki dönüşümü.	16
Şekil 2.7: Faster-RCNN mimarîsi.	17
Şekil 2.8: RFCN mimarîsi.	20
Şekil 2.9: Mask-RCNN mimarîsi.	22
Şekil 2.10: YOLOv1 mimarîsi.	24
Şekil 2.11: IoU metriğinin hesaplanması.	25
Şekil 2.12: Hedef nesne için sınırlayıcı kutu çizimi.	26
Şekil 2.13: YOLOv2 mimarîsi.	27
Şekil 2.14: YOLOv3 mimarîsi.	29
Şekil 2.15: YOLOv4 mimarîsi.	30
Şekil 2.16: YOLOv5 mimarîsi.	32
Şekil 2.17: YOLOv6 mimarîsi.	34
Şekil 2.18: YOLOv7 mimarîsi.	36
Şekil 2.19: YOLOv8 mimarîsi.	38
Şekil 2.20: Renk Skalası.	40

Şekil 2.21: PPI ve çözünürlük ilişkisi.....	40
Şekil 2.22: FSRCNN mimarîsi.	44
Şekil 2.23: ESPCNN mimarîsi.....	45
Şekil 2.24: EDSR mimarîsi.	46
Şekil 2.25: LapSRN mimarîsi.	47
Şekil 3.1: Veri seti oluşturma adımları.....	60
Şekil 3.2: a) Tüfek görüntüsünün etiketli hali, b) Silah ve insan görüntüsünün etiketli hali, c) Silah ve insan görüntüsünün etiketli hali, d) Etiket bilgileri	61
Şekil 3.3: Eğitim, validasyon ve test setinde yer alan nesnelerin dağılım grafiği.	63
Şekil 3.4: Orijinal, x_2 , x_3 ve x_4 oranındaki 1r “kelebek” görüntüleri.	63
Şekil 3.5: Artık blok yapısı.	64
Şekil 3.6: Kanal dikkat katmanı.....	65
Şekil 3.7: Örneklem katmanı (bo: büyütme oranı).	66
Şekil 3.8: Önerilen DCASR mimarîsi.	66
Şekil 3.9: 64x64 boyutlu görüntünün 3 farklı oranda DCASR modeli ile iyileştirilmesi.....	67
Şekil 3.10: İyileştirilen görüntülerin gürültüleri bf ile giderilmesi.	68
Şekil 3.11: Önerilen modelin mimarîsi.	69
Şekil 3.12: Karmaşıklık matrisi.	70
Şekil 3.13: Google Colab Pro çalışma zamanı tipleri.....	72
Şekil 3.14: Matplotlib örneği.	73
Şekil 3.15: Matplotlib ve Numpy örneği.....	74
Şekil 4.1: Farklı çözünürlük oranlarında SR ile iyileştirilen “kelebek” görüntüsü.	80
Şekil 4.2: Test_1 veri setinin sonuçları.	84
Şekil 4.3: Test_2 veri setinin sonuçları.	88
Şekil 4.3 (devam): Test_2 veri setinin sonuçları.....	89

TABLO LİSTESİ

Sayfa No

Tablo 2.1: YOLO mimarîlerinin karşılaştırması.	39
Tablo 2.2: Çalışmada yer alan SR algoritmalarının özellikleri.	43
Tablo 2.3: Geleneksel yöntemler ile yapılmış nesne tespit çalışmaları.	50
Tablo 2.4: DL tabanlı modern yöntemler ile yapılmış nesne tespit çalışmaları.	55
Tablo 2.4 (devam): DL tabanlı modern yöntemler ile yapılmış nesne tespit çalışmaları.	56
Tablo 2.5: Nesne tespiti ve SR uygulamaları.	58
Tablo 3.1: Kamuya açık tehlikeli veri setleri ve özellikleri.	59
Tablo 3.2: DMGDATA veri setinin özellikleri.	62
Tablo 3.3: Eğitim, validasyon ve test seti bilgileri.	62
Tablo 3.4: DCASR modelinin eğitim hiperparametreleri.	66
Tablo 3.4 (devam): DCASR modelinin eğitim hiperparametreleri.	67
Tablo 4.1: YOLO modellerin eğitim sonuçları.	77
Tablo 4.2: SR modellerinin performans değerlendirmesi.	79
Tablo 4.3: SR modellerinin Test_1 veri setini iyileştirirken harcadığı donanım ve zaman. ...	82
Tablo 4.4: Önerilen modelin Test_1 veri setinde nesne tespiti için harcadığı donanım ve zaman.	82
Tablo 4.5: Önerilen nesne tespit modelinin Test_1 veri setindeki sonucu.	83
Tablo 4.6: SR modellerinin Test_2 veri setini iyileştirirken harcadığı donanım ve zaman. ...	85
Tablo 4.7: Önerilen modelin Test_2 veri setinde nesne tespiti için harcadığı donanım ve zaman.	86
Tablo 4.8: Önerilen nesne tespit modelinin Test_2 veri setindeki sonucu.	87

SİMGE ve KISALTMA LİSTESİ

Kısaltmalar	Açıklama
ADAM	: Adaptive Moment estimation – Uyarlanabilir Moment Tahmini
B	: Blue – Mavi
BCE	: Binary Cross Entropy – İkili Çarpaz Entropi
BF	: Bilateral Filter – İkili Filtre
BO	: Büyütme Oranı
BoF	: Bag-of-Freebies – Hediye Çanta
BoS	: Bag-of-Specials – Özel Çanta
CCTV	: Close Circuit TeleVision – Kapalı Devre Televizyon Sistemleri
CIOU	: Complete Intersection Over Union – Birlik Üzerinde Tam Kesişme
CNN	: Convolutional Neural Networks – Evrişimli Sinir Ağları
CNTK	: Cognitive Toolkit – Bilişsel Araç Seti
CSP	: Cross Stage Partial – Kısmi Çarpaz Aşama
DCASR	: Deep Channel Attention Super Resolution – Derin Kanal Dikkat Süper Çözünürlüğü
DFL	: Directional Feature Learning – Yönsel Özellik Öğrenme
DL	: Deep Learning – Derin Öğrenme
DPM	: Deformable Part Model – Değişime Uğrayabilen Parça Bazlı Model
DSSD	: Deconvolutional Single Shot Detector – Evrişimsiz Tek Atım Dedektörü
EDSR	: Enhanced Deep Super-Resolution Network – Geliştirilmiş Derin Süper Çözünürlük Ağı
EELAN	: Extended Efficient Layer Aggregation Network – Genişletilmiş Verimli Katman Toplama Ağı
ES	: Ensemble Tree – Topluluk Ağacı
ESPCNN	: Efficient Sub-Pixel Convolutional Neural Network – Etkin Alt Piksel Evrişimli Sinir Ağı

FAST	: Features from Accelerated Segment Test – Hızlandırılmış Bölütleme Test
Faster-RCNN	: Faster Region Based Convolution Neural Networks – Daha Hızlı Bölge Tabanlı Evrişimli Ağ
Fast-RCNN	: Fast Region Based Convolution Neural Networks – Hızlı Bölge Tabanlı Evrişimli Ağ
FN	: False Negative – Yanlış Negatif
FP	: False Positive – Yanlış Pozitif
FPN	: Feature Pyramid Network – Özellik Piramit Ağ
FREAK	: Fast Retina Keypoint – Hızlı Retina Anahtar Noktası
FSRCNN	: Fast Super-Resolution Convolutional Neural Network – Hızlı Süper Çözünürlüklü Evrişimsel Sinir Ağ
FTT	: Feature Texture Transfer – Özellik Doku Aktarımı
G	: Green – Yeşil
GAN	: Generative Adversarial Network – Üretken Çekişmeli Ağları
GPU	: Graphics Processing Unit – Grafik İşleme Birimi
HOG	: Histograms of Oriented Gradients – Yönlü Gradyanlar Histogramı
HR	: High Resolution – Yüksek Çözünürlüklü
IoU	: Intersection Over Union – Birleşim Üzerinden Kesişme
KNN	: K Nearest Neighbors – K En Yakın Komşu
LapSRN	: Laplacian Pyramid Super-Resolution Network – Laplacian Piramidi Süper Çözünürlük Ağ
LR	: Low Resolution – Düşük Çözünürlüklü
MAE	: Mean Absoulte Error – Ortalama Mutlak Hata
mAP	: Mean Average Precision – Ortalama Hassasiyet
Mask-RCNN	: Mask Region Based Convolutional Neural Networks – Maske Bölge Tabanlı Evrişimli Sinir Ağ
MCGR	: GAN with Residual Feature Aggregation – Artık Özellik Toplama ile GAN
ML	: Machine Learning – Makine Öğrenmesi
MOBESE	: Mobil Elektronik Sistem Entegrasyonu
MRA	: Multi-Resolution Analysis – Çoklu Çözünürlük Analizi
MSE	: Mean Square Error – Ortalama Kare Hata
OAOD	: Orientation Aware Object Detector –Yönelim Duyarlı Nesne Dedektörü

PAN	: Path Aggregation Network – Yol Toplama Ağı
PPI	: Pixel Density / Pixels Per Inch – Piksel Yoğunluğuna
PSNR	: Peak Signal to Noise Ratio – Tepe Sinyalin Gürültüye Oranı
R	: Red – Kırmızı
RaD	: Relativistic Discriminator – Göreli Ayırıcı
RAM	: Random Access Memory – Rastgele Erişimli Belleğin
RCAN	: Residual Channel Attention Network – Artık Kanal Dikkat Ağı
RCNN	: Region Based Convolution Neural Networks – Bölge Tabanlı Evrişimli Ağı
RFA	: Residual Feature Aggregation – Artık Özellik Toplama
RFCN	: Region-based Fully Convolutional Networks – Bölge Tabanlı Tam Evrişimli Ağlar
RoI	: Region of Interest – İlgili Bölge
RPN	: Region Proposal Network – Bölge Öneri Ağını
RRDB	: Residual in Residual Dense Block – Artık Yoğun Bloкта Kalan
RSFR	: Random Forest Super Resolution – Rastgele Orman Süper Çözünürlüğü
SCSP-ResNet	: Uzamsal Çapraz Aşama Kısmi ResNet – Spatial Cross Stage Partial-ResNet
SIFT	: Scale-Invariant Feature Transform – Ölçek Değişmez Unsur Dönüşümü
SOTA	: CNN-Based Object Detectors And State of the Art – CNN Tabanlı Nesne Dedektörleri
SPP	: Spatial Pyramid Pooling – Mekansal Piramit Havuzlama
SR	: Super Resolution – Süper Çözünürlük
SS	: Selective Search – Seçici Arama
SSD	: Single Shot Multibox Detector – Tek Atışlı Multibox Dedektör
SURF	: Speeded Up Robust Features – Hızlandırılmış Sağlam Özelliklerin
SVM	: Support Vector Machine – Destek Vektör
TN	: True Negative – Doğru Negatif
TP	: True Positive – Doğru Pozitif
VDSR	: Very Deep Super Resolution – Çok Derin Süper Çözünürlük
VFL	: Vari Focal Loss – Değişken Odaklı Kayıp
YOLO	: You Only Look Once – Sadece Bir Kez Bak
YOLT	: You Only Look Twice – Sadece İki Kere Bak

DENKLEM LİSTESİ

- Denklem (2.1)** : Güven skoru denklemi
- Denklem (2.2)** : Düşük-yüksek çözünürlüklü görüntüler arasındaki ilişkinin denklemi
- Denklem (3.1)** : Kanal dikkat mekanizmasının denklemi
- Denklem (3.2)** : İkili filtrenin denklemi
- Denklem (3.3)** : Hassasiyet metriğinin denklemi
- Denklem (3.4)** : Duyarlılık metriğinin denklemi
- Denklem (3.5)** : mAP metriğinin denklemi
- Denklem (3.6)** : Tepe Sinyalin Gürültüye Oranının denklemi
- Denklem (3.7)** : Ortalama Kare Hatası metriğinin denklemi

ÖZET

[DOKTORA TEZİ]

TEHLİKELİ NESNELERİN TESPİTİ İÇİN SÜPER ÇÖZÜNÜRLÜK TABANLI HİBRİT MODELİN GELİŞTİRİLMESİ

Ayşe Berika VAROL MALKOÇOĞLU

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Doktora Programı

[Danışman: Prof. Dr. Rüya ŞAMLI]

Kamusal alanda yetkisiz kişiler tarafından taşınan ya da kullanılan tehlikeli nesnelerin varlığı halkın can güvenliğini tehdit ettiği bilinmektedir. Yetkililer güvenliğini sağlamak için kamusal alanda CCTV ve MOBESE gibi kamera sistemleri bulundurmaktadır. Fakat bu sistemler operatörler tarafından 7/24 izlendiği için olası insanî hatalar meydana gelebilmektedir. Bu sebepten tehlikeli nesneleri otomatik tanıyan operatörlere yardımcı sistemlerin geliştirilmesi oldukça önemlidir. Bu doğrultuda tez kapsamında tehlikeli nesneleri tespit etmek için dijital görüntü ve videolar üzerindeki hedeflenen nesneyi tespit eden YOLO ve görüntüyü iyileştiren SR algoritmasının birleşiminden oluşan hibrit bir model önerilmiştir. Önerilen hibrit modelin eğitimi için DMGDATA adında 18,231 adet veriden meydana gelen yeni bir veri seti hazırlanmıştır. Bu veri seti ile YOLO mimarisine ait seçilen versiyonlar eğitilmiş ve elde edilen metriklere göre tez için YOLOv8m modelinin uygun olduğu tespit edilmiştir. Önerilen hibrit model içerisinde yer alan SR katmanı için ise; kanal dikkat mekanizması ve artık blok katmanlarından meydana gelen DCASR adında yeni bir algoritma önerilmiştir. Bu algoritma benzerleri ile karşılaştırıldığında, hız ve PSNR değeri açısından daha başarılı sonuç ürettiği görülmüştür. DCASR modeli ile iyileştirilen görüntülerin BF filtresi ile gürültüsü giderildikten sonra tespit için YOLOv8m modeline gönderilmiştir. Önerilen YOLOv8m+DCASR+BF modeli iki farklı çözünürlükten oluşan test seti ve farklı SR algoritmaları ile test edilmiştir. Test işlemlerinin sonucunda DCASR algoritması kullanılarak oluşturulan hibrit modelin diğerlerine göre daha başarılı olduğu tespit edilmiştir. Önerilen hibrit model ile normal çözünürlüklü görüntülerde nesne tespit başarısında iyileşme sağlanmazken, düşük çözünürlüklü görüntülerde nesne tespit başarısının %4 ila %6 oranında iyileştiği görülmüştür. Bu sayede düşük kaliteli kaynaktan gelen görüntülerin algoritmik yöntemler kullanılarak daha başarılı nesne tespit işlemi gerçekleştirebileceği düşünülmektedir.

Haziran 2024, 121 sayfa

Anahtar kelimeler: Nesne Tespiti, Süper Çözünürlük, Bilgisayarla Görü

SUMMARY

[Ph.D. THESIS]

DEVELOPMENT OF SUPER RESOLUTION BASED HYBRID MODEL FOR THE DETECTION OF DANGEROUS OBJECTS

Ayşe Berika VAROL MALKOÇOĞLU

**Istanbul University-Cerrahpaşa
Institute of Graduate Studies
Department of Computer Engineering
Computer Engineering, Ph.D. Program**

[Supervisor: Prof. Dr. Rüya ŞAMLI]

The presence of dangerous objects carried or used by unauthorized persons in public spaces is known to threaten public safety. Authorities have camera systems such as CCTV and MOBESE in public spaces to ensure security. However, since these systems are monitored 24/7 by operators, possible human errors may occur. For this reason, it is very important to develop systems that help operators to automatically recognize dangerous objects. Accordingly, in this thesis, a hybrid model consisting of YOLO, which detects the targeted object on digital images and videos, and SR algorithm, which enhances the image, is proposed to detect dangerous objects. For the training of the proposed hybrid model, a new dataset called DMGDATA consisting of 18,231 data was prepared. With this dataset, selected versions of the YOLO architecture were trained and according to the metrics obtained, it was determined that the YOLOv8m model was suitable for the thesis. For the SR layer in the proposed hybrid model, a new algorithm called DCASR, which consists of channel attention mechanism and residual block layers, is proposed. When this algorithm is compared with its similar algorithms, it is observed that it produces more successful results in terms of speed and PSNR value. The images enhanced with the DCASR model were sent to the YOLOv8m model for detection after removing the noise with the BF filter. The proposed YOLOv8m+DCASR+BF model was tested with a test set of two different resolutions and different SR algorithms. As a result of the test procedures, the hybrid model created using the DCASR algorithm was found to be more successful than the others. With the proposed hybrid model, object detection performance in normal resolution images is not improved, while object detection performance in low resolution images is improved by 4% to 6%. In this way, it is thought that images from low-quality sources can perform more successful object detection using algorithmic methods.

June 2024, 121 pages

Keywords: Object Detection, Super Resolution, Computer Vision

1. GİRİŞ

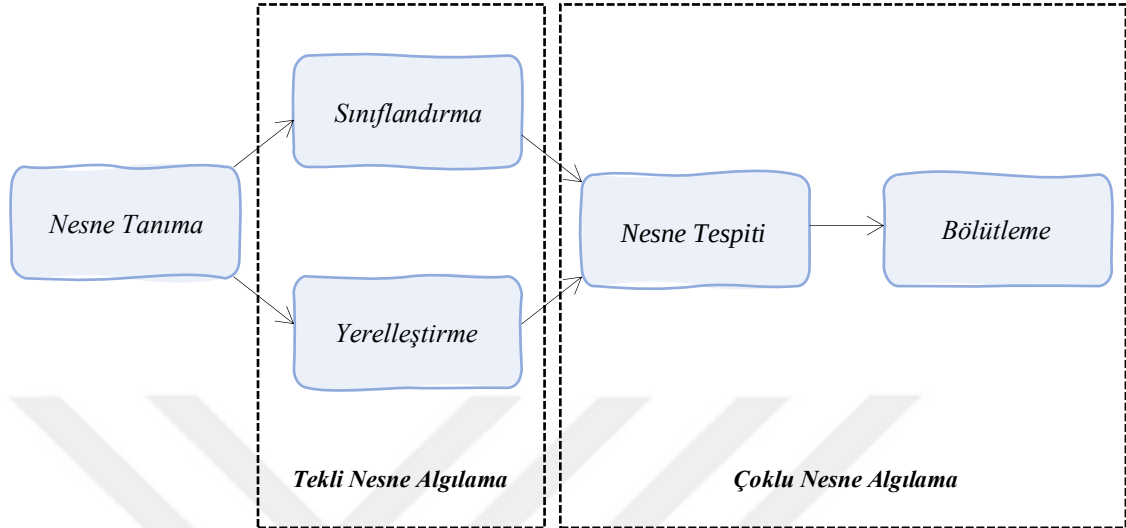
Nesne, Türk Dil Kurumuna (TDK) göre kendisine ait ağırlığı, hacmi, rengi ve dokusu olan cansız bir varlık olarak tanımlanmaktadır (<https://sozluk.gov.tr/>, Erişim tarihi: 7 Şubat 2024). Bilgisayarla görü alanında ise nesne kavramı bir görüntü içerisinde yer alan gerçekte canlı ya da cansız olabilen her türlü varlığın tanımı için kullanılmaktadır. Bilgisayarla görü alanında sıklıkla kullanılan nesnelerin örnek listesi aşağıda sunulmuştur:

- Araba
- İnsan yüzü
- Hayvan
- Bisiklet
- Silah
- Bıçak

TDK'ya göre, bu örnek listede yer alan “insan yüzü” ve “hayvan” ifadeleri nesne olarak tanımlanmamaktadır. Fakat bilgisayarla görü alanında bu liste içerisinde yer alan tüm örnekler birer nesnedir ve genellikle algoritmalar tarafından tespit edilmek için kullanılır. Buna göre bilişim dünyasında nesne tanımının daha geniş bir alana sahip olduğu söylenebilir.

Algoritmalar ile tespit işlemi görüntü ya da video içinde hedeflenen bir nesnenin ait olduğu sınıfın tahmin edilmesi ve sınırlayıcı bir kutu ile çerçevelendirilmesini kapsamaktadır. Nesne tespiti; hedeflenen bir nesnenin şeklinin, renginin, duruşunun, boyutunun, konumunun farklı örnekler üzerinden modele öğretilmesi ve sonrasında yeni gelen görüntü üzerinde hedeflenen nesne/nesnelerin tespit edilmesi işlemidir. Görüntüdeki nesneyi tespit etme sırasında “sınıflandırma + yerelleştirme” ya da “nesne tespit” işlemi kullanılabilir. Bu iki yöntem benzer mantıkta çalışmaktadır. Fakat sınıflandırma + yerelleştirme görüntüde tek bir nesne varsa ya da nesnelerin sayısı biliniyorsa doğru çalışabilmektedir. Eğer görüntüde birden fazla nesne varsa ve nesnenin mevcut sayısı bilinmiyorsa, sistemin üreteceği sınırlayıcı kutu sayısı dolayısıyla koordinat sayısı bilinemeyecektir. Bu noktada nesne tespit yönteminin kullanılması gerekmektedir (Fei-Fei vd., 2017; Karagiannakos, 2019).

Hedef nesnenin tespit edilebilmesi için tüm nesneleri tanıma ve konumlandırma işlemi yapan teknik görevler Şekil 1.1’de hiyerarşik bir şekilde gösterilmektedir (Salari vd., 2022; Russakovsky vd., 2015).



Şekil 1.1: Nesne tanıma işleminin görevleri.

Tekli Nesne:

Nesne Sınıflandırma: Etiketli veriler ile eğitilmiş modele gönderilen yeni girdinin hangi sınıfa ait olduğunun tahmin edilmesi işlemidir. Bu işleme ait bir örnek görüntü Şekil 1.2’de verilmiştir. Görüldüğü gibi görüntüde yer alan nesnenin silah (gun) olma olasılığı %97,1 olarak tahmin edilmişken, bıçak (knife) olma olasılığı %2,9 olarak tahmin edilmiştir.



Şekil 1.2: Görüntüdeki nesnenin sınıflandırılması.

- **Nesne Yerelleştirme:** Görüntüdeki nesnenin konumunu tespit ederek sınırlayıcı bir kutu ile çerçeveleme işlemidir. Bu işlem sırasında nesnenin çevresindeki sınırlayıcı kutunun x koordinatı, y koordinatı, yükseklik ve genişlik bilgileri olmak üzere 4 sürekli sayı kümesi tahmin edilmektedir. Bu eylem ancak sınıflandırma işleminden sonra hedef nesne/nesneler için gerçekleşir.
- **Sınıflandırma+Yerelleştirme:** Görüntüde yalnızca bir nesne varsa ya da verideki nesnelerin sayısı biliniyorsa yapılan işlemidir. Amaç nesnelerin yerlerini tespit ederek sınır kutusunun çizimini gerçekleştirmektir. Örnek bir görüntü Şekil 1.3'te verilmiştir.



Şekil 1.3: Görüntüdeki nesnenin sınıflandırılması ve yerleştirilmesi

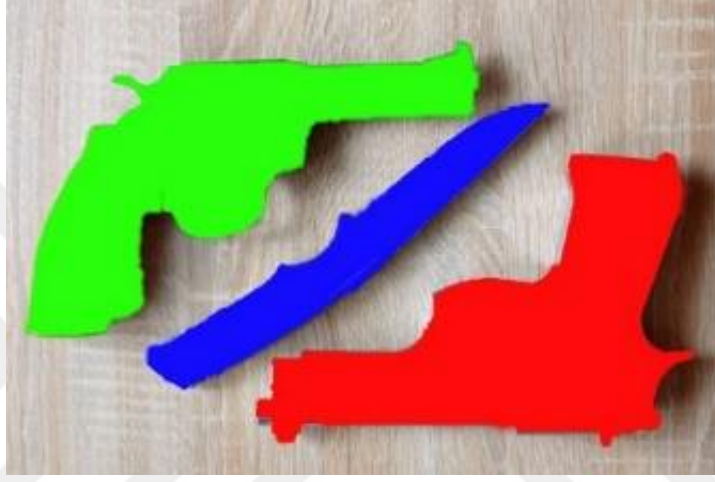
Çoklu Nesne:

- **Nesne Tespiti:** Görüntüdeki tüm nesnenin tespit edilip sınırlarının belirlenmesini sağlayan algoritmaların bulunduğu çatıdır. Örnek bir görüntü Şekil 1.4'te verilmiştir.



Şekil 1.4: Çeşitli nesnelerin belirli sınıflara göre tespit edilmesi.

- **Görüntü Bölütleme:** Görüntü bölütleme işlemi ise sınıflandırma ve nesne tespiti gibi işlemler için potansiyel olarak anlamlı alanları betimleyen bir yöntemdir. Görüntüdeki her nesne için oluşturulan piksel bazında maskeler ve bir nesnenin varlığının işaretlenmesini sağlar. Sınırlayıcı kutuya göre daha etkili bir nesne algılama yöntemidir. Örnek Bölütleme ve Semantik Bölütleme olarak kategorize edilebilir.
 - **Örnek bölütleme:** Aynı sınıfın birden çok örneği ayrı bölütlenir. Tüm nesneler aynı sınıfa ait olsalar bile farklı renklerle renklendirilir. Örnek bir görüntü Şekil 1.5'te verilmiştir.



Şekil 1.5: Nesnelerin örnek bölütlemeye göre maskelenmesi.

- **Semantik bölütleme:** Aynı sınıftaki tüm nesneler için tek bir sınıf oluşturulur, bu nedenle aynı sınıftaki tüm nesneler aynı renkle renklendirilir. Örnek bir görüntü Şekil 1.6'da verilmiştir.



Şekil 1.6: Nesnelerin semantik bölütlemeye göre maskelenmesi.

Birçok ülkede “bireysel silahsızlanmayı” özendirmek amacıyla çeşitli vakıflar kurulmasına rağmen silahlanma eylemi bir güvenlik sorunu olarak hala devam etmektedir (<http://www.umut.org.tr/>, Erişim tarihi: 24 Temmuz 2023; <https://unidir.org/>, Erişim tarihi: 24 Temmuz 2023; <https://www.un.org/en/>, Erişim tarihi: 24 Temmuz 2023). Yetkililer olası tehlikelerin önüne geçmek amacıyla kamu kuruluşlarında, alış-veriş merkezlerinde, bankalarda, otoparklarda, hastane vb. ortamlarda kapalı devre televizyon sistemleri (Close Circuit TeleVision – CCTV) ve şehir izleme sistemleri (Mobil Elektronik Sistem Entegrasyonu – MOBESE) kurmaktadır. Kurulan sistemlerden gelen veriler güvenlik operatörleri tarafından 7/24 izlenerek, tehlikeli durum oluşturma potansiyeli yüksek bireyler ve durumlar takip edilmektedir (Piza vd., 2019). Bu operatörler 4, 9, 16 ve daha fazla ekranı aynı anda izleyerek olası tehditleri tespit etme, kontrol etme, gözlemleme ve tehditlere karşı önlem alma gibi görevleri yerine getirerek insanların güvenliğini sağlamaktadır (Cohen vd., 2009). Fakat insan doğası gereği, aynı anda birden fazla video akışını izlemenin oldukça zor olduğu bilinmektedir. Tickner & Poulton (1973) tarafından yapılan çalışmaya göre operatörlerin algılama oranları yaşlarına bağlı olarak değişmekle birlikte, yaştan bağımsız her bireyin 1 saat sonra algılama oranının önemli ölçüde düştüğü yapılan çalışmada öne sürülmektedir. Dolayısıyla zaman içinde artan CCTV verilerinin manuel olarak incelenmesinin ve analiz edilmesinin oldukça zor olduğu düşünülmektedir.

Güvenlik amacı ile kamera izleme sistemlerinin otomatikleştirilerek operatörlere yardımcı yapılar haline getirilmesi bir ihtiyaç haline geldiği araştırmacılar tarafından fark edilmiş olmasına rağmen bu alandaki çalışmalar özellikle son 10 yılda ortaya çıkmıştır (Darker vd., 2007). Bu durumun temel sebebi görüntü akışlarının gerçek zamanlı işlenmesini kolaylaştıran Derin Öğrenme (Deep Learning – DL) modellerinin ortaya çıkışı, görüntülerin işlenmesini hızlandıran donanım araçları ve modellerin eğitilebilmesi için büyük verilerin saklanması kolaylaştıran depolama kapasitelerindeki artış olduğu görülmektedir (Zou vd., 2019). Dolayısıyla 10-15 sene önce donanım yetersizliğinden kaynaklanan problemlerin günümüzde teknolojinin gelişmesiyle aşıldığı ve güvenlik ya da kişisel amaçlarla çekilen video görüntülerinin artması sayesinde model eğitimlerini nispeten kolaylaşmıştır.

Bu tez çalışmasında tehlikeli nesnelerin tespit edilmesini sağlayan DL ve SR tabanlı model geliştirilerek güvenlik güçlerine yardımcı olabilecek bir sistem geliştirilmesi amaçlanmıştır. Bu doğrultuda, Sadece Bir Kez Bak (You Only Look Once – YOLO) mimarîleri incelenmiş ve uygulanmıştır. Mevcut modellerden daha başarılı sonuçlar üretmeyi başaran Derin Kanal Dikkat Süper Çözünürlüğü (Deep Channel Attention Super Resolution – DCASR)

adında yeni bir Süper Çözünürlük (Super Resolution – SR) algoritması önerilerek, uygun YOLO modeli ile birleştirilip hibrit bir model yapısı önerilmiştir. Önerilen modelin eğitiminin daha başarılı gerçekleştirilebilmesi için DMGDATA olarak isimlendirilen yeni bir veri seti hazırlanmıştır. Deneysel yöntemler ile elde edilen sonuçlar farklı modeller ve literatürdeki benzer çalışmalar ile kıyaslanarak gösterilmiştir. Bu tez çalışmasının literatüre olan katkısı şu şekilde özetlenebilir:

- Tehlikeli nesneler üzerinde yapılmış nesne tespit çalışmaları incelenerek özetlenmiştir.
- DMGDATA adında yeni bir veri seti hazırlanarak kamuya sunulmuştur.
- DCASR adında artık bloklu ve kanal dikkat mekanizmasına sahip yeni bir SR algoritması önerilmiştir.
- Önerilen DCASR algoritması ile YOLO algoritması birleştirilerek hibrit bir model oluşturulmuştur. Bu sayede düşük çözünürlüklü görüntülerdeki nesne tespit başarısı %4 – %6 oranında arttırılmıştır.
- SR algoritmaları literatürde ilk defa tehlikeli nesnelerin tespitini iyileştirmek için kullanılmıştır.

2. KAVRAMSAL ÇERÇEVE

Nesne tespiti, insanın görsel algılama yetisinden esinlenerek geliştirilmiş, görüntü ve videolardaki hedef nesneyi tespit etmeyi sağlayan bilgisayarla görü tekniğidir. Temel olarak, dijital görüntü veya videolardaki nesnelerin tanınması ve sınıflandırılması sürecini temsil etmektedir.

İnsanlar için nesneleri tanımak ve sınıflandırmak doğal bir süreç olsa da bu görev bilgisayarlar için oldukça zordur. Bilgisayarlar hedef nesneyi tespit etmek için farklı nesne tespit algoritmaları kullanmaktadır. Bu algoritmaların bir görüntüdeki ya da videodaki hedef nesneyi tanımlayabilmesi için geniş bir veri kümesi ile eğitilmesine ihtiyacı vardır. Hedef nesnelerden oluşan veri seti ile eğitilen model, nesnenin özelliklerini öğrenerek görüntüde hedeflenen nesnenin konumunu belirleyip bir çerçeve içine almaktadır. Ardından konumu belirlenen nesne, mevcut sınıflar arasından birine atanarak sınıfa aitlik oranı hesaplanmaktadır. Fakat nesne tespiti için sadece sınıflandırma işlemi ya da nesnenin konumunun bulunması yeterli değildir. Bu işlem için hem sınıflandırma hem de yerleştirme işleminin aynı model yapısı içerisinde yer alması büyük önem arz etmektedir.

Nesne tespit işlemleri temelde geleneksel ve modern yöntemler olarak iki ana bölüme ayrılmaktadır. Modern yöntemler ise İki Aşamalı ve Tek Aşamalı olarak iki alt bölüme ayrılmaktadır. Bu yöntemler şu şekilde kategorize edilebilir.

Geleneksel Yöntemler:

- **Haar:** Bir görüntüdeki iki veya daha fazla bitişik dikdörtgensel bölgenin piksel değerlerinin toplamalarının farkını hesaplayan özellik çıkarım algoritmasıdır. Görüntüyü dikdörtgenler yardımı ile tarayarak elde edilen değere belirli bir eşik değeri uygulanır ve nesne olup olmadığı tespit edilir (Papageorgiou vd., 1998).
- **Ölçek Değişmez Unsur Dönüşümü (Scale-Invariant Feature Transform – SIFT):** Gaussian filtresi ile farklı görünümeler arasındaki değişmeyen özelliklerin keşfedilmesini sağlayan algoritmadır. SIFT, ölçek değişikliklerine, dönüşümlere ve aydınlatma değişikliklerine karşı oldukça dayanıklıdır. Bu durum sayesinde nesne tespiti ve eşleştirme gibi uygulamalarda etkili sonuçlar üretmektedir (Lowe, 1999).

- **Viola & Jones:** Haar ile özellik çıkarımı gerçekleştirip AdaBoost ile sınıflandırma işlemini gerçekleştiren hızlı ve başarılı ilk yüz tanıma algoritmasıdır (Viola & Jones, 2004). Viola & Jones modeli, herhangi bir pencerenin insan yüzü içerip içermediğini tespit etmek için görüntüdeki olası tüm konumları ve ölçekleri gözden geçirir.
- **Yönlü Gradyanlar Histogramı (Histograms of Oriented Gradients – HOG):** Görüntüyü eşit büyüklüklerde hücreye ayırıp her hücredeki piksellerin gradyan yönünü gruplayan ve hepsini uç uca ekleyip tek bir özellik vektörü oluşturan algoritmadır (Dalal & Triggs, 2005). HOG çeşitli nesne sınıflarını tespit etmek için kullanılabilir olsa da temelde yaya tespit problemlerinde kullanılmaktadır.
- **Değişime Uğrayabilen Parça Bazlı Model (Deformable Part Model – DPM):** İlk olarak 2008’de HOG algoritmasının bir uzantısı olarak önerilmiştir. Bir görüntünün içinde yer alan nesnelerin özelliklerini böl ve yönet stratejisine göre çıkaran algoritmadır (Felzenszwalb vd., 2008; Felzenszwalb vd., 2010a; Felzenszwalb vd., 2010b).

İki Aşamalı Yöntemler:

- **AlexNet:** DL tarihinin en önemli ve ilk modellerinden biri olan AlexNet 7 katmanlı bir görüntü sınıflandırma algoritmasıdır. Başarılı performansları sayesinde nesne tespit işlemlerinde de DL tabanlı algoritmaların geliştirilmesinin yolunu açmıştır (Krizhevsky vd., 2012).
- **Bölge Tabanlı Evrişimli Ağ (Region Based Convolution Neural Networks – RCNN):** Derin evrişimli ağın bir görüntüdeki yüksek seviyeli özellik temsillerini öğrenebildiğini keşfeden araştırmacıların “nesne tespitini gerçekleştirebilir mi?” sorusunu sormalarının ardından nesne tespiti alanına öncülük eden DL tabanlı ilk algoritmalarından biri Girshick vd. tarafından 2014 yılında önerilmiştir. Önerilen RCNN algoritması bölge önerisi ve sınıflandırma+yerelleştirme adımlarından oluşan çok katmanlı bir Evrişimli Sinir Ağı (Convolutional Neural Networks – CNN) algoritmasıdır (Girshick vd., 2014).
- **OverFeat:** Nesne tespit işlemini DL tabanlı yöntemler ile gerçekleştiren diğer öncü algoritmadır. RCNN’deki önceden belirlenmiş bölge önerilerinin aksine farklı ölçek ve konumlarda nesneleri bulabilmek için ölçeklendirme ve kaydırma gibi teknikleri kullanır (Sermanet vd., 2014).
- **Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Fast Region Based Convolutional Neural Networks – Fast-RCNN):** RCNN algoritmasının iyileştirilmiş versiyonudur.

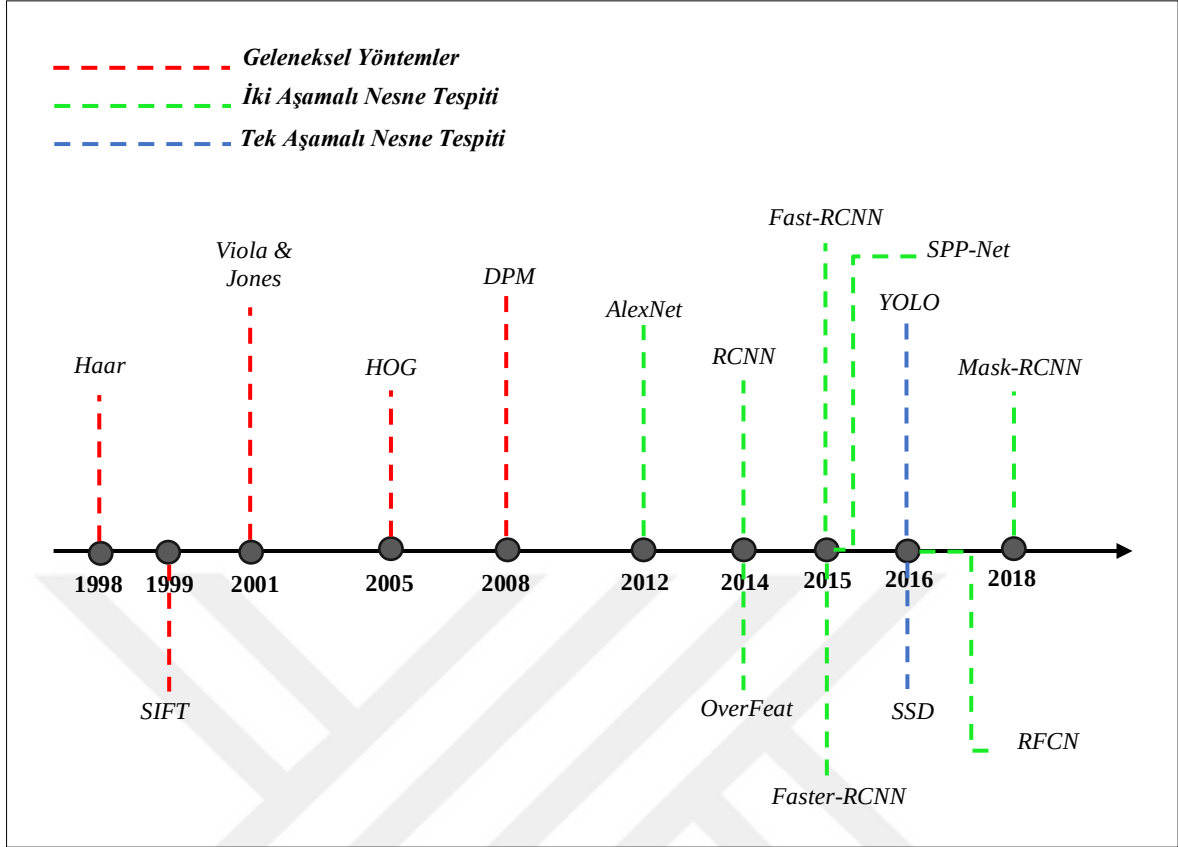
İlgili Bölge (Region of Interest -RoI) havuzlama katmanı kullanarak daha hızlı bölge tespiti yapar, depolamadan tasarruf sağlar ve maliyeti düşürür (Girshick, 2015).

