



T.C.
BATMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**DERİN ÖĞRENME ALGORİTMALARI
KULLANARAK İNSANSIZ HAVA ARAÇLARI
İLE SİLAH TESPİTİ**

Mustafa BURGAZ

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Temmuz-2020
BATMAN
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Mustafa BURGAZ tarafından hazırlanan “Derin Öğrenme Algoritmaları Kullanarak İnsansız Hava Araçları İle Silah Tespiti” adlı tez çalışması 09/07/2020 tarihinde aşağıdaki jüri tarafından oy birliği ile Batman Üniversitesi Fen Bilimleri Enstitüsü Elektrik–Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Doç. Dr. Ömer Faruk ERTUĞRUL

.....

Danışman

Dr. Öğr. Üyesi Cafer BUDAK

.....

Üye

Dr. Öğr. Üyesi Mehmet Emin ASKER

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Şahnaz TİĞREK

FBE Müdürü

TEZ BİLDİRİMİ

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

DECLARATION PAGE

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

Mustafa BURGAZ

Tarih: 09.07.2020

ÖZET

YÜKSEK LİSANS

DERİN ÖĞRENME ALGORİTMALARI KULLANARAK İNSANSIZ HAVA ARAÇLARI İLE SİLAH TESPİTİ

Mustafa BURGAZ

Batman Üniversitesi Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğretim Üyesi Cafer BUDAK

2020, 108 Sayfa

Jüri

Doç. Dr. Ömer Faruk ERTUĞRUL
Dr. Öğretim Üyesi Cafer BUDAK
Dr. Öğretim Üyesi Mehmet Emin ASKER

Derin öğrenme algoritmalarının günümüzde yaygınlaşması görüntü ve videolarda nesne tespit, tanıma uygulamalarının artmasına sebep olmuştur. Nesne tespit ve tanıma uygulamaları son yıllarda güvenlik, savunma, doğal afetler (sel, deprem ve yangın vb.), sağlık (salgınların yayılımının önlenmesi vb.), tarım, ormancılık alanlarında birçok probleme çözüm bulmaktadır. Nesne tespit, tanıma uygulamalarında oldukça yaygın kullanılan algoritmaların başında Bölgesel Tabanlı Konvolüsyonel Sinir Ağları (R-CNN) gelmektedir. R-CNN'nin geliştirilen tespit uygulamalarına yardımcı olmaları açısından Hızlı bölgesel tabanlı konvolüsyonel sinir ağları (Fast R-CNN) ve Daha hızlı bölgesel tabanlı konvolüsyonel sinir ağları (Faster R-CNN) algoritmaları geliştirilmiştir. Nesne tespit uygulamalarının başarısını daha da artırmak için kullanılan bir başka Konvolüsyonel sinir ağları (CNN) algoritması da ResNet101 algoritmasıdır. Özellikle görüntü tespitinde yaygın bir şekilde kullanılan ResNet101, R-CNN, Fast R-CNN ve Faster R-CNN algoritmalarının birbirleri arasındaki nesne tespit doğruluk oranı, nesne tespit zamanı gibi farkları en aza indirmek için tercih edilmiştir.

Bu çalışmada insansız hava aracı (İHA) ile havadan çekilmiş görüntülerden nesne (silah) tespiti yapılması amaçlanmıştır. Elde edilen görüntülerde R-CNN nesne (silah) tespitinde doğru tahmin oranının diğer R-CNN çeşitlerinden yüksek olması sebebiyle tercih edilmiştir. R-CNN algoritmalarının yanında doğru tahmin oranına katkısını görebilmek amacıyla ResNet101 algoritmasının kullanımı bu çalışmada denenmiştir. Bu kapsamda İHA ile havadan çekilmiş 200 adet görüntü kullanarak eğitim verileri ve test verileri oluşturulmuştur. Yapılan eğitim sonucunda veri seti üzerinde R-CNN mimarisi ve ResNet101 mimarisiyle %99 doğruluk oranı, hassasiyet ile görüntü tespit edilmiştir.

Söz konusu çalışma ile R-CNN mimarisinin ve ResNet101 mimarisinin İHA görüntülerinde nesne (silah) tespitinde ne kadar başarılı olduğu ortaya konulmuştur.

Anahtar Kelimeler: Bölgesel Tabanlı Konvolüsyonel Sinir Ağları, Derin Öğrenme, İnsansız Hava Araçları, Nesne Tespit, ResNet101 mimarisi

ABSTRACT

MS THESIS

WEAPON DETECTION WITH UNMANNED AIR VEHICLE IMAGES THROUGH DEEP LEARNING ALGORITHMS

Mustafa BURGAZ

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF BATMAN
UNIVERSITY**

**THE DEGREE OF MASTER OF SCIENCE
IN ELECTRICAL AND ELECTRONICS ENGINEERING**

Advisor: Dr. Öğretim Üyesi Cafer BUDAK

2020, 108 Pages

Jury

Doç. Dr. Ömer Faruk ERTUĞRUL

Dr. Öğretim Üyesi Cafer BUDAK

Dr. Öğretim Üyesi Mehmet Emin ASKER

The proliferation of deep learning algorithms today has led to an increase in object detection and recognition applications in images and videos. Object detection and recognition applications have found solutions to many problems in the fields of security, defense, natural disasters (flood, earthquake and fire etc.), health (prevention of the spread of outbreaks etc.), agriculture, forestry in recent years. Regional Based Convolutional Neural Networks (R-CNN) are among the most widely used algorithms in object detection and recognition applications. Region based Convolutional Networks (Fast R-CNN) and faster region based convolutional neural networks (Faster R-CNN) algorithms have been developed in order to assist the detection applications of R-CNN. Another Convolutional Neural Networks (CNN) algorithm used to further increase the success of object detection applications is the ResNet101 algorithm. Especially, ResNet101, which is widely used in image detection, has been preferred to minimize differences such as object detection accuracy rate, object detection time of R-CNN, Fast R-CNN and Faster R-CNN algorithms.

In this study, it is aimed to detect objects (weapons) from aerial images taken by unmanned aerial vehicle. In the images obtained, R-CNN were preferred because the correct prediction rate was higher than other R-CNN types in object (weapon) detection. In addition to R-CNN algorithms, the use of ResNet101 algorithm has been tried in this study in order to see its contribution to the correct prediction rate. In this context, training and test data sets were created using 200 images taken from the air with a drone. As a result of the training, the image based result was obtained with R-CNN architecture and ResNet101 architecture with 99% accuracy rate on the data set.

With this study, it has been demonstrated how successful the R-CNN architecture and ResNet101 architecture are in the detection of objects (weapons) in unmanned aerial images.

Keywords: Regional Based Convolutional Neural Networks, Deep Learning, Unmanned Aerial Vehicles, Object Detection, ResNet101 architecture

ÖNSÖZ

Tez çalışmasının tüm aşamalarında desteklerini esirgemeyen çok değerli bilim insanı; danışman hocam Dr. Öğretim Üyesi Cafer BUDAK'a yüksek lisans süresince desteklerini esirgemeyen, evlendiğim günden bu yana her türlü çabamın ortağı ve destekçisi olan değerli eşim Neslihan Burgaz'a gönülden teşekkür ediyorum.

Mustafa BURGAZ
BATMAN-2020

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
KISALTMALAR	ix
ŞEKİLLER LİSTESİ	x
TABLolar LİSTESİ	xii
1. GİRİŞ	1
1.1. Derin Öğrenme.....	2
1.2. Derin Öğrenmenin Tarihi Süreci.....	3
1.3. Derin Öğrenme Mimarileri	5
1.3.1. Konvolüsyonel sinir ağları	5
1.3.2. Tekrarlayan sinir ağları	6
1.3.3. Uzun kısa süreli hafıza ağları.....	7
1.3.4. Kısıtlı Boltzman makineleri	9
1.3.5. Derin inanç ağı	9
1.3.6. Derin oto-kodlayıcılar	10
1.4. Derin Öğrenme Kullanım Alanları	11
1.4.1. Doğal dil işleme	12
1.4.2. Biyomedikal işleme	14
2. KAYNAK ARAŞTIRMASI	15
2.1. Nesne Tespit Kavramı ve Tanıma Türleri.....	15
2.1.1. Yüz tanıma	16
2.1.2. Yaya algılama	17
2.2. Nesne Tespitte Bilgisayar Görüşü, Görüntü İşleme ve Örüntü Tanıma	18
2.2.1. Bilgisayar görüşü kavramı	18
2.2.2. Görüntü işleme kavramı.....	21
2.2.3. Örüntü tanıma	23
2.3. Derin Öğrenme Algoritmalarında Nesne Tespitinde Yer Alan Sinir Ağları	27
2.3.1. Derin öğrenmede konvolüsyonel sinir ağları.....	27
2.3.1.1. Konvolüsyonel sinir ağlarında aktivasyon fonksiyonları	28
2.3.1.2. Konvolüsyonel sinir ağlarının mimarisi	31
2.3.2. Bölgesel tabanlı konvolüsyonel sinir ağları.....	34
2.3.2.1. Bölgesel tabanlı konvolüsyonel sinir ağları mimarisi.....	35
2.3.2.2. Bölge tabanlı konvolüsyonel sinir ağlarının problemleri	38
2.3.3. Bölge tabanlı konvolüsyonel sinir ağları çeşitleri.....	39
2.3.3.1. Hızlı bölgesel tabanlı konvolüsyonel sinir ağları.....	39
2.3.3.2. Daha hızlı bölgesel tabanlı konvolüsyonel sinir ağları	41

2.3.4. ResNet ile görüntü sınıflandırma	42
2.4. Nesne Tespitinde Kullanılan Kütüphaneler	44
2.4.1. Theano	45
2.4.2. Caffe.....	45
2.4.3. TensorFlow	46
2.4.4. Torch.....	46
2.4.5. Digits.....	46
2.5. Nesne Tespitinde Kullanılan Araçlar.....	47
2.5.1. İnsansız hava araçlarının kavramı ve özellikleri.....	47
2.5.2. İnsansız hava araçlarının tarihçesi	50
2.5.3. Türkiye’de insansız hava araçları	52
2.5.4. İnsansız hava araçlarının kullanım alanları	53
2.5.5. İnsansız hava araçlarının nesne tanımadaki rolü	55
3. MATERYAL VE YÖNTEM.....	57
3.1. Nesne Tespitinde Kullanılan İnsansız Hava Aracının Kamera Özellikleri.....	57
3.2. Nesne Tespitteki Değerlendirme İçin Yöntemler	57
3.2.1. Nesne tespit değerlendirmesi için ortalama hassasiyet.....	57
3.2.2. Hassasiyet, hatırlama ve F1	58
3.2.3. Kesişim birliği.....	60
3.3. İnsansız Hava Araçları ile Nesne Tespit Aşamaları.....	61
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA	63
4.1. Eğitim ve Test Sonuçları ve Değerlendirme Adımları.....	63
4.1.1. Tensorflow nesne tespit kurulumu ve anaconda virtual environment oluşturma	63
4.1.2. Veri setini oluşturma ve etiketleme	65
4.1.3. Eğitim ayarları ve eğitimi gerçekleştirme.....	67
4.2. Eğitim Modellerinin Tensorboard Grafiklerinin Değerlendirilmesi.....	73
4.2.1. Tensorboard grafiklerinin değerlendirilmesi	74
4.2.1.1. Faster_rcnn_inception_v2_coco_2018 modeli grafiklerinin değerlendirilmesi	74
4.2.1.2. Rfcn_resnet101_coco_2018 modeli grafiklerinin değerlendirilmesi.....	77
4.2.2. Kullanılan modellerin tahmini ile gerçek değerlerin karşılaştırılması.....	80
4.2.3. Kullanılan modellerin hassasiyet ve hatırlama eğrisi	88
5. SONUÇLAR VE ÖNERİLER	94
5.1. Sonuçlar	94
5.2. Öneriler	95
KAYNAKLAR	96
ÖZGEÇMİŞ	108

KISALTMALAR

CNN	:Konvolüsyonel Sinir Ağları
DBN	:Derin İnanç Ağı
İHA	:İnsansız Hava Aracı
LSTM	:Uzun Kısa Süreli Bellek
NLP	:Doğal Dil İşleme
RBM	:Sınırlı Boltzman Makineleri
R-CNN	:Bölgesel Sinir Ağları
RE-LU	:Doğrusal Birim Katmanı
RNN	:Tekrarlayan Sinir Ağları
USA	:Amerika Birleşik Devletleri
YKİ	:Yer Kontrol İstasyonu
YVT	:Yer Veri Terminali

ŞEKİLLER LİSTESİ

Şekil	Sayfa
Şekil 1.1. Bilinen ilk derin ağ mimarisi.....	4
Şekil 1.2. LeNet mimarisi.....	5
Şekil 1.3. AlexNet mimarisi.....	6
Şekil 1.4. Basit tekrarlayan sinir ağları.....	7
Şekil 1.5. RNN’de kullanılan SRN birimi (solda) ve LSTM (sağdaki) ayrıntılı şeması. .	8
Şekil 1.6. DBN ve RBM mimarilerinin karşılaştırılması.....	10
Şekil 1.7. Derin oto-kodlayıcı algoritma şeması.....	11
Şekil 2.1. Yüz tanıma ile ilgili çalışma.....	16
Şekil 2.2. Bilgisayarın algıladığı görüntü.	19
Şekil 2.3. Pikselleri ifade eden matris.....	22
Şekil 2.4. Görüntü işleme akışı.....	23
Şekil 2.5. Örüntü tanıma sistemi.....	25
Şekil 2.6. Sigmoid aktivasyon fonksiyonu.....	29
Şekil 2.7. Tanh aktivasyon fonksiyonu.....	29
Şekil 2.8. ReLU aktivasyon fonksiyonu.....	30
Şekil 2.9. Konvolüsyon işlemi.....	32
Şekil 2.10. CNN çalışma mantığı.....	36
Şekil 2.11. CNN mimarisi.....	37
Şekil 2.12. Fast R-CNN mimarisi.....	40
Şekil 2.13. Faster R-CNN mimarisi.....	42
Şekil 2.14. R-CNN algoritmalarının hız karşılaştırmaları.....	42
Şekil 2.15. ResNet mantığı.	44
Şekil 2.16. Çeşitli İHA Türleri.....	49
Şekil 3.1. Aden E58 Drone.....	57
Şekil 3.2. Alıcı çalışma karakteristiği.....	60
Şekil 3.3. ResNet nesne tespit mimarisi.....	61
Şekil 3.4. Faster R-CNN nesne tespit mimarisi.....	62
Şekil 3.5. Nesne Tespit İşleyişi.....	62
Şekil 4.1. İHA’lardan elde edilen görüntüler.....	65
Şekil 4.2. Resimlerde etiketlenen nesneler-1.....	66
Şekil 4.3. Resimlerde etiketlenen nesneler-2.....	66
Şekil 4.4. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-1).....	68
Şekil 4.5. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-2).....	68
Şekil 4.6. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-3).....	68
Şekil 4.7. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-4).....	69
Şekil 4.8. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-5).....	69
Şekil 4.9. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-6).....	69
Şekil 4.10. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-7).....	69
Şekil 4.11. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-8).....	70
Şekil 4.12. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-9).....	70
Şekil 4.13. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-10).....	70
Şekil 4.14. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-11).....	70
Şekil 4.15. Rfcn_resnet101_coco_2018 (Test Resmi-1).....	71
Şekil 4.16. Rfcn_resnet101_coco_2018 (Test Resmi-2).....	71

Şekil 4.17. Rfcn_resnet101_coco_2018 (Test Resmi-3)	71
Şekil 4.18. Rfcn_resnet101_coco_2018 (Test Resmi-4)	71
Şekil 4.19. Rfcn_resnet101_coco_2018 (Test Resmi-5)	72
Şekil 4.20. Rfcn_resnet101_coco_2018 (Test Resmi-6)	72
Şekil 4.21. Rfcn_resnet101_coco_2018 (Test Resmi-7)	72
Şekil 4.22. Rfcn_resnet101_coco_2018 (Test Resmi-8)	72
Şekil 4.23. Rfcn_resnet101_coco_2018 (Test Resmi-9)	73
Şekil 4.24. Rfcn_resnet101_coco_2018 (Test Resmi-10)	73
Şekil 4.25. Rfcn_resnet101_coco_2018 (Test Resmi-11)	73
Şekil 4.26. Faster R-CNN Detectionboxes_Precision mAP grafiği	74
Şekil 4.27. Detectionboxes_Recall grafiği	75
Şekil 4.28. Detectionboxes_Recall (medium) grafiği	75
Şekil 4.29. Clipped gradient norm grafiği	76
Şekil 4.30. Global gradient norm grafiği	76
Şekil 4.31. Learning rate grafiği	77
Şekil 4.32. Loss fonksiyon grafiği	77
Şekil 4.33. ResNet101 Detectionboxes_Precision mAP grafiği	78
Şekil 4.34. Detectionboxes_Recall grafiği	78
Şekil 4.35. Clipped gradient norm grafiği	79
Şekil 4.36. Global gradient norm grafiği	79
Şekil 4.37. Learning rate grafiği	80
Şekil 4.38. Loss fonksiyon grafiği	80
Şekil 4.39. Birinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli	81
Şekil 4.40. İkinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli	81
Şekil 4.41. Üçüncü test resmi Faster_rcnn_inception_v2_coco_2018 modeli	82
Şekil 4.42. Dördüncü test resmi Faster_rcnn_inception_v2_coco_2018 modelinin	82
Şekil 4.43. Beşinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli	82
Şekil 4.44. Altıncı test resmi Faster_rcnn_inception_v2_coco_2018 modeli	83
Şekil 4.45. Yedinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli	83
Şekil 4.46. Sekizinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli	83
Şekil 4.47. Dokuzuncu test resmi Faster_rcnn_inception_v2_coco_2018 modeli	84
Şekil 4.48. Onuncu test resmi Faster_rcnn_inception_v2_coco_2018 modeli	84
Şekil 4.49. Birinci test resmi Rfcn_resnet101_coco_2018 modeli	84
Şekil 4.50. İkinci test resmi Rfcn_resnet101_coco_2018 modeli	85
Şekil 4.51. Üçüncü test resmi Rfcn_resnet101_coco_2018 modeli	85
Şekil 4.52. Dördüncü test resmi Rfcn_resnet101_coco_2018 modeli	85
Şekil 4.53. Beşinci test resmi Rfcn_resnet101_coco_2018 modeli	86
Şekil 4.54. Altıncı test resmi Rfcn_resnet101_coco_2018 modeli	86
Şekil 4.55. Yedinci test resmi Rfcn_resnet101_coco_2018 modeli	86
Şekil 4.56. Sekizinci test resmi Rfcn_resnet101_coco_2018 modeli	87
Şekil 4.57. Dokuzuncu test resmi Rfcn_resnet101_coco_2018 modeli	87
Şekil 4.58. Onuncu test resmi Rfcn_resnet101_coco_2018 modeli	87
Şekil 4.59. Faster R-CNN kesin referans değerleri	89
Şekil 4.60. ResNet kesin referans değerleri	91
Şekil 4.61. Faster R-CNN mimarisinin P ve R eğrisi	93
Şekil 4.62. ResNet Mimarisinin P ve R eğrisi	93

TABLÖLAR LİSTESİ

Tablo	Sayfa
Tablo 2.1. Derin öğrenme kütüphaneleri	45
Tablo 2.2. Derin öğrenme kütüphanelerinin kullandığı algoritmalar	47
Tablo 3.1. Karışıklık matrisi	58
Tablo 4.1. Faster R-CNN mimarisinin TP, FP VE FN değerleri.....	92
Tablo 4.2. ResNet mimarisinin TP, FP ve FN değerleri.....	92

1. GİRİŞ

Günümüz teknolojileri insanoğlunun hayalinin ötesine taşımakta ustalaşmaktadır. Durum böyle olunca son zamanlarda yapay zekâ adından sıkça söz ettirmektedir. Yapay zekâ hayatın her alanına hızlı bir şekilde girmesi ile birlikte hemen hemen her alanı etkilemeye başlamıştır. Teknolojik ve yapay zekâ alanındaki atılımları ile gelecekte insanın yerini makinelerle bırakacağını doğrulamaktadır.