- **Daha Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Faster Region Based Convolutional Neural Networks – Faster-RCNN):** Fast-RCNN algoritmasının iyileştirilmiş versiyonudur. Kullandığı bölge önerisi ağı algoritması ile daha hızlı nesne tespit işlemi gerçekleştirir (Ren vd., 2015).
- **Mekansal Piramit Havuzlama Ağı (Spatial Pyramid Pooling Networks – SPP-Net):** Değişken boyutlardaki girdi görüntülerini işleyebilme yeteneğine sahip CNN tabanlı nesne tespit algoritmasıdır (He vd., 2015).
- **Maske Bölge Tabanlı Evrişimli Sinir Ağı (Mask Region Based Convolutional Neural Networks – Mask-RCNN):** Nesne tespiti, sınıflandırma ve piksel bazlı nesne segmentasyonu gibi görevleri gerçekleştirmek için kullanılan bir CNN algoritmasıdır. Faster-RCNN mimarisine segmentasyon yönteminin eklenmiş versiyonu olarak da bilinir (He vd., 2018).
- **Bölge Tabanlı Tam Evrişimli Ağlar (Region Based Fully Convolutional Networks – RFCN):** Nesne tespiti ve konumlandırma görevlerinde geleneksel olarak kullanılan CNN algoritmasıdır. Puan haritaları oluşturarak her biri göreceli bir konuma göre hesaplanan özellik haritası sayesinde genellikle mevcut RCNN modellerinden daha başarılı çalışmaktadır (Dai vd., 2016).

Tek Aşamalı Yöntemler:

- **Sadece Bir Kez Bak (You Only Look Once – YOLO):** Görüntüdeki nesnelerin sınıfını ve koordinatlarını paralel olarak hesaplayan algoritmadır. 2016 yılından itibaren gelişmekte olan bu algoritmanın birçok modeli ve alt versiyonları bulunmaktadır (Redmon vd., 2016).
- **Tek Atışlı Multibox Dedektör (Single Shot Multibox Detector – SSD):** Temelde VGG16 mimarisi üzerine inşaa edilmiş olan SSD algoritması çok ölçekli özelliklerin çıkarılması ve katmanlardaki boyutların kademeli olarak azaltılmasına yardımcı olan bir algoritmadır. Benzer şekilde sınıflandırma ve yerelleştirme paralel olarak gerçekleştirilmektedir.

Şekil 2.1’de bu yöntemlere ait algoritmaların zaman çizelgesi detaylı bir şekilde gösterilmektedir.



Şekil 2.1: Geleneksel ve modern nesne tespit algoritmalarının zaman çizelgesi.

Bu tez kapsamında yapılan tüm uygulamalar modern yöntemler kullanılarak gerçekleştirilmektedir. Bunun temel sebebi modern yöntemlerin, el yapımı özelliklere dayalı geleneksel yöntemlere göre daha hızlı ve başarılı sonuç üretmesidir (Zou vd., 2023). Fakat bu iki yöntem kümesi arasındaki temel işleyişin anlaşılması adına çalışma kapsamında geleneksel yöntemlere de değinilmiştir.

2.1. GELENEKSEL NESNE TESPİT YÖNTEMLERİ

Geleneksel yöntemlerde; bölge seçimi, özellik çıkarımı ve nesne sınıflandırma olmak üzere üç temel bölüm yer almaktadır.

2.1.1. Bölge Seçimi

“Bölge” terimi bir görüntü içerisinde yer alan belirli bir alanı ifade etmektedir. Bu alan genellikle bir nesnenin yer aldığı ya da alabileceği piksel grubunu işaret eder. Dolayısı ile hedeflenen nesnenin tespit edilebilmesi için ilk olarak görüntü içerisinde bölge seçim

yöntemleri ile potansiyel nesne bölgelerinin belirlenmesi gerekmektedir. Geleneksel yöntemlerde bölge seçim işlemi 3 gruba ayrılabilir.

- 1. Renk ve Yoğunluk Tabanlı Yaklaşımlar:** Görüntü renk yoğunluğunun belirli eşik değerinde meydana gelen değişimlerine göre potansiyel bölge önerisini gerçekleştirir.
- 2. Kenar Tabanlı Yaklaşımlar:** Kenar algılayıcılar ile görüntüdeki kenarların tespitini gerçekleştirerek potansiyel bölge önerisinde bulunur.
- 3. Sürgülü Pencere Yaklaşımı:** Sabit boyutlu sürgülü pencere ile görüntü üzerinde belirlenen adım büyüklüğünde kaydırma işlemi gerçekleştirilerek potansiyel nesnenin boyutu ve konumu tespit edilir (Gürbüz & Gangal; 2014).

2.1.2. Özellik Çıkarımı

Bölge seçiminde belirlenen nesnenin ait olduğu sınıfı başarılı bir şekilde tespit edebilmek için renk, doku, şekil ve kenar gibi bölge özellikleri çıkartılır. Bu adımı el ile tek tek yapmak yerine Haar, SIFT, HOG ve DPM gibi teknikler tercih edilmektedir (Papageorgiou vd., 1998; Lowe 1999; Dalal & Triggs, 2005; Emrullah & Özerdem, 2016; Felzenszwalb vd., 2010). Bu yöntemler, nesneleri temsil eden özellik vektörlerini oluşturur.

2.1.3. Sınıflandırma

Özellik çıkarımı sayesinde bölgeye ait bilgiler elde edildikten sonra oluşan vektörler bir Makine Öğrenmesi (Machine Learning – ML) algoritmasına gönderilerek nesnenin sınıfı tahmin edilir. Bu kısımda genellikle Destek Vektör Makinesi (Support Vector Machine- SVM), Adaboost gibi algoritmalar tercih edilmektedir (Vapnik, 1995; Freund & Schapire, 1997).

2.2. DERİN ÖĞRENME TABANLI MODERN YÖNTEMLER

Nesne tespitinde modern yöntemler, DL tabanlı algoritmalar ile geliştirilmiş modelleri içermektedir. 2012 yılında Krizhevsky ve diğ. (2012) tarafından tasarlanan 7 katmanlı AlexNet mimarisinin önerilmesi ile DL kavramı bilgisayar bilimlerine girmiştir. AlexNet mimarisi görüntüleri sınıflandırmak için tasarlandığı bilinmesine rağmen araştırmacılar bu yöntemin nesne tespit işlemlerinde de kullanıp kullanamayacağını sorgulamaya başlamıştır. Bunun üzerine yapılan çalışmalar sonucunda 2014 yılında Bölge Tabanlı Evrişim Ağı (Region Based Convolution Neural Networks – RCNN) ve OverFeat adında iki farklı nesne tespit mimarisi tanıtılmıştır (Girshick vd., 2014; Sermanet vd., 2014). Tanıtılan bu iki mimari sayesinde nesne

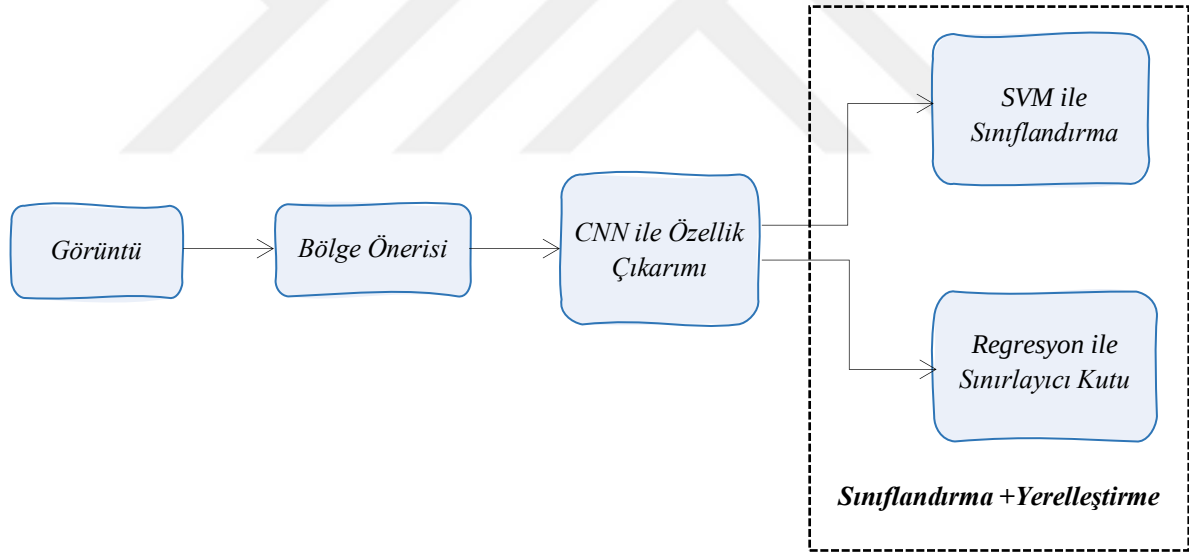
tespit işlemleri genellikle derin öğrenme yöntemlerine dayalı tekniklerle gerçekleştirilmeye başlanmıştır. Zaman içerisinde tasarlanan yeni algoritmalar ile nesne tespiti güçlü bir şekilde geliştirilmiş ve geliştirilmeye devam etmektedir. DL tabanlı modern yöntemler ise kendi içinde İki Aşamalı ve Tek Aşamalı olarak iki ana kısma ayrılmaktadır.

2.2.1. İki Aşamalı Nesne Tespiti

İki aşamalı nesne tespit algoritmaları, görüntü üzerinde nesneyi tespit edebilmek için iki adımdan oluşan bir yöntemi temsil etmektedir. Bu yöntem ile ilk olarak nesnenin bulunabileceği olası bölgeler önerilmektedir. Ardından önerilen bölge üzerinde sınıflandırma ve yerleştirme işlemleri gerçekleştirilmektedir (Du vd., 2020).

2.2.1.1. RCNN

RCNN, Girshick ve diğ. (2014) tarafından geliştirilmiş iki aşamalı nesne tespit mimarisidir. Bu mimari Şekil 2.2’de gösterildiği gibi temelde bölge önerisi, özellik çıkarımı ve sınıflandırma + yerleştirme olmak üzere 3 ana bölümden oluşmaktadır.



Şekil 2.2: RCNN mimarisi.

Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranır.

Bölge önerisi: Bir görüntüdeki olası nesnelerin yerini belirleme işlemidir. Bunun için çok sayıda öneride bulunulur. Öneri seçeneklerini daraltmak için seçilen bölgeler Seçici Arama (Selective Search – SS) algoritması ile 2000 aday bölgeye indirgenir (Uijlings vd., 2013; Du vd., 2020). Bu indirgeme işlemi Şekil 2.3’te gösterildiği gibi birbirine benzeyen bölgeler

birleştirilerek gerçekleştirilmektedir (Uijlings vd., 2013). SS algoritmanın çalışma prensibi aşağıdaki gibi ifade edilebilir:

- Görüntü bölütlenerek birçok aday bölge belirlenir.
- Açgözlü özelliğe sahip bir algoritma kullanılarak benzer renklerdeki bölgeler tekrar tekrar birleştirilir.
- Nihai aday bölge önerilerini üretmek için oluşturulan bölgeler kullanılır.



Şekil 2.3: SS algoritması ile bölge indirgeme.

CNN ile Özellik Çıkarımı: Bölge önerisi ile belirlenen yaklaşık 2000 adet aday bölge ayrı ayrı CNN modeline gönderilerek her bir bölge üzerinde sabit uzunlukta özellik çıkarır (Girshick vd., 2014).

SVM ile Sınıflandırma: Evrişimli katmanlardan elde edilen özelliklerin doğrusal SVM algoritmasına gönderilerek gerçekleştirilen sınıflandırılma işlemidir.

Regresyon ile Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir (Girshick vd., 2014).

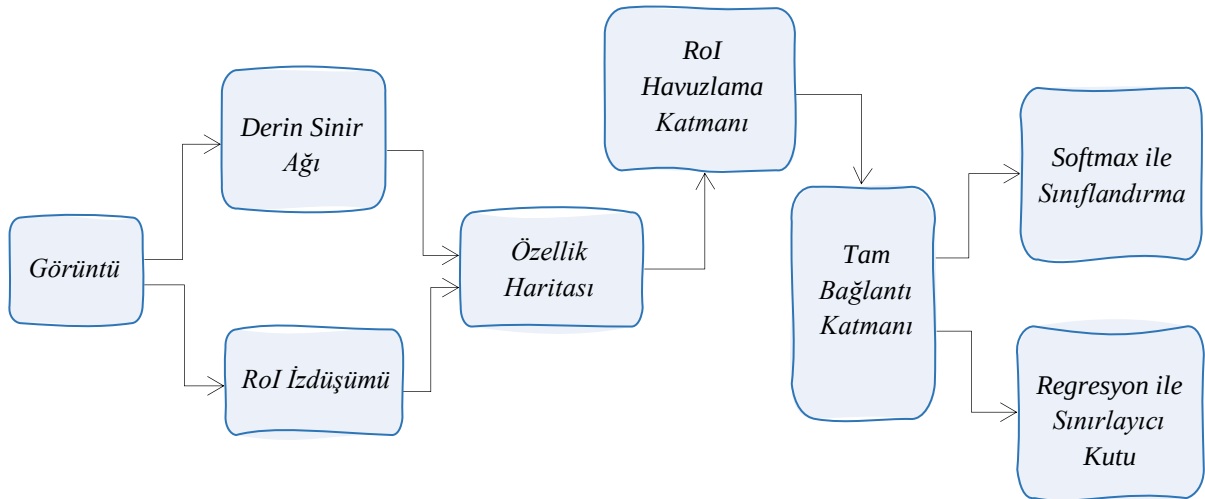
RCNN modeli derin öğrenme tabanlı nesne tespit algoritmalarının atası olarak bilinmektedir. Geleneksel nesne tespit algılarına göre daha başarılı sonuçlar ürettiği bilinse de bazı dezavantajlara sahip olduğu da bilinmektedir (Zou vd., 2019). RCNN mimarîsinin dezavantajları:

- Görüntüler bölge önerisi ile saklanması gerektiği için çok fazla depolama alanı gerektirmektedir (Girshick vd., 2014).

- Her görüntüden 2000 aday bölge önerisinin sınıflandırması, zaman kaybına sebep olmaktadır (Girshick vd., 2014).
- Model sadece 227×227 boyutunda girdi verisi alabildiği için nesne bilginin kaybolmasına yol açabilmektedir (Zou vd., 2019; Xiao vd., 2020).
- Bölge önerisi çıkarımı sırasında gereksiz birçok teklif ürettiği için zaman karmaşıklığının artmasına sebep olmaktadır (Mohan, 2020; Girshick vd., 2014).

2.2.1.2. Fast-RCNN

Fast-RCNN, RCNN'nin tasarımcısı Girshick tarafından 2015 yılında RCNN temelinde iyileştirme yapılarak tasarlanmıştır. RCNN mimarisinin 2000 adetlik aday bölge üzerinde işlem yapması zaman ve maliyeti arttırdığı için tasarlanan Fast-RCNN mimarisi görüntüyü bölge tavsiyelerine göre ayırmak yerine tüm görüntü ve bölge tavsiyelerini CNN mimarisine girdi olarak vermektedir. Şekil 2.4'te gösterildiği gibi görüntü CNN modeline gönderilerek özellik haritası üzerinde bölge önerilerinin izdüşümleri alınmaktadır. Buradan elde edilen bilgiler RoI havuzlama katmanına gönderilerek her bölge önerisi için sabit uzunlukta (3×3 , 5×5 , 7×7) bir özellik vektörü elde edilmektedir. Her özellik vektörü önce tamamlama katmanına sonra sınıflandırma için Softmax ve sınırlayıcı kutu için regresyon modeline gönderilmektedir. Böylece özellik haritalarını ayrıca saklamaya gerek kalmadığı için depolamadan da tasarruf sağlamaktadır (Girshick, 2015).



Şekil 2.4: Fast-RCNN mimarisi.

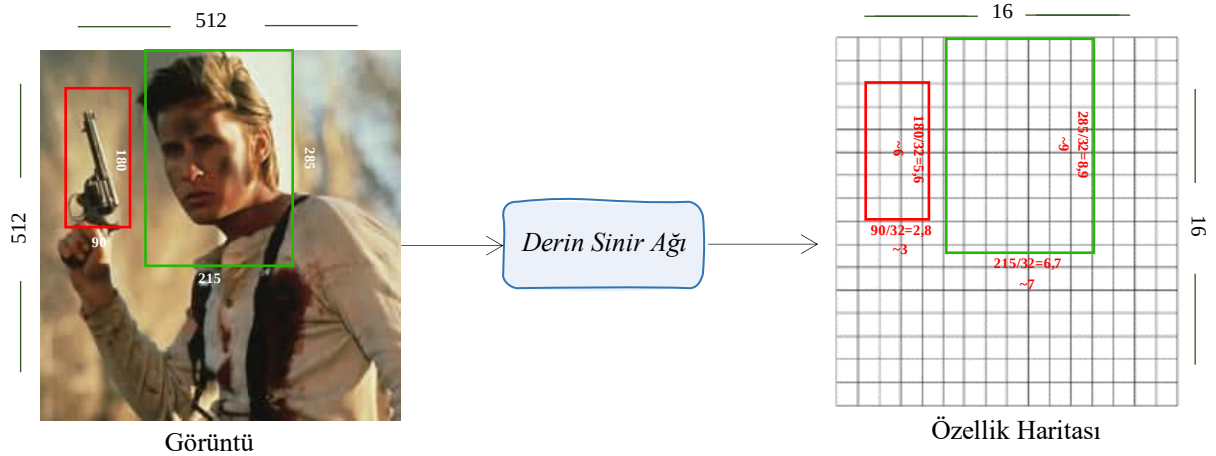
Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranır.

Derin Sinir Ağı: Çok katmanlı bir yapıya sahip olan DL modelidir. Verileri karmaşık ve yüksek düzeyde soyutlamalarla temsil etmek için birçok evrişim katmanı içerir.

RoI İzdüşümü: Görüntüden önerilen bölgeler RoI olarak adlandırılır. RoI izdüşümü ise nesnelerin özellik haritasına yansımalarını ifade eder.

Özellik Haritası: Girdi görüntüsünün farklı özelliklerini temsil eden bir dizi haritadır. Bu özellik haritaları, derin öğrenme modelinin daha yüksek seviyeli özelliklerini öğrenmek için kullanılır.

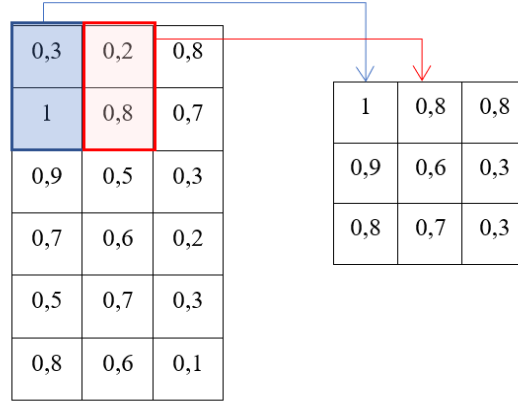
RoI Havuzlama Katmanı: Görüntü üzerinde nesne olabilecek olası bölgeleri çıkartmakla görevli bir katmandır. Orijinal görüntü bir derin sinir ağına (VGG16, ResNet-50 vs.) verildiğinde ilk olarak özellik haritası çıkartılır. Elde edilen bu özellik haritası üzerinde Şekil 2.5'te gösterildiği gibi olası nesnelerin izdüşümleri bulunarak RoI'ler elde edilir. Harita üzerindeki konumu ve sınırlayıcı kutunun büyüklüğü özellik haritası boyutunda hesaplanır. Bu yapının tam bağlantı katmanına gönderilebilmesi için farklı boyuttaki RoI'lerin aynı boyutta olması gerekmektedir. Şekil 2.5'te görüldüğü gibi belirlenen nesnelerin büyüklükleri birbirinden farklıdır. Her birinin sabit boyutta olabilmesi tam bağlantı katmanı için önemlidir. Bu sebepten elde edilen tüm RoI'ler, havuzlama katmanından geçirilerek eş boyutlara indirgenmektedir. RoI havuzu temelinde maksimum havuzlama yaparak farklı boyutlardaki girdileri sabit boyutlu (genelde 7×7) özellik haritalarına dönüştürme işlemidir.



Şekil 2.5: Özellik haritasında RoI'lerin izdüşümü.

Şekil 2.6'da bir RoI'nin havuzlama katmanından geçirilmesine ilişkin örnek görsel yer almaktadır. Şekil 2.5'te yer alan silah bölgesinin ya da başka bir deyiş ile RoI'nin harita üzerindeki izdüşümü $\sim 3 \times 6$ boyutunda olduğu görülmektedir. Bu RoI için özellik haritası 3×3 olarak belirlenmiştir. Görüntüden bu boyutta bir özellik haritası elde etmek için uygun

boyutlu filtre (bu örnek için 1×2) hesaplanarak maksimum özellikli havuzlama katmanına iletilir. Ardından elde edilen $3 \times 3 \times 512$ boyutlu bilgi, tam bağlantı katmanına gönderilir (Girshick, 2015).



Şekil 2.6: $\sim 3 \times 6$ boyutundaki RoI'nin havuzlama katmanından sonraki dönüşümü.

Özellik çıkarımı sırasında RoI'nin harita üzerindeki izdüşümü piksel bazında belirlenirken bazı veri kayıpları yaşanabilmektedir. Bu veri kayıpları piksellerin tam sayı ile ifade edilmesinden kaynaklanmaktadır. Bu sebepten RoI boyutları belirtilirken yaklaşık ($\sim$) ifadesi kullanılmıştır. Özellik çıkarımı sırasında oluşabilen bilgi kayıplarının yanı sıra bilgi kazanımları da yaşanabilmektedir. Fakat kayıplar nesne sınıflandırma için daha önemli olduğundan bu durumun önüne geçip daha hassas özellik haritalarının çıkartıla bilinmesi için nicelme yöntemi kullanmayan RoIAlign yöntemi tercih edilebilir (Erdem, 2020).

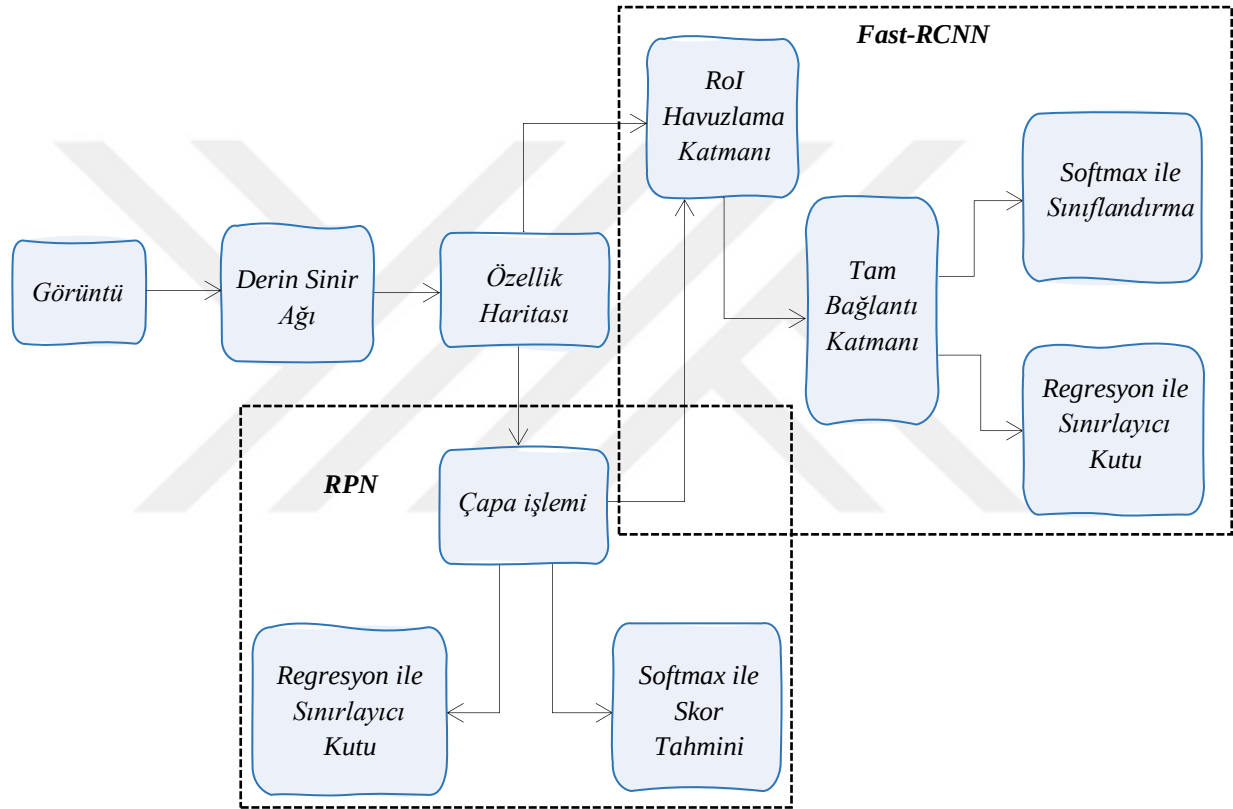
Tam Bağlantı Katmanı: Bir katmanın tüm çıkışlarını bir sonraki katmanın tüm girdilerine bağlandığı katmandır.

Softmax ile Sınıflandırma ve Regresyon: Tam bağlantı katmanından sonra düzleştirme katmanına gönderilerek tek boyutlu vektöre dönüştürülen bilgiler sınıflandırma için Softmax, sınırlayıcı kutu için doğrusal regresyon algoritmasına gönderilmektedir.

Fast-RCNN, RCNN'den daha yüksek algılama kalitesine ya da başka bir deyişle Ortalama Hassasiyete (Mean Average Precision – mAP) sahip olduğu bilinmektedir. Özellikle depolama alanından tasarruf ettiği ve görüntü başına yalnızca bir işlem yaptığı için daha hızlı eğitim sürecinin olduğu bilinmektedir. Fakat Fast-RCNN, RCNN'e benzer şekilde, bölge önerilerini bulmak için SS algoritması kullanmaktadır. Dolayısıyla RCNN'den daha hızlı olsa bile hala yavaş ve zaman alıcı bir sürece sahip olduğu bilinmektedir.

2.2.1.3. Faster-RCNN

Faster-RCNN, Ren ve diğ. tarafından 2015 yılında RCNN’de bir darboğaz haline gelen bölge önerisinin maliyetini ortadan kaldırmak için tasarlanmış Fast-RCNN tabanlı bir mimarıdır. Ren ve diğ. (2015) Bölge Öneri Ağını (Region Proposal Network – RPN) tasarlayarak bölge önerilerinin ürettiği maliyeti yok denecek kadar azaltmayı ve bölge öneri süresini 2 saniyeden 10 milisaniye’ye düşürmeyi başarmıştır. Fast-RCNN Şekil 2.7’de gösterildiği gibi RPN ve Fast-RCNN mimarîlerinin birleştirilmiş halini temsil etmektedir.



Şekil 2.7: Faster-RCNN mimarîsi.

Görüntü: Dijital formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranır.

Derin Sinir Ağı: Çok katmanlı bir yapıya sahip olan DL modelidir. Verileri karmaşık ve yüksek düzeyde soyutlamalarla temsil etmek için birçok evrişim katmanı içermektedir.

Özellik Haritası: Girdi görüntüsünün farklı özelliklerini temsil eden bir dizi harita içerir. Bu özellik haritaları, derin öğrenme modelinin daha yüksek seviyeli özelliklerini öğrenmek için kullanılır.

RPN: Nesne sınırlarını ve her bir konumdaki puanları aynı anda tahmin etme yeteneğine sahip olan tamamen evrişimli bir ağıdır. Herhangi bir ölçekteki görüntüden bölge önerileri çıkararak, bölgelerin doğruluğu için skor ataması gerçekleştirmektedir. RPN yönteminin temelinde çapa (anchor) kutuları yer almaktadır. Bu kutular sayesinde görüntüde algılanan farklı boyuttaki nesnelerin olası konumları tespit edilebilmektedir. Uçtan uca eğitilebilir bir model olduğu için SS algoritmasına göre daha iyi bölge önerileri sunmaktadır. Ayrıca RPN ve Fast-RCNN mimarisinde aynı evrişimli katmanlar kullanıldığı için eğitim tek seferde gerçekleşmektedir. Bu sayede Faster-RCNN mimarisi bölge önerisinde SS algoritması kullanan diğer RCNN mimarilerine göre daha hızlı bölge önerileri üretebilmektedir (Ren ve diğ., 2015).

Çapa İşlemi: Her bir özellik haritası hücrelerine önceden belirlenmiş farklı boyut ve oranlarda çapa kutuları yerleştirilir. Temel amacı görüntü içinde nesne içerebilecek yerlerde sınırlayıcı kutular oluşturmak ve sınıf olasılıklarını tahmin etmektir. Bu sayede daha hassas hesaplama yapılabilir.

Softmax ile Skor Tahmini: Her bir sınırlayıcı kutu içinde nesne varlığının olasılığının tahminlenmesi işlemidir.

Regresyon ile Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

RoI havuzlama katmanı: Görüntü üzerinde nesne olabilecek olası bölgeleri çıkartan katmandır.

Tam Bağlantı Katmanı: Bir katmanın tüm çıkışlarını bir sonraki katmanın tüm girdilerine bağlandığı katmandır.

Softmax ile Sınıflandırma: Evrişimli katmanlardan elde edilen özelliklerin Softmax algoritmasına gönderilerek gerçekleştirilen sınıflandırılma işlemidir.

Faster-RCNN modelinin çalışma şekli aşağıdaki gibi ifade edilebilir:

- Görüntünün herhangi bir derin sinir ağı (VGG16, ResNet50, AlexNet vs.) ile özellik haritası çıkartılır.
- RPN katmanına gönderilen özellik haritası $n \times n$ ($3 \times 3, 5 \times 5$ vs.) boyundaki filtre ile N (224, 512 vs.) adet evrişim katmanından geçirilir.
- Elde edilen bilgiler olası nesnenin tespiti için Softmax katmanına ve sınırlayıcı kutu için regresyon katmanına gönderilir.
- RPN katmanından çıkan veriler ve derin sinir ağından elde edilen özellik haritası, RoI havuzlama katmanına gönderilir.

- Farklı boyutlardaki verilerin tamamlama katmanı tarafından anlaşılabilirliği için her bölgeden sabit uzunlukta özellik vektörü çıkarılır.
- Çıkarılan özellik vektörleri Fast-RCNN'deki Softmax katmanı sayesinde sınıflandırılır, regresyon katmanı sayesinde ise sınırlayıcı kutular çizilir.

2.2.1.4. SPP-Net

SPP-Net, He ve diğ. tarafından 2015 yılında sabit boyutlu girdilere mecburiyeti ortadan kaldırmak için geliştirilen bir nesne tanıma mimarîsidir. Bu mimarînin temel amacı görüntü boyutu ne olursa olsun çıktı olarak sabit uzunlukta bir temsil oluşturabilmektir. Klasik CNN mimarîleri giriş verisi olarak görüntüleri sabit boyutta almaktadır. Fakat bu yaklaşımı gerçekçi bulmayan araştırmacılar tam bağlantı katmanından kaynaklanan sabit boyutlu girdi problemini ortadan kaldırmışlardır.

Mekansal piramit havuzlama: Rastgele boyutlarda olan görüntüler ile ağı besleyebilmek için son havuzlama katmanı ile mekansal piramit havuzlama katmanı yer değiştirilir. SPP katmanları yalnızca bir havuzlama işlemi uygulamaz. Birkaç farklı çıktı boyutunda havuzlama işlemi uygular ve sonuçları bir sonraki katmana göndermeden önce birleştirir. Bu sayede tam bağlantılı katman için (veya diğer sınıflandırıcılara) sabit uzunlukta çıktılar üretilir (He vd., 2015)

RCNN'deki özellik hesaplaması görüntü başına binlerce bölgenin ham piksellerine tekrar tekrar uyguladığı için zaman alır, oysa SPP-Net evrişim katmanlarını yalnızca bir kez çalıştırdığı için daha hızlıdır. Ayrıca RCNN'ye kıyasla sınırlayıcı kutu tahmin hızı da daha yüksektir. Bu iyileştirmelere rağmen mevcut dezavantajları aşağıdaki gibi ifade edilebilir:

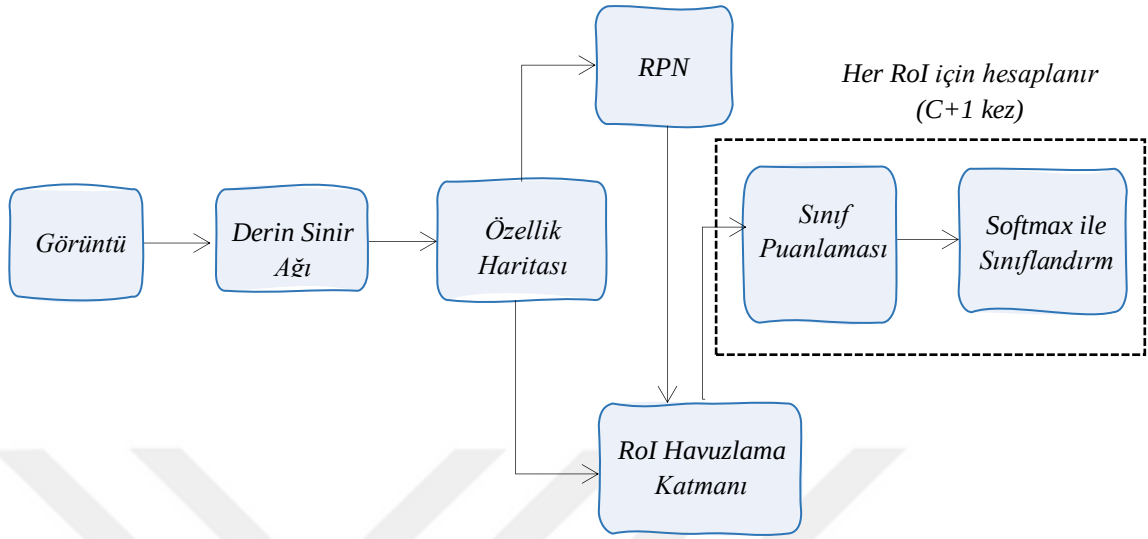
- Özellik çıkarma ve ağı ince ayarlarının yapılandırması zaman almaktadır.
- Bölge önerilerinden dolayı fazla yer kaplamaktadır.

RCNN gibi, özelliklerin çıkarılması, ince ayar yapılması, SVM ile eğitmeyi ve son olarak sınırlayıcı kutu yerleştirmeyi içeren çok aşamalı bir işlem hattından oluşmaktadır (Girshick, 2015).

2.2.1.5. RFCN

RFCN, Dai ve diğ. tarafından 2016 yılında tasarlanmış verimli bir nesne tanıma mimarîsidir. Özellik çıkarımı sırasında bir dizi özel evrişimsel katman ile konuma duyarlı puan haritası oluşturmaktadır. Öneri bölgelerinin tamamen evrişimli bir ağ içinde bulunan evrişim katmanlarından geçirilmesi sayesinde nesne konumlarının daha doğru bir şekilde belirlenmesi sağlanır. Bu sayede daha etkili bir sınıflandırma gerçekleştirerek Fast-RCNN ve

Faster-RCNN'den daha hızlı ve verimli çalışmaktadır. RFCN Şekil 2.8'de gösterildiği gibi RPN ve RoI havuzlama katmanının birleştirilmiş halini temsil etmektedir.



Şekil 2.8: RFCN mimarisi.

Görüntü: Dijital formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranır.

Derin Sinir Ağı: Çok katmanlı bir yapıya sahip olan DL modelidir. Verileri karmaşık ve yüksek düzeyde soyutlamalarla temsil etmek için birçok evrişim katmanı içermektedir.

Özellik Haritası: Girdi görüntüsünün farklı özelliklerini temsil eden bir dizi harita içerir. Bu özellik haritaları, derin öğrenme modelinin daha yüksek seviyeli özelliklerini öğrenmek için kullanılır.

RPN: Nesne sınırlarını ve her bir konumdaki puanları aynı anda tahmin etme yeteneğine sahip olan tamamen evrişimli bir ağıdır.

RoI havuzlama katmanı: Görüntü üzerinde nesne olabilecek olası bölgeleri çıkartmak için kullanılır.

Softmax ile Sınıflandırma: Tam bağlantı katmanından sonra düzleştirme katmanına gönderilerek tek boyutlu vektöre dönüştürülen bilgilerin Softmax katmanı ile sınıflandırılması işlemidir. Her nesnenin sınıfı bu katmanda belirlenir.

Sınıf Puanlaması: RPN işleminden sonra RoI havuzlama katmanına gönderilen bilgiler vasıtasıyla bir aday bölge belirlenir. Aday bölgenin 3x3'lük haritası oluşturulup skor haritası elde edilir. Skor haritasının aritmetik ortalaması alınarak sınıf puanı elde edilir.

Bu mimarînin çalışma prensibi şu şekilde ifade edilebilir:

- Görüntünün ResNet derin öğrenme ağı ile özellik haritası çıkartılır.

- Özellik haritası RPN ve RFCN arasında özellikler paylaşılır.
- RPN ile özellik haritalarında aday bölgeler belirlenir.
- Belirlenen her bir aday bölgenin 3×3 'lük haritası çıkartılır. Bu harita sol üst, orta üst, sağ üst, sol orta, sağ alt alanını algılayan 9 bölge tabanlı özellik haritasıdır.
- Elde edilen 3×3 'lük özellik haritasındaki hücrelerin ortalaması alınarak nesneyi algılama puanı elde edilir.
- C adet sınıf varsa, nesne olmayan yerler içinde bir sınıf eklenerek toplamda $C + 1$ tane 3×3 'lük harita oluşturulur ve $C + 1$ tane sınıf puanı elde edilir.
- Ardından, her bir sınıfın olasılığını hesaplamak için elde edilen puanlar Softmax algoritmasına gönderilir.
- RPN işleminden sonra RoI havuzlama katmanına gönderilen bilgiler burada bir süzgeçten geçirilir. Belirlenen bir aday bölgenin 3×3 'lük özellik haritasının elde edilip sınıf puanının hesaplanması sağlanır.

2.2.1.6. Mask-RCNN

Mask-RCNN, He ve diğ. tarafından 2018 yılında geliştirilmiş bir evrimsel sinir ağı mimarîsidir. Faster-RCNN'e segmentasyon işleminin eklenmesiyle elde edilen Mask-RCNN mimarîsi içerisinde Semantik ve Örnek olmak üzere 2 ana görüntü segmentasyon türü bulunmaktadır.

Şekil 2.9'da gösterildiği gibi Mask-RCNN temelde Faster-RCNN'de yer alan iki çıkışa ek olarak üçüncü bir çıkışın eklenmesi ile geliştirilmiştir. Bilinen çıkışlardan biri aday nesne, diğeri ise sınırlayıcı kutudur. Mask-RCNN ile bu çıkışlara ek olarak nesneye ait maskenin yer aldığı üçüncü bir çıkış eklenmiştir. Mask-RCNN mimarîsinin çalışma şekli iki aşamada açıklanabilir.