İnsanoğlunun yapmakta zorlandığı sınıflandırma, istatistik, matematik vb. işlemleri çeşitli yapay zekâ uygulamaları sayesinde makineler kolay bir şekilde yapmaktadır. Makine öğrenme tekniklerinin artmasıyla derin öğrenme kavramı da beraberinde gelmiştir. Derin öğrenmenin amaçlara yönelik geliştirilmesi sayesinde bankacılık, savunma sanayi, tıp, mühendislik vb. alanlarda büyük dönüşümlerin yaşanmasını sağlamıştır.

Derin öğrenme algoritmaları sayesinde geçmişte savunma sanayi de kullanılan ancak günümüzde hemen hemen her alanda kullanılan İHA’larda bu gelişmelerin bir çıktısıdır. İHA’ların otonom bir şekilde göreve yönelik kullanılması, meteorolojik olumsuzluklardan en az şekilde etkilenmesi ve insan hayatını riske atmadan yararlanılması ile hayatın hemen her alanında kullanılmaktadır.

İHA’lar genellikle keşif ve gözetleme amacıyla aktif olarak kullanılmaktadır. Bu kullanım amaçlarının önemli bir unsuru olan görüntüleme sayesinde nesne tespiti yapılmaktadır. Elde edilen görüntülerde tercih edilen derin öğrenme algoritmalarıyla nesnelerin kolaylıkla tespit edilmesini sağlanmaktadır.

Bu kapsamda çalışmada İHA’ların farklı açılardan ve farklı yükseklikte çekilen görüntülerle nesne tespitinde kullanılan derin öğrenme algoritmalarının başarısı analiz edilecek ve elde edilen görüntülerin birbirinden farklı olmasının nesne tespitindeki etkisi de böylelikle incelenmiş olacaktır.

Çalışma dört bölümden oluşmaktadır. Birinci bölümde derin öğrenme kavramı, derin öğrenmenin tarihi ve derin öğrenme mimarisi olarak adlandırılan sinir ağları sıralanarak açıklanacak ve derin öğrenmenin kullanım alanlarından kısaca değinilecektir. İkinci bölümde ise nesne tespit kavramı ve türleri, nesne tespitinde bilgisayar görüşü, görüntü işleme ve örüntü tanıma kavramı açıklanacaktır. Ayrıca nesne tespitinde yer alan sinir ağları ve yer alan kütüphaneler aktarılacaktır. Üçüncü bölümde ise İHA kavramı, kısaca tarihi, kullanım alanları nesne tanımadaki rolü anlatılacaktır. Dördüncü bölümde

ise İHA ile tespit edilen nesnelere dair takip edilecek olan adımlar anlatılacak ve yapılacak uygulama sonuçları aktarılacaktır.

1.1. Derin Öğrenme

Öğrenme, zekâ gibi, tam olarak tanımlanması zor olan çok çeşitli süreçleri kapsamaktadır. Bir sözlük tanımı, “ya da deneyime” ve “deneyime göre davranışsal bir eğilimin değiştirilmesi” gibi ifadeleri içermektedir. Makineler ile ilgili olarak ise, bir makinenin yapısını, programını veya verilerini, gelecekteki beklenen performansını artıracak şekilde değiştirdiğinde öğrenebilir olduğunu söylenmektedir (Nilson, 1998).

Makine öğrenimi yapay zekâdaki örüntü tanıma ve hesaplamalı öğrenme teorisi çalışmalarından gelişen bilgisayar bilimlerinin bir alt alanıdır. Makine öğrenimi, verilerden öğrenebilen, veriler üzerinde tahminlerde bulunabilen, algoritmaların inşasını ve çalışmasını araştıran bir alandır. Bu algoritmalar, veriye dayalı tahminler veya kararlar vermek için örnek girdilerden bir model oluşturarak çalışma sürecini ifade etmektedir (Akt. Wikipedia Guide).

Makine öğrenimi teknolojisi, modern toplumun birçok yönünü güçlendirmekte, web aramaları ve sosyal ağlarda içerik filtrelemeye, e-ticarette web siteleri önerilerden kameralar ve akıllı telefonlar gibi tüketici ürünlerine kadar yaygın olarak bulunmaktadır. Makine öğrenim sistemleri, görüntülerdeki nesneleri tespit etmek, konuşmayı metne dönüştürmek, haber öğelerini, yayınları veya ürünlerini kullanıcıların ilgi alanlarıyla eşleştirmek ve ilgili arama sonuçlarını seçmek için kullanılmaktadır. Giderek, bu uygulamalar derin öğrenme adı verilen bir teknik sınıfından yararlanmaktadır (LeCun, 2015).

Derin öğrenme insan beynin yapısı ve işlevinden esinlenerek oluşturduğu algoritmalar makine öğreniminin alt kümesidir (Brownlee, 2017). Bir başka tanıma göre ise işlenmemiş verilerle beslenen derin öğrenme verilerin aynı sınıflandırmasında olanları algılayan makine öğrenimi teknikleridir. Derin öğrenme sınıflandırma için kullanılacak katmanların tasarlanmasında mühendislerin rol oynamadığı bir öğrenme yöntemi kullanmak suretiyle öğrenmesidir (Lecun ve ark., 2015). Bir başka tanıma göre ise insan beyninin kabiliyetlerini tekrarlayan büyük verilerden özellikler çıkartan kontrollü ve kontrolsüz olarak çalışan, sınıflandırma yapabilen sıralı öğrenme teknikleridir (Kayaalp ve Süzen 2018).

Derin öğrenmede büyük veriler kullanılmaktadır. Derin öğrenmenin avantajı elde ettiği verilerin temsil etmede kullandığı özellik vektör ve kenar kümelerini denetimli veya denetimsiz şekilde öğrenecek şekilde algoritmalar kullanılmasıdır (Kayaalp ve Süzen 2018).

1.2. Derin Öğrenmenin Tarihi Süreci

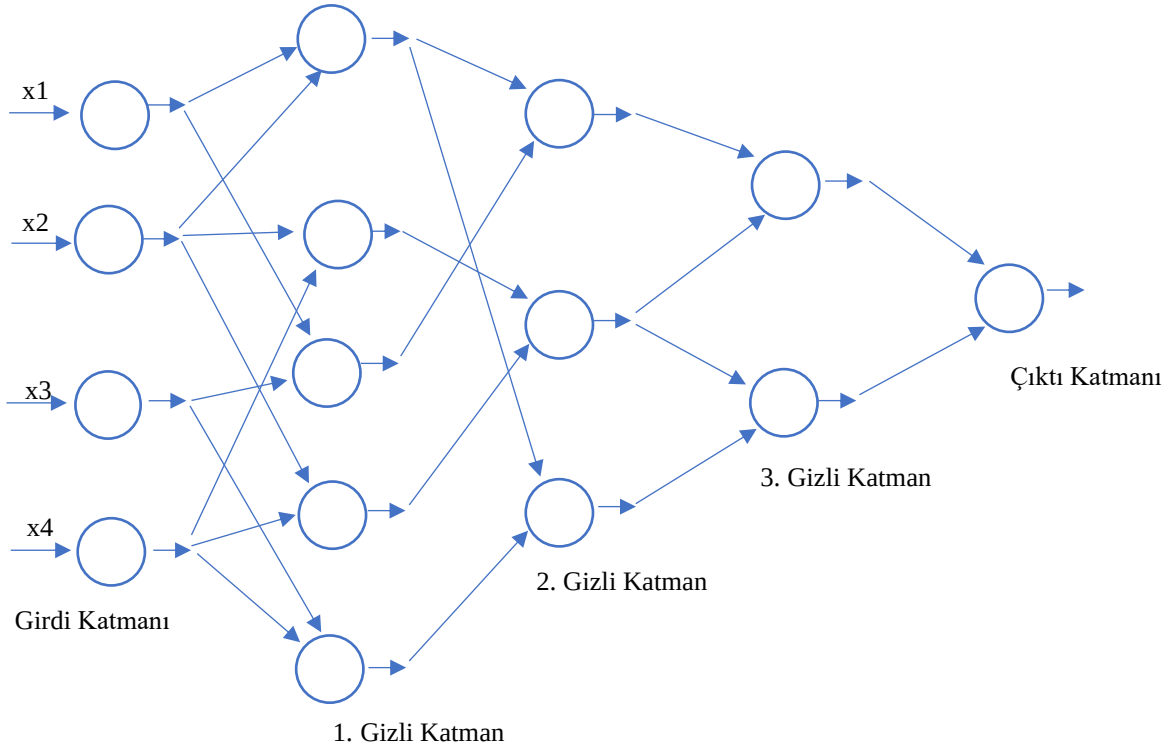
Makine öğrenmesi 2006'dan bugüne kadarki olan süreçte derin bir şekilde yapılandırılmış ya da genel olarak derin denilen öğrenme olarak ifade edilen hiyerarşik öğrenme biçiminde ortaya çıkmıştır (Deng Y, 2013).

Son yıllarda derin öğrenme teknolojisini birçok devlet, firma ve bilim dünyası yararlanmaktadır. Derin öğrenme insana ihtiyaç duyulmadan devasa miktardaki verilerden yararlanarak pek çok şeyi öğrenebilmekte, yapay zekâ alanı ve bilgisayarlar algoritmalarını insan beyninin deneyimlerinden öğrenmektedir. Derin öğrenme mimarisi bir vazifeyi her defasında girilen verileri kullanarak ve sonucunu geliştirerek yapmaktadır (Süberk, 2019).

Derin öğrenme tarihi Warren McCulloch ve Walter Pitts tarafından 1943 yılında oluşturulan hesaplama modeline dayanmaktadır. Düşünce sürecini taklit etmek için oluşturdukları algoritmalarla bu modeli oluşturmuşlardır (Akt. Kayaalp ve Süzen 2018).

Frank Rosenblatt tarafından 1958 yılında perceptron adlı model geliştirmiştir. Modelde kullanılan bilgisayarda yapay sinir ağı kontrollü ve öğretimli bir desen tanıma algoritması oluşturulmuştur. İstenilen verilerden öğrenmek istenilen bilgiyi elde etme aşamasında Rosenblatt tarafından geliştirilen iki katmanlı sinir ağları yetersiz olmuştur. Yapay öğrenme için oldukça yeni bir teknoloji olan derin öğrenme çok katmanlı sinir ağlarını kullanarak istenilen bilgiyi elde edilen verilerden ayırmayı sağlamaktadır. Derin öğrenmede elde etmek istenilen bilgi katman katman daha önceden eğitilmesi önemli bir üstünlük olarak karşımıza çıkmaktadır (Akt. Kayaalp ve Süzen 2018).

1965 yılında Ivakhnenko ve Lapa tarafından çok katmanlı algılayıcılar için derin öğrenme mimarileri geliştirilmiştir. Katmanlar arasında gerekli yöntemler özel teknikler seçilerek uçtan uca eğitilmiş ve bunun için en küçük kareler yönteminden yararlanılmıştır (Akt. Balık ve ark., 2017).



Şekil 1.1. Bilinen ilk derin ağ mimarisi (Ivakhnenko ve Lapa, 1966).

Bilinen ilk derin ağ mimarisi kendi kendini düzenleyen cognitron modelinin bir uzantısı olan neocognitron denetimsiz öğrenme yeteneğine sahiptir. Neocognitron çok katmanlıdır. Sınıflandırma aşamasında “öğretmene” gerek olmamakta ve sadece ağın giriş katmanına uyarıcı model koymak yeterli hale gelmektedir (Fukushima, 1980).

1993 yılında Jurgen Schmidhuber yaklaşık 1000 katmanın bir araya gelmesiyle oluşan devirli sinir ağı kullanmış ve bu ağ derin öğrenme alanında önemli bir yenilik olmuştur. 1997 yılında uzun-kısa dönem hafıza (LSTM) modeli ilk defa ortaya çıkmıştır. 1998 yılında geri yayımlı “öğrenme ile birleştirilen gradyan” temelli öğrenme modeli ortaya çıkmıştır. 2009-2011 döneminde ise derin öğrenme alanında bir üst seviyeye geçilmiş ve bu dönemde resim algılama ve kıvrımlı sinir ağları kullanılmıştır (Fogg, 2018).

2009 tarihinde Stanford Üniversitesi'nde profesör ve Yapay Zekâ Laboratuvarının başkanı olan Fei-Fei Li 2017 itibarıyla ImageNet'i başlatmıştır. ImageNet hem eğitimcilere hem de öğrencilere 14 milyondan fazla (14.197.122) görüntüleri etiketleyen veri tabanı sunmuştur. Alex Krizhevsky, 2011-2012 döneminde AlexNet'i yaratmıştır. AlexNet evrimsel sinir ağıdır. AlexNet makine ve derin öğrenme alanında uluslararası yarışmalar kazanmıştır. AlexNet LeNet5'i geliştirmiş ve ilk başlarda sadece 8 katmandan

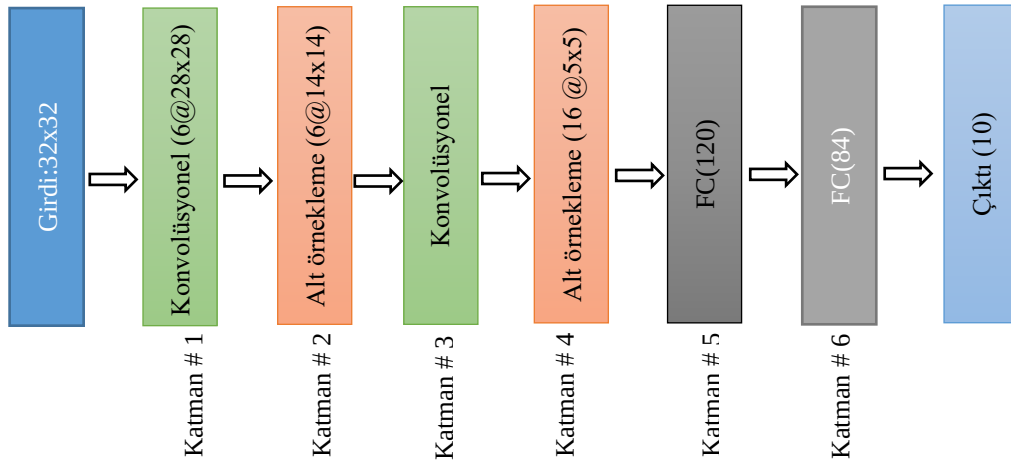
oluşmuş ve sonrasında bağlı katmanları düzelterek hızlandırılmasını sağlamaktadır (Akademi 4,0 Ekibi, 2019).

1.3. Derin Öğrenme Mimarileri

1.3.1. Konvolüsyonel sinir ağları

CNN sahip olduğu birden fazla gizli katman sayesinde görüntülerde bulunan nesnelerin niteliklerine göre ayırt edilmesini sağlayan sinir ağı olarak ifade edilmektedir. CNN özel bir mimari ile görüntüleri sınıflandırmayı başarmakta ve bunu sınıflandırmayı daha hızlı şekilde yapabilmektedir (Nielsen, 2015).

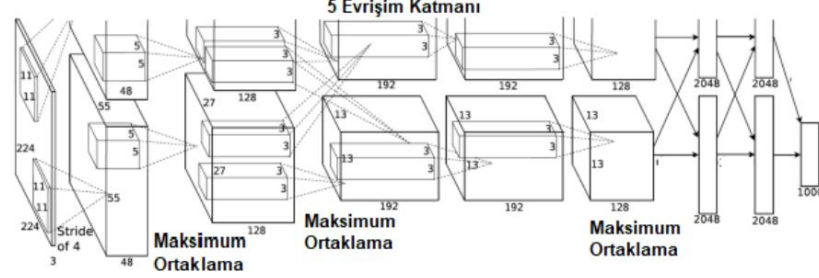
Le Cun ve ark. (1998) başarılı bir gradyan tabanlı öğrenme tekniğinin geri yayılım algoritması ile eğitilmiş çok katmanlı olan CNN'i kullanarak el yazısı tanıma işlemine geri yayılım ağlarını uygulamasını LeNet mimarisi ile birlikte uygulamışlardır.



Şekil 1.2. LeNet mimarisi (Alom ve ark., 2018)

Krizhevesky, Sutskever ve Hinton (2012) milyonlarca görüntüden binlerce nesne sınıflandırmak için büyük bir öğrenme kapasitesine duyulan ihtiyaç ve nesne tanıma görevinin büyük karmaşıklığı sorunu nedeniyle evrimsel sinir ağı modeli olan AlexNet'ten yararlanmışlardır. Kapasiteleri, derinliklerini ve genişliklerini değiştirerek kontrol edilebilir olması ve ayrıca görüntülerin doğası hakkında güçlü ve çoğunlukla doğru varsayımlar yapılabilmesi en büyük avantajıdır. Bu özellikleri sayesinde benzer boyutta katmanlara sahip standart ileri beslemeli sinir ağlarına kıyasla CNN'lerin çok daha az bağlantısı ve parametresi bulunması sebebiyle eğitilmesi daha kolaydır. AlexNet, ImageNet LSVRC-2010 yarışmasında 1,2 milyon yüksek çözünürlüklü görüntüyü 1000

farklı sınıfta sınıflandırmak için kullandığı geniş ve derin evrişimli sinir ağı ile hata oranı %37,5 iken, LSVRC-2012 yarışmasında hata oranı %16,4'e düşmesini sağlamış ve söz konusu yarışmada birincilik ödülünü kazanmıştır.



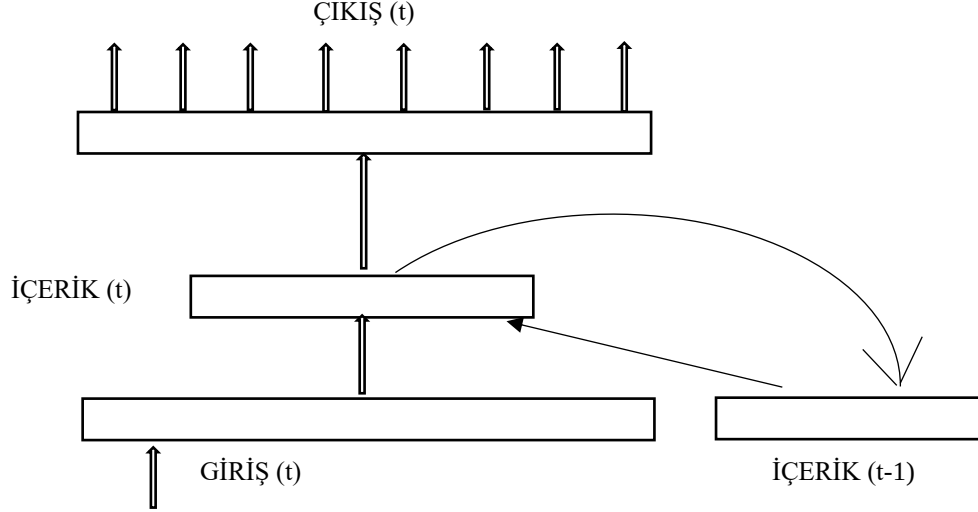
Şekil 1.3. AlexNet mimarisi

1.3.2. Tekrarlayan sinir ağları

Elman (1990) tarafından tekrarlayan sinir ağlarını (RNN) önerilmiştir. RNN de simülasyonun sinyalde öğrenilmesi gereken dilsel birimlerin sınırlarına bir ipucu olarak hizmet edebilecek bilginin olduğu ve basit RNN bu bilgileri çıkarma yeteneğine sahip olduğunu göstermektedir. Elman, ağı birkaç önemli kelime üzerinde kategorize etmiştir. Kategori fiiller ve isimler olarak ikiye ayrılmaktadır. Fiil kategorisini, doğrudan nesne gerektiren veya doğrudan nesnenin isteğe bağlı olduğu gruplara göre ayrılmıştır. İsim kategorisi ise iki ana gruba ayrılmaktadır. İnsan ve insan olmayan olmak üzere sınıflandırmaktadır. İnsan olmayanlar büyük hayvanlar ve küçük hayvanlar olarak ayrılmıştır. Ağ, kelimelerin kesin sırasını tahmin edememekte, ancak bu toplulukta, girdi olarak ifade edilen yani isimleri takip eden bir girdi sınıfı olarak ifade edilen fiiller olduğunu kabul etmektedirler.

RNN, gizli katman çıkışını yeniden gizli katmana giriş olarak yollayabilmeyi başaran ve ard arda gelen bilgilerden yararlanan derin öğrenme algoritması olarak ifade edilmektedir (Kayaalp ve Süzen, 2018).

RNN özünde sıralı halde bulunan bilgilerden yararlanmaktadır. Görüntü olarak kullanılan verilerde girdi veya çıktıların sıralı olmasının bir önemi bulunmamasına rağmen doğal dil işleme (NLP) gibi alanlarda sıralı olmasının önemi bulunmaktadır. Bir cümle de yer alan kelimenin öncesi ve sonrasında hangi kelimenin yer aldığına göre tahmin edilmesi RNN örnek olarak gösterilebilmektedir (Şeker ve ark. 2017). Şekil 1.4'te basit bir RNN gösterilmektedir.

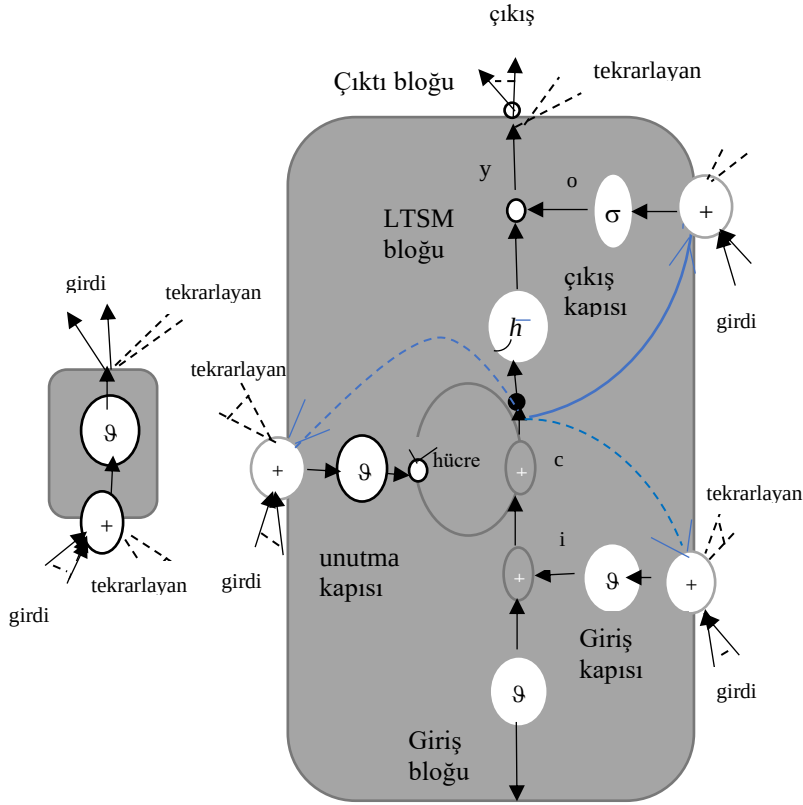


Şekil 1.4. Basit tekrarlayan sinir ağı

RNN kullanım bölgelerinde istenilen bilgiyi elde etmede başarısız olmazsa bu sorun için çok fazla algoritmayı hep birlikte kullanılarak çözüm sağlayabilmektedir (Kayaalp ve Süzen, 2018). Örnek verecek olursak cümle içinde bulunan kelime dizilimlerini hesaplamak, sınırsız etkileşimleri sağlamak ve sadeleştirmek, bağımlılık ağaçlarının hesaplamak için birden fazla algoritmalar (çift yönlü RNN) kullanılmaktadır (Karpathy ve Fei-Fei, 2015).

1.3.3. Uzun kısa süreli hafıza ağı

Hochreiter ve Schmidhuber, (1997) tarafından ifade edilen LSTM, RNN için önceki öğrenme algoritmaları tarafından çözülemeyen çok sayıda görevi çözebilmeyi amaçlamaktadır. LSTM uygun bir gradyan tabanlı öğrenme algoritması ile birlikte yeni bir RNN olarak ve hata geri akış problemlerinin üstesinden gelmek için tasarlanmıştır.



Şekil 1.5. RNN’de kullanılan SRN birimi (solda) ve LSTM (sağdaki) ayrıntılı şeması

LSTM mimarisinde giriş, unutma, çıkış ve hücre olmak üzere 4 tane katman bulunmaktadır. LSTM mimarisinde bulunan ve kapı olarak ifade edilen unsurlar sayesinde mimarinin anahtarı konumundaki hücreler bilgi tutabilmekte ya da bilgiyi silbilmektedir. Mimaride yer alan diğer katmanlar yani giriş, çıkış ve unutma katmanları hücre durumunda söz konusu ağı hafızasını meydana getirmektedir (Pervan, 2019).

En bilinen LSTM mimarisini (vanilya LSTM) ve bu mimarilerin 8 farklı modelini üç ana sorunda incelendiğinde bu sorunlar; akustik modelleme, el yazısı tanıma ve polifonik müzik modellemesi olarak sıralanmaktadır. (Greff ve ark., 2015). LSTM, müzik besteleme alanında kullanılmaktadır. Bu alanda devamlı hata akışını sağlayarak yok olma gradyanlarını imha ederek veride yer alan uzun vadeli bağımlılıkları bulabilmeyi sağlamaktadır. Bu sayede bir takım ritmik zamanlama ve sayma vazifelerini LSTM sayesinde çözüme kavuşmaktadır (Eck ve Schmidhuber, 2002).

Vanilya LSTM'den bulunan tek tek her varyant, tek bir farklılıkta değişiklik göstermektedir. Mimarinin performansının etkileyen her bir farklılığı LSTM yalıtmayı sağlamaktadır. Ayrıca tüm kapıların tekrarlanan girişlere tüm kapılardan tekrarlanan girdiler almaktadır (Greff ve ark., 2015).

1.3.4. Kısıtlı Boltzman makineleri

Kısıtlı Boltzmann makinesi, görünür birimler katmanından ve görünür-görünür ya da gizli-gizli bağlantılara sahip olmayan gizli birimler katmanından oluşmaktadır. Bir gizli katmanı öğrendikten sonra, gerçek birimler tarafından yönlendirildiklerinde gizli birimlerin aktivite vektörleri, başka bir Sınırlı Boltzmann Makinesinin (RBM) eğitimi için “veri” olarak ele alınmaktadır. Bu, istendiği kadar gizli katmanı öğrenmek için tekrarlanmaktadır. Bu şekilde birden fazla gizli katmanı öğrendikten sonra, tüm ağ tek birçok katmanlı üretken model olarak görülebilmekte ve her bir ek gizli katman, çok katmanlı modelin eğitim verilerini üretme olasılığı üzerinde bir alt sınır geliştirmektedir (Hinton, 2014; Hinton ve Salakhutdinov, 2006).

Her defasında bir gizli katmanı öğrenmek, birçok gizli katmanı ve milyonlarca ağırlığı olan derin sinir ağlarını öğrenmenin önemli bir yoludur. Öğrenme gözetimsiz olmasına rağmen, en ilerili seviyeli özellikler genellikle ham veri vektörlerinden çok daha fazla sınıflandırmak yararlıdır. Bu derin ağlar, geri yayılım algoritması faydalanan kategorize etme ya da boyut küçültmede daha etkili biçimde ayarlanmaktadır (Akt. Hinton, 2014).

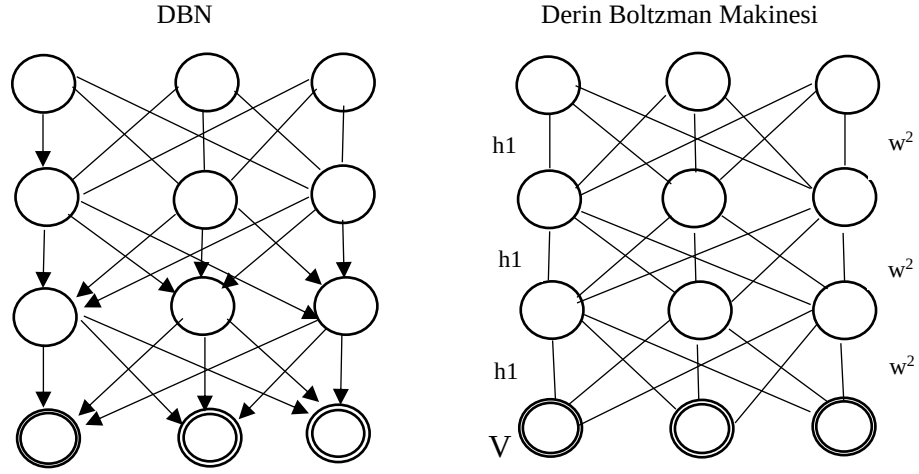
1.3.5. Derin inanç ağı

Derin İnanç Ağı (DBN) yapı taşı, bir seferde bir özellik katmanını keşfetmek için kullanılan RBM adı verilen olasılıklı bir modeldir. Bir DBN öğrenmek için, RBM'ler yığındaki bir sonraki RBM'yi eğitmek için veri görevi gören bir RBM tarafından üretilen özellik aktivasyonları ile tekrarlı olarak uygulanmaktadır (Akt. Sarıkaya ve ark. 2011).

RBM tek başına temsil edebileceği şeyler sınırlıdır. RBM'ler, birçok katmandan oluşan üretken bir model olan DBN oluşturmak için bir araya getirildiğinde gerçek gücü ortaya çıkmaktadır. Bir DBN'de, her katman bir grup ikili veya gerçek değerli birimler içermektedir. İki bitişik katman, aralarında tam bir bağlantı kümesine sahiptir, ancak aynı katmandaki iki ünite bağlı değildir (Lee ve ark., 2009).

Hinton ve ark. (2006) derin öğrenmeyi, DBN sayesinde her seferinde bir katmanın öğrenmenin mümkün olması olarak tanımlamışlardır. Bunu yapmanın en basit yolu, alt katmanları öğrenirken daha ileri katmanların bulunmadığını varsaymaktır. Her bir katman öğrenildikten sonra, ağırlıkları daha yüksek katmanlardaki ağırlıklardan

çözölmektedir. Daha ileri seviyeli ağırlıklar farklılaştıkça, alt katmanlar için öncelikler tamamlayıcı olmaktan çıkmaktadır.



Şekil 1.6. DBN ve RBM mimarilerinin karşılaştırılması

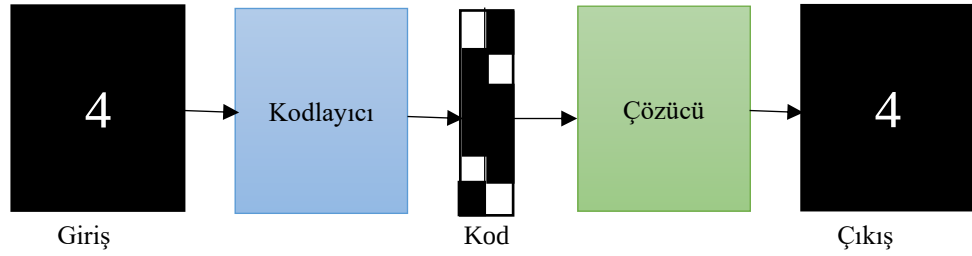
Şekil 1.6’da DBN ve RBM mimarilerinin karşılaştırılması yer almaktadır. DBM giriş katmanı ve çıkış katmanı olan yeni bir olasılıksal ileri besleme tahmin aracıdır. RBM de modelinin arkasındaki temel fikir için, sinir ağındaki ağırlıkların başlangıç değerlerini önceden eğitmek için katman katman denetimsiz öğrenme yöntemi kullanmaktır. Katman katman denetimsiz eğitim prosedürü, her katmanın bir öncekinin özelliklerini yakaladığını bir sonrakine aktarmaktadır. Her bir katman RBM kullanılarak önceden eğitilmiştir. Derin birbirine bağlı 2 farklı katmandan oluşmaktadır. Bir katmanın görünür düğümleri/nöronları ve diğer gizli düğümleri/nöronları bulunmaktadır. Her katmandaki düğümlerin aralarında sadece diğer katmanların birimleriyle bağlantısı yoktur. Tüm bu bağlantılar simetrik ve çift yönlüdür (Karathanasopoulos, 2017).

1.3.6. Derin oto-kodlayıcılar

Derin otomatik kodlayıcılar, istenen çıktı verisinin girdinin kendisi olduğu ileri beslemeli çok katmanlı bir sinir ağıdır. İlk bakışta, kimlik haritalamasında yeniden yapılandırma hatası olmayacağından bu işlem önemsiz görünebilmektedir. Bununla birlikte, kimlik haritasının ya bir tür düzenlileştirme yoluyla ya da daha da önemlisi mevcut türetme için, girdi verilerinin düşük boyutlu, doğrusal olmayan bir temsili olan gizli katmanlara sahip olması durumunda izin verilmediğinde otomatik kodlayıcılar önemsiz hale gelmektedir (Zhou ve Paffenrot, 2017).

Derin oto kodlayıcılar katman katman şeklinde öğrenme algoritması için çok etkili tahmin etme yoludur. Her bir özellik katmanı, bir alt katmandaki birimlerin faaliyetleri arasında güçlü, yüksek dereceli korelasyonları yakalamaktadır. Çok çeşitli veri setleri için bu, düşük boyutlu ve doğrusal olmayan yapıyı aşamalı olarak ortaya çıkarmanın etkili bir yolu olarak ortaya çıkarmaktadır (Hinton ve Salakhutdinov, 2006).

Otomatik kodlayıcılar, girdi verilerinin sıkıştırılmış bir sunumundan en iyi özellikleri öğrenmeyi amaçlamaktadır. İleriye dönük kendi ağ girişini çıkarmaya çalışan ve bu sebeple düğüm sayısı aynı giriş ve çıkış sayısına sahip bir sinir ağıdır. Gizli katmandaki düğüm sayısı fazla olması durumunda otomatik kodlayıcı bir giriş verisi sıkıştırılmış olarak göstermektedir (Şeker ve Yüksek, 2017).



Şekil 1.7. Derin oto-kodlayıcı algoritma şeması

Derin otomatik kodlayıcıların yapısı başlangıçta bir sinir ağı içindeki verilerin boyutsallığını azaltmayı amaçlamıştır. Şekil 1.7’de gösterildiği gibi, kod katmanının türetilmesindeki amaç gizli katmandan daha iyi performans gösterdiği için çok katmanlı bir kodlayıcı ve kod çözücü ağ yapısıdır (Chen ve Huang, 2019).

Derin otomatik kodlayıcılarda girdi ve çıktı da kullanılan veriler arasında benzerliğin çok olması durumunda derin otomatik kodlayıcı algoritmalarının başarılı olduğunu söylenmektedir.

1.4. Derin Öğrenme Kullanım Alanları

Bir önceki bölümlerde ifade edilen mimarilerde özellikle başarılı olunan ağlar incelenmiştir. Bu bölümde son yıllarda derin öğrenmenin kullanım alanlarından anlatılmaktadır. Derin öğrenmenin kullanım alanları olan NLP ve biyomedikal görüntüleme aşağıda başlıklarda detaylı olarak ifade edilecektir.

1.4.1. Doğal dil işleme

NLP işleme günümüzde yaygın olarak tartışılmış ve araştırılmıştır. Makine öğreniminin en eski araştırma alanlarından biri olduğu için, makine çevirisi konuşma, tanıma ve metin işleme gibi önemli alanlarda kullanılmaktadır. NLP, hesaplama ve yapay zekâ alanında büyük bir atılım getirmiştir (Jain ve ark., 2018).

NLP, konuşma ve metin gibi doğal dilin yazılım tarafından otomatik olarak manipüle edilmesi olarak ifade edilmektedir. NLP alanı 50 seneden daha fazla bir sürede var olmakta ve bilgisayarlar teknolojisinin ilerlemesiyle dilbilim uzmanlık alanından uzaklaşmıştır (Brownlee, 2017).

NLP, bilgisayarların faydalı işleri gerçekleştirmek amacıyla doğal dil metnini konuşmayı anlamak ve değiştirmek için nasıl yararlanacağını inceleyen araştırma ve uygulama alanıdır. NLP araştırmacıları, insanoğlunun dili nasıl anladığı ve kullandıkları hakkında bilgi toplamayı amaçlamaktadır. Böylece bilgisayar teknolojisinden beklenen vazifeleri yapmak için doğal dilleri kavramasını ve manipülasyonu sağlayacak gerekli araçlar ve teknikler geliştirilebilmektedir. NLP'nin temelleri bilgisayar ve bilgi bilimleri, dilbilim, matematik, elektrik ve elektronik mühendisliği, yapay zekâ ve robotik, psikoloji vb. bir dizi disiplin içindedir. NLP uygulamaları ise makine çevirisi, doğal dil metin işleme ve özetleme, kullanıcı ara yüzleri, çok dilli gibi bir takım çalışma alanını kapsamaktadır (Chowdhury, 2003).

NLP araştırmaları daha fazla yeni derin öğrenme yöntemlerinin kullanımına yoğunlaşmaktadır. On yıllardır NLP sorunlarını amaçlayan makine öğrenme yaklaşımları, çok yüksek boyutlu ve seyrek özellikler konusunda eğitilmiş sığ modellere dayanmaktadır. Son zamanlarda, yoğun vektör temsillerine dayanan sinir ağları çeşitli NLP görevlerinde başarılı sonuçlar vermektedir. Bu meyil, kelime düğümlerinin ve derin öğrenme yöntemlerinin başarısı ile desteklenmektedir. Derin öğrenme çok seviyeli otomatik özellik temsili öğrenmesini sağlamaktadır. Aksine, geleneksel makine öğrenimi tabanlı NLP sistemleri, el yapımı özelliklerle büyük ölçüde alakalıdır. Bu tür el yapımı özellikler zaman alıcıdır ve genellikle noksanıdır (Akt. Young ve ark., 2018).

Derin öğrenme yöntemleri, verilerin hiyerarşik temsillerini öğrenmek için birden fazla işlem katmanından yararlanmakta ve değişik uygulama alanlarında değerlendirilmektedir. Son birkaç yılda NLP alanında birçok model tasarımları ve yöntemleri meydana gelmiştir (Akt. Young ve ark., 2018). NLP alanında derin öğrenme,

daha fazla veri gerektirebilecek, ancak eğitilmesi ve işletilmesi için daha az dil uzmanlığı gerektiren modellerle daha iyi performans göstermektedir (Brownlee, 2017).

NLP için kullanılan çeşitli algoritmalar temel olarak RNN'ye bağlıdır (Jain ve ark., 2018). NLP de kelime tespiti ile ilgili yaklaşımlarda, anahtar kelime olanlar ve olmayanlar arasında eşdeğer anahtar kelimeler arasında ayırım yapılmasına imkân vermektedir. Sözkonusu ayırım için RNN algoritması kullanılmaktadır. RNN konuşma sinyalinde bir anahtar kelime nin olup olmasına bakılmaksızın tahmin yapılmasına olanak sağlamaktadır (Fernández ve ak., 2007).

Video, ses, metin gibi çeşitli verilerin işlenmesinde derin öğrenme metotları oldukça etkilidir. NLP alanında kullanılan bir yöntem video ve ses işleme alanında başarılı olamazken daha farklı bir yöntem video ve ses işlemede başarı elde edebilmektedir (Akt. Küçük ve Arıcı, 2018):

Aşağıda NLP sorunlarının ortaya çıkmasıyla sözkonusu sorunları çözebilmek için yararlanılan derin öğrenme algoritmaları yer almaktadır (Küçük ve Arıcı, 2018).

- **Metin sınıflandırma:** RNN, CNN,
- **Metin ayrıştırma:** CNN,
- **Duygu analizi:** Derin otomatik kodlayıcılar,
- **Bilgi çıkarımı:** Derin sinir ağları,
- **Varlık ismi tanıma:** CNN ve LSTM,
- **Zamansal ilişki çıkarımı:** CNN,
- **Olay çıkarımı:** CNN,
- **Sözcük türü ekleme:** Derin sinir ağları ve LSTM,
- **Metin sıralama:** CNN,
- **Otomatik harf çevirisi:** DBN'dir.