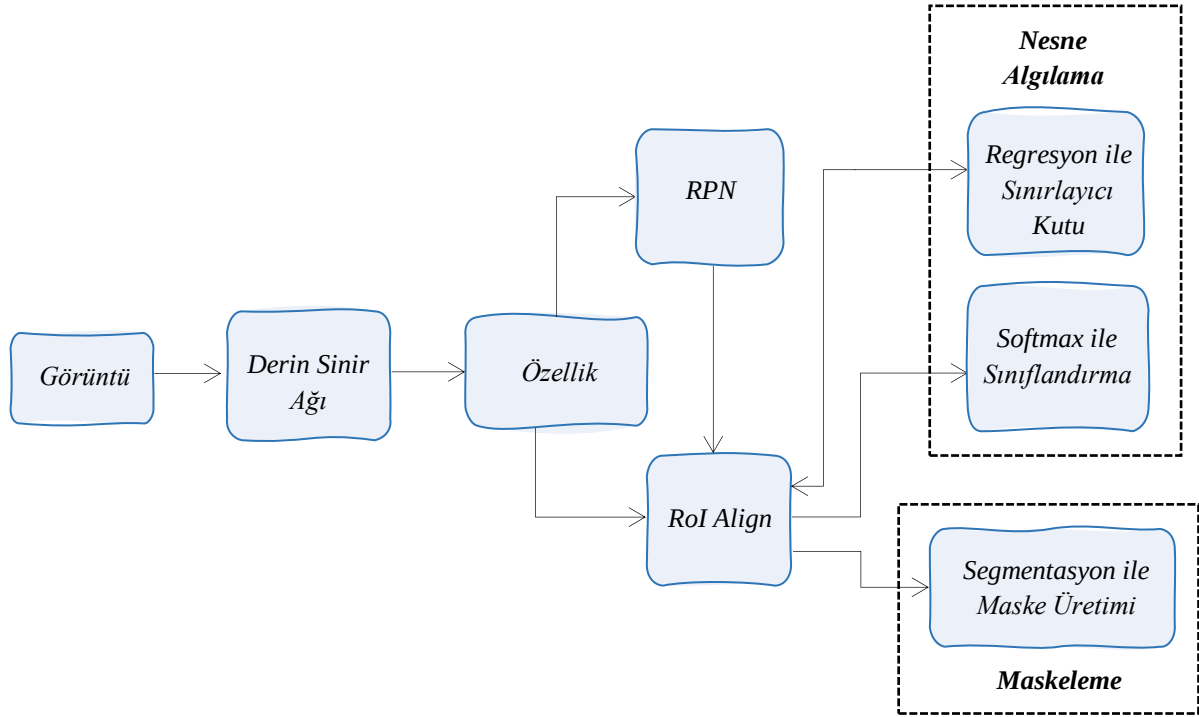
Birinci aşama:

- Görüntünün bir derin öğrenme ağı ile özellik haritası çıkarılır.
- Özellik haritası kullanılarak RPN ile görüntüdeki RoI'ler tespit edilir.

İkinci aşama:

- Önerilen bölgelerin her biri için sınırlayıcı kutular ve nesne sınıfının tahmini gerçekleştirir.
- Paralel olarak her bir RoI için $m \times m$ boyutlu ikili maske çıktısı üretilir.

Önerilen her bölge farklı boyutta olabilmektedir fakat ağlardaki tam bağlantı katmanları, tahminde bulunmak için her zaman sabit boyutlu vektör oluşturur. Bu yüzden önerilen bu bölgelerin boyutu, RoIAlign yöntemi kullanılarak sabitlenir.



Şekil 2.9: Mask-RCNN mimarîsi.

Görüntü: Dijital formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranır.

Derin Sinir Ağı: Çok katmanlı bir yapıya sahip olan DL modelidir. Verileri karmaşık ve yüksek düzeyde soyutlamalarla temsil etmek için birçok evrişim katmanı içermektedir.

Özellik Haritası: Girdi görüntüsünün farklı özelliklerini temsil eden bir dizi harita içerir. Bu özellik haritaları, derin öğrenme modelinin daha yüksek seviyeli özelliklerini öğrenmek için kullanılır.

RoIAlign: Bu işlem RoI havuzlama işlemine benzerdir. Aralarındaki temel fark ise RoIAlign'in niceleme gerçekleştirmemesidir. Niceleme işlemi büyük bir değer kümesindeki girdiyi ayrı bir kümeye sınırlama (yuvarlama) işlemidir. Sınırlama işlemi kolay hesaplama yapılmasını sağlarken veri kaybetmeye sebep olmaktadır. Bu durumu aşabilmek için RoIAlign yönteminde filtre boyutuna göre (3×3 ise 9, 4×4 ise 16 kutu vs.) tespit edilen nesneyi eşit parçalara bölme işlemi gerçekleştirilerek her birinin içine çift enterpolasyon uygulanmaktadır.

RoIAlign, ortalama olarak daha iyi hassasiyet sağladığı için nesne tespit işlemleri daha başarılı bir şekilde gerçekleştirilebilmektedir (Bozdağ, 2021).

Regresyon ile Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

Softmax ile Sınıflandırma: Tam bağlantı katmanından sonra düzleştirme katmanına gönderilerek tek boyutlu vektöre dönüştürülen bilgiler sınıflandırma için Softmax katmanına gönderilir. Her nesnenin sınıfı bu katmanda belirlenir.

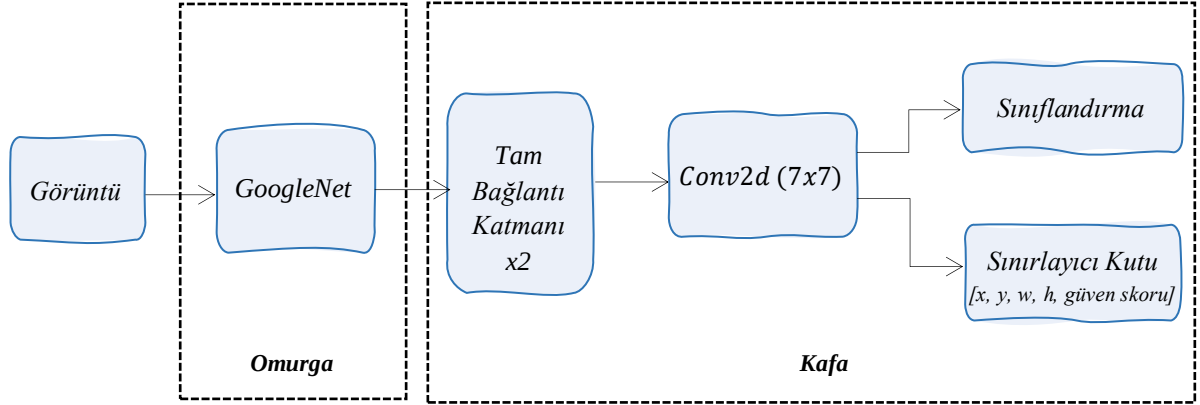
Segmentasyon ile Maske Üretimi: Segmentasyon, bir görüntüyü anlamlı bölgelere ayırma işlemidir. Bu katman algılanan her bir nesne için ayrı bir segmentasyon maskesi oluşturur. Maske, nesnenin görüntü üzerinde tam olarak nerede olduğunu ve hangi şekle sahip olduğunu belirtir. Bu maskeler, nesnenin sınırlarını ve şeklini çok daha ayrıntılı bir şekilde belirlenmesine yardımcı olur.

2.2.2. Tek Aşamalı Nesne Tespiti

Tek aşamalı nesne tespiti, nesne sınıflandırma ve algılama işlemlerini doğrudan görüntü seviyesinde gerçekleştiren yöntemdir. Nesnenin sınıflandırma doğruluğunu ve koordinat konumunu doğrudan elde edebildiği için hızlı tahmin yapma yeteneğine sahiptir (Popescu & Coatrieux, 2020).

2.2.2.1. YOLOv1

YOLO mimarîsi, Redmon ve diğ. tarafından 2016 yılında geliştirilen gerçek zamanlı nesne tespit mimarîsidir. İki aşamalı nesne tespit mimarîlerindeki bölge önerisi ve sınıflandırma + yerelleştirme adımlarını birleştirerek işlem yükünü azalmıştır. Bununla birlikte çalışma hızını iyileştirerek gerçek zamanlı tespit uygulamalarının gelişimine olumlu yönde katkı sağlamıştır (Redmon vd., 2016). YOLOv1 mimarîsi Şekil 2.10'da gösterildiği gibi omurga (backbone) ve kafa (head) kısmından meydana gelmektedir. YOLOv1, GoogleNet mimarîsinden ilham alarak geliştirilmiş 24 evrişimli ve 2 tam bağlantı katmanına sahip bir mimarîdir. İlk 20 katman bir omurga görevi görürken kalan katmanlar kafa görevi görmektedir.



Şekil 2.10: YOLOv1 mimarîsi.

Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

GoogleNet: Görüntü sınıflandırma, nesne tespiti gibi görevlerde kullanılan CNN tabanlı mimarîdir. YOLOv1 mimarîsinde omurga ağı olarak kullanılmaktadır.

Tam Bağlantı Katmanı: Bir katmanın tüm çıktılarını bir sonraki katmanın tüm girdilerine bağlandığı katmandır.

Conv2d: Görüntü üzerinde bir filtre veya çekirdek matrisini kaydırarak çıktı matrisi üreten işlemdir.

Sınıflandırma: Her nesnenin sınıfı bu katmanda belirlenmektedir.

Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

YOLO'nun temelini oluşturan YOLOv1 mimarîsi şu şekilde çalışmaktadır:

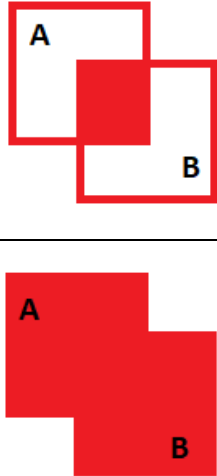
- Görüntü $S \times S$ boyutunda eşit büyüklükte ($3 \times 3, 5 \times 5, 7 \times 7$ gibi) parçalara (ızgara/hücre) bölünür. Şekil 2.12'de 512×512 boyutunda bir örnek gösterilmektedir.
- Hedef nesnenin orta noktası hesaplanarak denk geldiği hücre tespit edilir.
- Hedef nesnenin merkezinin bulunduğu hücre artık o nesnenin algılanmasından sorumlu olur. Bu sayede diğer hücrelerin nesneyi dikkate almaması sağlanarak çalışma maliyeti azaltılır.
- Hücreler, her nesnenin sınıfını belirlemek için tek bir evrimsel sinir ağından geçirilir.
- YOLO her hücre için ayrı bir tahmin vektörü $[x, y, w, h, \text{güven skoru}]$ oluşturur.
- Hücreler ağıdan geçirilirken nesnenin orta noktası hücre içinde olup olmadığı kontrol edilir.

- Eğer nesnenin orta noktası hücre içinde ise başka bir deyiş ile güven skoru 0'dan yüksek ise, tespit edilen nesnenin sol köşesinden merkeze olan konumu (x, y) , yüksekliği (h) ve genişliği (w) , hesaplanır. Eğer güven skoru 0 ise hücre içinde nesne olmadığı anlamına gelir bu sebepten diğer alanlara rastgele değerler atanabilir.
- Her bir bölgedeki nesneleri çevreleyen sınırlayıcı kutular hesaplanan koordinatlar $[x, y, w, h, \text{güven skoru}]$ sayesinde çizilir.
- Modelin ürettiği vektör çıktısı $SxSx(5xB + C)$ şeklinde hesaplanır. B değeri sınırlayıcı kutu sayısını verirken C değeri sınıf sayısını verir.
- Öngörülen sınırlayıcı kutu gerçek kutuyla aynıysa, Birleşim Üzerinden Kesişme (Intersection Over Union – IOU) 1'e eşittir. Gerçek kutuya eşit değilse sınırlayıcı kutular ortadan kaldırılır.

Güven Skoru: Sınırlayıcı kutu ile çerçevelenen hücre içindeki nesnenin içeriğinden ne kadar emin olduğunu gösteren skor türüdür. Başka bir deyiş ile hücrede nesne olup olmadığını söyleyen puan değeridir. Eğer hücrede nesne bulunmuyorsa skor değeri 0'dır.

P(nesne): Hücre içinde nesnenin bulunma olasılığını temsil eder.

IoU: Gerçekte nesnenin bulunduğu kutu ile tahmin edilen kutunun kesişimini temsil ederek modelin tahmin yeteneğini hesaplamaktadır. Şekil 2.11'de gösterildiği gibi sınırlayıcı kutular arasındaki örtüşme alanı, birleşim alanına bölünerek 0-1 arasında bir değer hesaplanmaktadır. Hesaplanan değer 0'a yaklaştıkça modelin tahmin doğruluğu arttığı bilinmektedir (Rahman; 2022). Ayrıca bu elemanların arasındaki ilişki aşağıdaki Denklem (2.11) ile ifade edilmektedir.

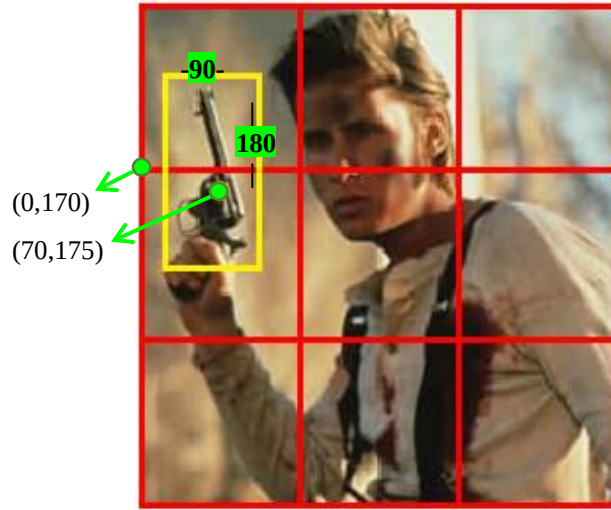


$$IoU = \frac{\text{Örtüşme Alanı}}{\text{Birleşme Alanı}} = \frac{A \cap B}{A \cup B}$$

Şekil 2.11: IoU metriğinin hesaplanması.

$$\text{Güven skoru} = P(\text{nesne}) \times \text{IoU} \quad (2.1)$$

Şekil 2.12’de gösterilen görselde silah nesnesinin orta noktası 4. hücrede yer almaktadır. Bu hücre içerisinde başka bir nesnenin de orta noktasının bulunması halinde sınırlayıcı kutunun sadece güven skoru yüksek olan için çizilmesi gerekmektedir (He vd., 2017). Bu durumda var olan diğer nesnenin sınırlayıcı kutusu göz ardı edilerek çizilmeyecektir. Şekil 2.12’deki görüntü 3x3’lük hücrelere bölünmüştür. Silah nesnesinin orta noktası ve koordinatları hesaplanmıştır. Sınırlayıcı kutu ile çerçevelenmiştir.



Şekil 2.12: Hedef nesne için sınırlayıcı kutu çizimi.

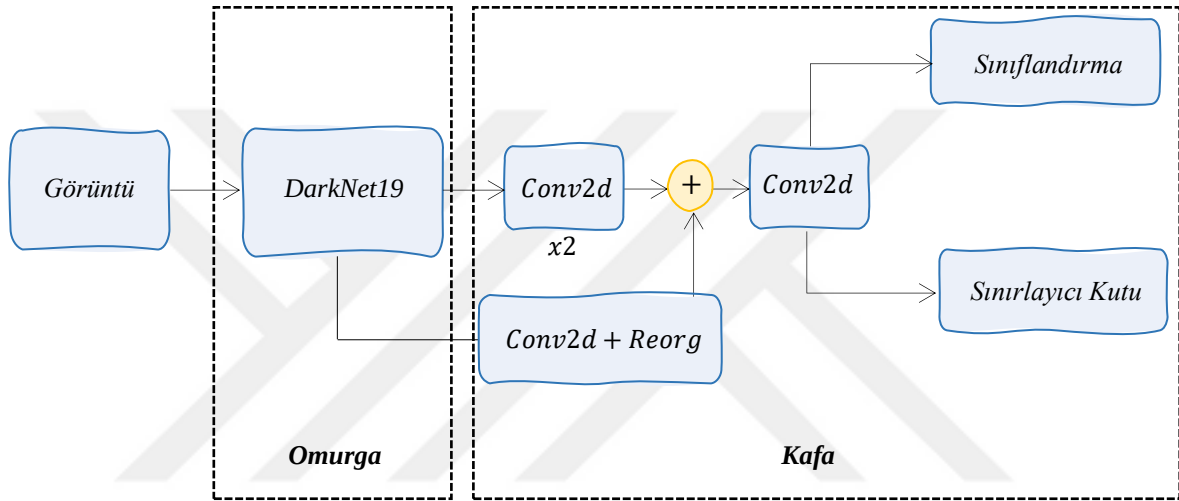
Mevcut görüntüde iki nesnenin hücre içinde bulunması bu nesnelerden sadece birini dikkate almayı gerektirmektedir. Bu durumda hedeflenen nesnelerden bazıları tespit edilemeyerek model performansını düşürmektedir. Bu durumda YOLOv1’in dezavantajları aşağıdaki gibi ifade edilebilir:

- Küçük nesneleri ya da birbirine yakın nesneleri tespit etmekte zorlanmaktadır.
- Genelleme yeteneği zayıftır (aynı nesnenin farklı boyutları için sınıflandırma yapamamaktadır).
- Birden fazla nesnenin orta noktası aynı hücrede ise birini seçmek zorunda kalmaktadır.

2.2.2.2. YOLOv2

Redmon ve Farhadi (2017) tarafından bir sonraki yıl YOLO9000 diğer adıyla YOLOv2 mimarisi tanıtılmıştır. Tanıtılan yeni mimarînin temel amacı modelin mevcut hızını ve

sınıflandırma doğruluğunu korurken önceki eksikliklerin giderilmesini ve yerleştirme başarısının iyileşmesini sağlamaktır. Bu amaçla ağ yüksek çözünürlüklü görüntülerde daha iyi çalışması için 448×448 boyutlu görüntüler ile eğitilmiştir. Ağ mimarisinde 19 evrişim katmanı ve 5 maksimum havuzlama katmanı olmak üzere toplamda 24 katman bulunmaktadır. Evrişim katmanları 3×3 ve 1×1 boyutundadır. YOLOv2 kafası 4 evrişim katmanı ve bir Reorg katmanından oluşmaktadır. Reorg katmanı, yeniden düzenlenmiş bir özellik haritası elde etmek için Darknet19'un 13. evrişim katmanının çıktısı özellik haritasını yeniden düzenlemektedir. YOLOv2 mimarisi Şekil 2.13'te gösterilmektedir.



Şekil 2.13: YOLOv2 mimarisi.

Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

DarkNet19: 19 evrişim katmanlı CNN mimarisi. YOLOv2 mimarisinde omurga ağı olarak kullanılmaktadır.

Tam Bağlantı Katmanı: Bir katmanın tüm çıkışlarını bir sonraki katmanın tüm girdilerine bağlandığı katmandır.

Conv2d: Görüntü üzerinde bir filtre veya çekirdek matrisini kaydırarak çıktı matrisi üreten işlemidir.

Reorg: Yeniden düzenlenmiş özellik haritasını oluşturmaktadır.

Sınıflandırma: Tespit edilen nesnenin sınıfı bu katmanda belirlenmektedir.

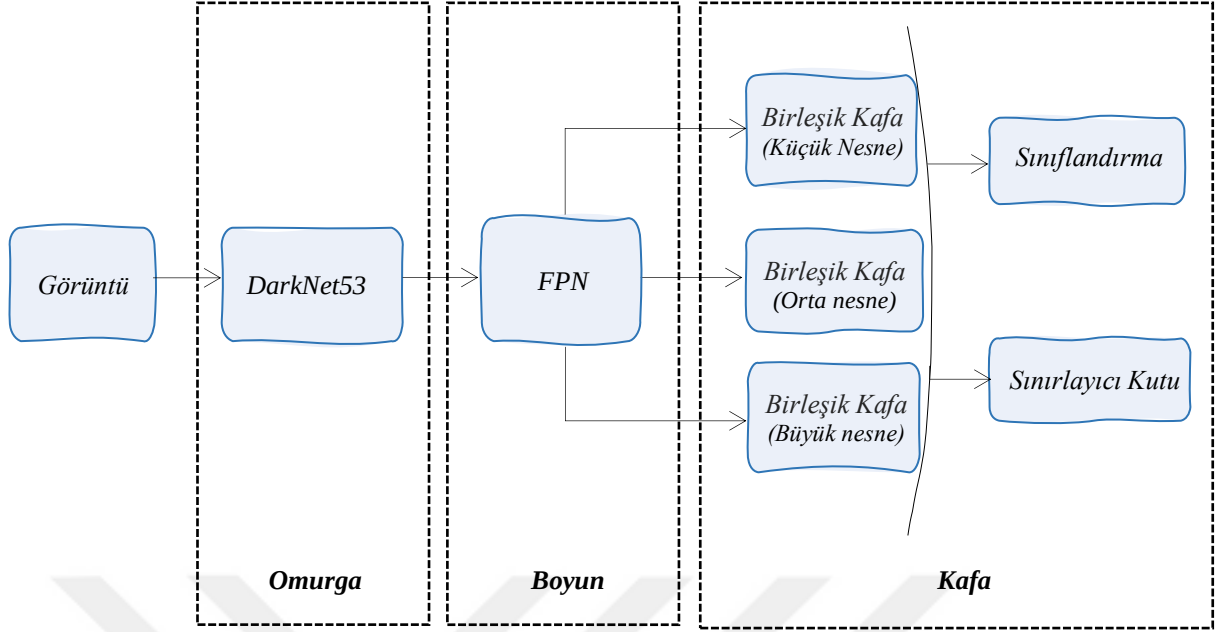
Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

Bu doğrultuda yapılan iyileştirme ve yenilikler şu şekilde özetlenebilir:

- Toplu normalleştirme yapılarak girdi katmanlarının dağılımı sabitlenir. Bu sayede daha hızlı ve kararlı yapıda bir eğitim modeli ortaya çıkmaktadır.
- Giriş boyutu yükseltilerek ortalama hassasiyet $\sim\%4$ oranında arttırılmaktadır.
- Birden fazla orta nokta durumunu aşmak için çapa (anchor) kutuları eklenmektedir. Bu sayede her bir hücre için vektör hesaplamak yerine her bir çapa kutusu için vektör hesaplanmaktadır.
- 13×13 'lük ızgaralar kullanarak görüntüdeki küçük nesnelerin tespiti gerçekleştirilmektedir.
- Çok ölçekli eğitim sayesinde nesnelerin genelleştirilmesi sağlanmaktadır. Başka bir deyiş ile giriş görüntüsünün boyutunu sabitlemek yerine, birkaç yinelemede bir ağı değiştirip, her 10 grupta bir ağ, rastgele yeni görüntü boyutları seçerek eğitimini tamamlar.
- Omurga olarak DarkNet19 mimarîsi kullanılmaktadır.

2.2.2.3. YOLOv3

Redmon ve Farhadi (2018) tarafından bir sonraki yıl YOLO mimarîsinin yeni versiyonu tasarlanarak sunulmuştur. YOLOv3'te, YOLOv2'nin küçük nesneleri algılamadaki eksikliği başta olmak üzere birçok eksiklik ele alınarak iyileştirmeler yapılmıştır. Ayrıca DarkNet19 ağına artık bloklar ve atlama bağlantıları eklenerek elde edilen 53 evrişim katmanına sahip DarkNet53 ağı omurga olarak kullanılmıştır. İlk kez bu YOLO versiyonunda boyun (neck) olarak kullanılan bir katman yer almaktadır. Fakat ilk boyun terimi YOLOv4 versiyonu ile kullanılmaya başlanmıştır. Katmanda Özellik Piramit Ağı (Feature Pyramid Network – FPN) kullanılmıştır. FPN ile giriş görüntüsünü 32, 16 ve 8 ölçeğine kademeli olarak düşürerek 3 farklı ölçekte tahmin yapılmaktadır. Bu sayede farklı tespit seviyeleri ile küçük hedeflerin tespit edilmesi gerçekleştirilir. Şekil 2.14'te YOLOv3 mimarîsi yer almaktadır.



Şekil 2.14: YOLOv3 mimarîsi.

Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

DarkNet53: 53 evrişim katmanlı CNN mimarîsidir. YOLOv3 mimarîsinde omurga ağı olarak kullanılmaktadır.

FPN: Bir girdi görüntüsünden çok ölçekli özellik haritaları üreten ağ mimarîsidir. Bu özellik haritaları nesne tespit veya segmentasyon görevlerinde kullanılmaktadır.

Birleşik Kafa (Coupled Head): Evrişim katmanları ile görüntü üzerinde bir filtre veya çekirdek matrisini kaydırarak çıktı matrisi üreten işlemdir. Kafa bölümündeki her katman 1×1 ve 3×3 filtre boyutlarına sahip *conv2d* bloklarından oluşmaktadır.

Sınıflandırma: Tespit edilen nesnenin sınıfı bu katmanda belirlenmektedir.

Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

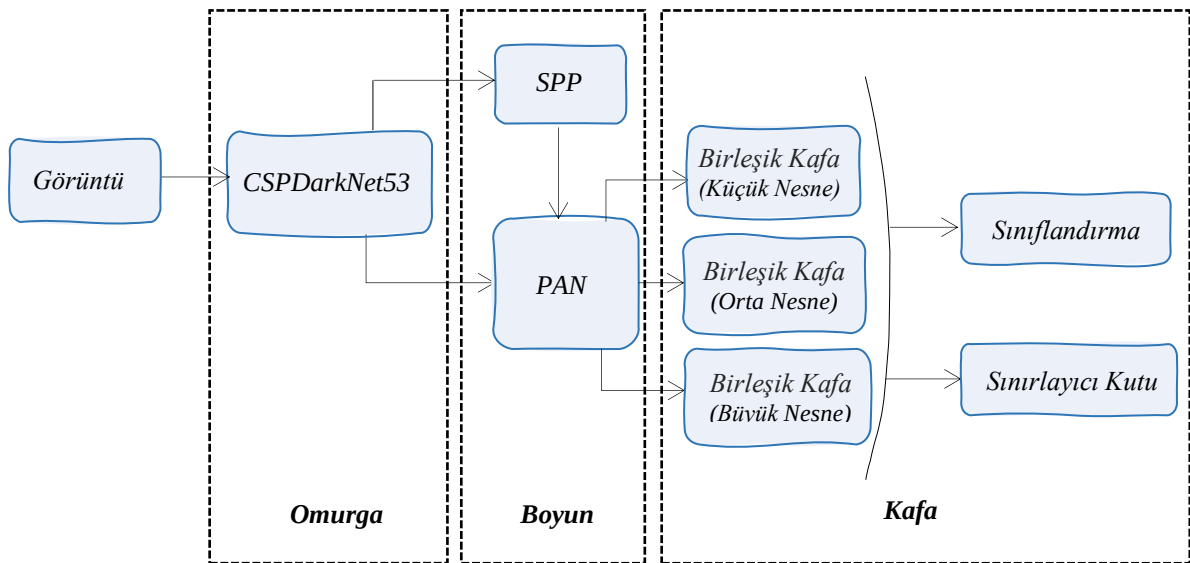
YOLOv3'teki iyileştirme ve yenilikler şu şekilde özetlenebilir:

- Her sınırlayıcı kutu için skor üretimi lojistik regresyon yöntemi ile gerçekleştirilir.
- YOLOv2'de kullanılan Softmax yerine her sınıf için lojistik sınıflandırıcılar kullanılır.
- Nesnelerin tahmini sırasında daha fazla bilgi elde etmek için ara katman eklenir. Bu görevi yerine getiren ara katmana YOLOv4 versiyonu ile “boyun” adı verilmiştir.

- YOLO mimarîsi omurga, kafa ve boyun bölümlerinden oluşmaktadır. Yeni eklenen boyun katmanını sayesinde görüntüden daha fazla bilgi elde edilerek modelin tahmin başarısı iyileştirilir.
- Farklı ölçeklerdeki görüntüleri işleyebilmek için FPN yöntemi kullanılır.
- YOLOv3'ün çıkış parametreleri $S \times S \times (B \times (5+C))$ şeklinde güncellenir.
- Omurga olarak DarkNet53 mimarîsi kullanılır.

2.2.2.4. YOLOv4

2020 yılının başlarında Redmon bilgisayarla görü alanındaki çalışmalarını durdurduğunu açıklamıştır. Bu süreçten sonra YOLO'nun geliştirilmesi diğer araştırmacılar tarafından gerçekleştirilmeye devam etmiştir. Redmon duyurusunu yaptıktan sonra ilk çalışma aynı yıl Bochkovskiy ve diğ. (2020) tarafından yapılmıştır. Yeni sürümün adını benzer şekilde devam ettirmek için YOLOv4 olarak isimlendirdikleri bu yeni mimarînin YOLOv3'e göre daha hızlı ve daha az maliyetli olduğunu belirtmişlerdir. Temelde yazarlar YOLOv3'ün omurga kısmını bilgi akışını daha etkili hale getiren Kısmi Çapraz Aşama (Cross Stage Partial – CSP) bloğu eklenmiş CSPResNext50, CSPDarknet53 ve EfficientNetB3 ağı ile çalıştırarak deneysel çalışmalar yapmışlardır. Bunun sonucunda CSPDarknet53 ağının YOLOv4 mimarîsinin omurga bölgesi olarak belirlenmiştir. Ağın boyun bölgesinde Yol Toplama Ağı (Path Aggregation Network – PAN) ve maksimum havuzlama katmanlarından meydana gelen Mekansal Piramit Havuzlama (Spatial Pyramid Pooling – SPP) kullanılmış olup kafa bölgesi YOLOv3 ile aynı şekilde tasarlanmıştır. Şekil 2.15'te YOLOv4 mimarîsi gösterilmektedir.



Şekil 2.15: YOLOv4 mimarîsi.

Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

CSPDarkNet53: DarkNet mimarîsinin bir türevi olan 53 evrişim katmanlı CNN mimarîsidir. CSP bloğu sayesinde özellik haritalarının işlenmesinde paralellik oluşturarak bilgi akışı ve özellik temsili daha etkili hale getirilmiştir. CSPDarkNet53, YOLOv4 mimarîsinde omurga ağı olarak kullanılmaktadır.

SPP: Ağa farklı boyutlardaki girdileri işleme yeteneği ekleyerek, girdi boyutları üzerinde bağımsızlık sağlamaktadır. Bu sayede ağ farklı boyutlardaki girdileri (nesneleri) işleyebilir.

PAN: Çeşitli boyutlardaki özellik haritalarını etkili bir şekilde birleştirerek daha kapsamlı ve keskin sonuçlar elde etmeyi sağlayan bir ağıdır. Farklı ölçekteki özellik haritalarını bir araya getirerek, bilgiyi ağın daha üst katmanlarına iletmektedir.

Birleşik Kafa (Coupled Head): Evrişim katmanları ile görüntü üzerinde bir filtre veya çekirdek matrisini kaydırarak çıktı matrisi üreten işlemdir. Kafa bölümündeki her katman 1×1 ve 3×3 filtre boyutlarına sahip *conv2d* bloklarından oluşmaktadır.

Sınıflandırma: Tespit edilen nesnenin sınıfı bu katmanda belirlenmektedir.

Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

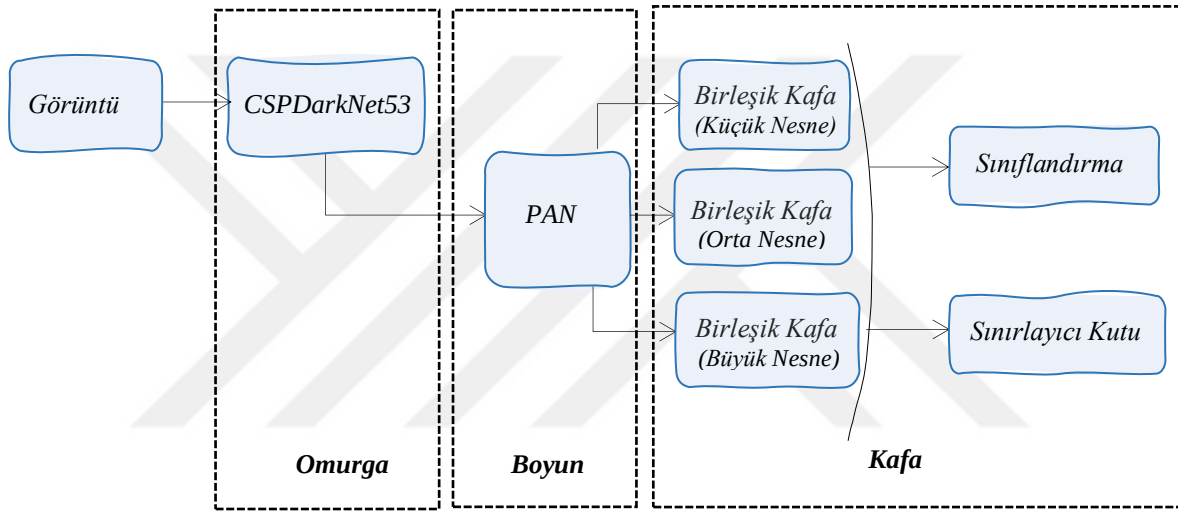
YOLOv4'teki iyileştirme ve yenilikler aşağıdaki şekilde ifade edilebilir:

- Özellik çıkarım süresini arttırmadan Hediye Çanta (Bag-of-Freebies – BoF) tekniği ile veri artırımı yapılmaktadır.
- Çıkarım maliyetini yalnızca küçük bir miktar artıran ancak nesne algılamanın doğruluğunu önemli ölçüde artırabilen Özel Çanta (Bag-of-Specials – BoS) tekniği kullanılmaktadır.
- YOLOv3'te boyun bölgesinde kullanılan FPN yerine PAN ve SPP kullanılmaktadır.
- Omurga olarak CSPDarkNet53 mimarîsi kullanılmaktadır.

2.2.2.5. YOLOv5

YOLOv4'ün tanıtıldığı aynı sene Jocher ve diğ. (2020) tarafından YOLOv5 mimarîsi GitHub üzerinden tanıtılmıştır. YOLOv5, diğer YOLO versiyonlarından farklı olarak bir PyTorch uygulaması olarak ortaya çıkmış ve akademik bir yayın yapılmamıştır. PyTorch uygulaması olarak tanıtılan YOLOv5 mimarîsi teorik olarak önceki YOLO versiyonları ile benzer niteliklere sahip olsa da farklı bir çerçevede sunulduğu için bu versiyonu diğerleri ile nesnel bir şekilde karşılaştırmak oldukça güçtür. Ayrıca YOLOv5 mimarîsi YOLOv5n,

YOLOv5s, YOLOv5m, YOLOv5l ve YOLOv5x olmak üzere 5 farklı versiyona sahiptir. Bu model versiyonları arasında, evrişim çekirdek sayısı, parametre sayısı, katman sayısı gibi farklılıklar mevcuttur. Temelde omurga bölgesi CSPDarknet53 ağından ve boyun kısmı PAN ağından meydana gelmektedir. Birleştirilmiş 3 evrişim katmanlı bir kafa bölümüne sahiptir. Sınıflandırma kaybı için İkili Çarpaz Entropi (Binary Cross Entropy – BCE), sınırlayıcı kutu kaybı için Birlik Üzerinde Tam Kesişme (Complete Intersection over Union – CIoU) ve nesne kaybı için ise BCE kullanılmaktadır. CIoU ile yalnızca iki kutu arasında örtüşme değil aynı zamanda aralarındaki en boy farklılıklarını da hesaba katarak sınırlayıcı kutu işleminin performansı arttırılmaktadır. Şekil 2.16’da YOLOv5 mimarisi gösterilmektedir.



Şekil 2.16: YOLOv5 mimarisi.

Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

CSPDarkNet53: DarkNet mimarisinin bir türevi olan 53 evrişim katmanlı CNN mimarisidir. CSP bloğu sayesinde özellik haritalarının işlenmesinde paralellik oluşturarak bilgi akışı ve özellik temsili daha etkili hale getirilmiştir. YOLOv5 mimarisinde omurga ağı olarak kullanılmaktadır.

PAN: Çeşitli boyutlardaki özellik haritalarını etkili bir şekilde birleştirerek daha kapsamlı ve keskin sonuçlar elde etmeyi sağlayan bir ağıdır. Farklı ölçekteki özellik haritalarını bir araya getirerek ağın daha üst katmanlarına iletmektedir.

Birleşik Kafa (Coupled Head): Evrişim katmanları ile görüntü üzerinde bir filtre veya çekirdek matrisini kaydırarak çıktı matrisi üreten işlemdir. Kafa bölümündeki her katman 1x1 ve 3x3 filtre boyutlarına sahip *conv2d* bloklarından oluşmaktadır.

Sınıflandırma: Tespit edilen nesnenin sınıfı bu katmanda belirlenmektedir.

Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

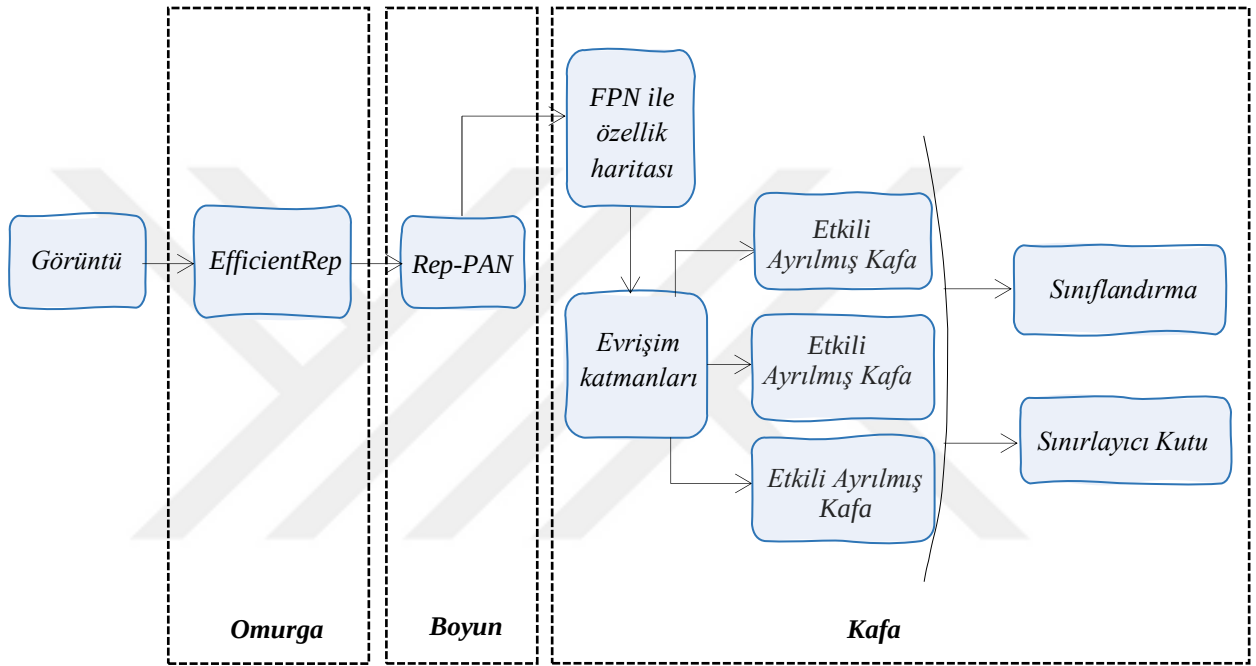
YOLOv5 mimarisine ait 3 ana bölüm aşağıdaki gibi ifade edilebilir:

- Omurga bölgesinde, Darknet çerçevesine dayanan CSPDarkNet53 modeli yer almaktadır. Bu model 53 evrişimli katmandan oluşmaktadır ve temel amacı verimsiz optimizasyon ve maliyetli çıkarım hesaplamaları ile sonuçlanan gereksiz gradyan problemini ortadan kaldırmak ve farklı katmanlarda öğrenilen özelliklerin değişkenliğini artırmaktır. Bunun için modelde aşamalar arası özellik birleştirme stratejisini ve kesik gradyan akışı kullanılmıştır. Bu sayede parametre sayısını azaltmaya ve ağ üzerinden bilgi akışını iyileştirmeye yardımcı olur (Wang vd., 2020).
- Boyun bölgesinde, farklı boyutlardaki girdilerin özellik haritalarını sabit boyutlu vektörlere dönüştürmek için SPP mimarisi kullanılmaktadır. Birkaç farklı boyutta havuzlama işlemi uygulayarak, sonuçları bir sonraki katmana göndermeden önce birleştirir. Bu sayede sabit uzunlukta çıktılar üretir (He vd., 2015). Ayrıca modelde özellik haritalarını daha iyi bir şekilde birleştirmek için PAN mimarisi kullanılmaktadır. Bu mimarî, bir çift özellik haritası oluşturarak daha yüksek özellik haritası ve daha düşük özellik haritası arasında bir bağlantı kurmaktadır. Yüksek özellik haritası, daha geniş bir bölgeye genel bir bakış sağlamak için kullanılırken, düşük özellik haritası daha detaylı bilgi sağlamak için kullanılmaktadır. Bu yöntem, birleştirilmiş özellik haritaları kullanarak daha başarılı nesne tespit işlemlerinin gerçekleştirilmesine yardımcı olmaktadır (Liu vd., 2018). PAN mimarisinin daha hızlı öğrenmesine yardımcı olmak amacıyla CSP mimarisi eklenerek CSP-PAN mimarisi oluşturulmuştur. CSP, girdi verilerini ağdaki farklı katmanlara paralel olarak dağıtma işlemini gerçekleştirerek katmanlar arası bağlantı kurmaktadır. Bu sayede çok ölçekli nesneleri tanıma başarısı ve eğitim hızı artırılmaktadır.
- Kafa bölgesi ise, detektör ve sınıflandırıcı olmak üzere ikiye ayrılmaktadır. Detektör, nesnelerin koordinatlarını tahmin etmek için kullanılırken, sınıflandırıcı nesnelerin sınıflarını belirlemek için kullanılmaktadır. Bu işlem için 1×1 ve 3×3 evrişim katmanı 3 kez tekrarlanmıştır.

2.2.2.6. YOLOv6

Çin merkezli Meituan Teknik Ekibinde yer alan Li ve diğ. tarafından 2022 yılında çapa içermeyen YOLOv6 mimarisi önerilmiştir. Bu mimarînin omurgasında yeniden

parametrelendirmeye izin veren ve Grafik İşleme Biriminin (Graphics Processing Unit – GPU) üzerinde işlem yapmayı destekleyen EfficientRep ağı kullanılmaktadır. Boyun bölgesinde PAN mimarisine dayanan Rep-PAN, kafa bölgesinde ise hibrit kanal stratejisine sahip ayrıştırılmış bir yapı yer almaktadır. Sınırlayıcı kutu kaybı için IoU ve sınıflandırma kaybı için Değişken Odaklı Kayıp (Vari Focal Loss – VFL) kullanılmıştır. Ayrıca YOLOv6 mimarisi YOLOv6-n, YOLOv6-tiny ve YOLOv6-s dahil olmak üzere çeşitli model boyutları sunmaktadır. Şekil 2.17’de YOLOv6 mimarisi gösterilmektedir.



Şekil 2.17: YOLOv6 mimarisi.

Görüntü: Dijital bir formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

EfficientRep: Çeşitli RepVGG katmanlarından, bir toplu normalleştirme katmanından ve bir ReLU aktivasyon biriminden oluşan RepConv ve RepBlock katmanlarını kullanmaktadır. Bu sayede daha başarılı ve hızlı çıkarım elde etmek için sinir ağının yeniden yapılandırılması sağlanmaktadır.

Rep-PAN: Boyun bölgesinde RepBlock'ları kullanabilen PAN topolojisine dayalı Rep-PAN kullanılmaktadır. PAN yapısı, nesne sınıfını ve maskeyi tahmin etmek için birleştirme ve füzyon kullanarak daha iyi nesne yerelleştirmeye yardımcı olmaktadır.

FPN: Bir girdi görüntüsünden çok ölçekli özellik haritaları üreten ağ mimarisidir. Bu özellik haritaları nesne tespit veya segmentasyon görevlerinde kullanılmaktadır.

Evrişim Katmanları: Farklı sayı ve filtre boyutlarındaki *conv2d* katmanları yer almaktadır.

Etkili Ayrılmış Kafa (Effective Decoupled Head): Hibrit kanal stratejisine sahip ayrıştırılmış bir yapıdır. Ayrıştırılmış kafa 128, 256 ve 512 kanala hitap eder, ağ gecikmesini %6,8 azaltır ve YOLOv6 kafasını daha kısa ve verimli hale getirir (Li vd., 2022).

Sınıflandırma: Tespit edilen nesnenin sınıfı bu katmanda belirlenmektedir.

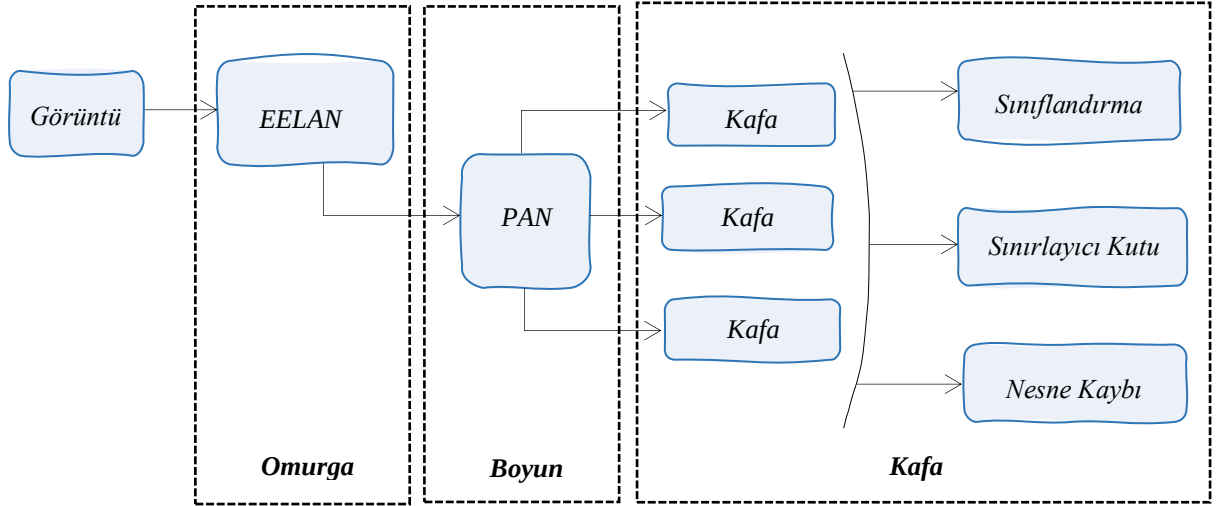
Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

YOLOv6'deki iyileştirme ve yenilikler aşağıdaki şekilde ifade edilebilir:

- Çapasız mimarî sunarak çapa tabanlı yapılara kıyasla hızda %51 oranında bir iyileşme sağlanmaktadır.
- Tespit doğruluğunu daha da geliştirmek için pozitif numuneleri dinamik olarak tahsis eden SimOTA etiket atama stratejisi kullanılmaktadır.
- Ağın öğrenmesini denetlemek için SIOU sınırlayıcı kutu regresyon kaybı kullanılmaktadır.

2.2.2.7. YOLOv7

YOLOv6 mimarîsinden 1 ay sonra YOLOv4'ün yazarları, Wang ve diğ. (2023a) tarafından YOLOv7 mimarîsi sunulmuştur. YOLOv6 ve YOLOv7 mimarîleri arasında GPU hızı geliştirmeye odaklı YOLOX ve YOLOR gibi çeşitli YOLO mimarîleri de tanıtılmıştır. Mimarî ve eğitimsel değişikliklere odaklanan YOLOv7 mimarîsi çıkarım hızını ve eğitim süresini etkilemeden doğruluğu arttıran birkaç mimarî değişiklik ve bir dizi eğitilebilir BoF önerisi içermektedir (Terven & Cordova-Esparza; 2023). Omurga ağı olarak Verimli Katman Toplama Ağı (Extended Efficient Layer Aggregation Network- EELAN) kullanılmıştır. Sınıflandırma kaybı için BCE, sınırlayıcı kutu kaybı için ise L1 kullanılmaktadır. Şekil 2.18'de YOLOv7 mimarîsi gösterilmektedir.



Şekil 2.18: YOLOv7 mimarisi.

Görüntü: Dijital formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

EELAN: Orijinal indirgeme yolunu bozmadan ağıın öğrenme yeteneğini sürekli olarak geliştirmek için genişletme, karıştırma ve birleştirme özelliklerini kullanan genişletilmiş ELAN ağıdır. EELAN, YOLOv7 mimarisinin omurgası olarak kullanılmaktadır (Wang vd., 2023a).

PAN: Çeşitli boyutlardaki özellik haritalarını etkili bir şekilde birleştirerek daha kapsamlı ve keskin sonuçlar elde etmeyi sağlayan bir ağıdır. Farklı ölçekteki özellik haritalarını bir araya getirerek ağıın daha üst katmanlarına iletmektedir.

Kafa: Evrişimli katmanlar ile görüntü üzerinde bir filtre veya çekirdek matrisini kaydırarak çıktı matrisi üreten işlemdir. Bu mimaride her bir kafa FPN katmanından gelen bilgileri sırasıyla 3×3 ve 1×1 *conv2d* katmanlarında işlemektedir.

Sınıflandırma: Tespit edilen nesnenin sınıfı bu katmanda belirlenmektedir.

Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

Nesne Kaybı: Nesnenin varlığına ilişkin tahminleri ve nesne sınıflandırma tahminlerini birleştirerek gerçek etiketlerle ne kadar uyumlu olduğunu ölçmektedir.

Mimarî değişiklikleri şu şekildedir:

- Sınırsız yığılmış hesaplama bloklarına sahip modeller için çalışan EELAN önerilmiştir. Orijinal gradyan yolunu bozmadan ağıın öğrenmesini geliştirmek için kardinaliteyi karıştırarak ve birleştirerek farklı grupların özelliklerini birleştirmesini sağlamaktadır.

- Modelin optimum yapısını korumak için bloğun derinliğinin ve genişliğinin aynı faktörle ölçeklendiği birleştirme tabanlı modelleri ölçeklendirmek için yeni bir strateji önermişlerdir.

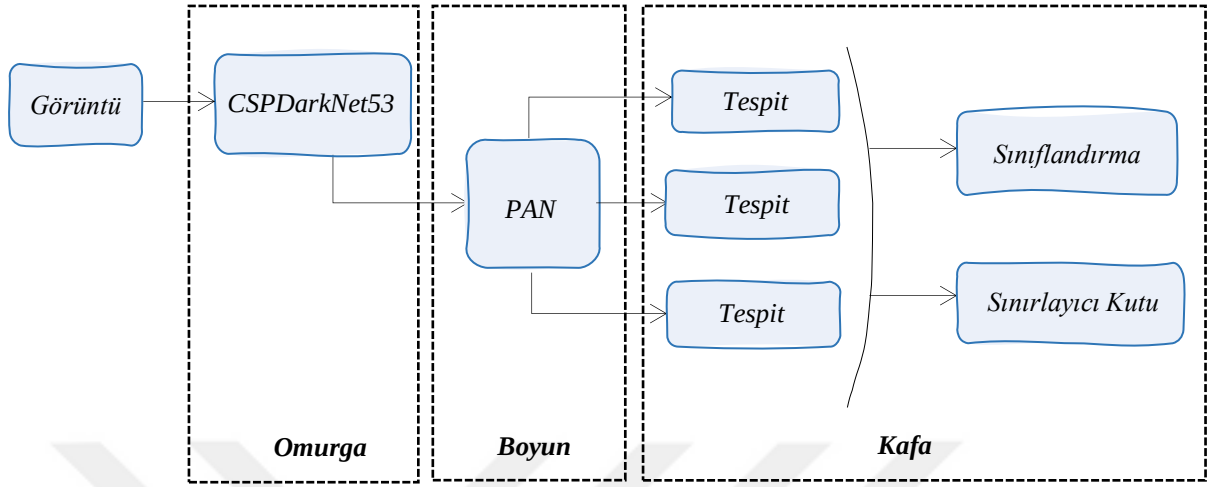
Eğitilebilir BoF önerileri şu şekildedir:

- Planlanmış yeniden parametrelendirilebilir evrişim katman modelini iyileştirmek için eğitimden sonra kullanılan bir tekniktir. Eğitim süresini arttırmasına rağmen modelin çıkarım sonuçlarını iyileştirir.
- Birden fazla kafa yapısına sahip olabilen YOLOv7’de eğitimde yer alan yardımcı kafa için kaba etiket ataması yapılırken nihai çıktıdan sorumlu lider kafa için ince etiket ataması yapılır.
- Toplu normalleştirmenin ortalamasını ve varyansını, çıkarım aşamasında evrişimli katmanın önyargısına ve ağırlığına entegre edilir.

2.2.2.8. YOLOv8

YOLOv5’i tanıtan Jocher ve diğ. tarafından 2023 yılında YOLOv8 mimarîsi tanıtılmıştır. Diğer sürümlerde olduğu gibi YOLOv8 mimarîsinde de farklı parametre büyüklüklerine sahip 5 adet versiyon bulunmaktadır. Bu versiyonlar: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l ve YOLOv8x olarak isimlendirilmiştir. YOLOv8 mimarisine ait n, s versiyonu genellikle mobil uygulamalarda kullanılan hızlı fakat daha düşük performanslı kompakt bir yapıya sahiptir; m versiyonu standart versiyon olarak bilinir ve ortalama bir başarıya sahiptir; gerçek zamanlı uygulamalar için hem başarı hem de hız açısından uygundur; l, x versiyonu ise doğruluğun çok önemli olduğu hızın önemli olmadığı uygulamalar için tasarlanmıştır. YOLOv8, nesne algılama, bölütleme, poz tahmini, izleme ve sınıflandırma gibi çoklu görme görevlerini desteklemektedir. YOLOv8’de, YOLOv5 mimarîsine benzer omurga yapısı kullanılmaktadır. Bu omurga modülü YOLOv5’teki C3 modülleri yerine C2f, ile yer değiştirerek algılama doğruluğunu iyileştirmek için yüksek seviyeli özellikleri bağlamsal bilgilerle birleştirmektedir. Boyun bölgesinde kendinden öncekilerde olduğu gibi PAN & FPN yapıları kullanılmaktadır. Ayrıca YOLOv8 nesnellik, sınıflandırma ve regresyon görevlerini bağımsız olarak işlemek için YOLOv7’dekine benzer bağlantısız bir başlığa sahip çapa içermeyen bir model kullanır. YOLOv8’in çıktı katmanında, sınırlayıcı kutunun bir nesne içermesi olasılığını temsil eden nesnellik puanı için aktivasyon fonksiyonu olarak Sigmoid, sınıf olasılıkları için Softmax fonksiyonu, sınırlayıcı kutu kaybı için CIoU ve Yönsel Özellik Öğrenme (Directional Feature Learning – DFL) kayıp fonksiyonları ve sınıflandırma kaybı için

BCE kullanılmıştır (Terven & Cordova-Esparza; 2023). Şekil 2.19’da YOLOv8 mimarîsi gösterilmektedir.



Şekil 2.19: YOLOv8 mimarîsi.

Görüntü: Dijital formatta depolanmış, piksellerden meydana gelen veri kümesidir. Görüntü içerisinde tespit edilmesi beklenen hedef nesne aranmaktadır.

CSPDarkNet53: Giriş görüntüsünü aşamalı olarak alt örnekleyerek birden fazla çözünürlükte özellikleri çıkaran 53 evrişim katmanlı CNN mimarîsidir. CSP bloğu sayesinde özellik haritalarının işlenmesinde paralellik oluşturarak bilgi akışı ve özellik temsili daha etkili hale getirmektedir. Her evrişim katmanı ardından bir $C2f$ modülü içeren dört bölümden oluşmaktadır. $C2f$ modül, bir ucunun darboğaz modülünden geçtiği bölmelerden oluşur (Solawetz, 2023).

PAN: PAN mimarîsi ise FPN’in geliştirilmiş versiyonudur ve atlama bağlantıları ile ağırlı farklı düzeylerdeki özellikleri bir araya getirerek birden fazla ölçek ve çözünürlükteki özellikleri daha iyi yakalayabilir. Bu sayede farklı boyut ve şekillerdeki nesnelerin doğru algılanması sağlanabilir (Reis vd., 2023).

Tespit: Görüntü üzerinde bir filtre veya çekirdek matrisini kaydırarak çıktı matrisi üreten 2 tane 3×3 $conv2d$ evrişim katmanlarından ve 4 tane $C2f$ katmanlarından oluşmaktadır.

Sınıflandırma: Tespit edilen nesnenin sınıfı bu katmanda belirlenmektedir.

Sınırlayıcı Kutu: Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin eden bir regresyon işlemidir.

Önceki YOLO versiyonlarına göre;

- Daha karmaşık bir mimarîye sahip olmasına rağmen işlem hızını koruyarak performans başarısını arttırmaktadır.

- Aynı görüntü içinde farklı ölçekteki nesneleri eş zamanlı olarak tespit edebilmektedir.
- Yanlış pozitiflerin azaltılması ve geliştirilmiş maksimum olmayan baskılama teknikleri sayesinde gerçek dünya uygulamalarının daha güvenilir olması sağlanmaktadır.

Tablo.2.1’de YOLO mimarilerinin yapısal farklılıklarına ilişkin bilgiler yer almaktadır. Boyun katmanı YOLOv3 versiyonunda ilk kez mimariye eklendiği için YOLOv1 ve YOLOv2’de bu katmanlar boş bırakılmıştır.

Tablo 2.1: YOLO mimarilerinin karşılaştırması.

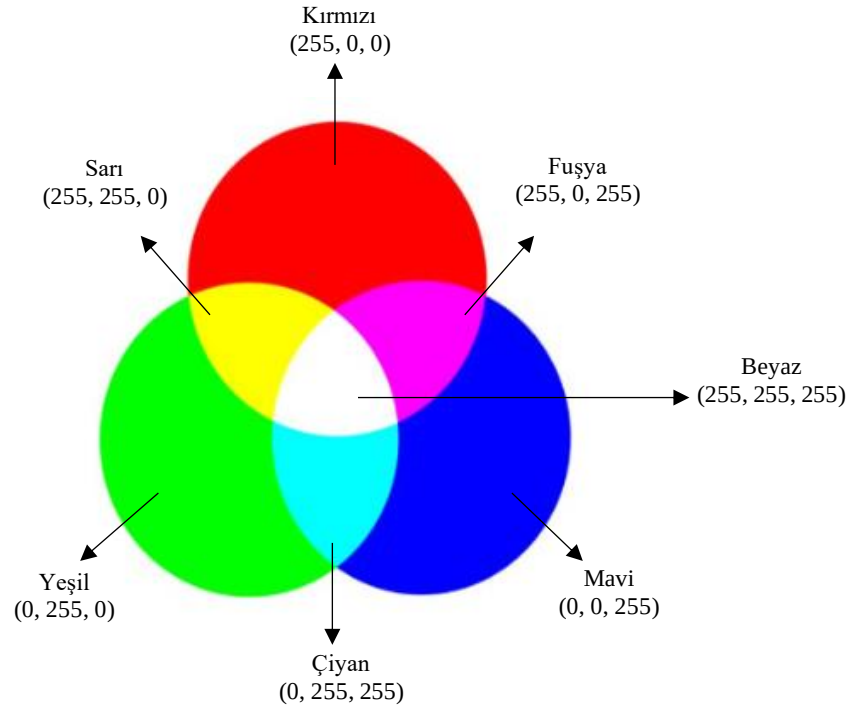
YOLO Mimarîsi	Omurga	Boyun
YOLOv1	GoogleLeNet	-
YOLOv2	DarkNet19	-
YOLOv3	DarkNet53	FPN
YOLOv4	CSPDarkNet53	SPP & PAN
YOLOv5	CSPDarkNet53	SPP & PAN
YOLOv6	EfficientRep	Rep-PAN
YOLOv7	EELAN	PAN
YOLOv8	C2f ile CSPDarkNet53	PAN

2.3. SÜPER ÇÖZÜNÜRLÜK TEKNİKLERİ

Türkçede “gözet” olarak isimlendirilen “piksel” kavramı bir görüntünün dijital ortama yansıtılmasını sağlamaktadır. Piksel kare şeklinde ve parçalanamaz yapısıyla görüntüyü oluşturan en küçük elemandır. Pikseller temelde Şekil 2.20’de gösterildiği gibi Kırmızı (Red – R), Yeşil (Green – G) ve Mavi (Blue – B) renklerinden meydana gelmekte ve her piksel sadece bir rengi temsil etmektedir. Bu renkler kısaca RGB olarak adlandırılmakta olup (R, G, B) parametreleri olarak temsil edilmektedir. (R, G, B) şeklinde temsil edilen piksel parametrelerinin her biri 0-255 arasında bir değer olarak yeni bir renk oluşturmaktadır. Farklı parametre değerleri sayesinde oluşan farklı renklerdeki piksellerin bir araya gelmesi ile de görüntü oluşmaktadır. Oluşan bu görüntülerin kalitesi Piksel Yoğunluğuna (Pixel Density/ Pixels Per Inch – PPI) ve çözünürlüğüne göre değişiklik göstermektedir.

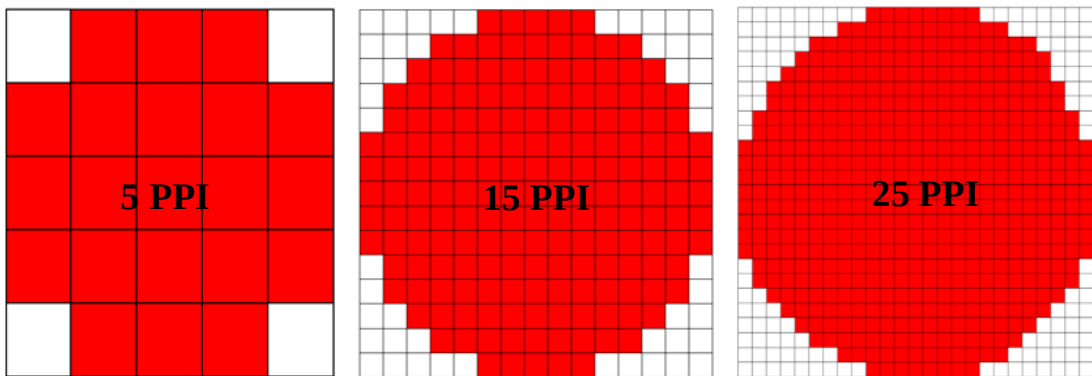
PPI: Bir inç (2,54 cm) başına düşen piksel sayısını temsil etmektedir.

Çözünürlük: Görüntünün yatay ve dikeyde sahip olduğu piksel sayısı ile ifade edilmektedir.



Şekil 2.20: Renk skalası.

Aynı alana düşen piksel sayısındaki değişim Şekil 2.21’de gösterildiği gibi görüntünün keskinliğini ve kalitesini etkilemektedir. Şekilde görüldüğü gibi daha çok piksel ile temsil edilmiş görüntü aynı çözünürlük içerisinde daha net bir görüntüye sahiptir. Görüntü kalitesindeki artış görüntüden yararlanarak işlem yapan algoritmaların performans başarısını arttırabileceği için görüntü kalitelerini iyileştirmek oldukça önemlidir.



Şekil 2.21: PPI ve çözünürlük ilişkisi.

SR algoritmaları, birim alana düşen piksel sayısını artırarak düşük çözünürlüklü bir görüntüden daha yüksek çözünürlüklü bir görüntü elde etmeyi sağlamaktadır. Çoğu SR algoritması denetimli öğrenmeye yaklaşımına sahip olma eğilimindedir. Bu durum SR

işleminin gerçekleştirilebilmesi için modelin nihai sonucu içeren örnekleri görmesi gerektiği anlamına gelmektedir. Bu sebepten model eğitimi için Düşük Çözünürlüklü (Low Resolution – LR) ve Yüksek Çözünürlüklü (High Resolution – HR) görüntülere ihtiyaç duyulmaktadır.

LR ve HR arasındaki “klasik” ilişki Denklem 2.2 ile ifade edilmiştir. Burada M çarpıtmayı, B bulanıklaştırmayı, D seyreltmeyi, n gürültüyü, x ise yüksek çözünürlüklü görüntüyü temsil etmektedir. Denklem, yüksek çözünürlüklü (x) görüntü üzerinde sırası ile çarpıtma (M), bulanıklaştırma (B), seyreltme (D) ve gürültü (n) ekleme işlemini gerçekleştirerek LR (y) değerinin elde edilmesi sağlanmaktadır. Bu denklem belirsiz tersine problem olarak adlandırılmaktadır ve ana amacı LR (y) görüntüden, HR (x) görüntüsünü elde etmektir (Anwar vd., 2020; Temiz, 2020).

$$y_k = DB_k M_k + n_k \quad k = 1, 2, \dots, p \quad (2.2)$$

SR algoritmaları arasında en yakın komşu interpolasyonu, doğrusal, çift doğrusal, çift kübik interpolasyon gibi klasik yöntemlere ek olarak 2014 yılından itibaren derin öğrenme tabanlı algoritmalar da yer almaktadır. DL tabanlı ilk SR algoritması Dong ve diğ. tarafından tanıtılan Süper Çözünürlüklü Evrişimsel Sinir Ağıdır (Super-Resolution Convolutional Neural Network – SRCNN) (Dong vd., 2014; Dong vd., 2015). Ardından Kim ve diğ. tarafından çok katmanlı VGG mimarisine dayanan Çok Derin Süper Çözünürlük (Very Deep Super-Resolution – VDSR) ağı tanıtılmıştır (Kim vd., 2016). Bu ağlar bilinen ilk DL tabanlı çözünürlük iyileştirme algoritmalarıdır ve birçok çalışmada sıklıkla tercih edilmektedir. İlerleyen zamanlarda ön örnekleme (pre-upsampling) ile yapılan çözünürlük artırma işlemlerinin hesaplama gücü daha yüksek olduğu için hesaplamayı önemli ölçüde azaltan örnekleme sonrası (post-upsampling) yöntemler de ortaya çıkmıştır. Ayrıca artık ağ yapıları, dikkat mekanizmaları, üretken ağlar gibi bilinen birçok mekanizma çözünürlük iyileştirme işlemleri için kullanılmış ve yeni modeller ortaya çıkmıştır (Anwar vd., 2020).

Genel olarak SR algoritmaları 9 alt gruba ayrılabilir:

1. **Ön örnekleme (Pre-upsampling):** Üst örneklenmiş bir görüntüyü iyileştirmek için çift kübik enterpolasyon ve derin öğrenme teknikleri kullanılmaktadır.
2. **Örnekleme sonrası (Post-upsampling):** Basit çift kübik enterpolasyon kullanmak yerine, ters evrişim/alt piksel evrişimi biçiminde öğrenilmiş bir üst örnekleme kullanılmaktadır. Bu işlemi sonda yaptığı için hesaplama maliyeti azalmaktadır.

3. **Artık Ağlar (Residual Networks):** Ağ tasarımı atlama katmanları kullanarak çok daha derin ağların kurulmasını sağlamaktadır. Tek aşamalı ve çok aşamalı olarak iki kısma ayrılmaktadır.
4. **Özyinelemeli Ağlar (Recursive Networks):** Tekrarlı olarak bağlı evrişim katmanları oluşturarak daha basit yapılar oluşturulmasını sağlamaktadır. Bu sayede bellek kullanımını azaltmaktadır.
5. **Aşamalı Yeniden Yapılanma Ağları (Progressive Reconstruction Networks):** Büyük ölçeklendirme ($\times 4, \times 8$) işlemlerinde yüksek çözünürlüklü bir görüntüyü daha kolay elde etmeyi sağlamaktadır. Bu işlemi görüntünün çözünürlüğünü adım adım arttırarak gerçekleştirmektedir.
6. **Yoğun Bağlantılı Ağlar (Densely Connected Networks):** DenseNet mimarisinin başarısından esinlenerek, performansı artırmak için yoğun şekilde bağlı CNN katmanlarından meydana gelerek yüksek esneklik ve zengin özellik haritası elde etmeyi sağlamaktadır.
7. **Çok Şubeli Ağlar (Multi-Branch Networks):** Daha derin bir ağın getirdiği başarıyla birlikte bazı problemlerin önüne geçmek adına bilginin geçebileceği birden fazla şube oluşturularak bilgi akışını iyileştirmeye çalışmaktadır. Birden fazla alıcı alandan gelen bilgilerin birleşmesi ile daha derin bir ağ ve daha iyi bir eğitim sağlanmaktadır.
8. **Dikkate Dayalı Ağlar (Attention-based Networks):** Tüm özelliklerin süper çözünürlük için gerekli olmadığını kanıtlayarak sadece modele katkı sağlayan özelliklerin seçilmesini sağlamaktadır. Başka bir deyişle bir görüntüdeki farklı bölgelere seçici düzeyde önem vererek daha başarılı sonuçlar elde edilmesini desteklemektedir.
9. **Üretken Çekişmeli Ağlar (Generative Adversarial Networks):** Bir üretici ve bir ayırmacı bileşenden meydana gelen kuramsal yaklaşım kullanmaktadır. Üretici bileşeni, bir ayırmacının fark edemeyeceği gerçeklikte SR görüntüleri oluşturarak algısal kaliteyi optimize etmektedir.

SR algoritmaları sayesinde LR bir görüntünün, HR bir görüntü haline getirilerek nesne sınıflandırma, nesne tanıma gibi problemlere üretilen çözümlerin performansını arttırdığı bölüm 2.4.3'te incelenen kaynaklar aracılığıyla gösterilmiştir. İncelenen bu çalışmalarda da görüldüğü gibi DL tabanlı algoritmaların çözünürlük artırma bakımından oldukça başarılı olduğu ve nesne tespiti ile SR algoritmalarının birleştirilmesine ilişkin çalışmaların kısıtlı bir

alanda işlendiği sonucuna varılmıştır. Bu çerçevede literatüre bakıldığında SR ile iyileştirilmiş tehlikeli nesneleri tespit eden hibrit bir model çalışmasına rastlanmamıştır.

Tez kapsamında artık ağ ve kanal dikkat mekanizması kullanılarak yeni bir SR algoritması önerilmiştir. Önerilen bu SR algoritması daha önceden OpenCV tarafından eğitilmiş ve açık kaynaklı bir şekilde sunulan 4 farklı SR algoritması ile karşılaştırılmıştır. Tablo 2.2 içerisinde karşılaştırma için kullanılan SR algoritmalarının özetleri yer almaktadır. Bu tabloda yer alan SR algoritmaları sırasıyla çalışmanın bir sonraki bölümünde detaylı olarak incelenmiştir.

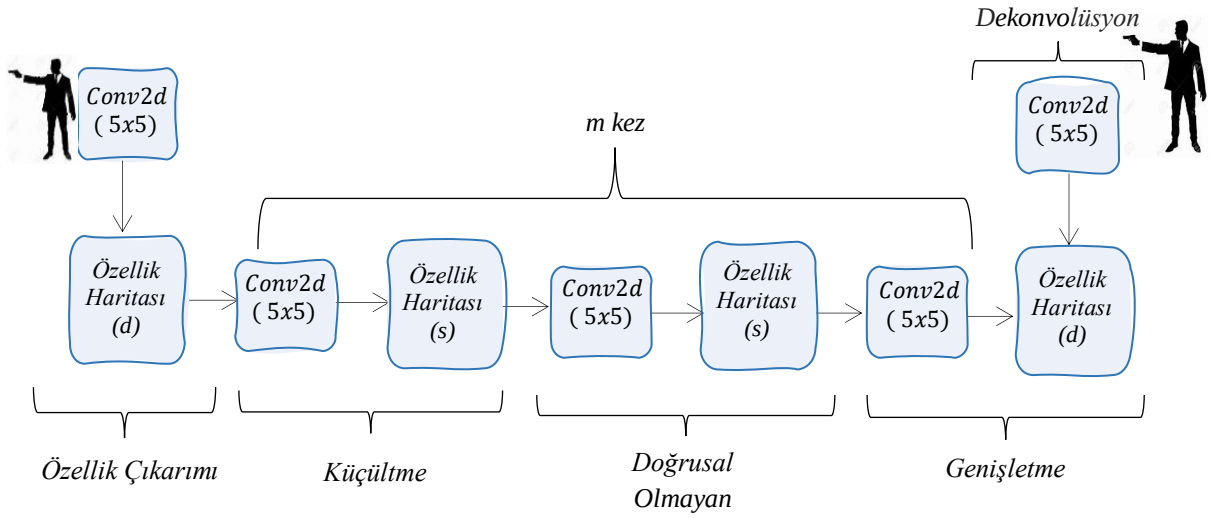
Tablo 2.2: Çalışmada yer alan SR algoritmalarının özellikleri.

Algoritma	Ağ Girişi	#Katman	Kayıp Fonksiyonu	Ağ Altyapısı	Sınıfı	#Parametre (B)
FSRCNN	LR	8	L2	CNN	Örnekleme Sonrası	12 B
ESPCNN	LR	3	L2	CNN	Örnekleme Sonrası	20 B
EDSR	LR	32	L1	ResNet	Artık Ağ	43.000 B
LapSRN	LR	27	Charbonnier	ResNet	Aşamalı Yeniden Yapılanma Ağı	812 B

2.3.1. Hızlı Süper Çözünürlüklü Evrişimsel Sinir Ağı

Hızlı Süper Çözünürlüklü Evrişimsel Sinir Ağı (Fast Super Resolution Convolutional Neural Network – FSRCNN), Dong ve diğ. tarafından 2016 yılında SRCNN modelini hızlandırmak için tasarlanmış yeni bir SR algoritmasıdır. SRCNN modelini hızlandırarak en az bir önceki model kadar başarılı sonuç üretmeyi hedefleyen araştırmacılar bu işlem için parametre sayısını düşürmeyi önermişlerdir. Ağ mimarisine ait parametrelerin sayısını düşürerek hesaplama süresi azaltılmış ve daha hızlı sonuçlar elde edilmesi sağlanmıştır.

SRCNN modelinin yeniden tasarlanması sonucunda daha küçük evrişim katmanları ile daha fazla özellik haritası ve ek ters evrişim katmanı eklenmiştir. Giriş boyutu ile düşük çözünürlüklü orijinal görüntünün boyutunun aynı kalması sağlanarak parametre sayısı azaltılmıştır. Şekil 2.22’de FSRCNN modelinin mimarisi gösterilmektedir.



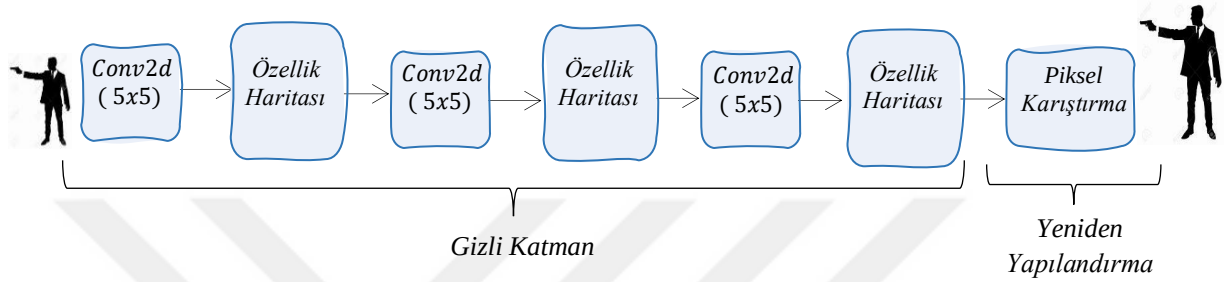
Şekil 2.22: FSRCNN mimarisi.

- **Özellik çıkarımı:** Giriş görüntüsünde 5x5 boyutlu filtreye sahip evrişimli katman kullanılmaktadır. Bu sayede giriş görüntüsünün yüksek frekans bileşenleri azaltılarak verilerin boyutsal karmaşıklığı azaltılmaktadır.
- **Küçültme:** Düşük çözünürlüklü giriş görüntüsünün boyutunu küçültmek için 1x1 boyutlu filtreye sahip evrişim katmanı kullanılmaktadır. Bu sayede özellik haritalarının büyüklüğü azaltılmaktadır.
- **Doğrusal olmayan haritalama:** FSRCNN'nin ana bileşenini oluşturan bu aşama, derin öğrenme bileşenlerini içermektedir. Ardışık olarak birkaç evrişimsel katman ve aktivasyon fonksiyonları bu aşamada yer almaktadır. Bu katmanlar, düşük çözünürlüklü özellikleri yakalar ve yüksek çözünürlüklü özellik haritalarını oluşturmaktadır.
- **Genişletme:** Ölçeklendirme faktörünü artırmak için kullanılmaktadır. Bu aşamada, ölçeklendirme faktörüne bağlı olarak giriş görüntüsünün boyutu tekrar ölçeklendirilir. Özellik haritalarını genişletmek için tekrar 1x1 evrişim katmanı kullanılmaktadır.
- **Yeniden oluşturma:** FSRCNN'nin çıkış katmanında yüksek çözünürlüklü sonuç üretmek için 9x9 boyutunda filtreler kullanılmaktadır.

FSRCNN, geleneksel süper çözünürlük yöntemlerine göre daha hızlı çalışabilen bir mimariye sahiptir. Daha düşük hesaplama maliyetiyle bile yüksek kalitede sonuçlar üretebilir ve gerçek zamanlı uygulamalarda kullanıma uygun bir modeldir.

2.3.2. Etkin Alt Piksel Evrişimli Sinir Ağı

Etkin Alt Piksel Evrişimli Sinir Ağı (Efficient Sub-Pixel Convolutional Neural Network – ESPCNN), Shi ve diğ. 2016 yılında SRCNN modelinde karesel olarak büyüyen karmaşıklık sorunun üstesinden gelmek için alt piksel evrişim ağı yapısını önermiştir. Önerilen alt piksel evrişim modülü SRCNN’de yer alan bikübik enterpolasyon yerine yerleştirilmiştir. Şekil 2.23’te ESPCNN algoritmasının mimarîsi yer almaktadır.



Şekil 2.23: ESPCNN mimarîsi.

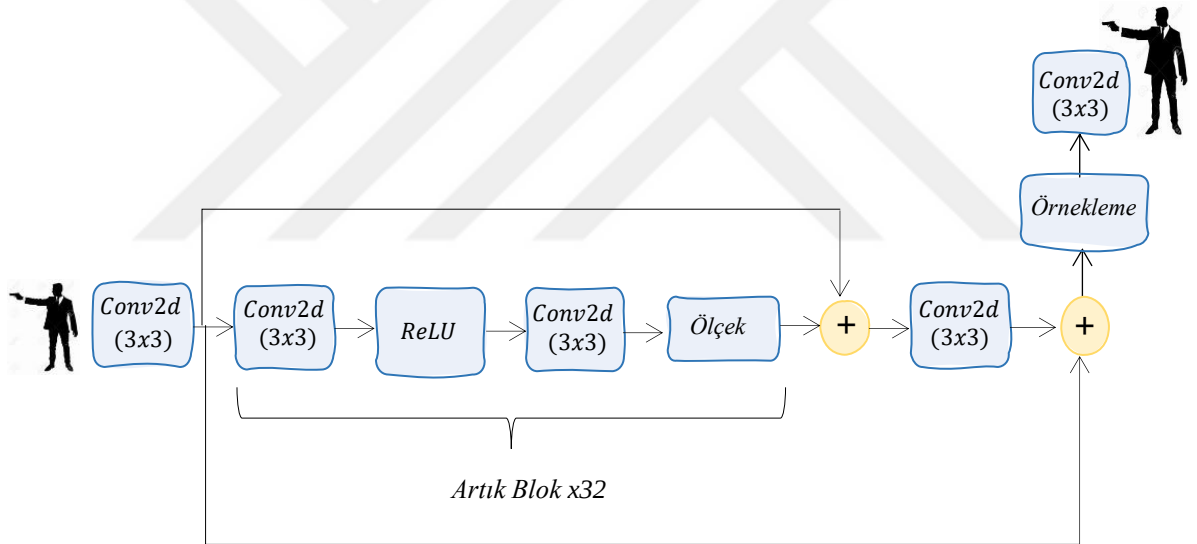
ESPCNN algoritması giriş katmanı, gizli katman ve yeniden yapılandırma katmanı olarak 3 ana bileşenden oluşmaktadır.