NLP alanında bugün dünyada en fazla kullanılan yöntemlerin başında 1994 yılında Cavnar, W. B. ve J. M. Trenkle tarafından geliştirilen N-GRAM yöntemi bulunmaktadır. Markov modeliyle yani bir durumun belirli bir istatistiksel değere göre ayırt edilmesi NLP alanında kullanılmasına olanak sağlamıştır. Ayrıca NLP alanında Ted Dunning, Compression Based Approach (PPM-Teahon) gibi yöntemler kullanılmaktadır (Tarcan ve Çakar, 2008).

1.4.2. Biyomedikal işleme

Tıbbi görüntüler göğüs röntgeni kadar basit veya fonksiyonel manyetik rezonans görüntüleme (MRI) gibi yaygın olmayan beyin görüntüleme kadar oldukça çok geniştir. Tıbbi görüntüleme, tüm radyasyon formlarının doku ile etkileşimi ve ilgili bilgilerin klinik olarak çıkarılmasıyla ilgili olsa da analizi, görüntülerden anatomik ve fizyolojik parametrelerin, görüntü işleme ve hareket ve görüntü dizilerinden değişiklik tespiti ile ilgilidir (Theis ve Meyer-Base, 2010).

X-ışınlarının keşfi Wilhelm Conrad Rontgen tarafından, tıbbi görüntülemenin ilk dönemidir. Wilhelm Conrad Rontgen, 1895 senesinde Hittorf- Crookes tüpüyle deney yaptığı esnada dalga boyu aralığında elektromanyetik radyasyon oluşumunu keşfetmiştir. Daha sonra William Coolidge tarafından, X-ışınları köklü anatomi ve tümörlerin daha iyi görüntülenmesini başaran Coolidge tüpünün icadı ile ilerleme sağlanmıştır. Coolidge tarafından Coolidge tüpünün tungsten filaman ile kullanılması, radyoloji alanındaki X-ışınlarının en büyük dönüm noktalarından biridir. 1946'da yoğunlaşmış bir maddede nükleer manyetik rezonans (NMR), Felix Bloch ve Edward Purcell tarafından keşfedilmiş ve MRI keşfinin ilk adımı olmuştur (Santhi, 2016).

NMR'nin keşfi ile Raymond Vahan Damadian tarafından 1969 yılında ilk MRI vücut tarayıcısı geliştirilmiştir. NMR sayesinde tümörün normal hücrelerden ayırt edilebileceğini bulmuştur. Godfrey Hounsfield, X-ışınları sayesinde nesnelerin farklı görüş açılarından dilimler şeklinde nesne yaratma fikrini ileri sürmüştür. İlk olarak bilgisayarlı tomografik (CT) tarayıcısını üretmiş ve insan beyninin ilk CT görüntüsünü almıştır. İlk MRI vücut tarayıcısı insanlar üzerinde 1977'de ortaya çıkmıştır. Matematik ve bilgisayar algoritmalarının yardımıyla dijital ve iletişim sistemindeki teknolojik genişleme ile yeni görüntüleme teknikleri geliştirilmiştir. Bu durum biyomedikal görüntülemeyi fizikçi, biyolog, matematikçi, farmakolog ve biyolog ile iş birliği yapan disiplinler arası bir alan olarak şekillendirmiştir (Santhi, 2016).

2. KAYNAK ARAŞTIRMASI

2.1. Nesne Tespit Kavramı ve Tanıma Türleri

Modern dünya büyük dijital görsel veri çemberiyle çevrilidir. Bu görsel veri okyanusunu incelemek, yorumlamak ve düzenlemek görüntü analiz teknikleri için oldukça çok önemlidir. Görüntülerin ya da videoların anlamsal kavramını otomatik şekilde analiz yapabilen yöntemler yararlı olmaktadır. Görüntülerde var olan içeriğin önemli bir unsuru görüntüde yer alan nesnelerdir. Nesne tespit tekniklerine ihtiyaç bulunmaktadır. Görüntü işleme ve bilgisayar görme alanında nesne tanıma ana vazifelerinden biridir. Özetle nesne tespiti bilinen bir etiket kümesinde yer alan görüntüde tespit edilen bir nesnenin kimliğinin belirlenmesi ile ilgilidir. Bireyler gerçek dünyada bulunan herhangi bir nesneyi emek harcamadan kolayca tanıyabilmekte iken makineler aksine tek başına nesneleri tanıyamamaktadırlar. Tanıma görevinin algoritmik açıklamaları ile makinelerde uygulanmakta ve bu durum oldukça karmaşık bir görev haline gelmektedir. Bu nedenle karmaşık olmayan ve verimliliği yüksek nesne tespit tekniklerinin geliştirilmesi gerekmektedir (Khurana ve Awasthi, 2013).

Gerçek dünya, hepsi birbirini tıkayan ve farklı pozlarda görünen nesnelerin karmakarışıklığından oluşmaktadır. Böylece makineler tanıma işleminde zorlanmaktadır. Bu tanıma problemi ne arandığını biliyorsak, hızlı bir şekilde tarama ve görüntülemenin belirlenmesini gerektiren bir nesne tespiti yapılmasını gerektirmektedir. En zorlu tanıma hayvanlar veya mobilyalar gibi çok çeşitli sınıfların örneklerini tanımayı içerebilen genel kategori tanımadır (Szeliski, 2010).

Elde edilen verilerin çeşitli özelliklerini dikkate alarak kendisine en yakın özellikleri içeren sınıfa dâhil edilmesine nesne tanıma denilmektedir. Nesne tanıma yaptığı işlem bakımdan nesnelerin sınıflandırılmasıdır. Günümüzde bilgisayar görüşünün gelişmesiyle fiziksel ve zihinsel işlemlerden alınan verilerin tanılandırılması ile nesne tanıma uygulamaları insan yüzü, parmak izi vb. alanlarda gelişmiştir. Aynı zamanda görüntü işleminin payı nesne tanımada fazlasıyla mevcuttur (Türkoğlu, 1996).

Nesne tanıma veri tabanında bulunan nesne modelleriyle nesne görüntüsünün eşleştirilmesiyle bu işlem gerçekleşmektedir. Önceden hazırlanan veri tabanı tanıma işlemi yapılacak her nesneye ilişkin verileri bulunmaktadır. Nesne türüne göre farklı tanımlama türleri bulunmaktadır (Türkoğlu, 1996).

2.1.1. Yüz tanıma

Yüz tanıma, görüntü işleme ve bilgisayarlı görmede uygulama alanlarından birini oluşturmaktadır. Bireylerin yüzleri tanıma yeteneğine denk otomatik bir sistem kurmak oldukça zordur. Bireyler tanıdık yüzleri tanımlamakta oldukça başarılı iken çok sayıda tanıdık olamayan yüzle uğraşmak konusunda yetenekli olduğu söylemek mümkün değildir (Marques, 2010).

Yüz tanıma sistemleri, yüz görüntü işleme alanının önemli bir parçasıdır ve son yıllarda uygulama alanı olarak önemi artmaktadır. Yüz tanıma sistemi, görüntü analizlerinde yüz tanıma ve çıkarma tekniklerinin bileşkesidir. Algılama uygulaması, belirli bir görüntüdeki yüzlerin konumunu bulmak için faydalıdır. Tanıma algoritması, bilgisayar görme alanının çoğunda oldukça fazla kullanılan, yapısal niteliklere sahip ve belirli görüntüleri kategorize etmek için yararlanılan sistemlerdir. Tanıma uygulamaları standart görüntülerinden yararlanmakta ve algılama algoritmaları yüzleri ve yüzün detayı olan gözleri, kaşları, burun ve ağzı tespit edebilmektedir (Gürel ve Erden, 2012).

Yüz tanıma sisteminin ilk basamağı bir kameradan görüntü olarak adlandırılan veriyi elde etmektir. İkinci basamak, sağlanan veriden yüzü tespit etmektir. Üçüncü basamak ise, tanıma kısmının bir sonucu olarak kişinin kimliğini tespit etmektir (Gürel ve Erden, 2012).



Şekil 2.1. Yüz tanıma ile ilgili çalışma (Szeliski, 2010)

Yüz tanıma, geçen yüzyılın altmışlı ve yetmişli yıllarına kadar uzanmakta ve onlarca yıl boyunca gelişim göstermektedir. Geleneksel yüz algılama yöntemi yüzün yapısal özelliklerine ve yüzün renk özelliklerine dayanmaktadır. Bazı geleneksel yüz tanıma amacıyla kullanılan algoritmaları yüz özelliklerini, yüzün bulunduğu görüntüden yer işaretleri ya da özellikler çıkararak tanımlamaktadır (Li ve Cha, 2019).

1970'lerin ilk yıllarında yüz tanıma 2B örüntü tanıma sorunu olarak kabul edilmiştir. Bilinen yüzleri tespit için yaralanılan kilit bölgeler arasındaki uzaklıklar misal olarak gözler veya öteki kilit bölgeler arasındaki mesafenin ölçülmesi veya yüzü oluşturan alanların farklı açıların ölçülmesi zordur. Fakat bu sebeple yüz tanıma sistemlerinin tam otomatik olması gerekmektedir (Parmar ve Mehta, 2013).

Eksiksiz bir yüz tanıma sistemi, yüz tanıma ve yüz algılama süreçlerini kapsamaktadır. Bu sebeple, yüz bölgesini yüz algılama işleminden çıkarmak ve yüzü arka plan deseninden ayırmak gerekmekte ve bu da yüz özelliklerinin sonraki çıkarımları için temel sağlamaktadır. Geleneksel yöntemle karşılaştırıldığında öğrenme algılama yöntemlerinin derinliğine dayanmakta ve yüz tanıma da doğru ve etkili bir şekilde geliştirmektedir. Yüz tanıma, görüntülerde yer alan insan yüzlerinin kimliğini tespit etmek amacıyla normalleştirilmiş yüz görüntülerinin bir özellik çıkarma ve kontrast tanımlama işlemidir (Li ve Cha, 2019).

2.1.2. Yaya algılama

Yaya algılama, nesne algılamanın standart bir örneğidir. Araç güvenliği, gözetleme ve robot teknolojisindeki doğrudan uygulamalar nedeniyle son yıllarda büyük ilgi görmüştür (Benenson ve ark., 2014).

Gerçek dünya görüntülerinde bulunan yayaları doğru şekilde algılama becerisi, video gözetimi ya da araçlarda yer alan otomatik sürücü destek sistemleri gibi farklı uygulamalar için enteresandır. Ayrıca yayalar nesne tanıma için en zorlu kategoriler arasında yer almaktadır. Yerel ve küresel görünümünde büyük bir değişkenliğe böylece sadece birkaç yerel bölge tüm kategori için gerçekten karakteristiktir. İlâveten küresel şekil, çeşitli eklemler ve bir yaya silüetini bozabilecek sırt çantaları, omuz çantaları ve el ya da alışveriş çantaları gibi çok sayıda tıkanıklık nedeniyle çok çeşitli dönüşümler geçirmektedir. Son olarak, birçok uygulamada, kısmen birbirini tıkayan ve zorluğa sebep olan aynı görüntü bölgesinde birkaç kişi bulunabilir (Leibe ve ark., 2005).

Geniş uygulama yelpazesi sebebiyle yaya tespiti bilgisayar görüşünde ciddi bir sorundur. Ana sorun, son derece güvenilir ve pratik bir yaklaşım keşfetmektir. Trafik sahneleri gibi gerçek yaşam senaryolarında yayaları tespit etmek oldukça karmaşıktır. Yayalar farklı özellikte kıyafetler giyer ya da aksesuarlar takarlar ve bu durum nesne tespitinde birtakım engellerin oluşmasına sebep olmaktadır. Bir görüntüde yer alan yayaların boyutu kamera ile mesafesi ile ilişkilidir. Mevcut en iyi performans sunan yaklaşımlar bile halen yayaları tespit etmekte istenen başarıyı sağlayamamıştır (Costea ve Nedeveschi, 2014).

2.2. Nesne Tespitte Bilgisayar Görüşü, Görüntü İşleme ve Örüntü Tanıma

2.2.1. Bilgisayar görüşü kavramı

Nesne tespitinde elde edilen görüntüler bilgisayar görüşünden faydalananarak elde edilmektedir. Elde edilen görüntüler ise görüntü işleme teknikleri yardımıyla nesne tespitinde oldukça başarılı olmaktadır.

Bilgisayar görüşü, dijital görüntülerden bilgi alan bilimsel bir alan olarak tanımlanabilmektedir. Bilgisayar görüşünün tanımlamanın bir başka ifadesi de uygulamalarıdır. Bilgisayar görüşü, görüntülerin içeriğini anlayabilen ve uygulamalar için kullanılabilir algoritmalar oluşturmaktır (Krishna, 2017).

Bilgisayar görüşü, görüntülerin otomatik olarak çıkarılmasıdır. Bilgi, 3B modeller, kamera konumu, nesne algılama ve tanıma ile görüntü içeriğini gruplama ve arama arasında bir anlam ifade edebilmektedir. Pratik bilgisayar görüşü, programlama, modelleme ve matematiğin bir karışımını içermekte ve bazen kavraması zordur (Solem, 2012).

Bilgisayar Görüşü bir fotoğraf ya da video kameradan elde edilen verinin bir karara ya da yeni bir veriye dönüştürülmesidir. Belirli bir amaca ulaşmak için yapılan bu dönüşümler bir kamera veya buna benzer bir aygıttan alınan verileri karar olarak görüntü içeriği hakkında bilgi sağlamaktır. Bir başka amacı ise renkli bir görüntüyü gri tonlamalı bir görüntüye çevirmek veya kamera hareketini bir görüntü dizisinden kaldırmaktır (Bradski ve Kaehler 2008).

İnsan beyni, görme sinyali farklı türden bilgiler aktaran birçok kanala bölmektedir. Ancak bilgisayar görüşünde gözün yaptığı işlemi bilgisayar gördüğü şeyi sadece bir sayı ızgarası olarak algılamaktadır (Bradski ve Kaehler 2008). Bilgisayarın

algıladıkları görüntüler sayısal bir ızgaraya dönüştürülmektedir. Bu dönüşümü Şekil 2.2’de görülmektedir.



Şekil 2.2. Bilgisayarın algıladığı görüntü (Bradski ve Kaehler, 2008)

Izgara içinde verilen herhangi bir sayı oldukça büyük bir gürültüye sahiptir ve bu yüzden kendi başına çok az bilgi vermektedir. Bilgisayarın görüşüyle ilgili sorun gürültüdür. Genelde istatistiksel yöntemleri kullanarak gürültüyü yok etmeye uğraşmaktadır. Bilgisayarların algıladığı görüntülerdeki bu sorunları çözmek için çeşitli görüntü işleme algoritmaları kullanılmaktadır (Bradski ve Kaehler, 2008).

Görüntüdeki gürültülerin giderilmesinin bu kadar zor olmasının nedeni, görüşün yetersiz bilgi aktarması, bilinmeyen bazı bilgileri bilinmeyen bir şekilde çözmeyi hedeflendiğinde ters bir problem oluşturmasıdır. Fakat, görsel hafızamızda bulunan dünyayı bütün karmaşıklığı ile modellemek, konuşulan sesleri ise üreten ses yolunu modellemekten çok daha zordur. Bilgisayarlı görmede kullanılan ileri modeller genellikle fizikte (radyometri, optik ve sensör tasarımı) ve bilgisayar grafikleriyle geliştirilmektedir. Her iki alan da nesnelerin nasıl hareket ettiği, canlandırdığı, ışığın yüzeyleri nasıl yansıttığı, atmosfer tarafından nasıl dağıldığı, kamera lenslerinden (veya insan gözlerinden) nasıl kırıldığı ve bir görüntü düzlemine nasıl yansıtıldığını göstermektedir. İnsanların ve hayvanların bu işlevi kusursuzca ve hiçbir emek haracamadan yapmaları emsalsizken, bilgisayar görme algoritmaları ise hataya oldukça açık ve yeteri kadar başarılı değildir (Akt. Szeliski, 2010).

Bilgisayar görüşü pikseller ve anlam arasında büyük bir boşluk olması sebebiyle modellemek zordur. Bilgisayarın 200×200 RGB görüntüde gördüğü şey 120,000 değer kümesidir. Bu rakamlardan anlamlı bilgiye giden yolu bulması çok zordur. Muhtemelen, insan beyninin görsel korteksi retinamıza yansıtılan ve nöron sinyallerine dönüştürülen görüntüleri anlama problemini zor olsa da çözmektedir. Bu sorunu çözmek için bilgisayar görüşünün iki avantajı bulunmaktadır. İlk olarak, bir algılama cihazı bir görüntüden mümkün olduğunca fazla ayrıntı yakalamaktadır. Göz, iris içinden gelen ışığı yakalayarak

özel hücrelerin beyne nöronlar aracılığıyla bilgi ileteceği retinaya yansıtmaktadır. Bir kamera da görüntüleri benzer şekilde yakalamakta ve pikselleri bilgisayara aktarmaktadır. Kameralar da kızılötesini görebildikleri, daha uzağı daha hassas ayarda görebildiği için kameralar insanlardan daha iyidir. İkinci olarak, yorumlama cihazı bilgiyi işlemek ve ondan anlam çıkarmak zorundadır. İnsan beyni bunu beynin farklı bölgelerinde birçok adımda çözmektedir. Bilgisayar görüşü hala bu alanda insan performansının gerisinde kalmaktadır (Krishna, 2017).

Kameralar her yerde ve internete yüklenen görüntü sayısı katlanarak arttırmaktadır. Instagram'da yer alan görüntüler, YouTube'da bulunan videolar, güvenlik kameralar, tıbbi ve bilimsel görüntüler bulunmaktadır. Bilgisayar görüşü çok önemlidir. Çünkü bu görüntüler arasında sıralama yapmak ve bilgisayarların içeriklerini anlamasını sağlamak gerekmektedir. Bilgisayar görüş uygulamalarının kapsamlı olmayan bir listesi olarak ifade edilmektedir (Krishna, 2017).

Bunlara örnek verecek olursak:

a) Optik Karakter Tanıma: Harfler üzerindeki el yazısı posta kodlarını ve otomatik plaka tanıma,

b) Makine Muayenesi: Kalite güvencesi için hızlı parça denetimi,

c) Perakende Satış: Otomatik ödeme şeritleri için nesne tanıma,

d) Tıbbi Görüntüleme: Ameliyat öncesi ve ameliyat sonrası görüntülerin kaydedilmesi veya insanların yaşlandıkça beyin morfolojisi ile ilgili uzun vadeli çalışmalar yapılması,

e) Otomotiv Güvenliği: Radar veya lidar gibi aktif görme tekniklerinin iyi çalışmadığı koşullar altında, sokakta yayalar gibi beklenmeyen engellerin tespit edilmesi,

f) Sürveyans: Davetsiz misafirleri izleme, karayolu trafiğini analiz etme ve boğulan kurbanlar için havuzları izlemedir.

Bilgisayar görüşü alanı nesne tespit probleminin çözümünde görüntü işleme teknikleriyle birlikte nesne tespitinde kolaylık sağlamaktadır. Görüntülerden elde edilen nesnenin içinde bulunduğu pencerelerden nesnenin tanımlanmasını nesne tespit algoritmaları kullanarak gerçekleştirmektedir.

Bilgisayarlı görüş uygulamasında ilk aşama yani görüntü işleme öncesi görüntü işleme sonrası ve görüntüyü uygun bir analize dönüştürmek için görüntü işleme kullanımı örnek olarak pozlama, düzeltme ve renk dengeleme, görüntü parazitinin azaltılması, keskinliğin artırılması veya görüntünün döndürülerek düzleştirilmesidir. Bazıları görüntü

işlemenin bilgisayar görüşünün dışında olduğunu düşünebilir, ancak bilgisayarlı fotoğrafçılık ve hatta tanıma gibi çoğu bilgisayar görme uygulaması kabul edilebilir sonuçlar elde etmek için görüntü işleme aşamalarını kullanmaya özen gösterilmektedir (Szeliski, 2010).