- **Giriş Katmanı:** Düşük çözünürlüklü girdi verilerinden yüksek seviyeli özellikleri öğrenebilmek için evrişim katmanları içerir. Evrişim katmanları, girdi verilerinin farklı özelliklerini çıkarmak için filtreleri kullanmaktadır. Bu filtreler, özellikleri daha iyi temsil edebilen ve çözünürlüğü artıran öğrenilmiş parametrelerle güncellenmektedir.
- **Gizli Katman:** Öğrenme katmanından gelen özellik haritalarını alır ve daha yüksek seviyeli özelliklerin çıkarılması için kullanır. Bu durum, özelliklerin daha iyi bir temsilini gerçekleştirerek daha fazla ayrıntının yakalanmasını sağlamaktadır.
- **Yeniden Yapılandırma Katmanı:** Düşük çözünürlüklü girdilerin yüksek çözünürlüklü çıktılara dönüştürülmesinden sorumludur. Bu dönüşüm, alt-piksel evrişim ağı kullanarak piksel değerlerini yeniden düzenler ve çıktı görüntüsünün mekansal çözünürlüğünü artırmaktadır.

ESPCNN ile hesaplama karmaşıklığı ve bellek maliyeti azaltılmaktadır, böylece verimlilik büyük ölçüde artırılabilir. Bu sayede ESPCNN, gerçek zamanlı olarak HR görüntüleri oluşturabilir, video verileri üzerinde kullanılabilir.

2.3.3. Geliştirilmiş Derin Süper Çözünürlük Ağı

Geliştirilmiş Derin Süper Çözünürlük Ağı (Enhanced Deep Super-Resolution Network – EDSR), Lim ve diğ. 2017 yılında çoklu artık bloklardan oluşan SRResNet mimarisine dayanan EDSR modelini önermiştir. Modelde geleneksel artık ağlardaki Toplu Normalleştirme (Batch Normalization – BN) katmanı kaldırılarak daha hızlı işlem gerçekleştiren bir yapı elde edilmiştir. Bu sayede EDSR modeli, SRResNet'e kıyasla eğitim sırasında bellek kullanımında yaklaşık %40 tasarruf sağlayarak sınırlı hesaplama kaynakları altında geleneksel ResNet yapısından daha iyi performansa sahip daha büyük bir model olmuştur. EDSR modeli sayesinde biküçük enterpolasyonun uygulandığı artık ağ içeren modellerdeki ağır bellek tüketimi sorununun üstesinden gelmek mümkün olmuştur. Fakat bu noktada da blok katmanlarının 32 kez tekrar etmesinden dolayı artan katman sayısı modelin hesaplama maliyetini artırmıştır. Şekil 2.24'te EDSR algoritmasının mimarisi yer almaktadır.

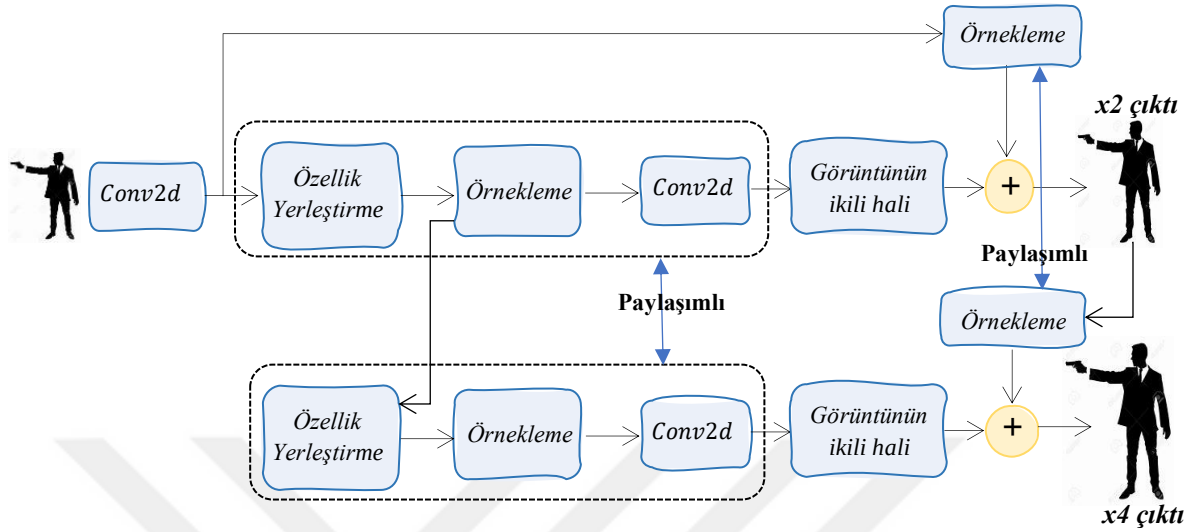


Şekil 2.24: EDSR mimarisi.

2.3.4. Laplacian Piramidi Süper Çözünürlük Ağı

Laplacian Piramidi Süper Çözünürlük Ağı (Laplacian Pyramid Super-Resolution Network – LapSRN), Lai ve diğ. tarafından 2017 yılında Laplacian piramidi kullanılarak HR görüntüleri hızlı bir şekilde üretmeyi sağlayan SR algoritması önerilmiştir. Laplacian yöntemi bir görüntünün birden fazla ölçekte temsil edilerek her ölçek üzerindeki ayrıntıların yakalanmasını sağlamaktadır. Bu algoritmanın öğrenme sürecinde derin gözetim ve Charbonnier kayıp fonksiyonu kullanılmıştır. Bu sayede yüksek kaliteli yeniden yapılandırma

sağlanarak LR giriş verileri HR görüntülere dönüştürülür. Şekil 2.25'te LapSRN algoritmasının mimarisi yer almaktadır.



Şekil 2.25: LapSRN mimarisi.

- **Laplacian Piramit Oluşturma:** LR görüntüsü, Laplacian piramitine dönüştürülür. Laplacian piramit, görüntünün farklı ölçeklerde temsilini sağlar ve yüksek frekanslı bileşenlerin ayrıştırılmasına olanak tanımaktadır. Her piramit seviyesinde, önceki seviyeye göre daha düşük çözünürlüklü bir özellik haritası elde edilmektedir.
- **Alt-Bant Artık Tahmini:** Her piramit seviyesinde, LapSRN modeli, kaba çözünürlüklü özellik haritalarını giriş olarak alır ve yüksek frekanslı alt-bant artıklarını tahmin etmektedir. Bu tahmin, yüksek çözünürlüklü girişin alt bantlarını yeniden oluşturmasını amaçlamaktadır.
- **Transpoz Konvolüsyon ve Yeniden Örnekleme:** Alt-bant artık tahminleri, daha ince seviyeye giderken transpoz konvolüsyonlar kullanılarak yüksek çözünürlüklü alt bantlar haline getirmektedir. Transpoz konvolüsyonlar, daha düşük çözünürlüklü bir özelliği, daha yüksek çözünürlüğe çıkarmak için kullanılmaktadır.
- **Piramit Birleştirme:** Alt-bantların yeniden örnekleme ile elde edilmesi ve önceki piramit seviyelerinden gelen alt bantlar ile birleştirilerek, yüksek çözünürlüklü çıktı görüntüsü elde edilmektedir.

2.4. LİTERATÜR TARAMASI

Nesne tespit etme işlemi birçok “bilgisayarla görü” uygulamalarının temelini oluşturmaktadır. Genellikle yüz tanıma, yaya tespiti, robotik, otonom sürüş gibi uygulamalarda tercih edilen bu yöntem tez kapsamında tehlikeli nesnelerin tespitinde kullanılmıştır (Yang vd., 2011; Chen vd., 2015; Ren vd., 2015; Chen vd., 2016; Yang vd., 2016; Coşkun vd., 2017; Kamencay vd., 2017; Wojke vd., 2017; Chen vd., 2018; Brunetti vd., 2018). Dolayısı ile bu bölümde tehlikeli nesnelerin (ateşli silahlar, kesici ve delici aletler vb.) tespitini gerçekleştiren çalışmalar incelenerek özetlenmiştir. Çalışmada kullanılan algoritmalara göre literatür taraması üç kısma ayrılmıştır. İlk kısımda geleneksel yöntemler ile gerçekleştirilen tehlikeli nesne tespit çalışmaları sunulurken, ikinci kısımda DL tabanlı modern yöntemler ile gerçekleştirilen tehlikeli nesne tespit çalışmaları; son kısımda ise SR algoritmalarının nesne tespitindeki etkisini gözlemleyen çalışmalar sunulmuştur.

2.4.1. Geleneksel Yöntemler ile Gerçekleştirilen Çalışmalar

Tehlikeli nesnelerin yetkisiz kişilerce kullanımının önüne geçmek amacıyla 2000li yılların başında gizli silah ve bıçak tespit sistemleri geliştirilmeye başlanmıştır. Bu sistemler kızılötesi görüntüleme, milimetre dalga görüntüleme gibi teknikleri kullanarak kişinin taşıdığı gizli nesneyi tespit edebilmektedir. Ancak nesnenin tehlikeli olup olmadığını ve türünü tespit edememektedir (Uner vd., 1997; Sheen vd., 2001; Xue vd., 2002; Blum vd., 2004; Upadhyay vd., 2014).

2007 yılında hem silahları hem de silahla suç işleme niyetinde olan kişileri CCTV aracılığıyla otomatik olarak tespit edilmesi amaçlanmış ve bu doğrultuda MEDUSA adında bir program oluşturulmuştur (Darker vd., 2007). Ertesi yıl operatörlerin silah tespit etme konusunda sınırlı görüş ve dikkate sahip olduğu belirtilerek ateşli silahların tespitini otomatikleştirmek için yeni model geliştirmişlerdir. Fakat bu model de aynı sahne içerisinde yer alan farklı silahları algılamada başarılı olamamıştır (Darker vd., 2008).

İlerleyen yıllarda yüksek kapasiteli donanımlara erişim, veri saklama miktarındaki artış ve DL tabanlı mimarilerin geliştirilmesiyle birlikte nesne tespit sistemleri daha da önem kazanmıştır. Bu süreç içerisinde nesne algılama yöntemleri kullanılarak yaya tespiti, yüz tanıma gibi çeşitli problemlere çözüm aranmıştır. Ancak nesne algılama yöntemleri kullanılarak tehlikeli nesnelerin tespit edilmeye başlanması, 2013 yılında Grega ve diğ. tarafından gerçekleştirilmiştir. Grega ve diğ. (2013) tarafından yeni veri setleri oluşturulup arka plan

algılama, kenar algılama, sürgülü pencere, temel bileşen analizi ve sinir ağları yöntemi ile silah tespit işlemi gerçekleştirilmiştir.

Ardından Tiwari ve Verma (2015a), görüntüden ilgisiz nesneleri ortadan kaldırmak için renk tabanlı bölütleme kullanmıştır. Hedef nesneyi bulmak için Harris ilgi noktası dedektörü ve Hızlı Retina Anahtar Noktası (Fast Retina Keypoint – FREAK) yöntemlerini kullanıp %84,26 sınıflandırma doğruluğu elde etmişlerdir. Buna rağmen arka plan değişimlerine duyarlı olduğu için gerçek zamanlı çalışabilecek bir sistem olamayacağı düşünülmektedir. Aynı yıl yazarlar benzer şekilde ilgisiz nesneleri ortadan kaldırmak adına tekrar renk tabanlı bölütlemeyi kullanmış ve bu sefer Hızlandırılmış Sağlam Özelliklerin (Speeded Up Robust Features – SURF) ilgi noktası dedektörünü parçalı görüntülerde hedef nesneyi bulmak için kullanmışlardır. Veri setini kendilerinin topladığını ve 15 adet silah, 10 adet silah olmayan görüntülerden oluştuğunu belirtmişlerdir. Sistemleri %88,67 oranında başarı elde etmesine rağmen veri setinin azlığı ve silahların sabit bir şablonla eşleştirilmesi sistemin gerçekçi çalışmadığını düşündürmektedir (Tiwari & Verma, 2015b).

Grega ve diğ. (2016) tarafından bıçak ve silah tespit sistemi geliştirilmiştir. Bu sistem için kendi verilerini toplayıp, nesne tespiti için sürgülü pencere ve Canny kenar algılama tekniklerinden yararlanmışlardır. Düşük çözünürlükteki verilerde çalışabildiği için gerçek zamanlı verilerde de çalışabileceğini belirtmişlerdir. Fakat sabit boyutlu sürgülü pencere kullanmaları hatalı tespitlere ve zaman kaybına yol açabilmektedir.

Vajhala ve diğ. (2016) öznitelik vektörünü çıkartırken HOG tekniğinden, sınıflandırma için ise yapay sinir ağından yararlanmıştır. Nesne tespiti sırasında %83 oranında doğruluk değeri elde ettiklerini belirtmelerine rağmen çalışmada kullanılan silah ve bıçaklardan oluşan veri seti dengesizliği, görüntü kalitesi yüksekliği ve arka plan sabitliği sebebiyle bu modelin gerçek zamanlı veriler üzerinde iyi bir sonuç üretemeyeceği düşünülmektedir.

Geleneksel yöntemlerden yararlanan bir diğer çalışmada ise Buckchash ve Raman (2017), farklı ölçek ve konumdaki bıçakların çevrimiçi videolarda tespiti için bir çerçeve sunmuştur. Çalışmada ön plan bölütleme ile özellik çıkarımı, Hızlandırılmış Bölütleme Test (Features from Accelerated Segment Test – FAST) tekniği ile nesne tespiti ve Çoklu Çözünürlük Analizi (Multi-Resolution Analysis – MRA) tekniği ile nesne sınıflandırma işlemini yaparak %98,48 doğruluk değerini elde ettiklerini belirtmişlerdir. Fakat eğitim ve test sırasında kullanılan veri sayısının azlığı üretilen performans sonucunun ve süresinin gerçekçi olmadığını düşündürmektedir.

Geleneksel yöntemler ile yapılan tehlikeli nesne tespit çalışmalarının özeti Tablo 2.3'te gösterilmektedir.

Tablo 2.3: Geleneksel yöntemler ile yapılmış nesne tespit çalışmaları.

Çalışma	Model	Veri Seti Boyutu	Performans Değerleri				
			<i>mAP50</i>	<i>Doğruluk</i>	<i>Duyarlılık</i>	<i>Hassasiyet</i>	<i>Özgüllük</i>
Grega vd. (2013)	Sürgülü p. +Canny+Sinir ağı	4500	-	-	-	-	0,966
Tivari vd. (2015a)	Harris+FREAK	89	-	0,842	-	-	-
Tivari vd. (2015b)	SURF	25	-	0,886	-	-	-
Grega vd. (2016)	Sürgülü p. +Canny	12,899	-	-	-	-	0,966
Vajhala vd. (2016)	HOG+YSA	10,508	-	0,830	-	-	0,857
Buckchash & Raman (2017)	FAST+MRA	1448	-	0,984	0,980	0,970	-

2.4.2. Derin Öğrenme Tabanlı Yöntemler ile Gerçekleştirilen Çalışmalar

2014 yılında tanıtılan RCNN ve OverFeat mimarîleri nesne tespit işlemlerinde çığır açarak tespit işlemlerinin başarısını ve hızını arttırmıştır. Bu dönem itibari ile yapılan çalışmaların birçoğu DL tabanlı mimarîler çerçevesinde gerçekleştirilmiştir. Sunulan bu mimarîler ile birçok nesne tanıma sistemi (yaya tespiti, otonom sürüş, yüz tanıma vs.) geliştirilmesine rağmen tehlikeli nesneler özelinde bilinen ilk çalışma Lai ve Mapes (2017) tarafından hazırlanmıştır. Yapının omurgası OverFeat mimarîsinden oluşturulmuş ve gerçek zamanlı sınıflandırma yapabilmek adına film sahnelerinden alınan veriler kullanılmıştır. En iyi performansı Overfeat-3'te %93 eğitim doğruluğu ve %89 test doğruluğu ile elde etmişlerdir. Ancak çalışmada da belirtildiği gibi her sınıflandırmanın yaklaşık 1,3 saniye sürmesi gerçek zamanlı gözetlemenin yapılamayacağını göstermektedir. Aynı yıl Verma ve Dhillon (2017) tarafından karmaşık sahnelerde otomatik silah algılama için Faster-RCNN tabanlı model geliştirilmiştir. Veri setlerini film, televizyon şovları, video oyunları ve animelerden oluşturarak önerdikleri model ile SVM, K En Yakın Komşu (K Nearest Neighbors – KNN) ve Topluluk Ağacı (Ensemble Tree – ES) modellerini karşılaştırmışlardır. En iyi sonucu %93 sınıflandırma doğruluğu ile topluluk ağaçları algoritmasıyla elde etmişlerdir.

Olmos ve diğ. (2018), sürgülü pencere ve bölge önerisi yaklaşımını karşılaştırmıştır. Her iki yaklaşımda yer alan modelleri hazırladıkları 3000 adetlik veri seti ile ince ayar yaparak

eğitmişlerdir. En iyi sonucu Faster-RCNN ile elde ettiklerini belirterek düşük çözünürlükteki videolarda dahi başarılı sonuçlar üretebildiklerini göstermişlerdir.

Akcaç ve diğ. (2018), X-RAY verilerini kullanarak sürgülü pencere tabanlı CNN, Faster-RCNN, RFCN, YOLOv2 modellerinin performansını karşılaştırdıklarında YOLOv2 modelinin hem doğruluk hem de hız açısından diğer modellerden daha iyi olduğunu göstermişlerdir.

Singleton ve diğ. (2018), çeşitli yönlerde, şekillerde ve boyutlarda silahların tanımlamak için yaklaşık 5 bin veri kullanarak MobileNetv1 ağını eğitmişlerdir. Modeli test ederken 30 silahlı fotoğraftan 26'sını başarı ile tespit ettiklerini belirterek %86,67 oranında sınıflandırma doğruluğu elde ettiklerini göstermişlerdir. Fakat kullandıkları test seti çok küçük olduğu için bu oranın gerçekçi olmadığı düşünülmektedir.

Gelena ve Yadav (2019), CCTV operatörlerini hem görsel hem de sesli bir şekilde uyarılması için CNN tabanlı model önerisinde bulunmuştur. Görüntü verilerini siyah-beyaza çevirip, filtreleme, kenar algılama, arka plan çıkarımı ve sürgülü pencere işlemlerini gerçekleştirerek elde ettikleri özellikleri CNN mimarisine gönderip bir nesnenin silah ya da silah olmamasını tahmin etmişlerdir.

Romero ve Salamea (2019), YOLO mimarisini kullanarak iki bölümden oluşan ateşli silah tespit modeli tasarlamışlardır. İlk bölümde YOLO ile nesne algılama ve yer belirleme işlemini, ikinci bölümde ise CNN ile ateşli silah algılama modelini gerçekleştirmişlerdir. CNN modeli olarak VGGNet ve ZFNet algoritmaları kullanmış ve performanslarını karşılaştırmışlardır. Buna göre gri tonlamalı görüntülerin renkli görüntülerden, VGGNet tabanlı modelin, ZFNet tabanlı modelden daha başarılı olduğunu göstermişlerdir.

Fernandez-Carrobles ve diğ. (2019), silah ve bıçak tespiti için Faster-RCNN tabanlı model önermişlerdir. Modelin omurgasında sırasıyla bir GoogleNet ve bir SqueezeNet mimarisi alınarak iki yaklaşımı karşılaştırmışlardır. Silah tespiti için en iyi sonuç %85,44 ortalama hassasiyet değeri ile SqueezeNet mimarisi kullanılarak elde edilirken, bıçak tespiti için en iyi sonucu %46,68 ortalama hassasiyet değeri ile GoogleNet yaklaşımı kullanarak elde etmişlerdir.

Iqbal ve diğ. (2019), silahların ve tüfeklerin algılanması için iki aşamalı Yönelim Duyarlı Nesne Dedektörü (Orientation Aware Object Detector – OAOD) modelini önermişlerdir. Aşama-1'de, nesnenin yönünü tahmin ederken, aşama-2'de yeniden sınıflandırılan ve yerleştirilen döndürülmüş nesne önerilerinden maksimum alan dikdörtgenlerini kırpmaaktadırlar. Önerilen OAOD modelini, aynı veri seti üzerinde YOLOv2, YOLOv3, SSD, Evrişimsiz Tek Atım Dedektörü (Deconvolutional Single Shot Detector – DSSD) ve Faster-RCNN dahil olmak üzere beş mevcut model ile değişen IoU ve mAP metriklerine göre

karşılaştırmışlardır. Sonuç olarak önerilen OAOD modelinin diğerlerinden daha başarılı sonuçlar ürettiğini göstermişlerdir. Fakat bu sistemin tüfek vb. uzun ve ince nesneler için daha uygun olduğu düşünülmektedir.

Azevedo Kanehisa ve Almeida Neto (2019) tarafından YOLO mimarîsinin ateşli silah tespitiindeki başarısı gözlemlenmiştir. Veri setini %90 eğitim %10 test şeklinde ayırıp validasyon seti kullanmadan eğittikleri modelin sonucunda %70,72 mAP, %95,73 duyarlılık, %97,30 özgüllük ve %96,26 sınıflandırma doğruluğu elde etmişlerdir.

Raturi ve diğ. (2019), gizli silahların tespiti ve sınıflandırılması için Faster-RCNN tabanlı bir model önermiştir. Modelin eğitimi için hazırlanan veri setindeki gürültüyü Canny kenar algoritması ile giderip nesne tespiti için de sürgülü pencere tekniği kullanarak yeni bir çerçeve önerdiklerini belirtmişlerdir.

González ve diğ. (2020), modelin öğrenme başarısını arttırmak için gerçek ve sentetik veri seti kullanarak Faster-RCNN-FPN tabanlı silah tespit modeli önermiştir. Omurgası ResNet50 olan Faster-RCNN mimarîsini FPN mimarîsi ile birleştirip daha hızlı çalışmasını sağlamışlardır. Sentetik veriler kullanılarak yapılan eğitimin, modele çeşitlilik sağlayıp doğruluğu artırdığını ve gerçek verilerin algılanmasını kolaylaştırdığını belirtmişlerdir. Fakat gerçek zamanlı görüntülerde uzaktaki nesneleri algılamayı çözemediklerini de ifade etmişlerdir.

Perez-Hernandez ve diğ. (2020), küçük nesneler (silah, akıllı telefon, bıçak, fatura, cüzdan ve kart) arasındaki algılamayı geliştirmek amacıyla iki aşamalı ODeBiC metodolojisini önermiştir. Aşama-1’de hedef nesneleri içeren aday bölgeleri elde etmek için omurgası ResNet101 olan Faster-RCNN mimarîsini eklemişlerdir. Aşama-2’de ise, her bir bölgeyi binarizasyon tekniği ile sınıflandırarak çıktı çerçevesini algılama sonuçlarıyla birlikte üretmek için bir toplama yöntemi izlemişlerdir.

Kayalvizhi ve diğ. (2020), X-ışını görüntüleme sistemlerinde bıçak, makas, İngiliz anahtarı, pense gibi nesnelerin tespiti için DL tabanlı bir çözüm önermiştir. YOLOv3 modelinin omurgasında DarkNet53 yerine Inceptionv3 ve ResNet50 kullanarak karşılaştırma yapmışlardır. ResNet50 omurgası kullanılarak elde edilen sonuçlar Inceptionv3’ten elde edilen sonuçlara göre daha başarılı olmasına rağmen mAP metriğinin herhangi bir nesne için dahi %70’i aşamadığı görülmüştür.

Rose ve diğ. (2020), tehlikeli nesnelerin tespitini kendi topladıkları veri setini kullanarak eğittikleri Faster-RCNN tabanlı model önermiştir. Gerçek gözetleme görüntülerinden alınan video karelerindeki ateşli silahların tespitini hassasiyet ve duyarlılık puanlarına göre

değerlendirdiklerinde sırasıyla; tabancalarda %93,9 ve %96,4- tüfeklerde %95,2 ve %94,6 sonucunu elde etmişlerdir.

Noor ve Isa (2021), YOLOv4 modeli ile silah ve bıçak gibi tehlikeli nesneleri tespit etmeye çalışmıştır. Tek sınıflı ve çok sınıflı iki eğitim seti oluşturarak sistemin etkinliğini test etmişlerdir. Tek sınıflı nesne algılamada %66 ila %77 arasında sınıflandırma doğruluğu elde ederken, çoklu sınıflı nesne algılamada %100'e kadar sınıflandırma doğruluğu elde etmeyi başarmışlardır.

Narejo ve diğ. (2021), daha az hesaplama kaynağı ile ateşli silahları kısa sürede tespit edebilen bir çerçeve önermiştir. IP kameralar aracılığı ile güvenlik güçlerine ulaşarak girişleri kilitleme sistemini programlamışlardır. Nesne tespiti için YOLOv3 modelini, YOLOv2 ile karşılaştırarak YOLOv3'ün daha başarılı sonuç ürettiğini göstermişlerdir.

Hashmi ve diğ. (2021), CCTV verilerini kullanarak görüntüdeki silahın varlığını tespit etmek için YOLOv3 ve YOLOv4 modellerinin karşılaştırmalı analizini yapmıştır. YOLOv4 modelinin işlem süresi ve hassasiyet açısından YOLOv3'ten bariz bir şekilde üstün performans gösterdiğini belirtmişlerdir.

Bhatti ve diğ. (2021), CCTV görüntülerinden silah tespitini başarabilen, düşük çözünürlük ve parlaklıkta bile iyi çalışan model geliştirmek için sürgülü pencere yaklaşımında VGG16, InceptionV3 Inception ResnetV2 sınıflandırıcılarını ve bölge önerisi yaklaşımında SSDMobileNetV1, Faster-RCNN Inception-ResnetV2 (FRIRv2), YOLOv3 ve YOLOv4 modellerinin karşılaştırmışlardır. Sonuçta, YOLOv4 mimarîsinin, diğer mimarîlerden daha başarılı olduğunu %91,73 mAP değeri ve %91'lik bir F1 puanı ile göstermişlerdir.

Salido ve diğ. (2021), silah tespiti için üç farklı mimarî kullanarak, bu mimarîlerin karşılaştırmalı analizini gerçekleştirmiştir. Çalışmada, omurgası VGG16 olan Faster-RCNN, omurgası ResNet50 olan RestinaNet, omurgası DarkNet53 olan YOLOv3 olmak üzere 3 model kullanmışlardır. ResNet50 omurgasıyla ince ayar yapılmış RetinaNet tarafından elde edilen en iyi ortalama hassasiyet değeri %96,36 ve duyarlılık değeri %97,23 olarak hesaplamışlardır. Ayrıca en iyi hassasiyet değerini %96,23 ve F1 puan değerini %93,3 ile YOLOv3 mimarîsini kullanarak elde etmişlerdir.

Iqbal ve diğ. (2021), geliştirdiği sistem ile drone gözetimi sonucunda herhangi bir olağandışı faaliyetin tespit edilmesinden sonra, sistemin güvenlik görevlileri için bir uyarı oluşturmasını sağlamışlardır. Gözetim görüntülerinin ilgili nesneleri belirleyebilmesi için Faster-RCNN modelini 4 farklı omurga (SqueezeNet, GoogleNet, ResNet-18 ve ResNet-50)

kullanarak karşılaştırmışlardır. Sonuçta ResNet50 omurgasına sahip Faster-RCNN modelinin en iyi sonucu ürettiğini gözlemlemişlerdir.

Sivakumar (2021), suç oluşumu hakkında güvenlik güçlerini uyaran bir sistem önermiştir. Sistemde YOLOv4 modelini kullanıp Faster-RCNN modeli ile performans karşılaştırması yapmıştır. Çalışmanın sonucunda YOLOv4 modelinin daha başarılı olduğunu göstermiştir.

Kaya ve diğ. (2021), DL yöntemi kullanarak 7 farklı silah türünün tespiti için yeni bir model önermiştir. Bu modelin VGGNet mimarisine dayalı silah sınıflandırmasına yeni bir yaklaşım sunduğunu belirterek, VGG16, ResNet50 ve ResNet101 modeli ile karşılaştırmışlardır. Karşılaştırma sonucunda önerilen modelin %98,40 sınıflandırma doğruluğuyla ve diğer metriklerle göre de en iyi sonuca sahip olduğunu göstermişlerdir.

Bushra ve diğ. (2022), web kameraları vasıtası ile silah tespiti yapıp suçluların yüzü ile birlikte tam konumunu ve meydana geldiği olay zamanını YOLOv5 modeli ile bulmayı hedeflenmişlerdir. Modelin düşük çözünürlüklü ve küçük nesneleri doğru bir şekilde yakalayabildiğini belirtmişlerdir.

Harvey ve diğ. (2023), sosyal medyada yer alan ateşli silah görüntülerini tespit etmek amacıyla VINOD adındaki kendi modellerini Twitter’ dan (X) topladıkları gerçek dünya verileri ile eğitmişlerdir. VINOD omurga yapısı olarak ResNeXt-152 C4 mimarisi kullanan 2 gizli katmana sahip DL algoritmasıdır. Performansını YOLOv5 modeli ile karşılaştırdıklarında önerdikleri modelin daha başarılı sonuç ürettiğini belirtmişlerdir.

Yellapragada ve diğ. (2023), CCTV kameralarından elde edilen görüntülerdeki silah tespit zorluklarını ele alarak mevcut 3 veri setini alıp içlerinden uygun çerçeveleri derleyerek yeni bir veri seti birleşimi önermişlerdir. Ayrıca Son Teknoloji CNN Tabanlı Nesne Dedektörleri (CNN-Based Object Detectors And State of the Art – SOTA) adını verdikleri farklı omurga ve çerçevedeki 5 nesne tespit algoritmalarını karşılaştırarak analiz etmişlerdir. Analizler veri setlerinin farklı kombinasyonları ile gerçekleştirmişlerdir. Sonuç olarak modellerin farklı veri setleri üzerinde ince ayarının yapılması karışık sonuçlar gösterdiğini ve net bir sonuç elde edemediklerini fakat model olarak ConvNext-t omurgasına sahip FasterRCNN+FPN çerçevesinin başarılı sonuç ürettiğini belirtmişlerdir.

Wang ve diğ. (2023b), CCTV otonom silah tespiti için gerçek zamanlı bir model geliştirmişlerdir. Bu model için Uzamsal Çapraz Aşama Kısmi ResNet (Spatial Cross Stage Partial-ResNet- SCSP – ResNet) içeren YOLOv4 omurgasını önermişlerdir. Ayrıca model eğitiminde sentetik ve gerçek dünya verileri kullanmışlardır. Sonuç olarak %4 ila %7 oranında bir iyileşme sağlamışlardır.

DL tabanlı modern yöntemler ile yapılan tehlikeli nesne tespit çalışmalarının özeti Tablo 2.4'te gösterilmektedir

Tablo 2.4: DL tabanlı modern yöntemler ile yapılmış nesne tespit çalışmaları.

Çalışma	Model	Veri Seti Boyutu	Performans Değerleri				
			mAP50	Doğruluk	Duyarlılık	Hassasiyet	Özgüllük
Lai & Maples (2017)	OverFeat	2.735	-	0,890	-	-	-
Verma & Dhillon (2017)	Faster-RCNN	-	-	0,931	-	-	-
Olmos vd. (2018)	Faster-RCNN+ VGG16	3.000	-	-	1,00	0,842	-
Akçay vd. (2018)	Sürgülü p.+ YOLOv2	19.398 - 9.680 13.770	0,974	-	-	-	-
Singleton vd. (2018)	MobileNetv1	4.794	-	0,866	-	-	-
Gelana & Yadav (2019)	Sürgülü p.+ CNN	5.869	-	0,977	-	0,938	0,997
Romero & Salamea (2019)	YOLO+VGGNet	17.684	-	0,861	0,860	0,860	-
Fernandez-Carrobles vd. (2019)	SqueezeNet	15.000	-	-	-	0,854*	-
Iqbal vd. (2019)	OAOD	10.973	0,899	-	-	~0,900*	-
Azevedo Kanehisa & Almeida Neto (2019)	YOLO	4.646	0,707	0,962	-	-	0,973
Raturi vd. (2019)	Canny+ Faster-RCNN	9.084	-	0,936	-	-	-
Gonzalez vd. (2020)	Faster-RCNN+ ResNet50+FPN	-	-	-	1,00	0,881	-
Perez-Hernandez vd. (2020)	Faster-RCNN+ ResNet101	-	-	-	0,930	0,938	-
Kayalvizhi vd. (2020)	YOLOv3+ ResNet50	1.059.231	0,637	-	-	-	-
Rose vd. (2020)	Faster-RCNN	11.652	0,672	-	0,964 (±0,2)	0,939 (±0,2)	-
Noor vd. (2021)	YOLOv4+ DarkNet	3.000	-	0,71 (±0,5)	-	-	-
Narejo vd. (2021)	YOLOv3+ DarkNet53	~500	-	0,989	-	-	-
Hashmi vd. (2021)	YOLOv4	7.800	0,848	-	0,780	0,850	-
Bhatti vd. (2021)	YOLOv4	1.732 5.254 8.237	0,917	0,990	-	0,930	-
Kaya vd. (2021)	VGGNet tabanlı model	5.214	0,873	0,984	-	-	0,992

Tablo 2.4 (devam): DL tabanlı modern yöntemler ile yapılmış nesne tespit çalışmaları.

Çalışma	Model	Veri Seti Boyutu	Performans Değerleri				
			mAP50	Doğruluk	Duyarlılık	Hassasiyet	Özgüllük
Bushra vd. (2022)	YOLOv5	2.954	0,950	-	0,870	0,980	-
Harvey vd. (2023)	VINOD	400	0,723	-	-	-	-
Yellapragada vd. (2023)	FasterRCNN+FPN	7.767	-	-	-	-	--
Wang vd. (2023b)	YOLOv4	10.231	0,817	-	-	-	-

*ifadesi değerin ortalama olduğunu belirtmek için kullanılmıştır.

IMFDB: Internet Movie Firearms Database- İnternet Film Ateşli Silahlar Veri tabanı

Geleneksel ve modern yöntemler kullanılarak tehlikeli nesneleri tespit etmeyi hedefleyen çalışmaların sınırlı sayıda olduğu görülmektedir. Özel olarak seçilen bu nesne grubuna ilişkin tespit çalışmalarının literatürde daha fazla yer alması için yapılması gereken en önemli şeyin kamuya açık veri setleri oluşturmak olduğu görülmüştür. Bu alanda literatürde yer alan çalışmaların çoğu veri setini taraflarınca oluşturmasına rağmen kamuya açık bir şekilde sunmadığı görülmüştür. Bu durum nesnel bir karşılaştırma yapılmasının önüne geçtiği gibi veri seti hazırlama işlemlerinin zorluğundan ötürü mevcut çalışmaların sayısının sınırlı sayıda kalmasına sebep olduğu düşünülmektedir.

2.4.3. SR algoritmaları ile Gerçekleştirilen Çalışmalar

Noh ve diğ. (2019), küçük nesneleri tespit edebilmek için Üretken Çekişmeli Ağ (Generative Adversarial Networks – GAN) tabanlı SR ve nesne algılama ağı arasında geri besleme döngüsü oluşturmuştur. Geri besleme ile iki ağın birlikte optimize edilmesini ve birbirlerine karşı adaptasyonlarını sağlayarak nesne tespitinin başarısını arttırdığını belirtmişlerdir. Faster R-CNN'nin çeşitli veri tabanlarında performansını iyileştirdiğini belirtmişlerdir. Ayrıca önerilen yöntemin küçük nesnelerin tespitinde daha iyi performans gösterdiğini ve diğer SR yöntemlerine kıyasla daha başarılı sonuçlar elde edildiğini belirtmişlerdir.

Shermeyer ve Van Etten (2019), uydu görüntülerini Çok Derin Süper Çözünürlük (Very Deep Super Resolution – VDSR) ve özel bir Rastgele Orman Süper Çözünürlüğü (Random Forest Super-Resolution – RSFR) tekniği ile farklı ölçeklerde iyileştirdikten sonra Sadece İki Kere Bak (You Only Look Twice – YOLT) ve Tek Atış Çoklu Kutu Dedektörü SSD modellerinin nesne tespitinde mAP metriğinin %13 ila %36 arasında arttığını gözlemlemiştir.

Fakat çözünürlük boyutunun çok arttığında mAP değerinin düştüğünü ya da hiçbir iyileşme olmadığını da kaydetmişlerdir.

Fredous ve diğ. (2019), uydu görüntülerini tespit edebilmek adına GAN tabanlı bir SR modeli kullanarak görüntüleri iyileştirdikten sonra nesne tespiti için SSD modelini kullanmışlardır.

Truong ve diğ. (2020), havadan çekilmiş görüntüleri dikkat tabanlı ağ tabanlı Artık Kanal Dikkat Ağı (Residual Channel Attention Network- RCAN) tekniği ile iyileştirmesinin ardından Fast-RCNN, Cascade-RCNN, DetectoRS, Retina, SSD modellerindeki başarının olumlu yönde arttığını gözlemlemiştir.

Courtrai ve diğ. (2020), uydu veya havadan çekilen görüntülerdeki küçük nesnelerin algılanması sorununu ele alarak YOLOv3, Faster-RCNN, RetinaNet ve EfficientDet modellerinin performans karşılaştırması gerçekleştir. Modellerin performansını arttırmak için EDSR tabanlı SR ağı önererek, EDSR modelinin boyutu ve artık bloklarının sayısı artırılmış ve daha fazla detay yakalaması sağlanmıştır. Daha sonra, döngü Wasserstein GAN modeli kullanılarak, düşük çözünürlüklü girdi görüntülerinden yüksek çözünürlüklü çıktı görüntüleri üreterek SR-WGAN modelini önermişlerdir. Sonuçta Faster-RCNN ile en başarılı nesne tespit işlemini gerçekleştirdiklerini belirtmişlerdir.

Rabbi ve diğ. (2020), uydu görüntülerinin tespit edilebilmesi için Faster-RCNN ve önerdikleri GAN tabanlı SR modelini kullanarak küçük nesnelerin tespitini gerçekleştirmişlerdir.

Bashir ve Wang (2021), küçük nesnelerin başarılı bir şekilde tespit edilebilmesi için SRCGAN ve EDSR ile Artık Özellik Toplama (Residual Feature Aggregation – RFA) başka bir deyiş ile EDSR-RFA SR algoritmalarını kullanarak çözünürlük üzerindeki etkilerini karşılaştırmıştır. Ardından ağı eğitimi için YOLOv3 modelini kullanıp en iyi çözünürlük sonucunu üreten SR tekniği ile birleştirerek SRCGAN-RFA-YOLO modelini önermişlerdir. Uydu ve havadan görüntüler üzerinde model eğitildiğinde nesne tespit performansının başarısının çift kübik enterpolasyon yöntemine göre AP değerini %54,3 oranında arttırdığını gözlemlemiştir.

Wang ve diğ. (2021), küçük nesnelerin tespitini başarılı bir şekilde gerçekleştirebilmek adına YOLOv4 tabanlı bir model geliştirerek modelin görüntüyü iyileştirebilmesi için Özellik Doku Aktarımı (Feature Texture Transfer – FTT) modülünü eklemiştir. Sonuç olarak düşük çözünürlükteki küçük nesneleri başarılı bir şekilde tespit edebildiklerini belirtmişlerdir.

Wang ve diğ. (2022), uydu görüntülerindeki küçük nesnelerin tespiti için kendi topladıkları veri setini kullanarak RFA tabanlı bir SR yöntemi kullanmayı ve döngüsel GAN tabanlı bir yaklaşım kullanarak YOLOv5 modelini iyileştirmeyi hedefleyen Artık özellik toplama ile GAN (GAN with Residual Feature Aggregation – MCGR) ağını önermiştir. Bunun sonucunda $x2$ ve $x4$ ölçek boyutları için mAP'de sırasıyla %5 ve %13 oranında standart YOLOv5'ten daha iyi performans gösterdiğini belirtmişlerdir.

Herhangi bir nesne üzerinde nesne tespiti ve SR algoritmalarını kullanılarak hazırlandığı modelleri içeren çalışmalarının özeti Tablo 2.5'te gösterilmektedir

Tablo 2.5: Nesne tespiti ve SR uygulamaları.

Çalışma	Model		Veri Seti Boyutu
	Nesne Tespit	SR	
Noh vd. (2019)	Faster-RCNN	GAN tabanlı SR	10.000
Shermeyer & Van Etten (2019)	YOLT, SSD	VDSR, RFSR	257.853
Ferdous vd. (2019)	SSD	GAN tabanlı SR	-
Truong vd. (2020)	FAST-RCNN, Cascade-RCNN, DetectoRS, Retina, SSD	RCAN	8.599
Courtrai vd. (2020)	YOLOv3, Faster-RCNN, RetinaNet ve EfficientDet	SR-WGAN, SR-CWGAN	10.000
Rabbi vd. (2020).	Faster-RCNN	GAN tabanlı SR	12.651
Bashir & Wang (2021).	YOLOv3	SRCGAN-RFA	2.244
Wang vd. (2021)	YOLOv4	FTT	2.200
Wang vd. (2022)	YOLOv5	RFA	22.091

3. YÖNTEM

Bu bölümde tez kapsamında hazırlanan veri seti, önerilen SR algoritması, önerilen hibrit nesne tespit modeli ve deneysel çalışmaların değerlendirilmesini sağlayan metrikler detaylı bir şekilde açıklanmaktadır.

3.1. VERİ SETİ

Hedeflenen nesnenin model tarafından tanınabilmesi için öncelikle bu nesneyi öğrenmesi gerekir. Öğrenimin başarılı bir şekilde gerçekleşmesi kurulan model yapısı ve veri setine bağlıdır. Dolayısı ile hedef nesneyi tespit etmeye çalışırken geliştirilen modelin başarısı; katman yapısı, katman sayısı, aktivasyon fonksiyonu, kayıp fonksiyonu, optimizasyon fonksiyonu ve diğer ara hiperparametre değerlerine ek olarak veri setinin içeriği, yapısı ve boyutuna bağlıdır.

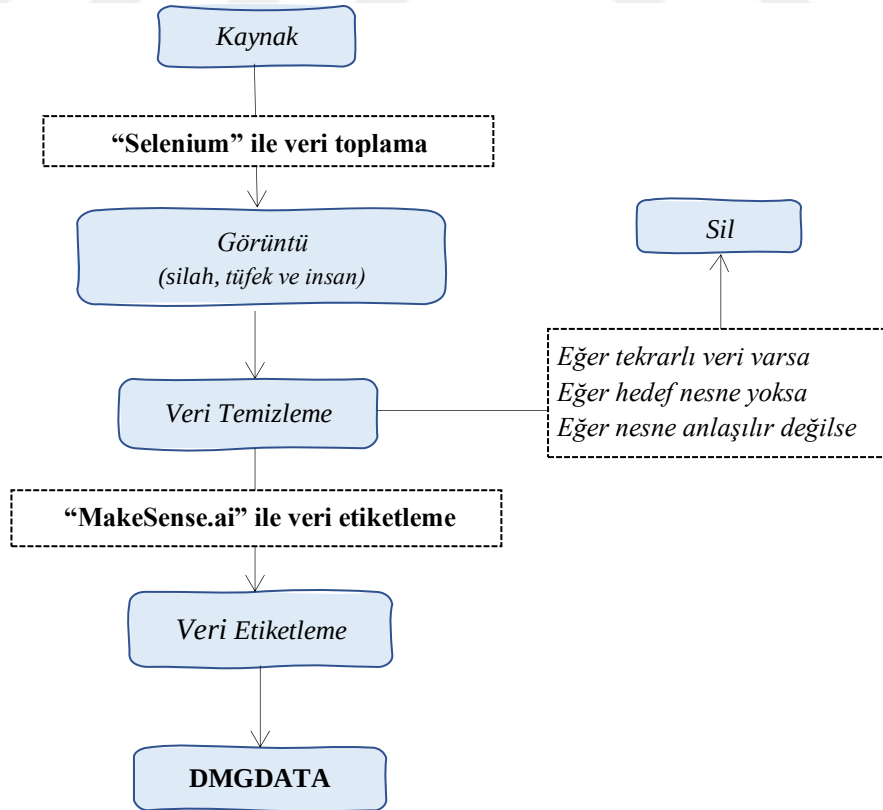
Tez çalışması kapsamında kullanabilmek için hedeflenen nesnelerin bulunduğu açık kaynaklı veri setleri araştırılmış fakat oldukça kısıtlı olduğu görülmüştür (Grega vd., 2016; Olmos vd., 2018; Olmos vd., 2019; Sasank 2019; <https://public.roboflow.com/object-detection/pistols/1>, Erişim tarihi: 9 Nisan 2023, 2020; Qi vd., 2021;). Kamuya açılmış bu içerikler incelendiğinde etiketsiz ya da istenilen nesnelerin (silah, tüfek ve insan) etiketlenmediği görülmüştür. Genel olarak veri setleri Tablo 3.1’de gösterildiği gibi silah etiketi ile etiketlenmiştir. Dolayısı ile silah etiketi ateşli silahların tüm türünü kapsamakta olup tabanca ya da tüfek gibi bir ayırım yapılmamıştır.

Tablo 3.1: Kamuya açık tehlikeli veri setleri ve özellikleri.

Kaynak	Veri Seti Büyüklüğü	Etiketleme Formatı	İçeriği
Grega et al. (2013)	4.500	Etiketsiz	Silah
Grega et al. (2016)	12.899	Etiketsiz	Silah ve bıçak
Olmos et al. (2018)	3.000	PASCAL VOC	Silah
Perez-Hernandez et al. (2020)	5.879	YOLO / PASCAL VOC	Silah, bıçak, akıllı telefon, fatura, cüzdan ve kart
Noor et al. (2021)	3.000	Belirsiz	Silah
Qi et al. (2021)	46.242	PASCAL VOC	Silah
Sasank (2024)	333	Belirsiz	Silah

Kamuya açık tehlikeli veri setleri incelendiğinde bu tez için yeterli olmadığı sonucuna varılmıştır. Bu doğrultunda tez kapsamında toplanan veriler Şekil 3.1’de gösterildiği gibi şu adımları içermektedir:

- Selenium test otomasyon kütüphanesi aracılığı ile Google Images, Yandex Images, Bing Images ve Internet Movie Firearms Database (IMFDB) kaynaklarından .jpg formatlı görüntüler toplanmıştır. Toplanan veriler hedef nesneyi içeren anahtar kelimeler yardımı ile keşfedilmiş olup silah, tüfek ve çeşitli ateşli silah türlerini taşıyan insanları içermektedir. Toplama işlemi sonucunda 24.322 adetlik bir veri seti oluşturulmuştur.
- Elde edilen veri seti üzerinde veri temizleme işlemi manuel olarak gerçekleştirilerek anlamsız ve tekrarlayan veriler çıkartılmıştır. Bu işlem sonucunda veri setinden 18.231 kullanılabilir veri tespit edilmiştir.
- Verilerin her biri manuel olarak tek tek etiketlenmiştir. Etiketleme işlemi kişisel ve profesyonel kullanım için kullanılan ve YOLO, VOC XML, CSV biçiminde dosya dışı aktarımını destekleyen açık kaynaklı online MakeSense.ai platformu üzerinde gerçekleştirilmiştir (<https://www.makesense.ai/>, Erişim tarihi: 13 Nisan 2023).



Şekil 3.1: Veri seti oluşturma adımları

Etiketleme işleminde veriler “insan”, “silah” ve “tüfek” olmak üzere 3 sınıfa ayrıştırılmıştır. Bu sınıflar 0, 1 ve 2 olarak temsil edilmektedir. Her bir görüntü içerisinde bulunan nesne Şekil 3.2’de görüldüğü gibi 4 koordinatlı bir çerçeve ile işaretlenmiştir.



Şekil 3.2: a) Tüfek görüntüsünün etiketli hali, b) Silah ve insan görüntüsünün etiketli hali, c) Silah ve insan görüntüsünün etiketli hali, d) Etiket bilgileri

İşaretlenen her bir görüntüden koordinat bilgileri elde edilir. Bu koordinatlar görüntü boyutuna göre 0-1 arasında normalleştirilerek depolanır. Elde edilen bilgi kümesinin temsili ise şu şekildedir:

< Nesne sınıfı > < X merkezi > < Y merkezi > < Kutu genişliği > < Kutu yüksekliği >

Belirtilen formatta tutulan bu bilgiler her bir görsel için .txt formatlı bir dosyaya aktarılmaktadır. Ortaya çıkan metin dosyası içerisinde görüntülerde etiketlenen her bir nesnenin sınırlayıcı kutu koordinatları ve sınıf bilgisi yer almaktadır. Son tahlilde elde edilen veri seti DMGDATA olarak isimlendirilmiş ve model eğitimlerinde kullanmaya hazır hale getirilmiştir. Etiketlenen veriler sonraki araştırmacılara kullanım kolaylığı sağlanması amaçlanarak Tablo 3.2’de gösterildiği gibi 3 farklı dosyaya yerleştirilmiştir.

- Silah içeren görüntüler “*gun*” olarak etiketlenmiş ve “Gun” klasörü altına yerleştirilmiştir.
- Tüfek içeren görüntüler “*rifle*” olarak etiketlenmiş ve “Rifle” klasörü altına yerleştirilmiştir.
- İnsan içeren görüntüler “*person*” olarak etiketlenmiş ve genellikle silah ve/veya tüfek taşıyan insanlardan oluştuğu için karma klasör olan “Person_g_r” klasörü altına yerleştirilmiştir.

Tablo 3.2: DMGDATA veri setinin özellikleri.

Veri Setindeki Dosya Adı	Veri Sayısı	Nesne Sayısı
Gun	5.008	11.667
Rifle	5.010	10.332
Person_g_r	8.213	9.689
Toplam	18.231	31.688

Bu klasörlerin ve etiketlerin İngilizce olarak hazırlanmasının temel sebebi uluslararası çalışmalarda kolaylıkla anlaşılabilir ve kullanılabilir olmasını sağlamaktır.

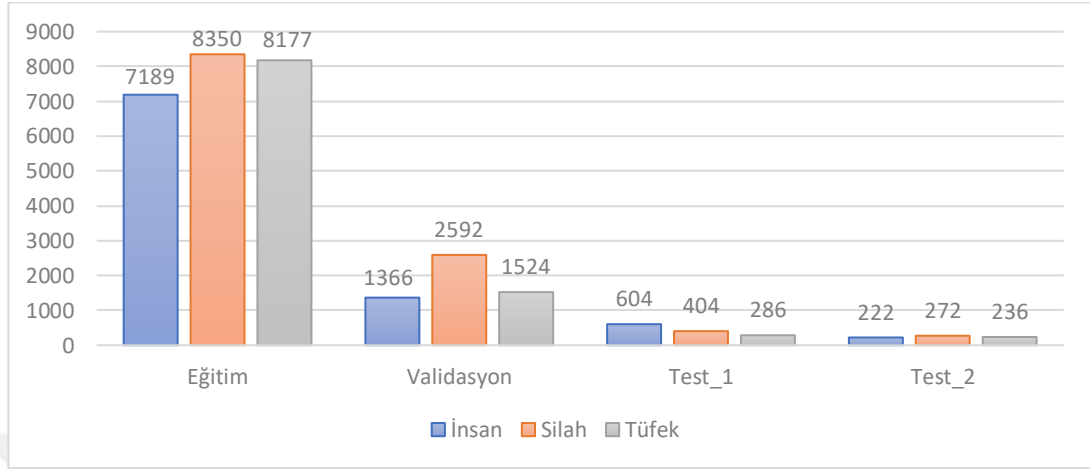
Tez kapsamında hazırlanan DMGDATA veri seti Tablo 3.3’te gösterildiği gibi eğitim, validasyon ve test seti olarak rastgele ayrıştırılmıştır. Test işlemi sırasında iki farklı veri seti kullanılmıştır. Test_1 olarak isimlendirilen veri seti DMGDATA içinden rastgele seçilmiş 500 görüntüyü içermektedir. Test_2 olarak isimlendirilen veri seti ise düşük çözünürlüklü görüntülerin iyileştirilmesi sonrası elde edilen nesne tespit başarısında artış olup olmayacağını gözlemlemek amacıyla Qi ve diğerlerinin (2021) hazırladığı sınıflandırma veri seti içerisinde rastgele seçilmiş 500 küçük resimden oluşmaktadır. Bu veri seti 46.232 adetlik nesne tanıma veri setinden bağımsız olarak hazırladıkları sınıflandırma veri setidir. Bu sınıflandırma veri seti 32x32, 48x48 ve 64x64 gibi farklı büyüklüklere sahip etiketsiz verilerden oluşmaktadır.

Tablo 3.3: Eğitim, validasyon ve test seti bilgileri.

Veri seti türü	Veri Sayısı
Eğitim	13.585
Validasyon	3.646
Test_1	500
Test_2	500

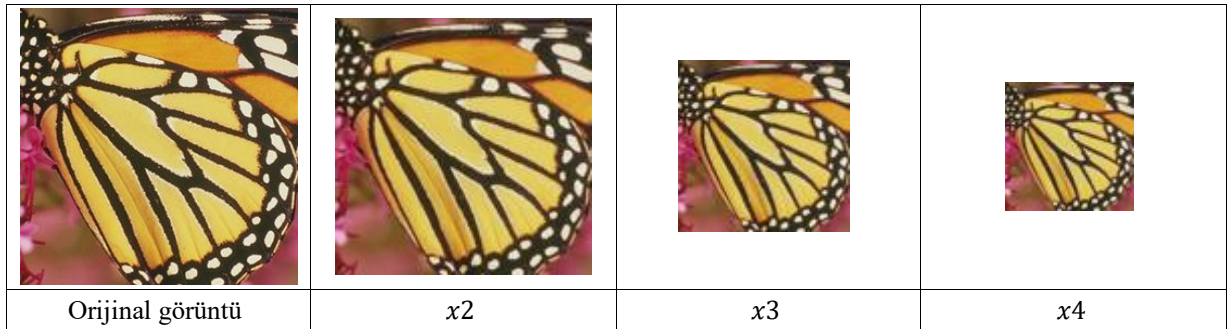
DMGDATA veri setinin eğitim, validasyon ve test setlerine ayrıştırılmasından sonra oluşan nesne dağılımı Şekil 3.3’te gösterilmektedir. Bu grafiğe bakıldığında rastgele

ayrıştırılmış görüntüler içerisinde yer alan nesnelerin dağılımının oldukça dengeli olduğu görülmektedir.



Şekil 3.3: Eğitim, validasyon ve test setinde yer alan nesnelerin dağılım grafiği.

Bu tez kapsamında YOLO modellerinin eğitimi için hazırlanan DMGDATA veri setine ek olarak SR algoritmalarının karşılaştırılması için literatürde sıklıkla kullanılan Set5 (Kim vd., 2016) veri seti de kullanılmıştır. Set5 veri seti içinde SR algoritmalarının performansını test etmek için kullanılan ve “bebek”, “kuş”, “kelebek”, “kafa” ve “kadın” görsellerinden oluşan 5 adet görüntü yer almaktadır. Tez kapsamında SR algoritmalarının testi için Şekil 3.4’te gösterilen orijinal, $x2$, $x3$ ve $x4$ oranında büyütülmesi için hazırlanan LR “kelebek” görüntüleri kullanılmıştır.



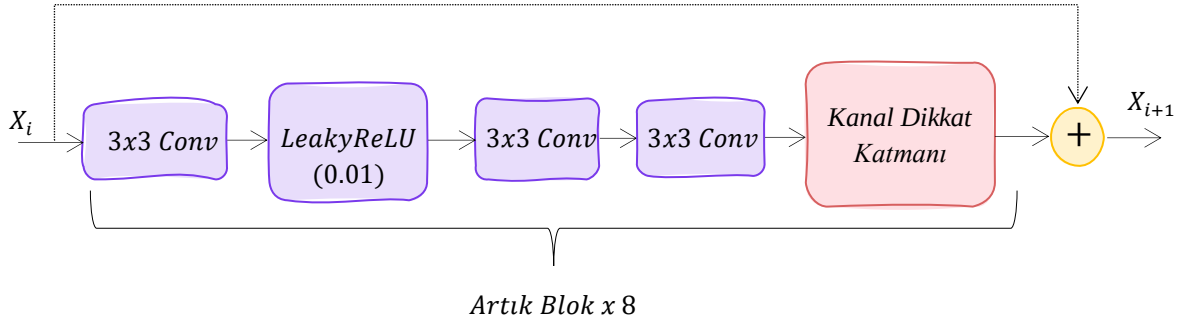
Şekil 3.4: Orijinal, $x2$, $x3$ ve $x4$ oranındaki LR “kelebek” görüntüleri.

3.2. DERİN KANAL DİKKAT SÜPER ÇÖZÜNÜRLÜĞÜ

Bilgisayarla görü tekniği alanında bir görüntünün düşük çözünürlüklü olması hedef nesneyi temsil eden piksel sayısının az olduğu anlamına gelmektedir. Bu durumun sebepleri; görüntünün küçük olması, çekim yapan cihazın kalitesinin düşük olması ya da hedef nesnenin

uzakta olması olabilir. Düşük çözünürlük, görüntüden elde edilen özelliklerin yetersiz olmasına ve hedeflenen nesnenin tespitin zorlaşmasına sebebiyet vermektedir. Birçok sebepten kaynaklanabilen bu problemi çözmek için algoritmik yöntemler kullanmak en az maliyet ve en hızlı çözüm olduğu düşünülmektedir. Bu doğrultuda tez kapsamında tespit edilmeye çalışılan nesnelerin görüntü kalitesini iyileştirmek için DCASR adında yeni bir SR algoritması geliştirilmiştir.

DCASR algoritması birden fazla artık blok ve bir kanal dikkat mekanizmasından oluşan yeni bir SR algoritmasıdır. Geleneksel artık blok yapısının aksine EDSR algoritmasına benzer şekilde artık ağlardaki her evrişim katmanından sonra BN katmanları ve aktivasyon katmanları kaldırılarak bellekten tasarruf sağlamaktadır. Aynı zamanda işlem sayısındaki azalma modelin daha hızlı çalışmasına yardımcı olmaktadır. Şekil 3.5'te gösterildiği gibi artık katman bloğu içerisinde 3 evrişim katmanı ve 1 LeakyRelu aktivasyon fonksiyonu yer almaktadır.

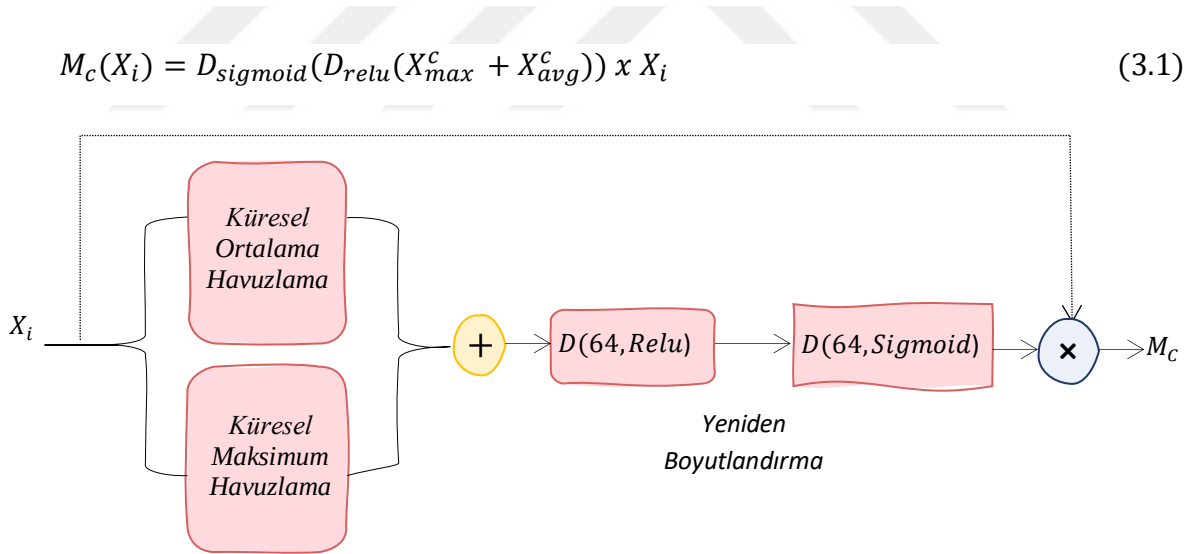


Şekil 3.5: Artık blok yapısı.

Artık blok katmanında her bir evrişim katmanı 64 adet 3×3 boyutlu filtreye sahiptir. İlk evrişim katmanının ardından aktivasyon fonksiyonu kullanılarak doğrusal olmayan bilgilerin öğrenilmesi ve ağırlık öğrenme yeteneğinin artırılması sağlanmıştır. Ayrıca LeakyRelu aktivasyon fonksiyonunun kullanılması ile ReLU'daki kaybolan gradyan sorunun önüne geçilerek negatif bölgelerdeki bilgilerin kullanılması sağlanmıştır. Bu sayede $x < 0$ değerlerinde bilginin kaybolması engellenerek tam öğrenme sağlanmıştır. Ayrıca, öğrenme yeteneklerini artırmak ve önemli bilgileri vurgulamak için her bir artık katman bloğundan sonra Şekil 3.6'daki gibi bir kanal dikkat katmanı eklenmiştir.

Dikkat Katmanı (Attention Layer), verimliliği artırmak ve kaynakları korumak için sınırlı alandan önemli bilgilerin çıkarılmasına yardımcı olan ağırlıklı ortalama azaltma işlemidir. Verilerdeki ayırt edici özelliklere daha çok odaklanarak modelin eğitimini iyileştirmektedir. Genellikle dil modelleri için kullanılan bu katmanın birçok farklı versiyonu mevcuttur. Bu tez

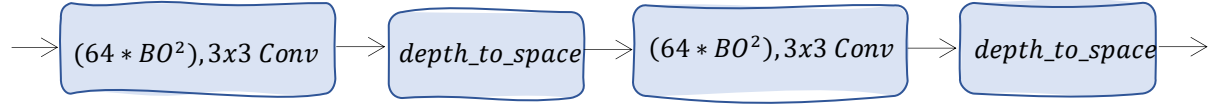
kapsamında özellikle görüntü kanallarındaki önemli bilgileri yakalamak için kullanılan kanal dikkat katmanını kullanılmıştır. Kanal Dikkat Katmanı (Channel Attention Layer) özelliklerin kanallar arası ilişkilerinden faydalanılarak üretilen bir özellik haritasıdır (Woo vd., 2018). Bu katman aldığı veriyi küresel maksimum X_{max}^c ve küresel ortalama X_{avg}^c havuzlama katmanlarından geçirmektedir. Ortalama havuzlama katmanını hedef nesnenin kapsamını etkili bir şekilde öğrenirken, maksimum havuzlama katmanını ince bilgileri çıkarıp hedef nesnenin ayırt edici özellikleri hakkında bilgi vermektedir. Bu iki havuzlama katmanından gelen bilgiler toplanarak farklı aktivasyon fonksiyonuna sahip iki yoğunluk katmanına D_{relu} , $D_{sigmoid}$ sırasıyla gönderilmektedir. Ardından elde edilen sonuç ve giriş verileri çarpılarak ağıın model eğitimi için bilgilendirici kanallara odaklanmasına yardımcı olan özellik haritası M_c üretilmektedir. Kanal dikkat katmanına ait Denklem 3.1’de yer almaktadır. Denklemde yer alan X girdiyi, D yoğunluk katmanını, c kanal sayısını ve M özellik haritasını temsil etmektedir. Bu katman ile aynı zamanda daha az ilgili olanları önemsizleştirerek gereksiz bilgilerin öğrenmesi engellenmektedir. Bu durum ağıın öğrenme sürecini iyileştirmektedir (Lu & Hu; 2022).



Şekil 3.6: Kanal Dikkat katmanı.

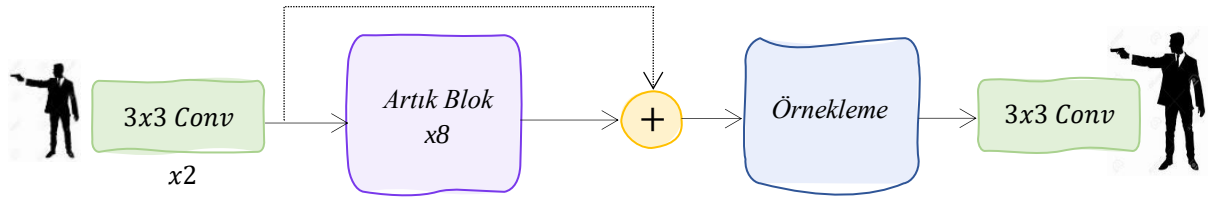
Örnekleme (Upsample) katmanı ise modelin son katmanından önce gelen ve görüntünün çözünürlüğünü, belirlenen büyütme oranında arttırma işlemidir. Bu katman daha büyük özellik haritaları elde etmek ve modelin daha fazla örüntü öğrenmesini sağlamak için kullanılmaktadır. Tez kapsamında örnekleme işlemi Şekil 3.7’de gösterildiği gibi DL yöntemleri ile gerçekleştirilmiştir. Bu katman $64 * BO^2$ adet 3×3 boyutlu filtreden oluşmaktadır. Her evrişim

katmanından sonra gelen TensorFlow kütüphanesine ait `depth_to_space` fonksiyonu ise uzamsal veri bloklarını yeniden düzenleyerek görüntüyü istenen çözünürlüğe ayarlamaktadır.



Şekil 3.7: Örneklem katmanı (BO: Büyütme Oranı).

Sonuçta önerilen DCASR algoritması Şekil 3.8’de gösterildiği gibi toplam 28 evrişim katmanına sahiptir. Her bir evrişim katmanı ise 3x3 boyutunda 64 filtreden oluşmaktadır.



Şekil 3.8: Önerilen DCASR mimarisi.

Bu algoritmanın eğitiminde DIV2K veri setine ait “bicubic_x2”, “bicubic_x3” ve “bicubic_x4” verileri kullanılmıştır (Agustsson & Timofte, 2017). Her bir veri kümesinin içerisinde 800 adet eğitim 100 adet test verisi bulunmaktadır. Modelin olası aşırı öğrenme probleminin önünü geçmek amacıyla veriler rastgele döndürülerek, kırılarak ve sağ-sola yansması alınarak artırılmıştır. Modelin eğitim sırasında HR görüntüleri 96x96 büyüklüğüne ayarlanırken LR görüntüleri x2 eğitimi için 48x48, x3 eğitimi için 32x32 ve x4 eğitimi için 24x24 büyüklüğüne ayarlanmıştır. Eğitim 50 epok ile gerçekleştirilmiştir. Her bir epokta 1.000 iterasyon yapılmıştır. Optimizasyon fonksiyonu olarak Uyarlanabilir Moment tahmini (Adaptive Moment estimation – ADAM) ve kayıp fonksiyonu olarak Ortalama Mutlak Hata (Mean Absoulte Error – MAE) kullanılmıştır. Öğrenme katsayısı ise 0,001 olarak belirlenmiştir. DCASR modelinin eğitim hiperparametreleri Tablo 3.4’te yer almaktadır.

Tablo 3.4: DCASR modelinin eğitim hiperparametreleri.

Hiperparametre		Değer
Veri seti		DIV2K
HR görüntü boyutu		96x96
LR görüntü boyutu	x2	48x48
	x3	32x32
	x4	24x24

Tablo 3.4 (devam): DCASR modelinin eğitim hiperparametreleri.

Hiperparametre	Değer
Epok	50
İterasyon	1.000
Yığın boyutu (Batch size)	16
Öğrenme katsayısı	0,001
Optimizasyon fonksiyonu	ADAM
Kayıp fonksiyonu	MAE

DCASR mimarisi $x2$, $x3$ ve $x4$ çözünürlük oranına göre ayrı ayrı eğitilmiştir. Eğitim sonuçlarını değerlendirmek adına 64×64 boyutundaki orijinal görüntü modele gönderilerek 3 farklı çözünürlük oranına göre iyileştirme gerçekleştirmiştir. Bu işlemin sonucunda $x2$, $x3$ ve $x4$ çözünürlük oranında iyileştirilen görüntüler ve orijinal görüntü Şekil 3.9’da gösterilmektedir.

**Şekil 3.9:** 64×64 boyutlu görüntünün 3 farklı oranda DCASR modeli ile iyileştirilmesi.

3.3. İKİLİ FİLTRE

İkili Filtre (Bilateral Filter – BF) görüntüdeki gürültüleri azaltmayı sağlayan aynı zamanda Gauss bulanıklaştırma filtresinin gelişmiş bir versiyonu olan çift taraflı yumuşatma filtresidir (Tomasi& Manduchi, 1998). Çift taraflı bir filtre kenarları keskin tutarken gürültüyü azalmaya yardımcı olmaktadır. Bu sayede nesneler arasında bulanıklığın oluşması önlenirken aynı alanlardaki gürültü ortadan kaldırılacaktır. Görüntüdeki her bir pikselin yoğunluğu Denklem 3.2’de gösterildiği gibi hedef piksele komşu olan diğer piksel değerlerinin ortalaması ile değiştirilmektedir. Başka bir deyiş ile pikseller arasındaki farklar azaltılarak görüntü yumuşatma işlemi gerçekleştirilmektedir. Tüm piksellerin konumunun yanı sıra hedef piksel ile komşuları arasındaki yoğunluğun eşitsizlik bilgisi de hesaba katılmaktadır. Yoğunluk bilgisinde dikkate alınması bu filtreyi Gauss filtresinden ayıran en önemli noktadır.

Bu sayede Gauss filtresinde yapılan eşit bulanıklaştırma işleminin önüne geçilerek sadece gürültü odaklı ikili filtre elde edilmektedir.

Denklem 3.2’de yer alan I giriş görüntüsünü, p filtrenin uygulanacağı pikselin koordinat bilgisini, Ω bu pikseli merkez alan pencereyi, f_r bir kenarın minimum genliğini belirten yumuşatma fonksiyonunu ve g_s çekirdeğin uzaysal boyutunu, başka bir deyiş ile komşuluğun boyutunu temsil etmektedir.

$$I_p = \frac{1}{w_p} \sum_{q \in \Omega} f_r(|I_p - I_q|) g_s(||p - q||) I_q \quad (3.2)$$

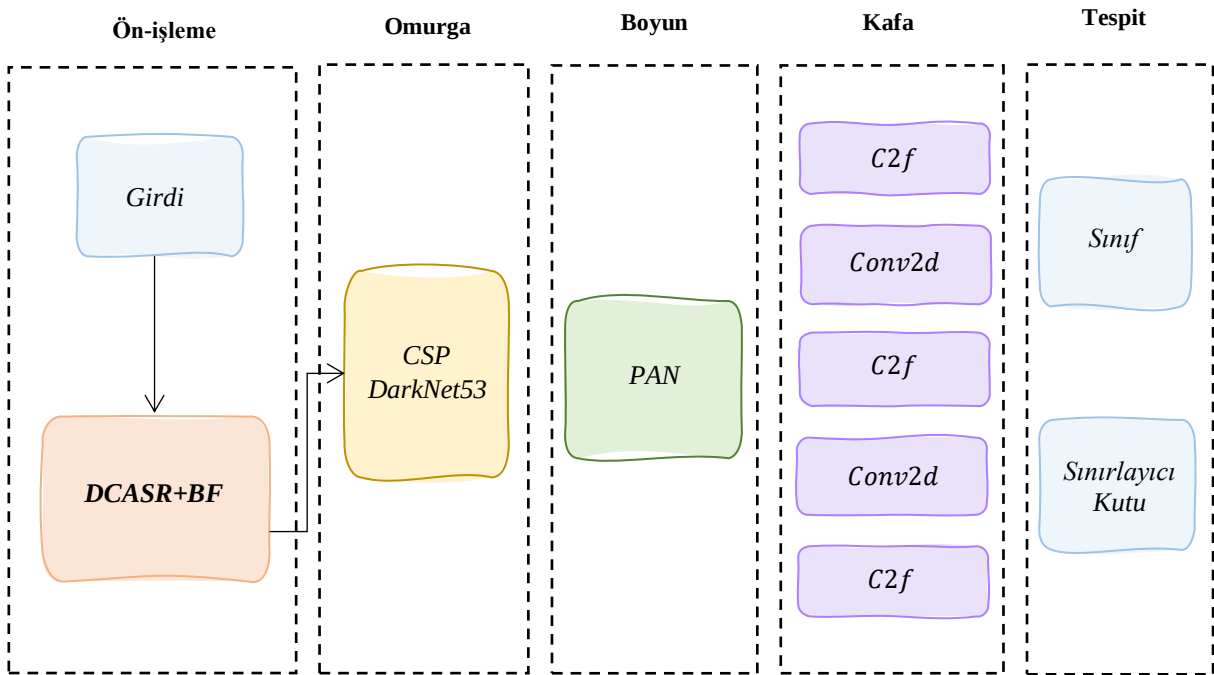
Bu tez kapsamında x_2 oranında çözünürlük arttırmak için (5, 225, 100), x_3 oranında çözünürlük arttırmak için (7, 225, 100) ya da (9, 225, 100), x_4 oranında çözünürlük arttırmak için (11, 225, 100) ölçeğinde BF uygulanmıştır. Burada ilk eleman pikselin komşuluk çapını, ikinci eleman renk uzayındaki değeri ve üçüncüsü ise koordinat alanındaki değeri temsil etmektedir. Çözünürlük boyutuna göre ilk elemanın değişmesindeki temel sebep ise piksel sayısının farklılığından kaynaklanmaktadır. Büyüme oranı arttıkça piksellerin komşuluk çapıda artırılarak oransallık korunmuştur. Sonuç olarak DCASR modeli ile iyileştirilen görüntülerin gürültüsünü azaltmak ve keskinliğini arttırmak için BF filtresi uygulanmıştır. Bu işlemin sonucu Şekil 3.10’da gösterilmektedir.

DCASR			
DCASR+BF			
	x_2	x_3	x_4

Şekil 3.10: İyileştirilen görüntülerin gürültüleri BF ile giderilmesi.

3.4. ÖNERİLEN MODEL

Tez kapsamında silah ve tüfek nesnelerinin video görüntüleri üzerinde tespit edilmesi ve varsa bu nesneleri taşıyan kişilerin tespit edilmesi hedeflenmektedir. Başka bir deyiş ile görüntüde yer alan silah, tüfek ve insan varlıklarının tespit edilerek etiketlenmesi amaçlanmaktadır. Bu doğrultuda belirlenen nesnelerin tespit edilebilmesini sağlayan hibrit bir nesne tespit modeli önerilmiştir. Önerilen model Şekil 3.11’de gösterilmektedir. Bu hibrit model temelde nesne tespiti ve iyileştirme olmak üzere iki yapıdan oluşmaktadır. İlk yapı tespit edilecek nesnelerin DMGDATA veri seti kullanılarak YOLO modelinin eğitiminden sorumludur. Eğitilen model kendisine gelen yeni nesneleri tespit etmeyi amaçlar. YOLO modellerinin birçok versiyonu olduğu için seçilen versiyonları DMGDATA veri seti ile eğitilmiş ve elde edilen performans sonuçları değerlendirilerek en uygun YOLO versiyonu seçilmiştir. Bu modelin seçimine karar verilirken elde edilen eğitim performans sonuçları bölüm 4.1’de yer almaktadır. İkinci yapı ise modelin performansını ve nesne tespit başarısını arttırmak için önerilen yeni bir SR algoritmasıdır. Önerilen SR algoritması DCASR modeli ile tespit için gönderilen yeni görüntülerin kalitesi iyileştirilip modele gönderir. Görüntü iyileştirme adımının ardında BF kullanılarak görüntü içerisindeki gürültüler azaltılır. Bu sayede özellikle düşük çözünürlüklü görüntülerde modelin nesne tespit performansının artması hedeflenmektedir.



Şekil 3.11: Önerilen modelin mimarîsi.

3.5. DEĞERLENDİRME METRİKLERİ

Tez kapsamında önerilen modelin nesnel bir şekilde değerlendirilebilmesi için genel kabul görmüş değerlendirme metrikleri kullanılmıştır. Bu bölümde nesne tespitin değerlendirilmesi ve görüntü kalitesindeki değişimin değerlendirilmesini sağlayan metrikler iki başlık altında sunulmuştur.

3.5.1. Nesne Tespitinin Değerlendirilmesi

Modelin performansını ve sınıflandırma başarısını ölçmek için modelin ürettiği tahmin sonuçlarına ihtiyaç duyulmaktadır. Bu tahmin sonuçlarının anlaşılabilir şekilde sunulduğu özet tabloya karmaşıklık matrisi denmektedir. Şekil 3.12'de gösterildiği gibi, bu tablo gerçek ve tahmin edilen verilere dayalı olarak dört farklı kategoride sınıflandırılmaktadır.

- Tahmin edilen doğru ve gerçekte de doğru (True Positive – TP)
- Tahmin edilen yanlış ve gerçekte de yanlış (True Negative – TN)
- Tahmin edilen doğru fakat gerçekte yanlış (False Positive – FP)
- Tahmin edilen yanlış fakat gerçekte doğru (False Negative – FN)

		GERÇEK	
		Pozitif	Negatif
TAHİMİ	Pozitif	TP	FP
	Negatif	FN	TN

Şekil 3.12: Karmaşıklık matrisi.

Precision: Kelime anlamı “*Hassasiyet*” olan bu metrik modelin veri kümesindeki pozitif sınıf etiketlerini tahmin etmedeki başarısını ölçmektedir. Bu metrik Denklem 3.3’te gösterildiği gibi öngörülen pozitif vakaların toplam öngörülen pozitif vakalara oranı şeklinde temsil edilmektedir. Doğru sınıflandırılan verilerin oranını göstermek için kullanılmaktadır.

$$Precision = \frac{TP}{TP+FP} \quad (3.3)$$

Sensitivity-Recall: Kelime anlamı “*Duyarlılık*” olan bu metrik modelin doğru tahmin edilen pozitif sınıf etiketlerini toplam pozitif sınıf etiketlerine oranlayarak pozitif tahmin değerini ölçmektedir. Bu metrik Denklem 3.4’teki gibi temsil edilmektedir. Yalnızca doğru tahmin edilip gerçekte de doğru olan verilerin oranını göstermek için kullanılmaktadır.

$$Recall = \frac{TP}{TP+FN} \quad (3.4)$$

Mean Average Precision (mAP): Kelime anlamı ile “*Ortalama Hassasiyet Değerlerinin Ortalaması*” olan bu metrik tüm sınıflar için Ortalama Hassasiyet (Average Precision – AP) değerlerinin ortalamasını temsil eder. AP, hassasiyet ve duyarlılık eğrisinin altında kalan değeri ifade eder. Yüksek AP, bir nesne algılama modelinin, nesnenin yüksek tahmin doğruluğunu gösterdiği anlamına gelmektedir. Çalışmalarda kullanılan AP50, %50 IoU eşiklerinde APS’yi ifade eder. mAP, belirli bir sınıf için tüm AP’nin ortalamasını hesaplamaktadır. Bu çıktı Denklem 3.5’teki gibi temsil edilmektedir.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP \quad (3.5)$$

3.5.2. Görüntü Kalitesinin Değerlendirilmesi

Tez çalışması kapsamında SR algoritmaları kullanılarak görüntülerin iyileştirilmesi gerçekleştirilmiştir. Görüntü çözünürlüğünün iyileştirilmesindeki temel amaç ise hedeflenen nesnenin daha başarılı tespit edilmesini sağlamaktır. Bu kapsamda yeni bir SR algoritması önerilmiştir. Önerilen SR algoritmasının benzerleri ile karşılaştırılması ve görüntü kalitelerindeki değişimleri gözlemlemek için en yaygın ölçüm tekniği olan Tepe Sinyalin Gürültüye Oranı (Peak Signal to Noise Ratio – PSNR) yöntemi kullanılmıştır.