2.2.2. Görüntü işleme kavramı

Kendini çeşitli formlar ve şekiller, renkler ve dokular, hareket ve huzur ile gösteren görsel olarak büyüleyici bir dünya içerisindedir. İnsan algısı, dünyada oldukça fazla olan görsel bilgiyi kazanma ve kazanımları yorumlama kapasitesi bulunmaktadır. İnsanın duyuşal dünyasında bulunan hem hareketsiz görüntülerde yer alan görel bilgileri hemde grafiklere, videolara veya hareketli görüntülerde bulunan görsel bilgileri yorumlamayı sağlayan yeteneklerini bir makineye yüklemek oldukça zordur. Bu sebeple, sözkonusu görsel bilgilerin depolanması, işlenmesi, iletimi, tanınması ve yorumlanmasını kavramak önemlidir (Acharya ve Ray, 2005).

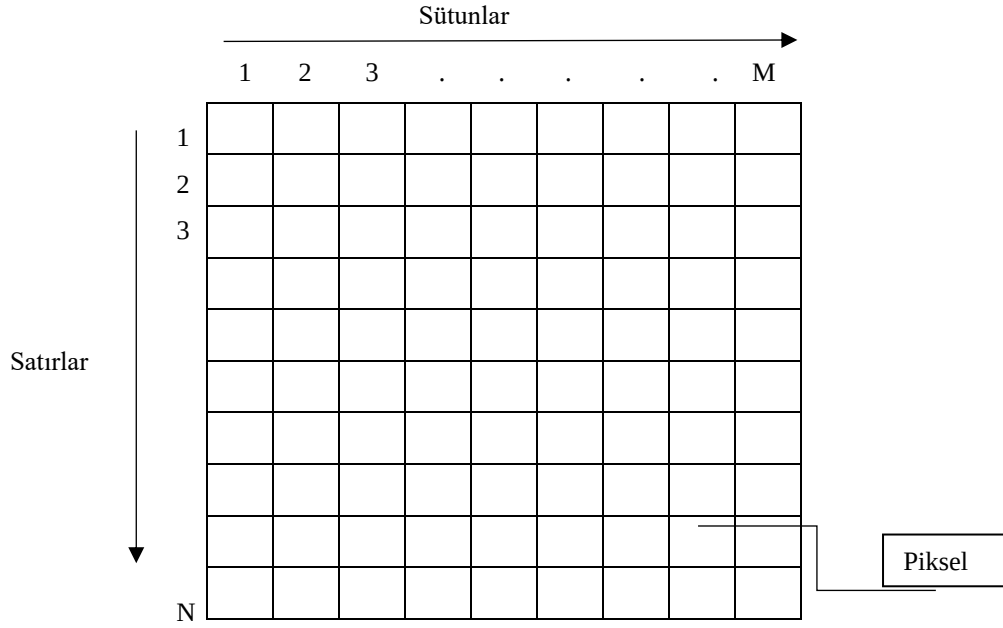
Bir görüntü, fotoğraf, resim, hatta rüya olabilmekte, ancak bilgisayar dünyasında "piksel" olarak adlandırılan noktaların bir koleksiyonudur. Böyle bir görüntü genellikle dijital görüntü olarak bilinmekte ve resmi olarak değerleri o piksel tarafından temsil edilen resim ögesindeki ışık yoğunluğunu gösteren bir piksel dizisi olarak tanımlanmaktadır (Acharya ve Ray, 2005).

Görüntü işleme terimi, bir görüntüde yer alan bilgilerin, genellikle görüntüyü görsel olarak geri yüklemek veya optimize etmek için değiştirildiği prosedürleri ifade etmektedir. Tipik örnekler, zayıf odaklanmanın neden olduğu görüntü keskinliğinin düzeltilmesi, lenslerin optik hatalarının düzeltilmesi, kontrastın düzeltilmesi, yoğunluk veya parlaklık, renk düzeltilmesi, orijinal görüntüde kolayca görülmeyen unsurları vurgulamak için görüntü yapısı geliştirme, arka plan gürültüsünün çıkarılması ve benzerleridir. Görüntü işleme, görüntüleri manuel veya otomatik olarak görüntü analizi için hazırlamaktadır (Stipaničev, 1994).

Bir görüntü işleme sisteminin çeşitli bileşenleri bulunmakta ancak en önemli ana bileşen üç boyutlu bir nesnenin görüntülerini yakalayan bir kameradır. Görüntü işlemede alanında görüntü ve video sıkıştırma etkin bir rol oynamaktadır. Görüntü ve video için sıkıştırma teknolojilerinin geliştirilmesi sayesinde multimedya iletişimi ve uygulamaları da ilerleme kaydedilmektedir. Depolama maliyeti son yirmi yılda önemli ölçüde azalmış

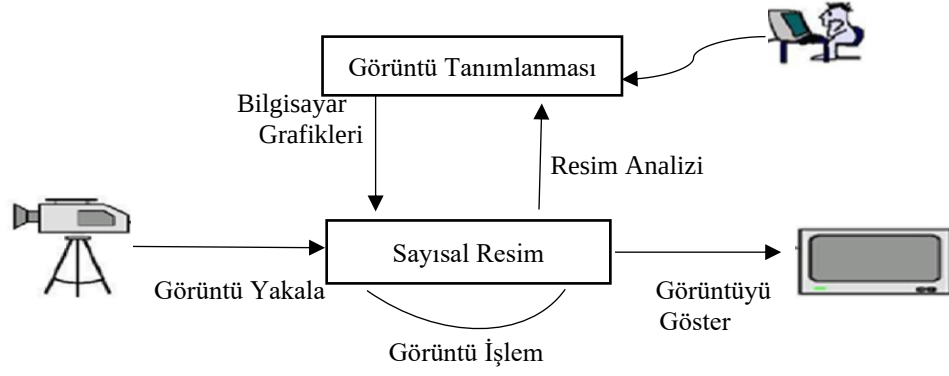
olsa da görüntü ve video sıkıştırma gereksinimi de katlanarak artmaktadır (Acharya ve Ray, 2005).

Günümüz teknolojileriyle birlikte iki boyutlu görüntülerle birlikte üç boyutlu görüntüler ve alternatif görüntü seçenekleri geliştirilmiştir. Görüntüleme imkânlarına göre ekran kare, dikdörtgen ve diğer seçeneklerle karşılaşılmaktadır. Boyutlarının farklılık göstermesinin yanında en yaygın kullanılanlardan biri 800 sütun ve 600 satırdan oluşan her elemanı bir pikseli ifade eden bir matristir. Şekil 2.3’de gösterilmektedir (Akıncı, 2017).



Şekil 2.3. Pikselleri ifade eden matris (Akıncı, 2017)

Sayısal bir resme dönüştürülmüş görüntülerin bir giriş verisi şeklinde işlenerek sözkonusu verinin özelliklerini ve görüntüsünü değiştirmek suretiyle yeni bir resim oluşturmaya görüntü işleme olarak tanımlanması sebebiyle (Karakoç, 2011), görüntü işleme görüntünün sayısallaştırılması ile başlamaktadır. Bu sürecin aşamaları Şekil 2.4’de yer almaktadır (Yaman ve ark., 2001).



Şekil 2.4. Görüntü işleme akışı

Görüntü işleme, savunma sanayi, güvenlik ve suç laboratuvarları, tıp, robotik, astronomi ve trafik, fotoğraf endüstrisi, petrol arama, biyomedikal, sanat alanlarında, uzaktan algılama, hava gözlem ve tahmin uygulamaları gibi alanlarda kullanılmaktadır. (Karakoç, 2011).

2.2.3. Örüntü tanıma

Örüntü tanıma nesneleri kategorize etmeye ya da sınıflara ayrılmasına katkı sağlayan bilimsel bir disiplin olarak adlandırılmaktadır. Uygulamaya ilgili olarak ise nesneler sinyal dalgası şekillerine, görüntülere, ya da kategorize edilmesi ihtiyaç duyulan ölçüm türü olabilmektedir (Teodoridis ve Koutroumbas, 2003).

Örüntü tanıma uzun bir geçmişe sahiptir, ancak 1960'lardan önce çoğunlukla istatistik alanında teorik araştırmanın çıktısı olarak kullanılmaktaydı. Bilgisayarların teknolojisinin ortaya çıkışı ve gelişmesi sayesinde örüntü tanımaya ait uygulamalarına ilgi oldukça artmış ve bu ilgi sayesinde daha fazla teorik ilerlemek için uygun imkanların yaratmasına olanak vermiştir. Toplumun sanayi aşamasından sanayi sonrası aşamaya doğru ilerledikçe sanayi üretiminde otomasyon, bilgi işleme ve erişim ihtiyacı giderek önem kazanmaktadır. Günümüzde bu dönüşüm ile örüntü tanımayı mühendislik uygulamalarının ve araştırmalarının en zirve noktasına ulaşmasını sağlamıştır (Teodoridis ve Koutroumbas, 2003).

Görüntü işlemede çevreyi gözlemlene, arka planlarından ilgili örüntüleri tanımayı ve örüntü sınıfları hakkında doğru kararlar alma bilimine örüntü tanıma denilmektedir. Duda ve Hart (1973) çalışmalarına göre örüntü tanıma işlevini, karmaşık ve gürültülü ortamlardaki anlam düzenlemelerinin makine tarafından tanınmasıyla ilgili bir alan olarak tanımlamıştır. Gonzalez ve Thomas (1978), örüntü tanıma işlevini çok

sayıda gürültülü veriden önem taşıyan özellikleri çıkararak girdi verilerinin sınıflandırması olarak tanımlamışlardır. Watanabe (1985), örüntü tanıma işleminin sınıflandırma problemi, yapı analizi, ayrımcılık yaklaşımı ve benzeri gibi hayal edilebileceğini belirtmiştir. Fukunaga (1990) örüntü tanımanın “yoğunluğun işlevlerini yüksek boyutlu bir uzayda değerlendirme olduğunu ifade etmiştir. Schalkoff (1992), örüntü tanıma işlevini “ölçümlerin sınıflandırılması veya tanımlanması ile ilgilenen bilim” olarak tanımlamıştır (Akt. Albera, 2016). Bir başka tanıma göre ise örüntü tanıma, gerçek ve gürültülü verilerle başa çıkmasını sağlayan bir süreç olarak kabul edilmektedir (Nandy ve Padariya, 2016).

Duin (2002) örüntü tanımanın bir mühendislik yaklaşımı olduğunu ve nihai hedefinin uygulama ile teori arasındaki var olan boşluğu çözebilecek bir makine tasarlamak olduğunu belirtmiştir. Sergios Theodoridis (2003) örüntü tanıma işlevini bilimsel bir disiplin olarak tanımlamıştır ve amacını nesneyi birçok sınıf kategorisinde sınıflandırmak olarak vurgulamıştır (Akt. Albera, 2016).

Örüntü tanıma, karmaşık ortamlarda bulunan gürültüyü anlam düzenlerinin makine tarafından tanınmasıyla ilgili bir alandır. Örüntü kelimesi, patron kelimesi ile aynı kökten türetilmiş ve taklit edilecek mükemmel bir örnek olarak yer almıştır. Birçok gürültülü verinin önemli özelliklerinin çıkarılması yoluyla girdi verilerinin sınıflandırılmasıdır. Örüntü tanıma, farklı alanlarda çok sayıda uygulamanın geliştirilmesini ima eden birçok yöntem içermektedir (Akt. Nayak ve ark., 2011).

Örüntü tanıma, gözlem yapılarını kullanan herhangi bir alanda kullanılabilir. Halen, örüntü tanıma, yüz tanıma, karakter tanıma, bilgisayar görüşü, konuşma tanıma, parmak izlerini tanıma, otomobil tipi kişisel tanımlama sistemlerini tanıma, araç sistemi için arıza teşhisi gibi askeri, sağlık ve imalat endüstrisi ile ilgili sayısız uygulamada kullanılmaktadır (Akt. Albera, 2016).

Örüntü bilgisi ile veri tabanında bulunan bilgiler sayesinde belirli bir ilişki kurularak bilgisayar tarafından değerlendirilmesi görüntü işleme tekniklerine yardım sağlanmaktadır. Fark etmeden hayatta karşılaşılan bazı durumlar örüntü tanıma işini sürekli şekilde yapmaktadır. Yolda ilerlerken karşılaşılan insanları tanımak, çevreden gelen sesleri ayırt etmek buna örnek olarak gösterilebilmektedir (Sert, 2010).

Örüntü tanıma, konuşma tanıma, yüz tanıma, el yazısı karakterlerin sınıflandırılması ve tıbbi tanı gibi çeşitli sorunlara çözüm sağlamaktadır. Örüntü tanımanın çeşitli uygulama alanları biyoinformatik, belge sınıflandırması, görüntü analizi, veri madenciliği, endüstriyel otomasyon, biyometrik tanıma, uzaktan algılama, el

yazısı metin analizi, tıbbi tanı, konuşma tanıma, CBS olarak sıralanmaktadır. Tüm bu uygulamalar arasındaki benzerlik, bir çözüm bulma yaklaşımı için özelliklerin çıkarılması ve daha sonra tanıma ve sınıflandırma amacıyla analiz edilmesi gerekmektedir (Asht ve Dass, 2012).

Örüntü tanıma görevinde üç işlem gerçekleşmektedir. İlk adım veri toplamadır. Veri toplama, bir formdaki (konuşma, karakter, resimler vb.) verileri, başka işlemler için hesaplama cihazı tarafından kabul edilebilir olması gereken başka bir forma dönüştürme işlemidir. Veri toplama genellikle sensörler, sayısallaştırma makinesi ve tarayıcılar tarafından sağlanmaktadır. İkinci basamak olarak veri analizi yapılmaktadır. Analiz görevi için öncelikle veri toplama işlemi yapılmaktadır. Veri analizi basamağında veriler ile ilgili öğrenme süreci başlamakta ve verilerdeki farklı olaylar ve kalıp sınıfları ile ilgili bilgi edinilmektedir. Verilerle ilgili bu bilgi veya bilgiler, daha sonraki işlemler yapılmak amacıyla edinilmektedir. Örüntü tanıma için yapılan üçüncü basamak işlem kategorize etmektir. Amacı, veri analizi sürecinden alınan bilgi temelinde yeni veri sınıfına karar vermektir (Asht ve Dass, 2012). Şekil 2.5’de görülen örüntü tanıma sisteminin temel yapısıyla görüntü işleme noktasında akışı gösterilmektedir (Sert, 2010).



Şekil 2.5. Örüntü tanıma sistemi

Örüntü tanıma sistemi aşağıdakilerden oluşmaktadır (Nayak ve ark., 2011):

1. **Bir sensör:** Sınıflandırılacak bilgileri toplamaktadır.
2. **Özellik seçici veya çıkarıcı:** Özellik seçimi, belirli bir değişkenler kümesinin alt kümesini seçme işlemidir. Özellik çıkarıcı mekanizması, yeni değişkenler oluşturmak için orijinal değişkenlerin doğrusal olmayan olası bir kombinasyonunu almaktadır.

3. **Bir sınıflandırıcı:** Çıkarılan özelliklere dayanan gözlemleri kategorize etmek ya da açıklamaktır. Örüntü sınıflamasının amacı, örüntü analizinden elde edilen bilgileri kategorize etmek için bilgisayarı disipline etmek amacıyla yararlanmaktır.

Örüntü tanıma sistemleri üçe ayrılmaktadır (Sert, 2010);

- a. **İstatistiksel Tanıma:** Bu yöntemde, istatistiksel analiz temelin dayandırılarak istatistiksel algoritmalar bulunmaktadır. Özellik olarak nitelendirilen ölçümler giriş

verilerinden çıkarılmaktadır. Özellik vektörleri her örüntüye denk gelmektedir. Çıktıyı oluşturan sınıflandırma özellikleri ve sınıflandırma yöntemleri kullanılmaktadır. Sınıflandırma yöntemleri işlenebilir örüntü bilgilerini bir araya getirmeyi esas almaktadır.

b. Yapısal Tanıma: Bu yöntemde verilen bir örüntü, şekilsel yapı karakteristik tanımlanma olarak algılanmaktadır. İşlenmemiş verilerden elde edilen bilgi veya bunların sentezinden çıkarılan bilgiler kullanılmaktadır. Yapısal yöntemde, basit alt örüntüler kullanılarak karışık örüntüler formüle edilmektedir. Yapısal tanıma yönteminde veriler arasındaki ilişki, büyük önem taşımaktadır ve gerçekleşen tanıma şekillerle gösterilmektedir. Ekrandaki bir sandalyeyi tanıma geometrik alan boyunca uzanan bacaklar olarak tanımlanabilmektedir. Bu tanımada, çevre uzunluğu, alan ve Fourier tanımlayıcıları gibi özellikler kullanılmaktadır.

c. Akıllı Tanıma: Bu yöntemde öğrenme esas alınarak verilerden öğrenilen bilgilerle karar verilmektedir. Akıllı tanıma sistemlerinde çoğunlukla yapay sinir ağları kullanılmaktadır. Eğitim ve tanıma aşamalarından oluşan örüntü tanıma algılama, özellik çıkarma ve veri tabanına bilgi gönderme işlemlerini gerçekleştirmektedir (Akt. Sert, 2010).

Örüntü tanımayla birlikte görüntü işleme tekniklerinin kullanım alanları artmaktadır. Böylece birçok amaç için bu teknikler kullanılmaktadır. Bu amaçları beş gruba ayrılmaktadır (Albera, 2016).

- **Görselleştirme:** Görünmez nesneleri gözlemlemek için,
- **Görüntü netleştirme ve yenileme:** Daha iyi bir görüntü oluşturmak için,
- **Görüntü alımı:** İlgili görüntüyü aramak için,
- **Desen ölçümü:** Görüntüdeki birkaç nesneyi ölçmek için,
- **Görüntü Tanıma:** Görüntüdeki nesneleri ayırt etmek içindir.

Örüntü tanıma ve görüntü işleme tekniklerinin faydalanarak geliştirilen uygulamalar aşağıda verilmiştir (Albera, 2016).

- **Akıllı Ulaşım Sistemleri:** Otomatik plaka tanıma, trafik işareti tanıma ve araç logosu tanıma işlemlerinde kullanılmaktadır.

- **Uzaktan Algılama:** Uzaktan algılanan uydularda veya bir uçakta monte edilmiş çok spektrumlu tarayıcılarda dünya yüzeyinin resimlerini çeker. Nesneleri ve bölgeleri yorumlamak için uygulanan teknikler taşkın kontrolü, şehir planlama, tarımsal üretim izleme vb. alanlarda kullanılmaktadır.

- **Hareketli Nesne Takibi:** Uygulama hareket parametrelerini ölçülmesine ve hareketli nesnenin görsel kaydını elde etmenize izin verir. Bir nesneyi izlemek için farklı yaklaşım türleri şunlardır: hareket tabanlı izleme ve tanıma tabanlı izlemedir.

- **Savunma Gözetimi:** Uygulama okyanus yüzeyindeki deniz taşıtlarının çeşitlerini ve oluşumlarını keşfetmek için kullanılır. Önemli görev görüntünün su kütlesi kısmında bulunan birkaç nesneyi bölmektir. Uzunluk, genişlik, alan, çevre, kompaktlık gibi farklı parametreler bölünmüş nesnelerin her birini sınıflandırmak için ayarlanmıştır.

- **Biyomedikal Görüntüleme Teknikleri:** Tıbbi tanı, X-ışını, ultrason gibi farklı görüntüleme araçlarında kullanılmaktadır.

- **Otomatik Görsel Muayene Sistemi:** Endüstrilerde ürünün mükemmelliğini ve verimliliğini kurtarmada kullanılmaktadır.

2.3. Derin Öğrenme Algoritmalarında Nesne Tespitinde Yer Alan Sinir Ağları

2.3.1. Derin öğrenmede konvolüsyonel sinir ağları

CNN, görüntü sınıflandırma ve algılama gibi bilgisayar görme görevlerinde doğru bir performans üretmek için kullanılan derin, ileri beslemeli yapay sinir ağı sınıfıdır. CNN'ler geleneksel sinir ağına benzemektedir, fakat daha derin katmanları bulunmaktadır. Doğrusal olmayan bir aktivasyon yoluyla ağırlıklara, sapmalara ve çıktılara sahiptir. CNN nöronları yükseklik, genişlik ve derinlik gibi hacimsel bir şekilde düzenlenmiştir (Galvez ve ark., 2018).