PSNR, görüntünün ne kadar az gürültüye sahip olduğunu Desibel(dB) cinsinden ölçmektedir. Temelde görüntü üzerinde piksel bazında karşılaştırma yaparak bir sinyalin olası maksimum değeri ile referansın kalitesini düşüren gürültü gücü arasındaki oranı temsil etmektedir. Başka bir deyişle, orijinal ve yeniden oluşturulmuş iki görüntü arasındaki benzerliği ölçmektedir. Bu değer yüksek olması görüntü kalitesinin de yüksek olduğu anlamına gelmektedir. PSNR Denklem 3.6’da temsil edilmektedir. Bu denklemde yer alan Ortalama Kare Hata (Mean Square Error – MSE) değeri yüksek çözünürlüklü görüntü ile süper çözünürlüklü görüntü arasındaki “*Ortalama Kare Hatasını*” temsil etmektedir. Denklem 3.7’de ise MSE metriği temsil edilmektedir. Burada gösterildiği üzere M ve N satır ve sütun girdilerini, f ve g değerleri sırasıyla referans ve test görüntülerini, x ve y ise sırasıyla referans ve test

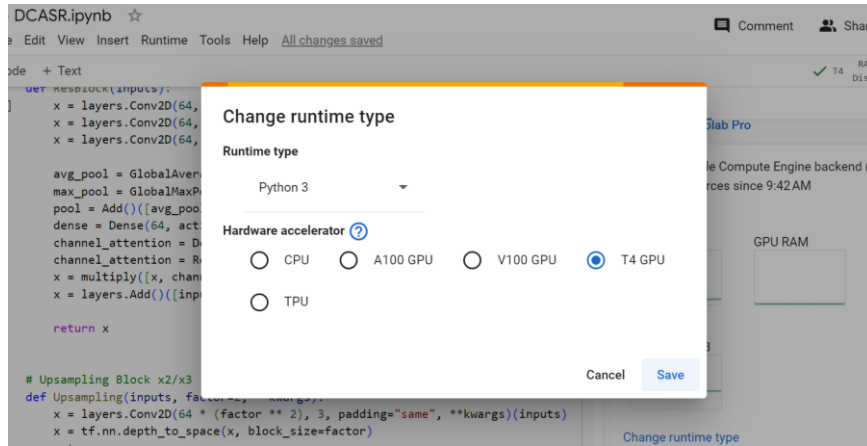
görüntülerinin aynı piksel numaralarındaki değerlerini temsil etmektedir (Kahveci, 2020; Şenalp & Ceylan, 2022).

$$PSNR = 10 \log \frac{(\text{Max Pixel Value})^2}{MSE} \quad (3.6)$$

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{i=0}^{N-1} (f(x_i, y_i) - g(x_i, y_i))^2 \quad (3.7)$$

3.6. UYGULAMA PLATFORMU ve KÜTÜPHANELERİ

Tez kapsamında ilk olarak Monster Tulpar v 20.3 Notebook cihazı kullanılmıştır. Fakat görüntü iyileştirme adımında GPU ve Rastgele Erişimli Belleğin (Random Access Memory – RAM) yetersiz kalmasından dolayı Google Colab Pro platformuna geçiş gerçekleştirilmiştir. Google Colab’ın 5 farklı kullanım seçeneği mevcuttur. Bunlardan ilki ücretsiz ve herkes tarafından kullanıma açıktır. Fakat donanım desteğinin düşük olmasından dolayı bu tez için yeterli ortamı sağlamamıştır. Diğer 4 seçenek ise belirli bir ücret karşılığında sunduğu farklı donanım destek türlerini ve ek hizmetleri içermektedir. Tez kapsamında Colab Pro versiyonu seçilerek 100 işlem birimi boyunca 166,77 GB disk kapasitesi, yüksek performanslı T4 GPU ve 50,99 GB RAM’e sahip çalışma ortamı sağlanmıştır. Donanım için farklı çalışma zamanı seçenekleri Şekil 3.13’te gösterildiği gibi yer almaktadır. Burada tez için uygun olan T4 GPU modeli seçilmiştir.



Şekil 3.13: Google Colab Pro çalışma zamanı tipleri.

Modelin geliştirilmesi sırasında veri ön işleme, veri etiketleme, veri görselleştirme, ML & DL modellerini tasarlama, eğitme ve sınıflandırma işlemlerinin kolaylıkla gerçekleştirildiği Python dili kullanılmıştır. Python mevcut kütüphaneleri ve kullanım kolaylığı

sayesinde yapay zeka dünyasında en çok tercih edilen dillerden biridir. Tez kapsamında kullanılan kütüphane ve çerçeve(ler) şu şekildedir:

- **Matplotlib:** Verileri iki ya da üç boyutlu grafiklerde göstermeye yardımcı olan temel Python kütüphanesidir. Çizgi, saçılım, sütun, pasta gibi çeşitli grafik türlerini desteklemenin yanı sıra grafiklerin renkleri, çizgi stilleri, işaretleyicileri, etiketleri ve açıklamaları özelleştirilebilir. Genellikle NumPy ve Pandas gibi kütüphanelerle birlikte kullanılır (Hunter vd., 2024). Şekil 3.14'te bu kütüphanenin örnek bir kullanımı mevcuttur.

```
import matplotlib.pyplot as plt

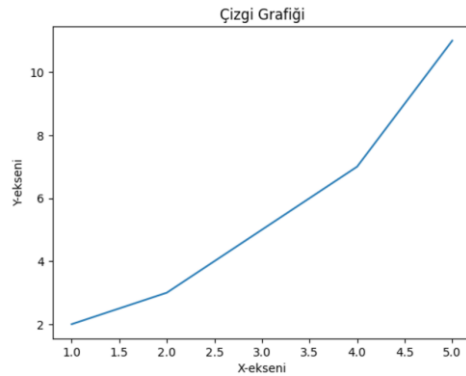
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

plt.figure()

plt.plot(x, y)

plt.xlabel('X-ekseni')
plt.ylabel('Y-ekseni')
plt.title('Çizgi Grafiği')

plt.show()
```



Şekil 3.14: Matplotlib örneği.

- **NumPy (Numerical Python):** Mantıksal ve matematiksel işlemleri gerçekleştirmek için kullanılan temel Python kütüphanesidir. Temel özellikleri aşağıdaki gibi ifade edilebilir:
 1. Çok boyutlu diziler ve matrisler üzerinde yüksek performanslı matematiksel işlemler yapmak için kullanılmaktadır.
 2. NumPy vektörize edilmiş matematiksel işlemler yapmak için optimize edilmiştir. Bu sayede, Python döngülerine göre çok daha hızlı çalışmaktadır.
 3. Farklı boyutlardan oluşan diziler arasında işlem yapmayı kolaylaştıran broadcasting özelliğini destekler.
 4. Doğrusal cebir, fourier dönüşümü ve matrisler için bir dizi fonksiyon sağlamaktadır.

Kısaca NumPy, bilimsel hesaplamalar, veri analizi, makine öğrenimi, yapay zeka gibi pek çok alanda yaygın olarak kullanılan bir kütüphanedir (<https://numpy.org/doc/stable/>, Erişim tarihi: 4 Mart 2024). Şekil 3.15'te bu kütüphanenin Matplotlib ile örnek bir kullanımı mevcuttur.

```
import matplotlib.pyplot as plt
import numpy as np

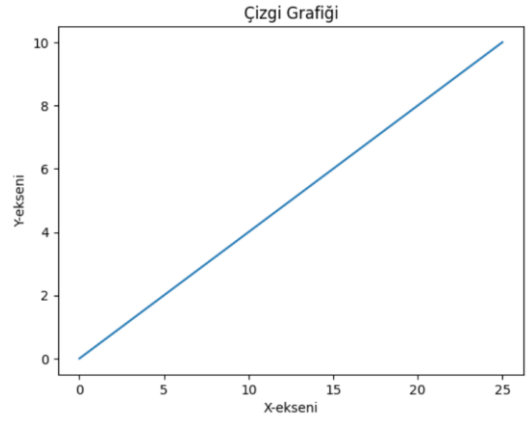
x = np.array([0, 25])
y = np.array([0, 10])

plt.figure()

plt.plot(x, y)

plt.xlabel('X-ekseni')
plt.ylabel('Y-ekseni')
plt.title('Çizgi Grafiği')

plt.show()
```



Şekil 3.15: Matplotlib ve NumPy örneği.

- **OpenCV:** Görüntüleri, videoları işlemek ve analiz etmek için kullanılan açık kaynaklı bir kütüphanedir. Görüntü işleme, nesne tespiti, nesne izleme, stereo görüntüleme ve 3B modelleme gibi birçok bilgisayarla görü görevini gerçekleştirmek için kullanılabilir. Ayrıca bu kütüphane Python, C++, Java ve MATLAB gibi birçok programlama dilinde kullanılabilir.
- **TensorFlow:** Google tarafından ML & DL modelleri geliştirmek için hazırlanmış açık kaynaklı bir matematik kütüphanesidir. Yapay sinir ağı modelini oluşturmak, eğitmek ve dağıtmak için kullanılır (<https://www.tensorflow.org/tutorials>, Erişim tarihi: 4 Mart 2024). Temel özellikleri aşağıdaki gibi ifade edilebilir:
 1. Çeşitli cihaz ve platformlarda çalıştırılabilen esnek bir mimariye sahiptir.
 2. Farklı karmaşıklık düzeylerindeki model mimarilerini oluşturmak ve uygulamak için geniş bir API seti sunmaktadır.
 3. Kullanıcılar, özel model mimarileri oluşturabilir veya TensorFlow Hub gibi kaynaklardan önceden eğitilmiş modelleri kullanabilir.
 4. Büyük veri kümeleri üzerinde eğitim yapmak için optimizasyonlar sağlamaktadır.
 5. Modelin dağıtımını ve hizmet olarak sunumunu kolaylaştıran araçlar sunmaktadır.
 6. Model eğitimini izlemek ve analiz etmek için TensorBoard adında bir görselleştirme araç desteği verir.
- **Keras:** Veri işleme, görselleştirme gibi işlemleri gerçekleştirmek ve sinir ağı modelleri kurmak için kullanılan Python dilinde yazılmış açık kaynaklı bir yapay sinir

ağı kütüphanesidir (<https://keras.io/about/>, Erişim tarihi: 4 Mart 2024). Temel özellikleri aşağıdaki gibi ifade edilebilir:

1. Keras, katmanları, kayıp fonksiyonlarını, optimizasyon algoritmalarını ve metrikleri kolayca birleştirmeye yardımcı olan modüler bir yapıya sahiptir.
2. Çeşitli optimizasyon algoritmalarını (örneğin, SGD, Adam, RMSprop vb.) ve farklı kayıp fonksiyonlarını (örneğin; MSE, MAE vb.) dahili olarak sunmaktadır.
3. TensorFlow, Theano, PyTorch ve Bilişsel Araç Seti (Cognitive Toolkit –CNTK) üzerinde çalışabilen bir API'dir.
4. Basit iş akışları sayesinde anlaşılır ve kullanımı kolaydır.

Keras, TensorFlow'un TensorFlow 2.x sürümünden itibaren entegre edilmiştir. Bu sebeple, kolay kullanımının yanısıra TensorFlow'un gücünde dahil edilerek daha başarılı modeller geliştirilebilir.

4. BULGULAR

Tez çalışması kapsamında tehlikeli nesnelerin tespiti için yeni hibrit bir model önerilmiştir. Önerilen hibrit model 2 temel algoritmanın birleşiminden meydana gelmektedir:

1. YOLO algoritması
2. SR algoritması

YOLO algoritmasına ait model seçimini gerçekleştirmek için mAP50, mAP50-95, parametre sayısı, katman sayısı ve eğitim süresi parametreleri kullanılarak değerlendirme işlemi gerçekleştirilmiştir. SR algoritmasına ait model seçimi ise OpenCV tarafından sunulan *cv2.dnn_superres* kütüphanesindeki EDSR, FSRCNN ESPCNN ve LapSRN modelleri ve tez kapsamında tasarlanan DCASR modelinin PSNR ve görüntü iyileştirme süresine göre karşılaştırılması ile gerçekleştirilmiştir. Ardından oluşturulan hibrit YOLOv8m+DCASR+BF modelinin performansı, mAP50, mAP50-95, hassasiyet, duyarlılık, test süresi, bellek, CPU ve GPU kullanım bilgisine göre değerlendirip karşılaştırılmıştır.

4.1. YOLO MODELLERİNİN PERFORMANS DEĞERLENDİRMESİ

Nesne tespit işlemi için oluşturulan hibrit modelin yapı taşlarından ilki YOLO algoritmasıdır. YOLO algoritması hedef nesnenin tespit edilmesi için literatürde sıklıkla kullanılan en başarılı algoritmalarından biridir. Bu sebepten tez kapsamında YOLO algoritması ile nesne tespit işlemi gerçekleştirmeye karar verilmiştir. Fakat YOLO algoritmasına ait birçok model ve bu modellere ait çeşitli versiyonlar bulunmaktadır. Tez kapsamında hibrit modelde kullanılacak nesne tespit algoritmasının belirlenmesi için hazırlanan DMGDATA veri seti ile YOLO algoritmasına ait seçilen modeller eğitilmiştir. Eğitim işlemi Google Colab Pro platformunda gerçekleşmiş olup her bir modele 50 epok, 16 topluluk boyutu verilmiştir. Seçilen her bir YOLO modelinin çeşitli versiyonlarında gerçekleştirilen eğitim işlemi Tablo 4.1’de gösterildiği gibi mAP-50, mAP50-95 metriğine ve eğitim süresine göre değerlendirilmiştir. Genel olarak sonuçlar mAP50 metriğine göre incelendiğinde her bir modelin son versiyonunun en başarılı sonucu ürettiği gözlemlenmiştir. Bununla birlikte başarılı sonuçlar üreten son versiyonlar en fazla parametre ve katman sayısına sahip olduğu için eğitim süresinin daha uzun

sürdüğü farkedilmiştir. Sonuçlara istinaden hibrit model için seçilecek YOLO versiyonu oluşturulan model dinamiğine bağlı olarak 2 farklı şekilde seçilebilir.

1. Tasarlanan model yapısı düzenli olarak yeni veriler ile eğitilecek ise; eğitim süresi görece daha az olan düşük parametre ve katman sayısına sahip modeller tercih edilmelidir. Bu durumda zamandan, maliyetten tasarruf sağlayan ve nispeten diğerlerine göre daha başarılı olan s ya da m versiyonları tercih edilmelidir. YOLO modellerinin m serisine bakıldığında YOLOv5m modeli mAP50 metriğinde 0,879 oranında ve YOLOv6m modeli 0,849 oranında değer üretirken YOLOv8m versiyonu 0,884 oranında daha başarılı bir sonuç üretmiştir. Ayrıca YOLOv8m modelinin eğitim süresi YOLOv6m'e göre daha kısa sürmüştür. Bu durumda YOLOv8m versiyonunu seçmenin hız ve performans açısından daha mantıklı olacağı düşünülmektedir.
2. Eğer tasarlanan model yapısı düzenli olarak eğitime ihtiyacı duymuyorsa, sadece bir kez eğitilip daha sonra modele entegre ediliyorsa bu durumda en başarılı sonucu üreten modeli seçmek daha doğru olacaktır. Çünkü eğitim maliyeti sadece bir kez verilecektir. Bu durumda YOLOv8x versiyonunu seçmenin daha mantıklı olacağı düşünülmektedir.

Tablo 4.1: YOLO modellerin eğitim sonuçları.

Model	mAP50	mAP50-95	Parametre sayısı	Katman	Eğitim süresi (saat)
YOLOv5n	0,865	0,580	1.763.224	157	1,069
YOLOv5s	0,876	0,609	7.018.216	157	1,435
YOLOv5m	0,879	0,628	20.861.016	212	2,399
YOLOv5l	0,879	0,632	46.119.048	267	3,727
YOLOv5x	0,882	0,637	86.186.872	322	6,066
YOLOv6n	0,846	0,581	4.630.000	-	1,662
YOLOv6s	0,848	0,582	18.500.000	83	2,513
YOLOv6m	0,849	0,585	34.800.000	167	4,055
YOLOv6l	0,866	0,610	59.600.000	255	4,825
YOLOv7	0,880	0,616	36.492.560	314	5,295
YOLOv7-x	0,886	0,662	70.795.920	362	7,383
YOLOv8n	0,872	0,628	3.011.417	225	1,691
YOLOv8s	0,881	0,638	1.1126.745	225	1,843
YOLOv8m	0,884	0,647	25.841.497	295	3,104
YOLOv8l	0,886	0,650	43.608.921	365	4,552
YOLOv8x	0,889	0,651	68.126.457	365	7,261

Yukarıda bahsedilen iki durumdan birincisi genellikle gerçek dünya modellerine ait bir durumdur. Eğer geliştirilen model bir ürün haline getirilirse konsept kaymalarını engellemek, aşırı öğrenmeyi azaltmak ve modelin güncelliğinin korunması için tasarlanan model düzenli olarak yeni veriler ile eğitimine devam ettirilmelidir.

Tez kapsamında gerçek dünyada yer alan veriler test edilerek modelin yapısı ve performansı değerlendirileceği için YOLOv8m modeli kullanılmaya karar verilmiştir. Modelin eğitim süresi ve performans başarısı gerçek dünya modeli için uygundur. Benzer şekilde test verileri ile performansı değerlendirip gerçek dünya modellerine entegre edilebilip edilemeyeceği hakkında sonuca varılmıştır.

4.2. DCASR MODELİNİN PERFORMANS DEĞERLENDİRMESİ

Tasarlanan modelin bir diğer yapı taşı ise SR algoritmasıdır. SR algoritmaları görüntü kalitelerini iyileştirmek için kullanılmaktadır. Hibrit model kapsamında nesne tespit modeli ile birleştirilmesinin temel sebebi; görüntü kalitesinin iyileştirilerek modelin nesne tespit başarısını arttırmaktır. Özellikle düşük çözünürlükte kaybolan nesnelerin piksel sayısının artırılması ile hedef nesnenin belirgenleşmesi sağlanarak tespit başarısının artırılması amaçlanmaktadır. Bu işlem için OpenCV tarafından sunulan *cv2.dnn_superres* kütüphanesindeki EDSR, FSRCNN ESPCNN ve LapSRN modelleri ve tez kapsamında geliştirilen DCASR modeli kullanılmıştır.

Seçilen SR algoritmalarının performansını değerlendirebilmek için PSNR metriği ve tek bir görüntüyü iyileştirmek için modelin harcadığı süre saniye cinsinden hesaplanmıştır. Bu hesaplama işlemleri görüntü iyileştirme modellerini karşılaştırmak için sıklıkla kullanılan Set5 (Kim vd., 2016) veri setindeki “kelebek” görüntüsü üzerinde gerçekleştirilmiştir. Belirlenen görüntünün farklı SR algoritmaları ile $x2$, $x3$ ve $x4$ çözünürlük oranında iyileştirilmesine ilişkin sonuçlar Tablo 4.2’de yer almaktadır.

Tablo 4.2’de yer alan sonuçlara bakıldığında:

- $x3$ ve $x4$ oranında DCASR modelinin en başarılı PSNR değerini ürettiği görülmektedir.
- $x2$ oranında ise EDSR modelinin 0,091 gibi küçük bir farkla DCASR modelinden daha başarılı PSNR değeri ürettiği tespit edilmiştir. Fakat EDSR modelinin tek bir görüntüyü iyileştirilmek için harcadığı süreye bakıldığında bu sürenin oldukça uzun olduğu görülmektedir. EDSR modelinin $x2$ oranında ürettiği PSNR sonucu önerilen

DCASR'ye göre yüksek olmasına rağmen üretim süresinin ~44 kat daha uzun sürdüğü hesaplanmıştır.

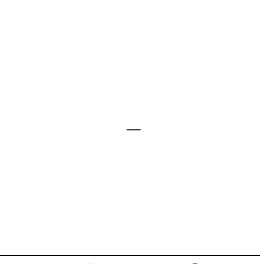
- Ayrıca diğer ölçeklere bakıldığında EDSR modelinin DCASR modeline oldukça yakın PSNR değeri ürettiği fakat tek bir görüntüyü iyileştirilmek için harcadığı sürenin oldukça yüksek olduğu tespit edilmiştir. Başka bir deyiş ile EDSR modelinin görüntü iyileştirme süresi diğer modellerden oldukça yavaş görülmektedir. Dolayısı ile EDSR modelinin özellikle gerçek zamanlı modellerde kullanılmasının uygun olmadığı düşünülmektedir. Bu sebepten hem PSNR metriği hem de iyileştirme süresi açısından DCASR algoritmasının karşılaştırılan diğer SR algoritmalarından daha başarılı olduğu söylenebilir.

Tablo 4.2: SR modellerinin performans değerlendirmesi.

SR Algoritmaları	Çözünürlük Oranı	PSNR	Tek görüntü için üretim süresi (s)
FSRCNN	x2	30.550	0,020
	x3	30.231	0,018
	x4	29.813	0,027
ESPCNN	x2	30.540	0,020
	x3	30.221	0,012
	x4	29.836	0,013
EDSR	x2	30.635	3,087
	x3	30.357	1,863
	x4	30.061	1,287
LAPSRN	x2	30.525	0,045
	x4	29.918	0,064
DCASR (Önerilen SR modeli)	x2	30.540	0,069
	x3	30.544	0,065
	x4	30.469	0,064

Elde edilen sonuçlara istinaden DCASR modeli ile iyileştirilen görüntülerin PSNR metriklerinin ve sürelerinin daha başarılı bir performans sağladığı gözlemlenmiştir. Bu sonuçların çıktıları her bir çözünürlük büyütme oranı ve kullanılan algoritmalar özelinde sırasıyla Şekil 4.1'de gösterilmiştir. Sonuçlar gözle görülür bir farkla iyileşme olduğunu ve DCASR modelinin ürettiği sonucun diğerlerine göre daha başarılı olduğunu göstermektedir.

Gözlemlenen bu sonuçlara istinaden tez kapsamında tasarlanan hibrit modelin görüntü iyileştirme aşamasında önerilen DCASR modeli kullanılmaya karar verilmiştir.

		
Original $x2$	Original $x3$	Original $x4$
		
FSRCNN $x2$	FSRCNN $x3$	FSRCNN $x4$
		
ESPCNN $x2$	ESPCNN $x3$	ESPCNN $x4$
		
EDSR $x2$	EDSR $x3$	EDSR $x4$
		
LAPSRN $x2$	LAPSRN $x3$	LAPSRN $x4$
		
DCASR $x2$	DCASR $x3$	DCASR $x4$

Şekil 4.1: Farklı çözünürlük oranlarında SR ile iyileştirilen “kelebek” görüntüsü.

4.3. YOLO+SR MODELLERİNİN PERFORMANS DEĞERLENDİRMESİ

Test sonuçları neticesinde hibrit modelin nesne tespit aşaması için YOLOv8m, görüntü iyileştirme aşaması için DCASR algoritmasının kullanımına karar verilmiştir. Eğitilen YOLOv8m modeli, tasarlanan DCASR modeli ile birleştirilerek yeni hibrit bir model oluşturulmuştur. Elde edilen hibrit modele gönderilen görüntülerin gürültüsünü azaltmak için ayrıca bir BF filtresi eklenmiştir. Önerilen modelin performansını değerlendirmek için iki farklı test seti ve diğer SR algoritmaları kullanılmıştır. Değerlendirme işlemi sırasında YOLOv8m+SR+BF şablonu kullanılmıştır. Modelin SR kısmına FSRCNN, ESPCNN, EDSR, LapSRN ve DCASR algoritmaları sırasıyla yerleştirilmiş ve hassasiyet, duyarlılık, mAP50 ve mAP50-95 metriklerine göre üretilen sonuçlar değerlendirilmiştir. Bunun yanı sıra hem SR algoritmalarının görüntü iyileştirme sırasında harcadığı donanım ve zaman maliyeti hem de hibrit modelin nesne tespiti sırasında harcadığı donanım ve zaman maliyeti hesaplanmıştır. Donanım maliyeti için harcanan bellek miktarı, disk kapasitesi, CPU ve GPU miktarları ölçülürken, zaman maliyeti için harcanan süre saniye cinsinden ölçülmüştür. Test_1 veri seti için görüntü iyileştirme sırasında harcanan donanım ve zaman Tablo 4.3'te, hibrit modelin harcadığı donanım ve zaman Tablo 4.4'te, hibrit modelin performans sonuçları ise Tablo 4.5'te gösterilmiştir. Tablo 4.3'teki ve Tablo 4.4'teki Test_1 veri setinin harcadığı donanım ve zaman bilgilerine ait sonuçlara bakıldığında:

- EDSR modeli büyük boyutlu görüntüleri iyileştirmek için uzun süre çalışmasına rağmen tüm görüntüleri iyileştiremediği için sonuç üretememiştir. Mevcut donanımın EDSR ile Test_1 veri seti üzerindeki görüntü iyileştirme işlemi başarısız olmuştur.
- Belleğin görüntü iyileştirme ve nesne tespiti sırasında en fazla DCASR algoritması tarafından kullanıldığı görülmüştür.
- Tüm modellerde hem görüntü iyileştirme sırasında hem de nesne tespiti sırasında kullanılan disk kapasitelerinin benzer olduğu görülmüştür.
- DCASR'nin hem görüntü iyileştirme sırasında hem de nesne tespiti sırasında en az CPU kullanan algoritma olduğu ve çözünürlük oranı arttıkça arttığı tespit edilmiştir.
- GPU'nun görüntü iyileştirme sırasında yalnızca DCASR modeli tarafından kullanıldığı tespit edilirken, nesne tespiti sırasında diğer modeller tarafından da kullanıldığı gözlemlenmiştir.
- Ayrıca tez kapsamından kullanılan tüm SR algoritmalarının DCASR algoritmasına göre daha fazla zaman harcadığı görülmüştür.

Tablo 4.3: SR modellerinin Test_1 veri setini iyileştirirken harcadığı donanım ve zaman.

Model	Çözünürlük Oranı	Görüntü iyileştirme için				
		Bellek Kullanımı (GB)	Disk Kullanımı (GB)	CPU Kullanımı (%)	GPU Kullanımı (%)	Test Süresi (s)
FSRCNN	x2	1,62	26,30	16,30	0	80,47
	x3	1,54	26,32	22,9	0	110,71
	x4	1,65	26,36	63,6	0	242,48
ESPCNN	x2	1,31	26,23	62,8	0	67,57
	x3	1,29	26,26	35,8	0	96,55
	x4	1,25	26,30	85,7	0	245,91
EDSR	x2	-	-	-	-	-
	x3	-	-	-	-	-
	x4	-	-	-	-	-
LAPSRN	x2	1,19	26,36	55,2	0	437,93
	x4	1,10	26,34	86,1	0	2.503,36
DCASR (Önerilen SR modeli)	x2	3,47	27,34	12,70	13,00	78,64
	x3	3,51	27,36	7,60	10,00	75,68
	x4	3,53	27,39	12,90	11,00	174,89

Tablo 4.4: Önerilen modelin Test_1 veri setinde nesne tespiti için harcadığı donanım ve zaman.

Model	Çözünürlük Oranı	Nesne tespiti için					
		Bellek Kullanımı (GB)	Disk Kullanımı (GB)	CPU Kullanımı (%)	GPU Kullanımı (%)	Test Süresi (s)	Toplam harcanan süre (s)
YOLOv8m	Orjinal	1,85	29,06	42,7	15,0	93,65	71,05
YOLOv8m+ FSRCNN+BF	x2	1,69	29,36	55,20	33,0	54,59	135,06
	x3	1,76	29,41	20,5	0	45,67	156,38
	x4	1,70	29,50	17,3	9,0	49,30	291,78
YOLOv8m+ ESPCNN+BF	x2	1,23	28,95	30,1	5,0	32,68	100,25
	x3	1,56	29,02	80,5	28,0	41,51	138,06
	x4	1,58	29,04	69,6	0	36,01	281,92
YOLOv8m + EDSR+BF	x2	-	-	-	-	-	-
	x3	-	-	-	-	-	-
	x4	-	-	-	-	-	-
YOLOv8m + LapSRN+BF	x2	1,83	29,09	51,0	0	32,10	470,03
	x4	1,83	29,11	63,2	0	50,62	2553,98
YOLOv8m + DCSAR+BF (Önerilen hibrit model)	x2	3,49	27,34	24,40	0	27,15	105,79
	x3	3,50	27,37	14,10	24,0	24,62	100,30
	x4	3,50	27,39	28,60	0	26,94	201,83

Test_1 veri seti özelinde YOLOv8m ve YOLOv8m+SR+BF hibrit modeline ait sonuçlar Tablo 4.5'te gösterilmektedir. Sonuçlara bakıldığında DMGDATA veri setinden ayrılmış Test_1 veri setindeki görüntüler ile yapılan iyileştirmenin nesne tespit modelinin başarısını yeterince arttırmadığı hatta bazı SR algoritmaları kullanıldığında azaldığı gözlemlenmiştir. EDSR modeli ile oluşturulan hibrit modelin ise hiçbir sonuç üretmediği gözlemlenmiştir. Bu durumun büyük boyutlu görüntüleri işlemekte zorlanan donanımdan kaynaklandığı düşünülmektedir.

Kısaca, tez kapsamında kullanılan SR algoritmalarının Test_1 veri setindeki hedef nesnelerin tespit başarısını arttırmadığı, fakat önerilen DCASR modeli ile iyileştirilen görüntülerdeki nesne tespit başarısının %0,3 oranında dahi olsa arttırdığı gözlemlenmiştir.

Tablo 4.5: Önerilen nesne tespit modelinin Test_1 veri setindeki sonucu.

Model	Çözünürlük oranı	Hassasiyet	Duyarlılık	mAP50	mAP50-95
YOLOv8m	Orjinal	0,856	0,803	0,852	0,543
YOLOv8m + FSRCNN BF	x2	0,864	0,793	0,849	0,542
	x3	0,872	0,781	0,842	0,539
	x4	0,866	0,773	0,837	0,537
YOLOv8m + ESPCNN+BF	x2	0,862	0,795	0,845	0,540
	x3	0,865	0,789	0,843	0,540
	x4	0,865	0,744	0,813	0,522
YOLOv8m + EDSR+BF	x2	-	-	-	-
	x3	-	-	-	-
	x4	-	-	-	-
YOLOv8m+ LapSRN+BF	x2	0,866	0,789	0,844	0,539
	x4	0,868	0,775	0,839	0,537
YOLOv8m + DCSAR+BF (Önerilen hibrit model)	x2	0,864	0,800	0,854	0,544
	x3	0,862	0,797	0,854	0,544
	x4	0,871	0,795	0,855	0,544

Elde edilen Test_1 veri setinin örnek görüntü sonuçları ise Şekil 4.2'de gösterilmektedir. Şekilde gözlemlendiği üzere hibrit modelin orijinal YOLOv8m modelinin sonucuna oldukça yakın değerler üretildiği görülmüştür. Buradan yola çıkarak görüntünün belirli bir çözünürlük değeri üzerinde olmasının iyileştirme algoritmalarının nesne tespit başarısını yeterince arttırmadığı söylenebilir.

		
Original		
		
YOLOv8m+FSRCNN+BF x2	YOLOv8m+FSRCNN+BF x3	YOLOv8m+FSRCNN+BF x4
		
YOLOv8m+ESPCNN+BF x2	YOLOv8m+ESPCNN+BF x3	YOLOv8m+ESPCNN+BF x4
-		
-		
-		
YOLOv8m+EDSR+BF x2	YOLOv8m+EDSR+BF x3	YOLOv8m+EDSR+BF x4
	-	
YOLOv8m+LAPSRN+BF x2	YOLOv8m+LAPSRN+BF x3	YOLOv8m+LAPSRN+BF x4
		
YOLOv8m+DCASR+BF x2	YOLOv8m+DCASR+BF x3	YOLOv8m+DCASR+BF x4

Şekil 4.2: Test_1 veri setinin sonuçları.

Test_2 veri seti için görüntü iyileştirme sırasında harcanan donanım ve zaman Tablo 4.6'da hibrit modelin harcadığı donanım ve zaman Tablo 4.7'de, hibrit modelin performans sonuçları ise Tablo 4.8'de yer almaktadır.

Tablo 4.6'daki ve Tablo 4.7'deki Test_2 veri setinin harcadığı donanım ve zaman bilgilerine ait sonuçlara bakıldığında:

- Görüntü iyileştirme sırasında en fazla bellek EDSR tarafından kullanılmıştır.
- Tüm modellerde hem görüntü iyileştirme sırasında hem de nesne tespiti sırasında kullanılan disk kapasitelerinin benzer olduğu görülmüştür.
- CPU'nun özellikle görüntü iyileştirme sırasında EDSR ve LapSRN modeli tarafından daha çok kullanıldığı tespit edilmiştir.
- GPU'nun görüntü iyileştirme sırasında yalnızca DCASR modeli tarafından kullanıldığı görülürken, nesne tespiti sırasında diğer modeller tarafından da kullanıldığı tespit edilmiştir.
- FSRCNN ve ESPCNN algoritmaları görüntü iyileştirirken daha az zaman harcamıştır. Fakat nesne tespit işlemi sırasında harcadıkları süre DCASR algoritmasına göre daha fazla olduğu için ortalamada benzer süre harcadığı görülmüştür.

Tablo 4.6: SR modellerinin Test_2 veri setini iyileştirirken harcadığı donanım ve zaman.

Model	Çözünürlük Oranı	Görüntü iyileştirme için				
		Bellek Kullanımı (GB)	Disk Kullanımı (GB)	CPU Kullanımı (%)	GPU Kullanımı (%)	Test Süresi (s)
FSRCNN	x2	2,25	26,24	1,8	0	10,80
	x3	1,58	26,21	9,6	0	21,76
	x4	1,50	26,24	2,4	0	26,49
ESPCNN	x2	1,52	26,27	2,1	0	8,86
	x3	1,52	26,25	2,2	0	14,85
	x4	1,49	26,24	2,9	0	25,43
EDSR	x2	2,91	26,31	37,4	0	677,93
	x3	4,22	26,35	33,2	0	672,85
	x4	4,22	26,40	42,2	0	707,06
LAPSRN	x2	1,36	26,24	28,8	0	55,14
	x4	1,30	26,22	64,7	0	235,29
DCASR (Önerilen SR modeli)	x2	2,83	27,26	15,40	2,00	36,24
	x3	2,76	27,27	16,10	3,00	38,64
	x4	2,86	27,28	7,70	2,00	53,20

Tablo 4.7: Önerilen modelin Test_2 veri setinde nesne tespiti için harcadığı donanım ve zaman.

Model	Çözünürlük Oranı	Nesne tespiti için					
		Bellek Kullanımı (GB)	Disk Kullanımı (GB)	CPU Kullanımı (%)	GPU Kullanımı (%)	Test Süresi (s)	Toplam harcanan süre (s)
YOLOv8m	Orjinal	1,85	28,68	18,3	4,0	30,97	30,97
YOLOv8m + FSRCNN+BF	x2	1,69	29,11	18,7	23,0	65,01	75,81
	x3	1,61	28,93	48,3	25,0	74,37	96,13
	x4	1,61	29,02	53,1	0	39,16	65,65
YOLOv8m + ESPCNN+BF	x2	1,54	29,14	48,6	0	35,43	44,29
	x3	1,58	29,09	43,7	0	38,02	52,87
	x4	1,65	29,05	32,4	0	42,68	68,11
YOLOv8m + EDSR+BF	x2	1,58	29,17	45,7	0	24,03	701,96
	x3	1,60	29,21	17,5	0	36,47	709,32
	x4	1,61	29,23	15,4	0	37,78	744,84
YOLOv8m + LapSRN+BF	x2	1,68	29,29	71,0	25,0	30,04	85,18
	x4	1,66	29,26	13,2	0	40,69	275,98
YOLOv8m + DCASR+BF (Önerilen hibrit model)	x2	2,80	27,26	22,60	0	24,13	60,37
	x3	2,83	27,27	22,90	0	23,94	62,58
	x4	2,82	27,25	22,70	0	20,44	73,64

Test_2 veri seti özelinde YOLOv8m ve YOLOv8m+SR+BF hibrit modeline ait sonuçlar Tablo 4.8’de gösterilmektedir. Sonuçlara bakıldığında düşük çözünürlüklü görüntülerden oluşan Test_2 veri seti ile yapılan iyileştirmenin nesne tespit modelinin başarısını arttırdığı gözlemlenmiştir. Her çözünürlük oranı için en başarılı mAP50 değerini üreten sonuca bakıldığında:

- x2 ve x4 çözünürlük oranında DCASR algoritmasının nesne tespit başarısını yaklaşık %4 ila %6 oranında iyileştirdiği gözlemlenmiştir.
- x3 çözünürlük oranında ise EDSR algoritmasının nesne tespit başarısını yaklaşık %5 iyileştirdiği gözlemlenmiştir.

Fakat hem nesne tespit başarısı hem de harcanan donanım ve zaman açısından bakıldığında EDSR algoritmasının gerçek dünyadaki uygulamalarda kullanıma uygun olmadığı düşünülmektedir. Çünkü x3 oranında DCASR algoritması ile oluşturulan hibrit model en iyi ikinci mAP50 sonucunu üreterek daha az donanım ve zaman harcamıştır. Bu durumda; YOLOv8m+EDSR+BF ve YOLOv8m+DCASR+BF modeli arasında x3 çözünürlüğünde mAP50 metriğine göre %1 başarı farkı olmasına rağmen zaman açısından YOLOv8m+EDSR+BF modeli yaklaşık 11 kat daha yavaş çalışmaktadır. Buna göre

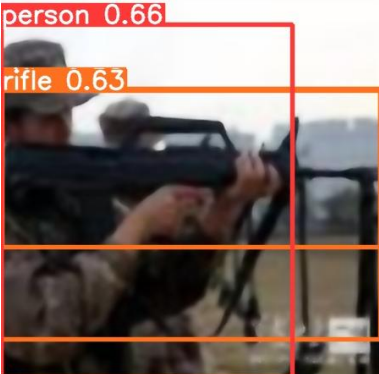
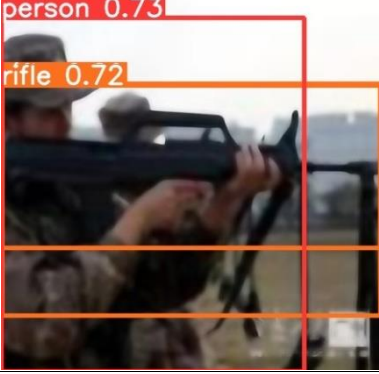
değerlendirme sırasında zaman parametreside göz önüne alındığında $x3$ oranında da YOLOv8m+DCASR+BF modelinin daha başarılı sonuç ürettiği ortaya çıkmaktadır.

Sonuç olarak, düşük çözünürlüklü görüntülerde YOLOv8m+DCASR+BF modeli kullanılarak nesne tespit başarısını %4 ila %6 oranında arttırılabileceği söylenebilir.

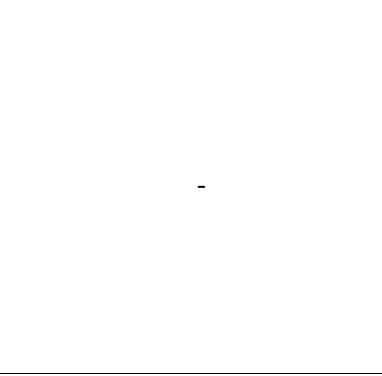
Tablo 4.8: Önerilen nesne tespit modelinin Test_2 veri setindeki sonucu.

Model	Çözünürlük oranı	Hassasiyet	Duyarlılık	mAP50	mAP50-95
YOLOv8m	Orjinal	0,734	0,626	0,703	0,408
YOLOv8m + FSRCNN+BF	$x2$	0,727	0,642	0,719	0,426
	$x3$	0,751	0,656	0,733	0,437
	$x4$	0,784	0,645	0,743	0,448
YOLOv8m + ESPCNN+BF	$x2$	0,759	0,618	0,716	0,421
	$x3$	0,747	0,655	0,732	0,437
	$x4$	0,718	0,631	0,706	0,419
YOLOv8m + EDSR+BF	$x2$	0,772	0,630	0,723	0,429
	$x3$	0,743	0,672	0,751	0,451
	$x4$	0,761	0,649	0,739	0,438
YOLOv8m+ LapSRN+BF	$x2$	0,726	0,633	0,712	0,419
	$x4$	0,735	0,640	0,716	0,423
YOLOv8m + DCSAR+BF (Önerilen hibrit model)	$x2$	0,730	0,679	0,744	0,440
	$x3$	0,758	0,668	0,741	0,442
	$x4$	0,796	0,648	0,763	0,459

Elde edilen Test_2 veri setinin örnek görüntü sonuçları Şekil 4.3'te gösterilmektedir. Şekilde görüldüğü üzere hibrit modelin orijinal YOLOv8m modeline göre daha başarılı nesne tespit işlemi gerçekleştirdiği ve düşük çözünürlüklü görüntüler üzerinde kullanılan tüm SR algoritmalarının nesnenin tespit başarısını arttırdığı gözlemlenmiştir. Özellikle YOLOv8m+DCASR+BF modelinin nesne tespitinde diğer tüm modellerden daha başarılı sonuç ürettiği tespit edilmiştir.

		
Original		
		
YOLOv8m+FSRCNN+BF x2	YOLOv8m+FSRCNN+BF x3	YOLOv8m+FSRCNN+BF x4
		
YOLOv8m+ESPCNN+BF x2	YOLOv8m+ESPCNN+BF x3	YOLOv8m+ESPCNN+BF x4
		
YOLOv8m+EDSR+BF x2	YOLOv8m+EDSR+BF x3	YOLOv8m+EDSR+BF x4

Şekil 4.3: Test_2 veri setinin sonuçları.

 person 0.62 rifle 0.71	 -	 person 0.69 rifle 0.67
YOLOv8m+ LAPSRN+BF x2	YOLOv8m+ LAPSRN+BF x3	YOLOv8m+ LAPSRN+BF x4
 person 0.75 rifle 0.77	 person 0.78 rifle 0.76	 person 0.77 rifle 0.77
YOLOv8m+ DCASR+BF x2	YOLOv8m+ DCASR+BF x3	YOLOv8m+ DCASR+BF x4

Şekil 4.3 (devam): Test_2 veri setinin sonuçları.

5. TARTIŞMA

Tez kapsamında tasarlanan DCASR algoritmasının görüntüleri iyileştirme başarısı yapılan karşılaştırmalar sonucunda FSRCNN, ESPCNN, EDSR ve LapSRN algoritmasından daha yüksek olduğu görülmüştür (Dong vd., 2016; Shi vd., 2016; Lim vd., 2017; Lai vd, 2017). Bu sebepten DCASR algoritması tehlikeli nesnelerin tespit edilmesini sağlayan hibrit modelin yapı taşlarından biri olarak seçilmiştir. Bu algoritma sayesinde görüntülerin tespit algoritmasına gönderilmeden önce iyileştirilmesi sağlanarak nesne tespit başarısı artırılır ve net olmayan nesneler iyileştirilerek tespit edilmesi sağlanır.

Tez kapsamında hazırlanan DMGDATA ile eğitilen YOLO versiyonları karşılaştırılarak hibrit model için uygun nesne tespit algoritması belirlenmiştir. Karşılaştırma işleminin sonucunda gerçek zamanlı olarak kullanılacak hıza ve başarıya sahip olan YOLOv8m versiyonu seçilmiştir.

Hibrit modelin tasarlanarak deneysel çalışmaların gerçekleştirilmesi sonucunda görüntü kalitesi düşüklüğünden kaynaklanan başarısız nesne tespit işlemlerini algoritmik yöntemler ile çözerek hem depolama hem de donanım maliyetinden tasarruf edileceği görülmüştür. Başka bir deyiş ile kalitesiz görüntü üreten bir kaynaktan gelen verilerin SR algoritmaları ile iyileştirilerek işlenmesi nesne tespitindeki başarıyı arttıracakı söylenebilir. Fakat yüksek çözünürlüklü görüntüler üzerinde yapılan testlerde görüntü iyileştirmeden sonra nesne tespit başarısının artmadığı hatta bazı modellerde sonucu kötüleştirdiği gözlemlenmiştir. Bu durumda düşük çözünürlüklü görüntüler üzerinde yapılan iyileştirmelerin nesne tespit başarısını daha çok etkilediği söylenebilir.

Bu tezin literatüre katkıları şu şekilde özetlenebilir:

- Tehlikeli nesneler özelinde nesne tespit çalışmalarında kullanılacak DMGDATA adında etiketlenmiş yeni bir veri seti hazırlanmıştır.
- Kanal dikkat mekanizması kullanılarak görüntüleri x_2 , x_3 ve x_4 oranında çözünürlüklerini arttıran ve görüntü kalitesini iyileştiren DCASR adında yeni bir SR algoritması önerilmiştir. Önerilen algoritma OpenCV tarafından sunulan başka SR algoritmaları ile karşılaştırılmıştır ve daha başarılı olduğu ispat edilmiştir.
- Hazırlanan DMGDATA veri seti kullanılarak YOLOv5, YOLOv6, YOLOv7 ve YOLOv8'e ait tüm versiyonlar eğitilerek performans karşılaştırması

gerçekleştirilmiştir. En başarılı modelin YOLOv8 olduğu tespit edilerek tez için en uygun versiyon seçilmiştir.

- Önerilen DCASR modeli, seçilen YOLOv8m ve BF filtresi birleştirilerek literatüre yeni hibrit bir model sunulmuştur.
- Önerilen YOLOv8m+DCASR+BF modeli YOLOv8m ve YOLOv8m+SR(FSRCNN/ESPCNN/EDSR/LapSRN)+BF ile karşılaştırılmıştır. Sonuçta hibrit modelin düşük çözünürlüklü görüntülerde YOLOv8m modeline göre her zaman daha başarılı sonuç ürettiği görülmüştür. En başarılı sonucu üreten YOLOv8m+DCASR+BF modelinin düşük çözünürlüklü görüntülerde nesne tespit başarısını %4 ila %6 oranında arttırdığı tespit edilmiştir. Fakat standart kaliteli görüntülerde nesne tespit başarısını %0,3 oranında dahi olsa arttıran tek modelin DCASR algoritması kullanılarak tasarlanan hibrit model ile gerçekleştirildiği görülmüştür.
- Elde edilen sonuçlara bakıldığında düşük kaliteli görüntüler üreten kaynağın değiştirilmesi yerine algoritmik yöntemler kullanılarak donanım ve depolama alanından tasarruf edilebileceği gösterilmiştir.
- Ayrıca güvenlik kameralarını izleyen operatörlere yardımcı bir sistem olabileceği düşünülmektedir.

6. SONUÇ VE ÖNERİLER

Tehlikeli nesnelerin kamusal alanlarda yetkisiz kişiler tarafından kullanılması halkın güvenliğini tehdit etmektedir. Bu sebepten çoğu yerde operatörler tarafından sürekli izlenen kameralar mevcuttur. Fakat bu görüntüleri dikkatli bir şekilde sürekli takip etmenin zor olduğu bilinmektedir. Bu sebepten tez kapsamında operatörlere yardımcı tehlikeli nesneleri tespit eden hibrit bir model önerisi sunulmuştur. Hibrit modelin temel amacı; kalitesiz görüntüleri algoritmik yöntemler ile iyileştirerek mevcut nesne tespit yöntemlerinin başarısını arttırmaktır.

Bu doğrultuda öncelikle literatürde yer alan tehlikeli nesneler özelinde gerçekleştirilmiş çalışmalar incelenmiş ve özetlenmiştir. Kullandıkları veri setleri incelenerek kamuya açık olanlar tespit edilmiştir. Literatür taraması sonucunda kamuya açık ve etiketli olan az sayıda veri seti olduğu, mevcutların ise ya az sayıda veri barındırdığı ya da istenilen etiketlere sahip olmadığı farkedilmiştir. Bu sebepten tez kapsamında silah, tüfek ve bu ateşli nesneleri taşıyan insanı tespit etmek amacıyla açık kaynaklı platformlardan veriler toplanmış, temizlenmiş ve etiketlenmiştir. Sonuçta literatüre 18.231 adetlik silah, tüfek ve insan etiketlerine sahip DMGDATA adında yeni bir veri seti sunulmuştur. Bu veri seti sayesinde nesne tespit modellerinin nesnel olarak karşılaştırılmasının gerçekleştirilmesi hedeflenmiştir. Hazırlanan bu veri setinin %75'i tehlikeli nesnelerin tespiti için tasarlanan hibrit modelin eğitiminde, %15'i modelin validasyon işleminde ve %5'i test işlemlerinde kullanılmıştır. %5'lik test veri setinin yarısı DMGDATA'da yer alan rastgele görüntü kalitesi ve büyüklükte olan görüntülerden oluşurken diğer yarısı Qi ve diğerleri (2021) tarafından hazırlanan görüntü kalitesi düşük silah sınıflandırma veri seti içerisinden alınmıştır.

Tehlikeli nesneleri tespit etmek için önerilen hibrit model iki temel yapıdan meydana gelmektedir. İlki nesneleri tespit etmek için kullanılan YOLO mimarîsidir. Tasarlanan yapı için YOLO mimarîsine ait model seçilirken YOLOv5, YOLOv6, YOLOv7 ve YOLOv8 modellerine ait tüm versiyonlar DMGDATA ile eğitilmiş ve mAP50, mAP50-95, parametre sayısı, katman sayısı ve eğitim süresi parametrelerine göre değerlendirilmiştir. Değerlendirme sonucunda YOLOv8 modeline ait m versiyonunun mAP metriklerine göre başarısı ve eğitim süresi diğer modellerin sonuçlarına göre daha başarılı ve dengeli olduğu görülmüştür.

Buna istinaden nesne tespit aşamasında DMGDATA ile eğitilmiş YOLOv8m modeli kullanılmıştır.

Hibrit modelin diğer yapı taşı ise görüntü üzerinde nesne tespit başarısını arttırmak için görüntüleri iyileştiren SR mimarisidir. Model içerisinde SR mimarisinin kullanılmasının temel amacı görüntü kalitesini iyileştirip hedef nesneyi temsil eden piksel sayısını artırarak YOLO modelinin hedef nesneyi daha kolay ve yüksek başarı ile tespit etmesini sağlamaktır. Bu doğrultuda tez çalışmasında dikkat katmanı ve artık bloklardan meydana gelen hızlı ve etkili DCASR adında yeni bir SR algoritması önerilmiştir. Önerilen DCASR algoritmasının performansını test etmek için OpenCV tarafından sunulan *cv2.dnn_superres* kütüphanesindeki EDSR, FSRCNN, ESPCNN ve LapSRN algoritmaları ile PSNR metriğine ve üretim süresine göre karşılaştırma gerçekleştirilmiştir. Karşılaştırma işlemi görüntünün $x2$, $x3$ ve $x4$ oranında büyütülmüş hali ile gerçekleştirilmiştir. Buna göre; $x3$ ve $x4$ oranında DCASR, $x2$ oranında ise EDSR modelinin diğerlerine göre daha başarılı PSNR değeri ürettiği görülmüştür. Fakat görüntünün iyileştirilmesi için harcanan süreye bakıldığında EDSR modelinin $x2$ oranında ürettiği PSNR sonucu önerilen DCASR'ye göre yüksek olmasına rağmen üretim süresinin ~44 kat daha uzun sürdüğü görülmektedir. Bu sonuçlara istinaden DCASR modelinin gerçek zamanlı nesne tespit modellerinde kullanılmasının uygun olduğu düşünülmektedir.

YOLOv8m ile DCASR algoritması birleştirilerek hibrit model oluşturulmuştur. Ayrıca bu modele DCASR algoritmasından sonra BF katmanı eklenerek görüntüdeki nesnelerin keskinliği artırılmış gürültüsü azaltılmıştır. Bu sayede nesnelerin daha başarılı bir şekilde tespit edilmesi sağlanmıştır.

Önerilen YOLOv8m+DCASR+BF modeli iki farklı test kümesi (Test_1 ve Test_2) kullanılarak, YOLOv8m ve YOLOv8m+SR(FSRCNN/ESPCNN/EDSR/LapSRN)+BF ile karşılaştırmalı değerlendirilmiştir. Test_1 veri setinde standart kalitedeki (örn; 200×250) görüntüler yer alırken; Test_2 veri setinde kalitesi düşük (örn; 32×32) görüntüler yer almaktadır. Değerlendirmenin temel amacı SR katmanının nesne tespiti üzerindeki etkisini gözlemlemektir. Bunun için değerlendirme sırasında hassasiyet, duyarlılık, mAP-50 ve mAP50-95 metrikleri kullanılmıştır. Ayrıca maliyet ve zaman kontrolü yapabilmek adına harcanan bellek, disk, CPU, GPU miktarı ve zaman miktarı hesaplanmıştır. Sonuçlara göre özellikle bellek ve disk kullanımı her iki test setinde de benzer sonuçları ürettiği görülmüştür. GPU sadece DCASR modeli tarafından desteklendiği için diğer SR algoritmaları tarafından kullanılamamıştır. Çalışmanın sonuçları zaman açısından değerlendirildiğinde ise benzer şekilde görüntü boyutunun büyük olduğu için Test_1 veri setinin görüntü iyileştirme süresi

Test_2 setine göre daha çok zaman almıştır. Ayrıca EDSR algoritması ile oluşturulan hibrit modelin Test_1 veri setinde sonuç üretmediği Test_2 veri setinde ise en fazla süreyi harcayarak ortalama bir sonuç ürettiği gözlemlenmiştir. Bu durumda görüntü iyileştirme süresinin ve donanım maliyetinin görüntü büyüklüğüne bağlı değiştiğini söylemek mümkündür. Diğer metriklerden elde edilen sonuçlara bakıldığında ise; Test_1 veri seti üzerinde yapılan iyileştirmenin nesne tespit modelinin başarısını arttırmadığı aksine YOLOv8m+DCASR+BF hariç tüm modellerde azalttığı görülmüştür. YOLOv8m+DCASR+BF modeli ise yaklaşık %0,3'lük bir performans artışı sağlamıştır. Başka bir deyiş ile Test_1 veri setine göre YOLOv8m modelinin mAP50 değeri %85,2 iken; YOLOv8m+DCASR+BF modelinin $\times 2$, $\times 3$ oranında %85,4 ve $\times 4$ oranında %85,5 sonucunu ürettiği görülmüştür.

Test_2 veri seti üzerinde yapılan iyileştirmenin nesne tespit modelinin başarısını mAP50 metriğine göre yaklaşık %4 ila %6 oranında arttırdığı gözlemlenmiştir. Test_2 veri seti ile YOLOv8m modelinde mAP50 değeri %70,3'tür. Test sonuçlarına göre $\times 2$ ve $\times 4$ oranında en iyi mAP50 değeri sırasıyla %74,4 ve %76,3 ile YOLOv8m+DCASR+BF modeli aracılığıyla elde edilirken; $\times 3$ oranında en iyi mAP50 değeri %75,1 ile YOLOv8m+EDSR+BF modeli aracılığıyla elde edilmiştir. Fakat hem nesne tespit başarısı hem de harcanan donanım ve zaman açısından bakıldığında YOLOv8m+EDSR+BF modelinin gerçek dünyadaki uygulamalarda kullanıma uygun olmadığı düşünülmektedir. YOLOv8m+EDSR+BF ve YOLOv8m+DCASR+BF modeli arasında $\times 3$ ölçeğinde mAP50 metriğine göre %1 başarı farkı bulunmaktadır. Fakat zaman açısından değerlendirildiği YOLOv8m+EDSR+BF modeli yaklaşık 11 kat daha yavaş çalışmaktadır. Bu durumda sonuçlar zaman parametresi ile tekrar değerlendirildiğinde $\times 3$ oranında da YOLOv8m+DCASR+BF modelinin en başarılı sonucu ürettiği söylenebilir.

Sonuç olarak bu tez çalışması ile literatüre tehlikeli nesneler özelinde yeni bir veri seti, kanal dikkat ve artık blok katmanına sahip yeni bir SR algoritması ve YOLOv8m+DCASR+BF adında yeni bir hibrit model sunulmuştur. Geliştirilen hibrit modelin özellikle düşük kalitedeki kaynaklardan gelen görüntüler üzerinde iyileştirme işlemi yaparak donanım maliyetinden tasarruf ettireceği düşünülmektedir. Ayrıca operatörlere yardımcı bir sistem olarak kullanılabileceği ve bu sayede olası insan hatalarının önüne geçilebileceği düşünülmektedir.

KAYNAK

- Agustsson, E. & Timofte, R., 2017, Ntire 2017 challenge on single image super-resolution: Dataset and study, In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 126-135.
- Akcay, S., Kundegorski, M. E., Willcocks, C. G. & Breckon, T. P., 2018, Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery, *IEEE transactions on information forensics and security*, 13(9), 2203-2215.
- Anwar, S., Khan, S., & Barnes, N., 2020, A deep journey into super-resolution: A survey. *ACM Computing Surveys*, 53(3), 1-34.
- Bashir, S. M. A., & Wang, Y., 2021, Small object detection in remote sensing images with residual feature aggregation-based super-resolution and object detector network, *Remote Sensing*, 13(9), 1854.
- Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J., 2021, Weapon detection in real-time cctv videos using deep learning”, *IEEE Access*, 9, 34366-34382.
- Blum, R., Xue, Z., Liu, Z., & Forsyth, D. S., 2004, Multisensor concealed weapon detection by using a multiresolution mosaic approach, In *Proceedings of the Vehicular Technology Conference*, 4597-4601.
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M., 2020, YOLOv4: Optimal speed and accuracy of object detection, *arXiv preprint arXiv:2004*, 10934.
- Bozdağ, Z., 2021, Histopatolojik Görüntülerde Tümör Bölütlenmesi, *Doktora Tezi*, İnönü Üniversitesi.
- Brunetti, A., Buongiorno, D., Trotta, G. F. & Bevilacqua, V., 2018, Computer vision and deep learning techniques for pedestrian detection and tracking: A survey, *Neurocomputing*, 300, 17-33.
- Buckchash, H. & Raman, B., 2017, A robust object detector: application to detection of visual knives, In *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*, 633-638.
- Bushra, S. N., Shobana, G., Maheswari, K. U. & Subramanian, N., 2022, Smart Video Surveillance Based Weapon Identification Using Yolov5, In *Proceedings of the International Conference on Electronic Systems and Intelligent Computing*, 351-357.

- Castillo, A., Tabik, S., Pérez, F., Olmos, R. & Herrera, F., 2019, Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos using deep learning, *Neurocomputing*, 330, 151-161.
- Chen, C., Seff, A., Kornhauser, A. & Xiao, J., 2015, Deepdriving: Learning affordance for direct perception in autonomous driving, In *Proceedings of the IEEE international conference on computer vision*, 2722-2730.
- Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S. & Urtasun, R., 2016, Monocular 3d object detection for autonomous driving, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2147-2156.
- Chen, Y., Zhao, D., Lv, L., & Zhang, Q., 2018, Multi-task learning for dangerous object detection in autonomous driving, *Information Sciences*, 432, 559-571.
- Cohen, N., Gattuso, J. & MacLennan-Brown, K., 2009, *CCTV Operational Requirements Manual*, St. Albans: Home Office Scientific Development Branch, United Kingdom, ISBN: 978-1-84726-902-7
- Coşkun, M., Uçar, A., Yildirim, Ö. & Demir, Y., 2017, Face recognition based on convolutional neural network”, In *International Conference on Modern Electrical and Energy Systems*, 376-379.
- Courtrai, L., Pham, M. T. & Lefèvre, S., 2020, Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks, *Remote Sensing*, 12(19), 3152.
- Dai, J., Li, Y., He, K. & Sun, J., 2016, R-FCN: Object detection via region-based fully convolutional networks, *Advances in neural information processing systems*, 29.
- Dalal, N. & Triggs, B., 2005, Histograms of oriented gradients for human detection, In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, 886-893.
- Darker, I. T., Gale, A. G. & Blechko, A., 2008, CCTV as an automated sensor for firearms detection: Human-derived performance as a precursor to automatic recognition, In *Unmanned/Unattended Sensors and Sensor Networks V*, 7112, 208-219.
- Darker, I., Gale, A., Ward, L. & Blechko, A., 2007, Can CCTV reliably detect gun crime?, In *Proceedings of the Annual IEEE International Carnahan Conference on Security Technology*, 264-271.
- de Azevedo Kanehisa, R. F. & de Almeida Neto, A., 2019, Firearm Detection using Convolutional Neural Networks, In *Proceedings of the International Conference on Agents and Artificial Intelligence*, 707-714.
- Dong, C., Loy, C. C. & Tang, X., 2016, Accelerating the super-resolution convolutional neural network. In *Proceedings of the Computer Vision–ECCV: 14th European Conference*, 391-407.

- Dong, C., Loy, C. C., He, K. & Tang, X., 2015, Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2), 295-307.
- Dong, C., Loy, C. C., He, K. & Tang, X., 2014, Learning a deep convolutional network for image super-resolution. In *Proceedings of the Computer Vision: 13th European Conference*, 184-199.
- Du, L., Zhang, R. & Wang, X., 2020, Overview of two-stage object detection algorithms, In *Journal of Physics: Conference Series*, 1544(1), 1-6.
- Emrullah, A. C. A. R. & Özerdem, M. S., 2016, Tarımsal imge dokularından HOG algoritması ile öz nitelik çıkarımı ve öz nitelik tabanlı toprak neminin tahmini, *Anatolian Science-Bilgisayar Bilimleri Dergisi*, 1(1), 1-7.
- Erdem K., 2020, *Understanding Region of Interest (RoI Pooling)*, <https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling>, [Ziyaret tarihi: 21 Ağustos 2023].
- Fei-Fei, L. & Johnson, J. & Yeung, S., 2017, *Lecture 11: Detection and Segmentation*, Stanford University, https://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf, [Ziyaret tarihi: 10 Aralık 2022]
- Felzenszwalb, P. F., Girshick, R. B. & McAllester, D., 2010, Cascade object detection with deformable part models, In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2241-2248.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D., 2010, Object detection with discriminatively trained part-based models, In *Proceedings of the IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627-1645.
- Felzenszwalb, P. F., McAllester, D. & Ramanan, D., 2008, A discriminatively trained multiscale deformable part model, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1-8.
- Ferdous, S. N., Mostofa, M. & Nasrabadi, N. M., 2019, Super resolution-assisted deep aerial vehicle detection, *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications* 11006, 432-443.
- Fernandez-Carrobbles, M., Deniz, O. & Maroto, F., 2019, Gun and knife detection based on faster R-CNN for video surveillance, In *Proceedings of the Iberian conference on pattern recognition and image analysis*, 441-452.
- Freund, Y. & Schapire, R. E., 1997, A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1), 119-139.
- Gelana, F. & Yadav, A., 2019, Firearm detection from surveillance cameras using image processing and machine learning techniques, *Smart innovations in communication and computational sciences*, 25-34.
- Girshick, R., 2015, Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, 1440-1448.

- Girshick, R., Donahue, J., Darrell, T. & Malik, J., 2014, Rich feature hierarchies for accurate object detection and semantic segmentation, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580-587.
- González, J. L. S., Zaccaro, C., Álvarez-García, J. A., Morillo, L. M. S. & Caparrini, F. S., 2020, Real-time gun detection in CCTV: An open problem, *Neural networks*, 132, 297-308.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. & Bengio, Y., 2014, Generative Adversarial Nets, *Advances in neural information processing systems*, 2672-2680.
- Grega, M., Łach, S. & Sieradzki, R., 2013, Automated recognition of firearms in surveillance video, In *Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, 45-50.
- Grega, M., Mاتیolański, A., Guzik, P. & Leszczuk, M., 2016, Automated detection of firearms and knives in a CCTV image, *Sensors*, 16(1), 47.
- Gürbüz, M. E. & Gangal, A., 2014, Döndürülmüş Kayan Pencereleer Kullanarak İyleştirilmiş Hibrid Nesne Tespit Yöntemi, In *Proceedings of the Eleco 2014 Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*, 573-577.
- Harvey, R., Lebre, R., Massonnet, S., Aberer, K. & Demartini, G., 2023, Firearms On Twitter: A Novel Object Detection Pipeline, In *Proceedings of the International AAAI Conference on Web and Social Media*, 1128-1132.
- Hashmi, T. S. S., Haq, N. U., Fraz, M. M. & Shahzad, M., 2021, Application of Deep Learning for Weapons Detection in Surveillance Videos, In *Proceedings of the International Conference on Digital Futures and Transformative Technologies*, 1-6.
- He, K., Gkioxari, G., Dollár, P. & Girshick, R., 2017, Mask R-CNN, In *Proceedings of the IEEE international conference on computer vision*, 2961-2969.
- He, K., Zhang, X., Ren, S. & Sun, J., 2015, Spatial pyramid pooling in deep convolutional networks for visual recognition, In *Proceedings IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904-1916.
- Hunter, J., Dale, D., Firing E., Droettboom, M. & The Matplotlib development team, 2024, *Using Matplotlib*, <https://matplotlib.org/stable/users/index> [Ziyaret tarihi: 4 Mart 2024].
- Iqbal, J., Munir, M. A., Mahmood, A., Ali, A. R. & Ali, M., 2019, Orientation aware object detection with application to firearms, *arXiv preprint arXiv:1904.10032*, 22.
- Iqbal, M.J., Iqbal, M.M., Ahmad, I., Allassafi, M.O., Alfakeeh, A.S. & Alhomoud, A., 2021 Real-Time Surveillance Using Deep Learning, *Security and Communication Networks*, 1-17.
- Jocher, G., Chaurasia, A. & Qiu, J., 2023, *YOLO by Ultralytics*. *GitHub*. <https://github.com/ultralytics/ultralytics>, [Ziyaret tarihi: 10 Ağustos 2023].
- Jocher, G., Nishimura, K., Mineeva, T. & Vilariño, R., 2020, *YOLOv5*, <https://github.com/ultralytics/yolov5>, [Ziyaret tarihi: 28 Temmuz 2023]

- Kahveci, S., 2020, *Derin Öğrenme Algoritmaları Kullanarak Uydu Görüntüleri Zenginleştirme*, Yüksek lisans tezi, Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü.
- Kamencay, P., Benco, M., Mizdos, T. & Radil, R., 2017, A new method for face recognition using convolutional neural network, *Advances in Electrical and Electronic Engineering*, 15(4), 663-672.
- Karagiannakos S., 2019, *Localization and Object Detection with Deep Learning*, <https://towardsdatascience.com/localization-and-object-detection-with-deep-learning-67b5aca67f22>, [Ziyaret tarihi: 11 Ağustos 2023].
- Kaya, V., Tuncer, S. & Baran, A., 2021, Detection and classification of different weapon types using deep learning, *Applied Sciences*, 11(16), 7535.
- Kayalvizhi, R., Malarvizhi, S., Choudhury, S. D., Topkar, A. & Vijayakumar, P., 2020, Detection of sharp objects using deep neural network based object detection algorithm, In *Proceedings of the International Conference on Computer, Communication and Signal Processing*, 1-5.
- Kim, J., Lee, J. K. & Lee, K. M., 2016, Accurate image super-resolution using very deep convolutional networks, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1646-1654.
- Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2012, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 25(1), 1097-1105.
- Lai, J. & Maples, S., 2017, Developing a real-time gun detection classifier, *Course: CS231n, Stanford University*, 1-4.
- Lai, W. S., Huang, J. B., Ahuja, N. & Yang, M. H. 2017, Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 624-632.
- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Xiaolin, W., Xiaolin, W. & Wei Meituan Inc., 2022, YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 1-17.
- Lim, B., Son, S., Kim, H., Nah, S. & Mu Lee, K., 2017, Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 136-144.
- Liu, S., Qi, L., Qin, H., Shi, J. & Jia, J., 2018, Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8759-8768.
- Lowe, D. G., 1999, Object recognition from local scale-invariant features, In *Proceedings of the seventh IEEE international conference on computer vision*, 1150-1157.

- Mohan S., 2020, *6 Different Types of Object Detection Algorithms in Nutshell*, <https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/>, [Ziyaret tarihi: 11 Aralık 2022].
- Narejo, S., Pandey, B., Rodriguez, C. & Anjum, M. R., 2021, Weapon detection using YOLO V3 for smart surveillance system, *Mathematical Problems in Engineering*, 1-9.
- Noh, J., Bae, W., Lee, W., Seo, J. & Kim, G., 2019, Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9725-9734.
- Noor, W. E. I. B. W. & Isa, N. M., 2021, Object Detection: Harmful Weapons Detection using YOLOv4, In *Proceedings of the IEEE Symposium on Wireless Technology & Applications*, 63-70.
- Olmos, R., Tabik, S. & Herrera, F., 2018, Automatic handgun detection alarm in videos using deep learning, *Neurocomputing*, 275, 66-72.
- Olmos, R., Tabik, S., Lamas, A., Perez-Hernandez, F. & Herrera, F., 2019, A binocular image fusion approach for minimizing false positives in handgun detection with deep learning, *Information Fusion*, 49, 271-280.
- Papageorgiou, C. P., Oren, M. & Poggio, T., 1998, A general framework for object detection, In *Proceedings of the Sixth International Conference on Computer Vision*, 555-562.
- Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H. & Herrera, F., 2020, Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance, *Knowledge-Based Systems*, 194.
- Piza, E. L., Welsh, B. C., Farrington, D. P. & Thomas, A. L., 2019, CCTV surveillance for crime prevention: A 40-year systematic review with meta-analysis, *Criminology & Public Policy*, 18(1), 135-159.
- Popescu, A. & Coatrieux, G., 2020, *Large-Scale Object Detection for Social Media User's Visual Privacy Protection*, Master Thesis, Paris-Saclay University.
- Qi, D., Tan, W., Liu, Z., Yao, Q. & Liu, J., 2021, A Dataset and System for Real-Time Gun Detection in Surveillance Video Using Deep Learning, In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 667-672.
- Rabbi, J., Ray, N., Schubert, M., Chowdhury, S. & Chao, D., 2020, Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. *Remote Sensing*, 12(9), 1432.
- Rahman, M. T., 2022, Driving-Scene Image Classification Using Deep Learning Networks: Yolov4 Algorithm, Master Thesis, Upsala Universite.
- Raturi, G., Rani, P., Madan, S. & Dosanjh, S., 2019, ADoCW: An automated method for detection of concealed weapon, In *Proceedings of the Fifth International Conference on Image Information Processing*, 181-186.

- Redmon, J. & Farhadi, A., 2017, YOLO9000: better, faster, stronger, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263-7271.
- Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016, You only look once: Unified, real-time object detection, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.
- Reis, D., Kupec, J., Hong, J. & Daoudi, A., 2023, Real-Time Flying Object Detection with YOLOv8. *arXiv preprint arXiv:2305.09972*.
- Ren, S., He, K., Girshick, R. & Sun, J., 2015, Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems*, 28, 91-99.
- Romero, D., & Salamea, C., 2019, Convolutional models for the detection of firearms in surveillance videos, *Applied Sciences*, 9(15), 2965.
- Rose, J., Bourlai, T. & Loudermilk, J. A., 2020, Assessment of data augmentation techniques for firearm detection in surveillance videos. In *Proceedings of the International Conference on Big Data*, 1838-1846.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L., 2015, Imagenet large scale visual recognition challenge, *International journal of computer vision*, 115(3), 211-252.
- Salari, A., Djavadifar, A., Liu, X. R., & Najjaran, H., 2022, Object recognition datasets and challenges: A review, *Neurocomputing*, 129-152.
- Salido, J., Lomas, V., Ruiz-Santaquiteria, J. & Deniz, O., 2021, Automatic handgun detection with deep learning in video surveillance images, *Applied Sciences*, 11(13), 6085.
- Sasank, S., 2023, *Gun Object Detection*, <https://www.kaggle.com/datasets/issaisasank/guns-object-detection>, [Ziyaret tarihi: 9 Nisan 2023]
- Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R. & LeCun Y., 2014, Overfeat: Integrated recognition, localization and detection using convolutional networks, *arXiv preprint arXiv:1312.6229*, 1-16.
- Sheen, D. M., McMakin, D. L. & Hall, T. E., 2001, Three-dimensional millimeter-wave imaging for concealed weapon detection, *IEEE Transactions on microwave theory and techniques*, 49(9), 1581-1592.
- Shermeyer, J. & Van Etten, A., 2019, The effects of super-resolution on object detection performance in satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1432-1441.
- Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D. & Wang, Z. 2016, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874-1883.

- Singleton, M., Taylor, B., Taylor, J. & Liu, Q., 2018, Gun identification using tensorflow, *In International Conference on Machine Learning and Intelligent Communications*, 3-12.
- Sivakumar, P., 2021, Real Time Crime Detection Using Deep Learning Algorithm, *In Proceedings of the International Conference on System, Computation, Automation and Networking*, 1-5.
- Solawetz, J., 2023, What is YOLOv8?, *The Ultimate Guide. Roboflow Blog*: <https://blog.roboflow.com/whats-new-in-yolov8/>, [Ziyaret tarihi: 4 Mart 2024]
- Şenalp, F. M. & Ceylan, M., 2022, Termal yüz görüntülerinden oluşan yeni bir veri seti için derin öğrenme tabanlı süper çözünürlük uygulaması. *Politeknik Dergisi*, 1-1.
- Temiz H., 2020, Ultrasound image super resolution with deep learning approach, *Doktora tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü.
- Terven, J. & Cordova-Esparza, D., 2023, A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv preprint arXiv:2304.00501*.
- Terven, J., Córdova-Esparza, D. M. & Romero-González, J. A., 2023, A comprehensive review of yolo architectures in computer vision: From yolov1 to YOLOv8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716.
- Tickner, A. H. & Poulton, E. C., 1973, Monitoring up to 16 synthetic television pictures showing a great deal of movement, *Ergonomics*, 16(4):381-401.
- Tiwari, R. K. & Verma, G. K., 2015, A computer vision based framework for visual gun detection using harris interest point detector, *Procedia Computer Science*, 54, 703-712.
- Tiwari, R. K. & Verma, G. K., 2015, A computer vision based framework for visual gun detection using SURF, *In Proceedings of the International Conference on Electrical, Electronics, Signals, Communication and Optimization*, 1-5.
- Tomasi, C., & Manduchi, R., 1998, Bilateral filtering for gray and color images, *In Proceedings of the Sixth international conference on computer vision*, 839-846.
- Truong, N. T., Vo, N. D. & Nguyen, K., 2020, The effects of super-resolution on object detection performance in an aerial image, *In Proceedings of the Conference on Information and Computer Science*, 256-260.
- Uijlings, J. R., Van De Sande, K. E., Gevers, T. & Smeulders, A. W., 2013, Selective search for object recognition, *International journal of computer vision*, 104(2), 154-171.
- Uner, M. K., Ramac, L. C., Varshney, P. K., & Alford, M. G., Concealed weapon detection: an image fusion approach, *In Investigative image processing*, 2942:123-132. (1997).
- Upadhyay, E. M. & Rana, N. K., 2014, Exposure fusion for concealed weapon detection, *In Proceedings of the International Conference on Devices, Circuits and Systems*, 1-6.
- Vajhala, R., Maddineni, R. & Yeruva, P. R., 2016, *Weapon detection in surveillance camera images*, Master Thesis, Blekinge Institute of Technology.

- Vapnik, V. N., 1995, *The Nature of Statistical Learning Theory*, 2nd ed., Springer Science & Business Media, New York, ISBN: 978-1-4757-3264-1.
- Verma, G. K. & Dhillon, A., 2017, A handheld gun detection using Faster R-CNN deep learning, In *Proceedings of the 7th international conference on computer and communication technology*, 84-88.
- Viola P. and Jones M. J., 2004, Robust real-time face detection, *International Journal of Computer Vision*, 57(2), 137-154.
- Viola, P. & Jones, M., 2001, Rapid object detection using a boosted cascade of simple features, In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1-9.
- Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M., 2023a, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464-7475.
- Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W. & Yeh, I. H., 2020, CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 390-391.
- Wang, G., Ding, H., Duan, M., Pu, Y., Yang, Z. & Li, H., 2023b, Fighting against terrorism: A real-time CCTV autonomous weapons detection based on improved YOLOv4, *Digital Signal Processing*, 132, 103790.
- Wang, Y., Bashir, S. M. A., Khan, M., Ullah, Q., Wang, R., Song, Y., Guo, Z. & Niu, Y., 2022, Remote sensing image super-resolution and object detection: Benchmark and state of the art, *Expert Systems with Applications*, 197, 116793.
- Wang, Z. Z., Xie, K., Zhang, X. Y., Chen, H. Q., Wen, C. & He, J. B., 2021, Small-object detection based on yolo and dense block via image super-resolution, *IEEE Access*, 9, 56416-56429.
- Wojke, N., Bewley, A. & Paulus, D., 2017, Simple online and realtime tracking with a deep association metric, In *Proceedings of the IEEE international conference on image processing*, 3645-3649.
- Woo, S., Park, J., Lee, J. Y. & Kweon, I. S., 2018, Cbam: Convolutional block attention module, In *Proceedings of the European conference on computer vision*, 3-19.
- Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S. & Lan, X., 2020, A review of object detection based on deep learning, *Multimedia Tools and Applications*, 79(33), 23729-23791.
- Xue, Z., Blum, R. S., & Li, Y., 2002, Fusion of visual and IR images for concealed weapon detection, In *Proceedings of the Fifth International Conference on Information Fusion*, 1198-1205.
- Yang, B., Huang, C. & Nevatia, R., 2011, Learning affinities and dependencies for multi-target tracking using a CRF model, In *CVPR*, 1233-1240.

- Yang, Z. & Nevatia, R., 2016, A multi-scale cascade fully convolutional network face detector, In *Proceedings of the International Conference on Pattern Recognition*, 633-638.
- Yellapragada, S., Li, Z., Doshi, K. B., Mhasakar, P. M., Fan, H., Wei, J., Blash, E., Zhang, B. & Ling, H. 2023, CCTV-Gun: Benchmarking Handgun Detection in CCTV Images. *arXiv preprint arXiv:2303.10703*.
- Zou, Z., Shi, Z., Guo, Y. & Ye, J., 2019, Object detection in 20 years: A survey, *arXiv preprint arXiv:1905.05055*.



İNTİHAL RAPORU İLK SAYFASI

BERİKA VAROL

ORJİNALLİK RAPORU

%**4**

BENZERLİK ENDEKSİ

%**3**

İNTERNET KAYNAKLARI

%**2**

YAYINLAR

%**1**

ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1

acikbilim.yok.gov.tr

İnternet Kaynağı

<%**1**

2

abakus.inonu.edu.tr

İnternet Kaynağı

<%**1**

3

Submitted to University Politehnica of
Bucharest

Öğrenci Ödevi

<%**1**

4

dergipark.org.tr

İnternet Kaynağı

<%**1**

5

nek.istanbul.edu.tr:4444

İnternet Kaynağı

<%**1**

6

ichi.pro

İnternet Kaynağı

<%**1**

7

Submitted to Cankaya University

Öğrenci Ödevi

<%**1**

8

www.seruvenyayinevi.com

İnternet Kaynağı

<%**1**

9

doaj.org

İnternet Kaynağı

<%**1**