Derin öğrenme algoritmalarının nesne tespitinde kullanılmasıyla birlikte CNN ana unsur olarak görüntüleri kategorize etmek benzerlikle kümelemek (fotoğraf arama) ve sahnelerde nesne tespit etmek için kullanılan yapay sinir ağlarıdır. Standart çok katmanlı sinir ağı ile bir veya daha fazla bağlı katlardan oluşan yaygın bir şekilde alt örnekleme adımını kullanan bir veya daha fazla sarmal katmanın bir araya gelmesine CNN denilmektedir (Akt. Deperlioğlu ve Köse, 2018). Bu ağlar görüntü işlemede ve görüntü sınıflandırma da oldukça başarılıdır.

Nesne tespiti için en gelişmiş yöntemler, CNN'leri bölge tekliflerini arka plana veya nesne sınıflarından birine sınıflandırmak üzere eğitmektedir (Kang, 2016). Derin öğrenme, özellikle CNN, çeşitli görüntü tanıma sorunları için son teknoloji haline gelmiştir. Sadece hesaplama süresini azaltmakla kalmaz, aynı zamanda çok daha büyük

ağları dikkate almayı da sağlamaktadır. Bu nedenle, günümüzde bilgisayarlar daha derin, daha geniş ve daha güçlü modeller kullanabilmektedir. Son teknoloji CNN'ler, el yazısı karakter tanıma, yüz tanıma, sahne etiketleme, nesne algılama ve görüntü sınıflandırma gibi çeşitli tanıma görevlerinde insan benzeri bir performans elde etmişlerdir. Bu arada, mobil cihazlar CNN modellerini neredeyse gerçek zamanlı olarak dağıtmak için gerekli hesaplamaları kaldırabilecek kadar güçlü hale gelmiştir (Tobias ve ark., 2016).

2.3.1.1. Konvolüsyonel sinir ağlarında aktivasyon fonksiyonları

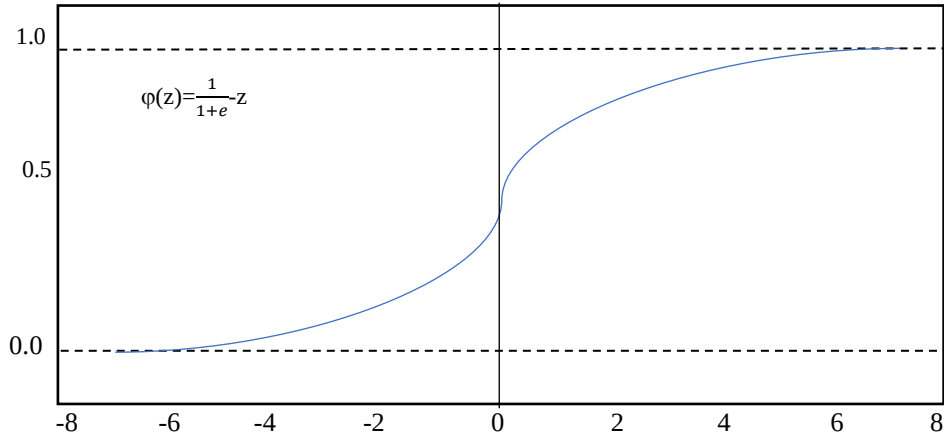
Görüntülerin vektörel ağırlıklarını, eğilim değerlerini güncelleştirebilmek için aktivasyon fonksiyonları kullanılmaktadır. Ağın başarı olması için giriş ve çıkış birimleri arasında eşleşmeyi doğru yapabilen aktivasyon fonksiyonu seçilmektedir (Demirtaş, 2018).

Sinir ağlarının hızlı hesaplama yapabilmesi için fonksiyonların türevlenebilir olması, sinir ağlarının daha hızlı hesaplamalar yapması için gereken avantajı sağlamaktadır. Bu sebeple, sıklıkla kullanılan aktivasyon fonksiyonları sigmoid, tanjant hiperbolik, doğrultulmuş lineer ünite ve sızdırılmış doğrultulmuş lineer ünite aktivasyon fonksiyonları olarak sınıflandırılmaktadır (Kuş, 2019).

2.3.1.1.1. Sigmoid aktivasyon fonksiyonu

Aktivasyon fonksiyonları ağırlık ve eğilim (bias) değerlerini en uygun düzeye ayarlamak ve nöronlar arasındaki iletişimi başlatmak için kullanılmaktadır (Demirtaş, 2018).

Sigmoid aktivasyon fonksiyonu, türevi alınabilmektedir. Türevi daimidir. Kendisinin de türevi alınabilmektedir. Bu sebeple en çok kullanılan fonksiyonlardandır (Yetkin, 2014).

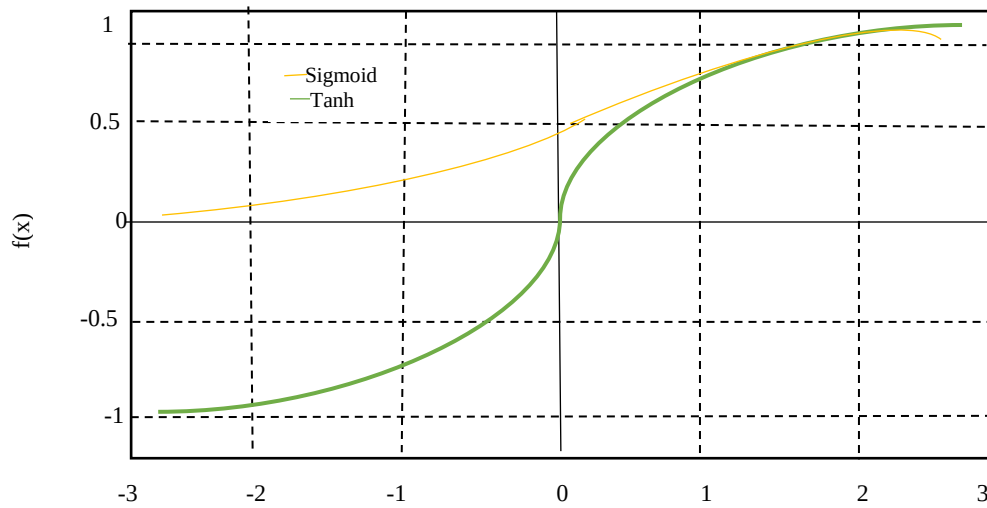


Şekil 2.6. Sigmoid aktivasyon fonksiyonu

2.3.1.1.2. Tanjant hiperbolik aktivasyon fonksiyonu

Tanjant hiperbolik aktivasyon (tanh) fonksiyonu sigmoid fonksiyonuna benzemekte ve daha avantajlıdır. Tanh fonksiyonu negatif değişkenleri kapsayan zaman serilerini modellemesi ile sigmoid tipli fonksiyona göre daha üstündür (Taşova, 2011).

Tanh, özellikle iki sınıfa ayırmada kullanırken benzeri olan sigmoid fonksiyonundan farklı girdi değerleri olarak -1 ile 1 arasında değerler üretmektedir (İpek, 2018).

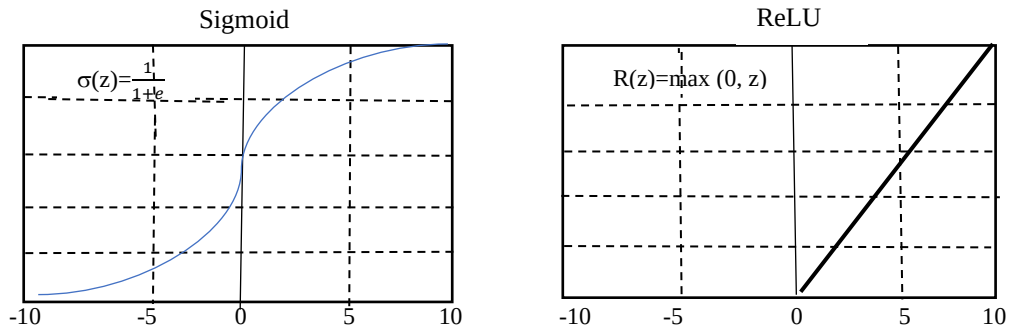


Şekil 2.7. Tanh aktivasyon fonksiyonu

2.3.1.1.3. Doğrultulmuş lineer ünite

Aktivasyon fonksiyonlarında girdi aralıkları 0 ile sonsuz olmasından dolayı doyuma ulaşmayan fonksiyon olarak adlandırılmaktadır. Bu fonksiyonlar doyuma ulaşmadığı için daha hızlı çalışmaktadır (Gülcü ve Kuş, 2019).

Doğrusal birim katmanının (ReLU) elde edilen verilerin negatif değerlerini birkaç matematiksel işlem yaparak sıfır olarak almaktadır. Bu yüzden derin öğrenmede oldukça sıklıkla rastlanmaktadır. Sıfır olarak verilerin olumsuz etkilerini daha hızlı olmasını sağlamıştır (Akt. Demirtaş, 2018).



Şekil 2.8. ReLU aktivasyon fonksiyonu

Aktivasyon fonksiyonu olumsuz değerleri sıfır diğer girişleri olduğu gibi almasından dolayı biyolojik nöronlara benzemektedir. Elde edilen verileri sıfıra yaklaştırmadığı için nöron kaybına uğramaktadır. Oluşan nöron kayıplarını en aza indirmek için Leaky ReLU fonksiyonu geliştirilmiştir (Demirtaş, 2018).

2.3.1.1.4. Sızdırılmış doğrultulmuş lineer ünite

ReLU aktivasyon fonksiyonun özelliklerine ilave olarak negatif değerlere çok küçük değerler verdiği için nöronların kaybolmasını engellemektedir (Demirtaş, 2018).

ReLU'dan farklı olan aktivasyon fonksiyonuna negatif değerler için eğim eklenilmesini sağlayan bir değer seçilerek bu değer (a_i), sabit, küçük (0.001) bir değer olduğunda aktivasyon fonksiyonu Leaky-ReLU olarak adlandırılmaktadır (Akt. Kuş, 2019).

$$Y_i = x_i, \quad \text{eğer} \quad x_i > 0 \quad (2.1)$$

$$Y_i = a_i x_i \quad \text{eğer} \quad 0 \geq x_i \quad (2.2)$$

2.3.1.2. Konvolüsyonel sinir ağlarının mimarisi

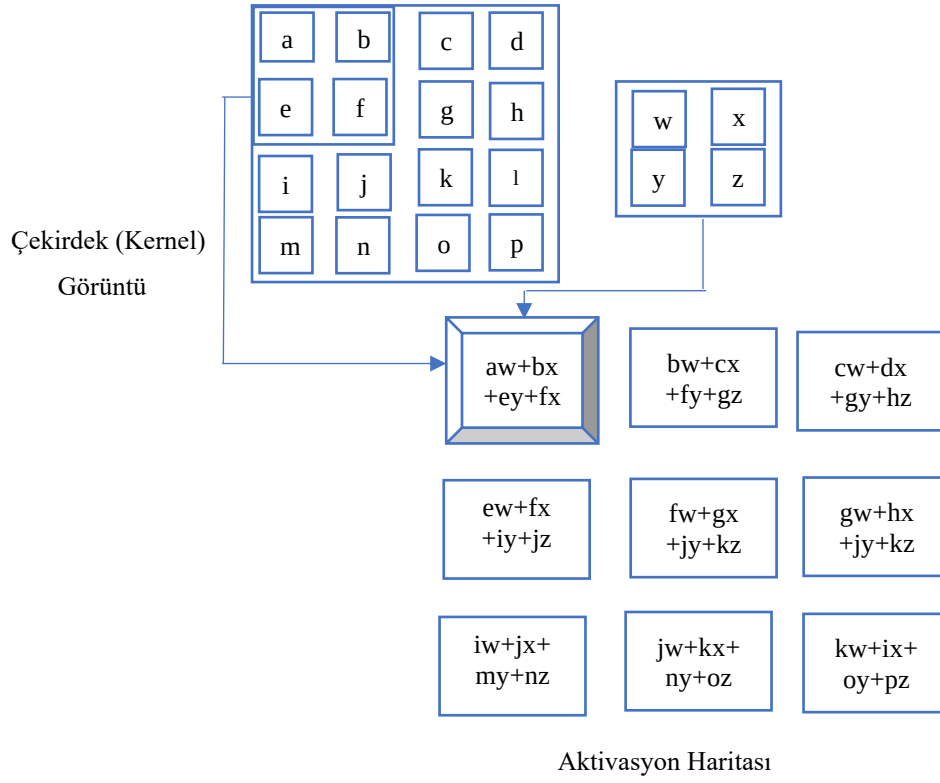
CNN katmanları konvolüsyonel katmanı, havuzlama katmanı ve tam bağlantılı katman olmak üzere üç temel bileşenden oluşmaktadır. Aşağıda bu katmanlar hakkında bilgi verilecektir.

2.3.1.2.1. Konvolüsyon katmanı

Konvolüsyon katmanına sunulan resimlerdeki özellikler her biri iç içe geçmiş filtrelerden geçirilerek çıkartılmaktadır (MathWorks, 2018). Bu katman CNN mimarisi ana katmanı olarak kabul etmektedir. Konvolüsyon katmanı bir önceki katmanlarla ilişkisi içerisindedir. Bir önceki katmanda bölgesel şekilde birbirine bağlı nöronların bir kısmını kullanmak suretiyle giriş verisine dönüştürme işlevini konvolüsyon katmanı yapmaktadır (Özdağ, 2019).

Konvolüsyon katmanı tüm uzamsal konumların aynı konvolüsyon çekirdeğini paylaşmaktadır, bu da konvolüsyon katmanı için gereken parametre sayısını büyük ölçüde azaltmaktadır. Örneğin, bir girdi görüntüsünde birden fazla köpek görünürse, aynı "köpek kafasına benzer desen" özelliği, farklı köpeklerin kafalarına karşılık gelen birden fazla konumda etkinleştirilebilmektedir (Wu, 2017).

CNN esas yapısını konvolüsyonel katman, iki matrisin arasındaki hesaplamayı bir nokta çıkarımı ile gerçekleştirir. Kernel ilk matrisin öğrenilebilir özellik kümesidir ve girdi alanının kısıtlı bir kısmıdır (Mishra, 2019).



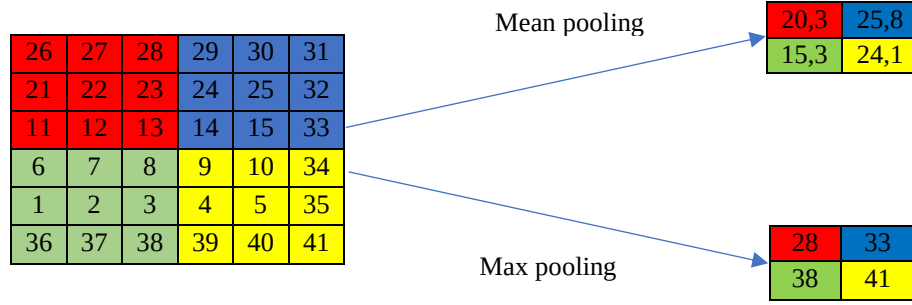
Şekil 2.9. Konvolüsyon işlemi (Goodfellow ve ark., 2016; Ghoury, 2019)

Konvolüsyonel işlemi sırasında oluşturulan kernelin yükseklik ve genişliğinin ileri geçiş sırasında her adımda görüntünün aktivasyon haritası üretilmesine adım adı verilmektedir (Mishra, 2019).

Konvolüsyonel katmanı CNN'nin ana katmanıdır. Bu katman CNN özelliklerinin algılanmasında sorumlu olan katmandır. Görüntüde ki mevcut yüksek ve düşük seviyedeki özellikleri çıkarmak için söz konusu katmanda resme filtreler uygulanmaktadır. Uygulanan filtreler hem çok boyutludur hem de piksel değerler bulunmaktadır (Serdoğan, 2020).

2.3.1.2.2. Havuzlama katmanı

Havuzlama katmanı sıralı şekilde bulunan konvolüsyonel katmanı arasına yerleştirilmektedir. Verinin genişlik ve yüksekliğini kademeli şekilde azaltmaktadır. Bu azalış sayesinde aşırı şekildeki uyum kontrol altına alınmaktadır. Havuzlama katmanında verinin genişlik ve yüksekliğini yeni bir boyut vermek amacıyla “max” işlemi uygulanmaktadır. Bu işlem de 2×2 boyutlu bir filtre ve filtre bölgesindeki 4 rakamın en büyüğü alınmaktadır (Özdağ, 2019).



Şekil 2.10. Max pooling ve Mean pooling

Havuzlama katmanının işlevi elde edilecek girdinin nitelik özelliklerinden hem verinin boyutunu hem de hesaplama ve parametre sayısını sayısını azaltmaktadır. Bunun için konvolüsyon katmanını çıktısında yer alan mekânsal boyutu küçültülmesi gerekmektedir. Havuzlama katmanı tek başına kullanılamamakta iki konvolüsyon katmanı ya da tam bağlı katman arasında yer alarak kullanılmaktadır (Sertkaya, 2018).

Havuz katmanları, verimli bir eğitim süreci sağlamak için CNN'nin çok önemli bir parçasıdır. Küçük bir kayıpla daha iyi doğruluk elde etmek için alt örnekleme önemli bir rol oynamaktadır. Genellikle, CNN havuzlama teknikleri maksimum ve ortalama havuzlama yöntemidir. İşlem sürecinde, en yüksek havuz oluşturma havuz penceresindeki maksimum puanı alır ve ortalama havuz, havuz penceresinin ortalama puanını seçmektedir. Bununla birlikte, bu çalışma çok fazla düşünmeden havuz türlerini seçmektedir (Wanda ve Jin Jie, 2020).

2.3.1.2.3. Tam bağlantılı katman

Tam bağlantılı katmanlarda bir nöron bir önceki katmandaki nörona bağlanmaktadır. Bu bağlantıların ağırlıkları bulunmakta ve katmanlar bağlantı modelidir (Boyacı, 2017). Bu katman CNN son katmanı olarak yer almaktadır. Bu katmandan önceki katmanlarda girdi verilerinin özellik çıkarımı, boyut düşürme ve normalizasyon işlemleri gerçekleştirilmektedir. Çıktı verileri için hata hesaplaması yapmakta ve bu hata değerin göre ağırlıklar yeniden hesaplanmaktadır (Gülcü ve Kuş, 2019).

Tam bağlantılı katmanlarda özellikle belli eşik değerin altındaki düğümlerin eksiltilmesiyle başarı artmaktadır. Böylece zayıf bilgilerin göz ardı edilmesi öğrenime katkı sağlamaktadır. Buna eksiltme işlemi denilmektedir. Eksiltme işleminin aşağıda

belirtilen özellikler ile giriş ile çıkış arasındaki eşleşmeye katkı sağlamış olacaktır (Kılınç, 2018).

- Eksiltme değeri olarak 0,5 kullanılmaktadır. Farklı kullanıldığı durumlar da mevcuttur.
- Eksiltme için rastgele eleme yöntemi de kullanılmaktadır.
- Eksiltme değeri eşik değeri olarak $[0,1]$ aralığında kullanılmaktadır.
- Katmanlarda eksiltme değeri kullanmak mecburi değildir.

2.3.2. Bölgesel tabanlı konvolüsyonel sinir ağları

Nesne tespit için bir katman ve ardından standart bir CNN devam edememesi çıktı katmanının uzunluğunun değişken olmasından kaynaklanmaktadır. Bunun sebebi değişkenlik nesnelerin oluşum sayısının sabit olmamasından ötürüdür. Bu problemi çözmek amacıyla yeni bir yaklaşım olarak görüntüden değişik bölgeleri belirlemek ve sözkonusu bölgede yer alan nesnenin mevcudiyetini kategorize etmek amacıyla yeni bir CNN önerilmiştir. Yeni yaklaşımdaki problem ise görüntü içinde yer alan nesnelerin farklı konum ve en-boy oranlarının bulunmaktadır. Bu sebeple, çok sayıda bölge seçerek söz konusu sorun çözülmektedir. Bu nedenle, R-CNN, FAST R-CNN, FASTER R-CNN, MASK R-CNN ve YOLO vb. algoritmalar bu sorunu bulmak ve hızlı tespit için geliştirilmiştir (Gandi, 2018).

R-CNN'ler son zamanlarda nesne tespiti alanında başarı göstermektedir. Derin öğrenmede en gelişmiş nesne algılama yaklaşımlarından olan R-CNN nesne tespiti için hesaplanan görsel nesne algılama sistemidir. Nesne algılama problemlerini çözmek için kullanılan R-CNN'ler görüntü üzerine sınırlayıcı kutular çizmektedir. Bu aşamayı teklif edilen bölge ve sınıflandırma yapılan bölge olarak ikiye ayırmaktadır. Herhangi bir bilinmez bölgenin büyüklüğü derin modelin çoklu katmanlarından geçmekte ve ağırlı çıkışı tekrar ağırlı girişini beslemektedir. Böylece görüntülerdeki tespit daha isabetli olmaktadır (Aalami, 2019).

R-CNN, CNN kullanarak görüntünün çeşitli bölgelerini sınıflandırmaktadır. Tespitlerin hesaplama maliyetlerini azaltmak amacıyla R-CNN'ler, tespit ağıyla beslenen çeşitli ilgi alanlarını gerçek zamanla ve doğru performansla elde eden evrişimli hesaplamaları ile paylaşmaktadır (Ren ve ark., 2016). Nesne tanıma ve tespiti için çok katmanlı evrişim ağlarını bir nesne algılama ve segmentasyon sistemini (sınırlayıcı

kutuları veya piksel düzeyinde segmentasyon maskelerini algılama) görüntü bölgelerini sınıflandırmada kullanılmaktadır (Girshick ve ark. 2014).

Görüntü sınıflandırmasından farklı olarak algılama, bir görüntüdeki nesnelerin yerleştirilmesine ihtiyaç duyulmaktadır. Çerçeve tespitini bir regresyon problemi olarak incelemektedir. Bu formülasyon tek bir nesneyi yerleştirmek için iyi çalışabilir ancak birden fazla nesneyi tespit etmek karmaşık çözümler veya görüntü başına nesne sayısı hakkında geçici bir varsayım gerektirir. Bir alternatif, kayar pencere detektörü oluşturmaktır. CNN'ler bu şekilde en az yirmi yıldır, tipik olarak yüzler, eller ve yayalar gibi kısıtlanmış nesne kategorilerinde kullanılmıştır. Bu yaklaşım hesaplama verimliliği açısından caziptir, ancak basit uygulaması tüm nesnelerin ortak bir en boy oranını paylaşmasını gerektirmektedir (Girshick ve ark. 2014).

2.3.2.1. Bölgesel tabanlı konvolüsyonel sinir ağları mimarisi

Nesne algılama, görüntüdeki nesneleri bulma ve sınıflandırma işlemidir. Bir derin öğrenme yaklaşımı R-CNN sahip bölgeler, dikdörtgen bölge önerilerini CNN özellikleri ile birleştirmektedir. R-CNN iki işlemli bir algılama algoritmasıdır. İlk işlem, görüntüdeki nesne içerebilecek bölgelerin bir alt kümesini tanımlamaktadır. İkinci işlem her bölgedeki nesneyi kategorize etmektir (MathWorks, 2020).

CNN'leri olan bölgeleri kullanarak nesne algılama modelleri aşağıdaki üç işlemi temel almaktadır:

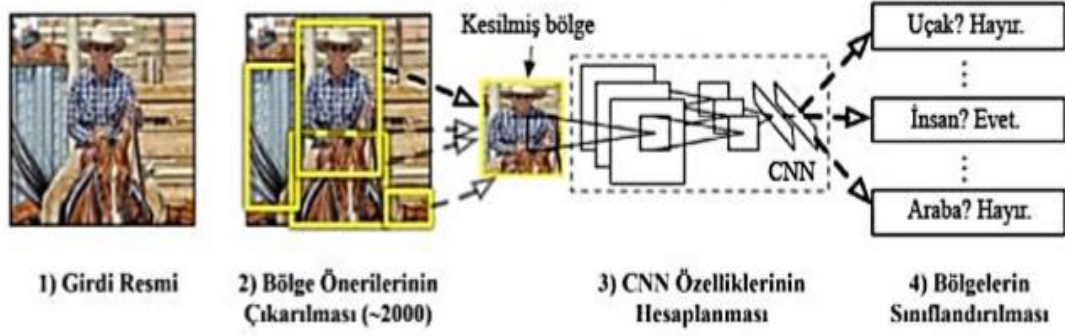
- Görüntüde nesne içerebilecek bölgeleri bulmaktır. Bu bölgelere bölge teklifleri adı verilmektedir.
- CNN özelliklerini bölge tekliflerinden çıkarmaktır.
- Nesneleri çıkarılan özellikleri kullanarak sınıflandırmaktır.

Çok sayıda bölge seçme sorununu için Ross Girshick ve ark. görüntüden sadece 2000 bölgeyi çıkarmak için seçici arama kullanılan bir yöntem önermiş ve onlara bölge önerileri adını vermişlerdir. Bu nedenle çok sayıda bölgeyi sınıflandırmaya çalışmak yerine sadece 2000 bölgeyle çalışmışlardır. Bu 2000 bölge teklifleri aşağıda yazılı olan seçici arama algoritması kullanılarak oluşturulmuştur (Gandi, 2018).

Seçici arama;

- İlk alt bölümlere oluşturmak ve birçok aday bölge oluşturmak,
- Benzer bölgeleri özyinelemeli olarak benzer bölgeleri daha büyük olanlara birleştirmek için algoritma kullanmak,

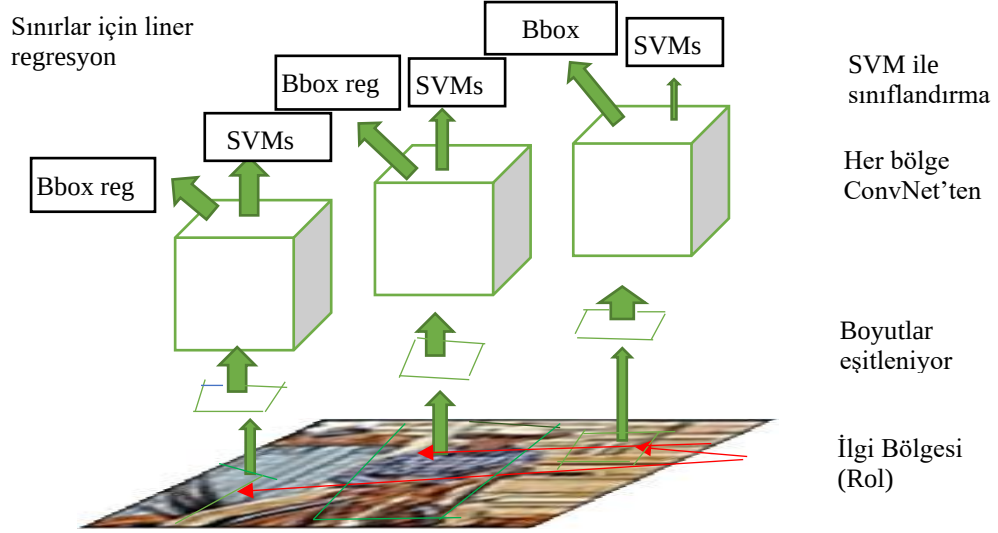
- Nihai aday bölge teklifleri oluşturmak için oluşturulan bölgeleri kullanmaktır.



Şekil 2.10. CNN çalışma mantığı

R-CNN detektörü ilk olarak kenar kutuları gibi bir algoritmalarından yararlanarak bölge önerileri üretmektedir. Söz konusu teklif bölgeleri görüntüden kırılıp ve yeniden boyutlandırılmaktadır. Daha sonra CNN kırılan ve yeniden boyutlandırılan bölgeleri kategorize etmektedir. Son olarak, bölge teklif sınırlama kutuları CNN özellikleri kullanılarak eğitilmiş bir destek vektör makinesi (SVM) tarafından rafine edilmektedir (MathWorks, 2020).

Nesne tespit sistemi üç aşamadan oluşmaktadır. İlki, bağımsız bölge önerileri oluşturmaktır. Bu öneriler, dedektör tarafından kullanılabilecek aday tespit kümesini tanımlamaktadır. İkinci aşama, sabit uzunluklu bir özelliği çıkaran geniş konvolüsyonel sinir ağıdır. Üçüncü aşama her bölgeden sınıfa özgü bir doğrusal SVM kümesi oluşturmaktır. Bölge önerileri bağımsız bölge önerileri oluşturmak için yöntemler sunmaktadır. R-CNN belirli bölge teklif yöntemine agnostik olsada, önceki ile kontrollü bir karşılaştırma sağlamak için seçici arama kullanmaktadır. İleriye doğru yayılarak hesaplanan test görüntüsü üzerinde seçici arama yapmaktadır. 2000'e yakın bölge önerisi çıkarmak için ileri özellikleri hesaplama CNN üzerinden yapılmaktadır. Her sınıf için eğitilmiş SVM'yi kullanarak çıkarılan özellik vektörünü görüntüdeki puanlanmış bölgeler göz önüne alınarak daha yüksek puanla seçilen bölge ile bir kesişim aşımı (IoU) varsa bir bölgeyi reddeden maksimum olmayan bir eşikten algılamayı verimli hale getirmektedir (Girshick, Donahue ve ark. 2014).



Şekil 2.11. CNN mimarisi (Girshick, Donahue ve ark. 2014)

Seçici arama algoritması 2000 aday bölge teklifi bir kareye çarpıtılmış ve çıktı olarak 4096 boyutlu bir özellik vektörü üreten CNN beslenmiştir. CNN, bir özellik çıkarıcı görevi görmekte ve çıktı yoğun katmanı görüntüden çıkarılan özelliklerden oluşmaktadır. Çıkarılan özellikler aday bölge teklifi içindeki nesnenin mevcudiyetini kategorize etmek amacıyla bir SVM kullanılmaktadır. Bölge teklifleri içinde bir nesnenin varlığını tahmin etmenin yanı sıra algoritma sınırlama kutusunun hassasiyetini artırmak için denkleştirme değerleri olan dört değeri de tahmin etmektedir. Örnek verilecek olursak bir bölge teklifi önerisi verildiği zaman algoritma bir kişinin mevcudiyetini tahmin etmekte fakat sözkonusu bölge önerisi kişinin yüzü yarı yarıya kesilebilmektedir. Bu nedenle denkleştirme değerleri bölge önerilerinin kısıtlayıcı kutusunun ayarlanmasına katkıda bulunmaktadır (Gandi, 2018).

R-CNN nesnelerin özelliklerine göre bölge önerisi yaparken sınıflandırma yaptığı esnada içinde bulundurduğu özellikleri kullanmaktadır. Bu işlemleri yaparken aradığı özellikleri sayısallaştırması resim üzerindeki aranılan nesnenin sayısal karşılığı ile mümkün hale gelmektedir. Bu yüzden nesne tanıma ve tespiti için R-CNN kütüphaneleri farklı boyutlarda bölge önerileriyle sinir ağlarından geçirmektedir. Sinir ağından geçilirken SVM sınıflandırma için kullanılmaktadır. Kullanılan R-CNN algoritmaları nesne tespitinde bölgede aradığı nesneyi x, y koordinatları ve uzunluk, derinlik olarak tespit etmeye çalışmaktadır. Bu denkleştirme değerleriyle tespit etmek istediğimiz nesne ile arama yapılan bölgenin arasında doğrusal bir ilişki aranarak nesne tespit edilmektedir (Demiroğ, 2019).

2.3.2.2. Bölge tabanlı konvolüsyonel sinir ağlarının problemleri

R-CNN nesne önerilerini sınıflandırmak için derin bir ConvNet kullanarak mükemmel nesne algılama doğruluğu elde etmektedir (Girshick, 2014). Bununla birlikte R-CNN'nin önemli dezavantajları bulunmaktadır:

- Eğitimin çok aşamalı olması sorunsalı: R-CNN ilk olarak bir ConvNet'i kullanarak nesne teklifleri üzerinde netleştirmektedir. Ardından SVM'leri ConvNet özelliklerine dönüştürmektedir. Bu SVM'ler, nesne ayarlama ile öğrenilen softmax sınıfının yerini alan nesne dedektörleri olarak işlev görmektedir. Üçüncü eğitim aşamasında sınırlayıcı geriye yönelik öğrenmektedir.

- Eğitimin alan ve zamanda maliyetli olması sorunsalı: SVM ve sınırlayıcı kutu geriye yönelik eğitim için özellikler görüntüdeki nesne teklifinden çıkarılmakta ve diske yazılmaktadır. VGG16 gibi çok derin ağlarda, bu işlem VOC07 tren seti setinin 5k görüntüsü için GPU ile 2,5 gün sürmektedir. Bu özellikler yüzlerce gigabayt depolama alanı gerektirmektedir.

- Nesne algılamasının yavaş olması sorunsalı: Test zamanında özellikler her test görüntüsünde her bir nesne teklifinden çıkarılmaktadır. VGG16 ile algılama 47s / görüntü (GPU'da) sürmektedir.

R-CNN ile ilgili sorunları özetleyecek olursak; ağı eğitmek, görüntü başına 2000 bölge teklifini sınıflandırma gerektiği için bu durumun çok vakit alması, her test görüntüsünün yaklaşık 47 saniye sürmesi sebebiyle gerçek zamanlı olarak uygulanamaması ve seçici arama algoritması sabit bir algoritma olması nedeniyle bu süreçte herhangi bir öğrenme gerçekleşmemesi nedeniyle kötü aday bölge önerilerinin oluşturulmasına yol açmasıdır.

R-CNN hesaplamayı paylaşmadan her nesne teklifi için bir ConvNet ileri geçişi gerçekleştirmesinden dolayı oldukça yavaştır. Uzamsal piramit havuzlama ağlarının (SPPnet) hesaplamayı paylaşarak R-CNN'yi hızlandırmaları önerilmektedir. SPPnet yöntemi, tüm girdi görüntüsü için konvolüsyonel bir özellik eşlemesi hesaplamakta ve ardından her bir nesne teklifini paylaşılan özellik eşleminden çıkarılan bir özellik vektörü kullanarak sınıflandırmaktadır. Bir nesne teklif için özellikler, özellik haritasının teklif içindeki kısmı sabit boyutlu bir çıktıya (ör: 6×6) çıkartılarak oluşturulmaktadır. Birden çok çıktı boyutu toplanmakta ve daha sonra uzaysal piramit havuzunda olduğu gibi birleştirilmektedir (Lazebnik, Schmid ve Ponce, 2006). SPPnet, test zamanında R-CNN'yi

10 ila 100 kat hızlandırmaktadır. Daha hızlı teklif özelliklerinin çıkarılması nedeniyle eğitim süresi de 3 kat azalmaktadır (Girshick, 2014).

SPPnet'in de önemli sakıncaları bulunmaktadır. R-CNN gibi, eğitim de özelliklerin çıkarılması, günlük kaybı olan bir ağın ince ayarının yapılması, SVM'lerin eğitilmesi ve son olarak sınırlayıcı regresörleri içeren çok aşamalı bir ağıdır. Çıkarılan özellikler diske de yazılmaktadır. Ancak R-CNN'den farklı olarak önerilen ince ayar algoritması, uzaysal piramit havuzlamasından önceki evrimsel katmanları güncelleyememektedir (Girshick, 2014; He, Zhang ve ark., 2014).

2.3.3. Bölge tabanlı konvolüsyonel sinir ağları çeşitleri

Bir R-CNN'nin iki çeşidi bulunmaktadır. Bunlar Fast R-CNN ve Faster R-CNN algoritmalarıdır. Her R-CNN bu işlemlerden bir veya daha fazlasının sonuçlarını optimize etmeye, hızlandırmaya veya geliştirmeye çalışmaktadır (MathWorks, 2020).

2.3.3.1. Hızlı bölgesel tabanlı konvolüsyonel sinir ağları

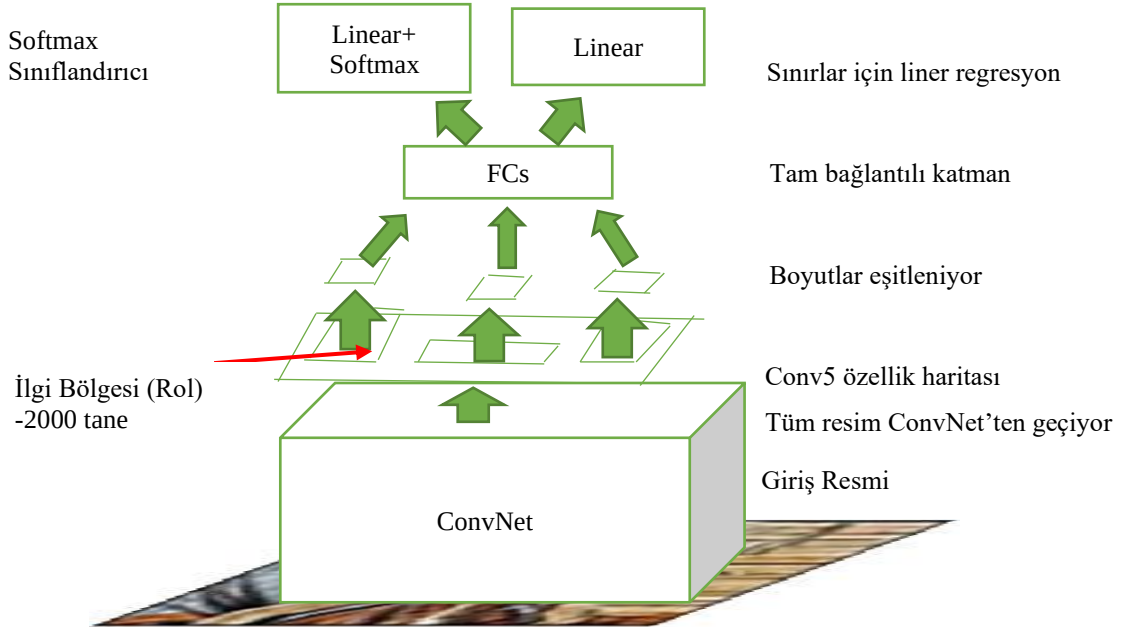
Hız ve doğruluklarını geliştirirken R-CNN ve SPPnet'in dezavantajlarını indirgeyen yeni bir eğitim algoritması önerilmektedir. Eğitmek ve test etmek nispeten hızlı olduğu için bu yöntem Fast R-CNN denilmektedir (Girshick, 2014). Fast R-CNN, Python ve C ++ dilinde yazılmıştır (Jia ve ark., 2014).

Fast R-CNN yönteminin çeşitli avantajları aşağıdaki gibi sıralanmaktadır;

- R-CNN, SPPnet'ten daha yüksek algılama kalitesi,
- Eğitim çok aşamalı bir kayıp kullanarak tek aşamalı olması,
- Eğitim tüm ağ katmanlarını güncelleyebilmesi,
- Özellik önbelleğe almak için disk depolama gerekmemesidir.

Bölge tabanlı algoritmalar tarafından kullanılan evrim özellik haritaları Fast R-CNN bölge önerileri oluşturmak için de kullanılmaktadır. Bu dönüşüm özelliklerinin üstünde, iki ilave dönüşüm katmanı ekleyerek bölge önerileri oluşturmaktadır. Her dönüşüm haritasının konumunu özellik vektörü çeşitli ölçeklere ve yönler göre bölge teklifleri için nesne puanlama ve gerileme sınırlarını oranlamaktadır (Ren, He ve ark., 2016).

R-CNN algoritması gibi Fast R-CNN algoritması da bölge önerileri oluşturmak amacıyla kenar kutuları gibi bir algorithmadan yararlanmaktadır. Bölge tekliflerini kırıp yeniden boyutlandıran R-CNN algoritmasının aksine, Fast R-CNN algoritması görüntünün tamamını işlemektedir. Bir R-CNN algoritması her bir bölgeyi kategorize etmesi gerekirken, Fast R-CNN her bir bölge teklifine karşılık gelen CNN özelliklerini havuzlamaktadır. Fast R-CNN, R-CNN'den daha verimlidir, çünkü Fast R-CNN algoritmasında çakışan bölgeler için hesaplamalar paylaşılmaktadır (MathWorks, 2020).



Şekil 2.12. Fast R-CNN mimarisi (Girshick ve ark., 2014)

Şekil 2.12 Fast R-CNN mimarisini göstermektedir. Bir Fast R-CNN ağı girdi olarak görüntünün tamamını ve bir dizi nesne teklifini almaktadır. Ağ ilk olarak tüm görüntüyü çeşitli evrişimli ve maksimum havuzlama katmanlarıyla işlemektedir. Daha sonra her nesne teklifi için ilgilenilen bir bölge havuz katmanı özellik haritasından sabit uzunluklu bir özellik vektörü çıkarmaktadır. Her özellik vektörü iki kardeş çıkış katmanını beslemekte ve son olarak dallanan tam olarak bağlanmış katmanlar dizisine beslenmektedir. Nesne sınıfları üzerinde softmax olasılık tahminleri tümünü arka plan ve dört gerçek çıktı veren 4 değer kümesi için önerilen sınırlama kutusu konumlarını kodlamaktadır (Girshick, 2014).

Fast R-CNN, R-CNN algoritmasına benzemektedir. Ancak bölge tekliflerini CNN'ye beslemek yerine, evrişimli bir özellik haritası oluşturmak için girdi görüntüsünü CNN ile beslemektedir. Evrişimsel özellik haritasından, tekliflerin bölgesini belirleyip

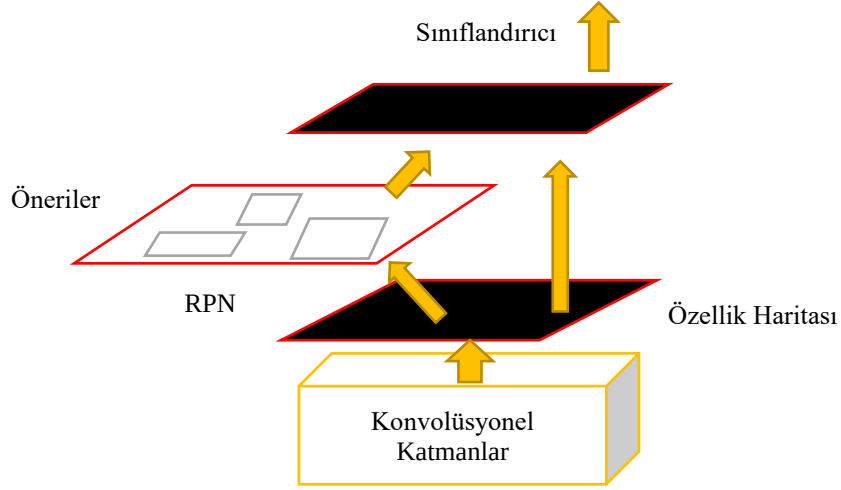
karelere dönüştürmekte ve bir havuzlama katmanı kullanarak bunları tamamen bağlı bir katmanla beslenebilmesi için sabit bir boyutu yeniden şekillendirmektedir. Özellik vektöründen teklif edilen bölgenin sınıfını ve ayrıca sınırlama kutusunun değişkenlik değerlerini bulmak amacıyla bir softmax katmanı kullanmaktadır. R-CNN'nin'den Fast R-CNN daha yavaş olmasının sebebi Fast R-CNN her defasında kıvrımlı sinir ağına 2000 bölge önerisi beslemesini gerekmemesidir (Gandi, 2018).

2.3.3.2. Daha hızlı bölgesel tabanlı konvolüsyonel sinir ağları

R-CNN ve Fast R-CNN bölge tekliflerini tespit etmek amacıyla seçici arama kullanmaktadır. Seçici arama yavaş ve zaman alıcı bir işlem olması sebebiyle ağın performansını etkilemektedir. Bu nedenle Faster R-CNN seçici arama algoritmasını yok eden ve ilgili ağı bölge tekliflerini öğrenmesini fayda sağlayan bir nesne tespit algoritmasıdır. Faster R-CNN'de Fast R-CNN'e benzer şekilde görüntü evrişimli bir özellik haritası sağlayan evrişimsel bir ağı girişi olarak sağlanmaktadır. Bölge tekliflerini belirlemek amacıyla özellik haritasında seçici arama algoritması kullanmak yerine bölge tekliflerini tahmin etmek amacıyla farklı bir ağ kullanmaktadır. Tahmin edilen bölge tekliflerini ilerleyen süreçte önerilen bölgede yer alan görüntüyü kategorize etmek için ve kısıtlayıcı kutularının değişiklik değerlerini tahmin etmek için yararlanılan ROI havuzlama katmanı ile tekrardan şekillendirilmektedir (Gandi, 2018).

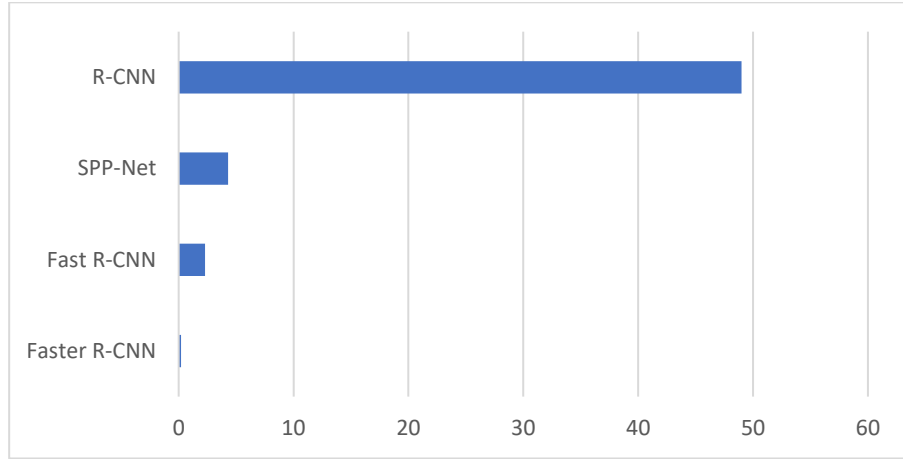
Faster R-CNN adı verilen nesne tespit sistemi iki aşamadan oluşmaktadır. İlk aşama bölgeleri öneren tamamen kıvrımlı derin bir ağıdır. İkinci aşama ise önerilen bölgeleri kullanan Fast R-CNN algoritmalarıdır. Nesne tespiti için bölge teklif önerisiyle nereye bakılacağını söylemektedir. Bölge teklif ağı (RPN) bir görüntüyü (herhangi bir boyutta) girdi olarak almakta ve her biri bir nesne puanına sahip bir dizi dikdörtgen nesne teklifi çıkarmaktadır. Bu işlemi tam bir evrişim ağı ile modeller ve hesaplamayı bir Fast R-CNN nesne algılama ağı ile paylaşmaktadır (Ren, S. ve ark., 2016).

Şekil 2.13'de Faster R-CNN mimarisi gösterilmektedir. Faster R-CNN algoritması kenar kutuları gibi harici bir algoritma kullanmak yerine bölge tekliflerini doğrudan ağda oluşturmak için bir RPN eklemektedir. RPN nesne tespiti için bağlantı kutuları kullanmaktadır. Ağda bölge teklifleri oluşturmak, verilere göre daha hızlı ve daha iyi ayarlanmaktadır (MathWorks,2020).



Şekil 2.13. Faster R-CNN mimarisi

RPN ilave edildikten sonra geri kalan işlemler R-CNN ile aynıdır. Üç değişik model PASCAL VOC veri seti üzerinde test edilmiş ve Şekil 2.14’de yer alan sonuçlar yer almaktadır. Aşağıdaki grafikten Faster R-CNN'nin öncekilerden çok daha hızlı olduğunu görülmektedir. Bu sebeple Faster R-CNN gerçek zamanlı nesne tespiti için bile kullanılabilir (Gandi, 2018)



Şekil 2.14. R-CNN algoritmalarının hız karşılaştırmaları (Gandi, 2018)

2.3.4. ResNet ile görüntü sınıflandırma

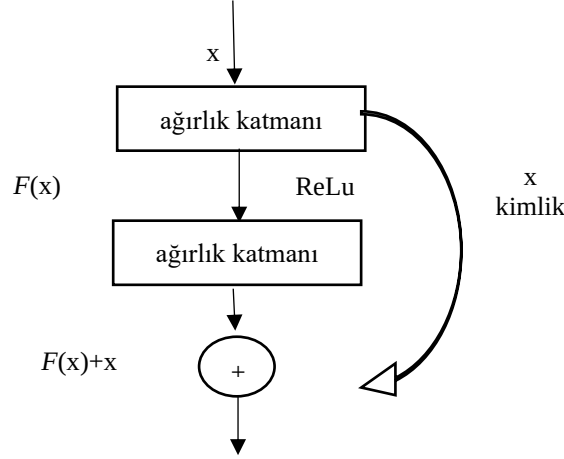
Derin ağların birleşmeye başlamasıyla bozulma sorunu ortaya çıkmaktadır. Ağın derinliği arttıkça, doğruluk doymuş hale gelmektedir. Böylece ağın yapısı hızla bozulmaktadır. Microsoft Research ve Facebook AI araştırması bu sorunu çözebilmek

için ResNet ve ResNeXt Architectures gibi son teknoloji image sınıflandırma ağlarını ortaya koymuşlardır (Jay, 2018).

Büyük bir veri seti gerektiğinde, CNN'yi sıfırdan eğitmek zor ve zaman alıcı olmaktadır. Bu nedenle son yıllarda birçok araştırmacı ImageNet veri kümeleri üzerinde eğitilen Resnet, VGG, AlexNet, vb. bu modelleri, tıbbi görüntülerin otomatik özellik çıkarımı ve sınıflandırılması için kullanmaktadır. Bu modeller 1000 kategoriye sınıflandırmakta ve yaklaşık 1,2 milyon eğitim görüntüsü üzerinde eğitilmektedir (Gopal ve Dessai, 2019).

ResNet-101, 101 katman derinliğinde kıvrımlı bir sinir ağından oluşmaktadır. Bu ağ ImageNet veritabanında bir milyondan fazla görüntü üzerinde eğitilmiştir. Böylece ağ çok farklı görüntüler için zengin özellik temsillerini öğrenmiştir. Ağın görüntü giriş boyutu 224×224 'tür. Hazırlanan ağların çoğu, ImageNet büyük ölçekli görsel tanıma zorluğunda ILSVRC kullanılan ImageNet veritabanının alt kümesinde eğitilmiştir. Bu ağlar bir milyon üzerinde görüntüyle eğitilen ve görüntüleri kalem, kahve kupası, klavye ve birçok hayvan gibi 1000 nesne kategorisinde sınıflandırmaktadır. Önceden eğitilmiş bir ağ kullanmak, bir ağı sıfırdan eğitmekten çok daha hızlı ve kolay olmaktadır. Tahmin süresi, en hızlı ağa göre ölçülmektedir (MathWorks, 2020).

ResNet, serebral kortekste piramidal (beynin en gelişmiş) hücrelerinden bilinen yapılara dayanan yapay bir sinir ağıdır. Tipik ResNet modelleri, ReLU ve aralarında toplu normalizasyon içeren çift veya üç katmanlı atlamalar uygulamaktadır. Atlama ağırlıklarını öğrenmek için ek bir ağırlık matrisi kullanmaktadır. Bu modeller otayol ağları olarak bilinmektedir. Birkaç paralel atlama içeren modellere DenseNets adı verilmektedir. Eğitim sırasında, ağırlıklar yukarı akış katmanını susturmak için önceden atlanan katmanı güçlendirmektedir. Atlama, ilk eğitim aşamalarında daha az katman kullanarak ağı etkili bir şekilde basitleştirmektedir. Bunun sonucunda ilerlemek için daha az katman ile yok olan düşüşlerin etkisini azaltarak öğrenmeyi hızlandırmaktadır. Daha sonra ağ, özellik alanını öğrenirken atlanan katmanları yavaş yavaş geri yüklemektedir. Eğitimin sonuna doğru, tüm katmanlar genişletildiğinde, daha hızlı öğrendiği gözükmemektedir (Wikipedia, 2020). Aşağıdaki Şekil 2.15'de ResNet mantığı gösterilmektedir.



Şekil 2.15. ResNet mantığı (He, Zhang ve ark., 2016)

X , bu katmanların ilkinde girişleri göstermektedir. Doğrusal olmayan çoklu katmanların asimptotik olarak karmaşık fonksiyonları yaklaşık olarak tahmin edebildikleri varsayıldığında giriş ve çıkışın aynı boyutlarda olduğu görülmektedir. Bu nedenle yığılmış katmanların yaklaşık olmasını beklemek yerine bu katmanların artık $f(x)$ fonksiyonuna yaklaşmasına izin verilmektedir. Böylece orijinal fonksiyon $f(x) + x$ olacaktır. Eklenen katmanlar karşılığında büyük olmayan eğitim hatasına neden olmaktadır. Bozulma problemi çoklu eşleme olmayan katmanlar tarafından kimlik eşlemelerine yaklaştırmakta zorluk çekebileceğini göstermektedir. ResNet101’de dikkat çekici bir şekilde derinlik artmasına rağmen bozulma problemi görülmemektedir (He, Zhang ve ark., 2016).

2.4. Nesne Tespitinde Kullanılan Kütüphaneler

Nesne tespitinde kullanılan birçok derin öğrenme algoritma arasında R-CNN, Fast R-CNN, Faster R-CNN, Single Shot Multibox Detector ve Mask R-CNN vardır. Çalışmada kullanılan R-CNN görüntülerde nesne tespitinde oldukça başarılı sonuçlar vermektedir. Bu derin öğrenme algoritmaları birçok derin öğrenme kütüphanelerinde kullanılmaktadır. Tensorflow kütüphanesi de bunlardan bir tanesidir. Aşağıda verilen derin öğrenme kütüphaneleri birçok uygulamada kullanılmakta ve kullanılmaya devam edilecektir (Balık ve ark., 2017).

Tablo 2.1. Derin öğrenme kütüphaneleri

Kütüphane	Yazıldığı Dil	Geliştirici
Theano	Python	MILA Lab.
Caffe	Python	Berkeley Vision and Learning Center
Torch	Lua	Ronan Collobert, Clement Farabet ve diğerleri
Digits	C++	NVIDIA
TensorFlow	Python	Google
DeepLearning4j	Java	Adam Gibson
KNET	Julia	Deniz Yuret

2.4.1. Theano

Theano, Montreal Üniversitesi'ndeki Yoshua Bengio tarafından bulunmuş ve MILA tarafından eksikliği giderilerek geliştirilmiştir. Theano GPU'larda veya CPU'larda oldukça etkili olan Python kütüphanesidir. Theano üzerinde Keras, Lazanya, Blocks, Pylearn2 gibi API'lerde bulunmaktadır (Balık ve ark, 2017)

Theano, çok boyutlu dizileri içeren matematiksel ifadeleri tanımlamanıza, optimize etmenize ve değerlendirmenize olanak tanıyan bir Python kütüphanesidir. Theano özellikleri (LISA lab, University of Montreal, 2017):

- NumPy ile sıkı entegrasyon içerisindedir.
- Yoğun veri kullanan hesaplamaları içim bir CPU'dan çok daha hızlı işlem gerçekleştirmektedir.
- Birçok hata türünü tespit ve teşhis etmektedir.

2.4.2. Caffe

Caffe Model Zoo üzerinden indirilebilen, hemen çalıştırılabilen, daha önceden eğitilmiş ağların tespit edilmesi ve bilgisayar görme verileri ile çalışma olanağı sağlayan Python platformudur (Jia ve ark., 2014).

Caffe, multimedia bilim insanlarına ve uygulayıcılarına en yeni derin öğrenme mimarileri ve referans modelleri koleksiyonu için net ve değiştirilebilir bir çerçeve sunmaktadır. Çerçeve, genel amaçlı evrişimli sinir ağlarının ve diğer derin modellerin emtia mimarileri üzerinde verimli bir şekilde eğitimi ve dağıtımı için Python ve MATLAB bağlarına sahip BSD lisanslı bir C++ kütüphanesidir (Jia ve ark., 2014).

2.4.3. TensorFlow

TensorFlow makine öğrenimi ve derin sinir ağı araştırmaları yapmak için Google Brain Machine tarafından Google'ın Machine Learning Intelligence araştırma organizasyonu içinde geliştirilen bir yazılım kütüphanesidir. TensorFlow ana özellikleri aşağıda yer almaktadır. Bunlar (Zaccone, 2016);

- Çok boyutlu dizileri (tensörler) içeren matematiksel ifadeleri tanımlama, optimize etme ve verimli bir şekilde hesaplama
- Derin sinir ağları ve makine öğrenimi programlama desteği teknikleri.
- Makinelerde ve devasa veri kümelerinde yüksek hesaplama ölçeklenebilirliğidir.

TensorFlow, sunucu ya da mobil cihazda yer alan CPU'ya veya GPU'ya dağıtma imkânı vermektedir. Derin öğrenme çalışma grubu makine öğrenimini tek bir API ile hesaplamasını yapmak ve derin sinir ağları kullanması sağlamak için geliştirilmiştir (Balık ve ark., 2017).

2.4.4. Torch

Google DeepMind, Twitter ve Facebook AI laboratuvarı alanlarında yapılan çalışmalar neticesinde geliştirilen Torch, C / C ++ kitaplıkları kullanılmaktadır. İşlemi oldukça kolay ve sade hale getiren Torch bilimsel algoritmalarını yapma konusunda yeterli esnekliğe ve hıza sahiptir. Torch'un meşhur uygulamaları, CNN ve derin pekiştirme öğrenmesi olan daha karmaşık bölgelerde görüntü sorunları nedeniyle tercih edilmektedir (Brownlee J., 2019).

2.4.5. Digits

NVIDIA Derin Öğrenme GPU Eğitim Sistemi (DIGITS), mühendislerin ve veri bilimcilerin sayesinde görüntüleri kategorize etme, bölümlendirme ve nesne tanıma için oldukça hassas derin sinir ağını hızlı ve etkili şekilde eğitmek için yararlanılmaktadır. DIGITS, veri yöntemi çoklu GPU sistemlerinde sinir ağları tasarlama ve gelişmiş görselleştirmelerle gerçek zamanlı oldukça başarılı performansı gösteren modeli seçme gibi genel derin öğrenme vazifelerini basitleştirmektedir (NVIDIA DIGITS).

Derin Öğrenme kütüphaneleri kullandıkları derin öğrenme algoritmaları ve birbirleri arasında kullandıkları ilgi alanları net bir biçimde ayrılmaktadır. Aşağıdaki Çizelge 2.2’de kütüphanelerin kullandıkları derin öğrenme algoritmaları yer almaktadır.

Tablo 2.2. Derin öğrenme kütüphanelerinin kullandığı algoritmalar

	CNN	RNN	RBM	LSTM	Derin Oto-Kod
Theano	+	+	+	+	+
TensorFlow	+	+	+	+	+
Torch	+	+	-	+	+
Caffe	+	+	+	+	+
MXnet	+	+	+	+	+
Neon	+	+	-	+	+
Microsoft Cognitive Toolkit	+	+	+	+	+

2.5. Nesne Tespitinde Kullanılan Araçlar

2.5.1. İnsansız hava araçlarının kavramı ve özellikleri

İHA kendi iç dinamikleri ile kendini idame edebilen, birçok çeşit faydalı yük imkânı bulunan yarı otonom veya tam otonom uçabilen pilotsuz hava araçlarıdır. Bunun yanında seyrüsefer füzeleri balistik füzeler bu kategoride yer almamaktadır. İHA teknolojisinin öncülerinden sayılan ABD İHA’lar için drones, robot uçak, pilotsuz uçak olarak da adlandırılmaktadır. İHA sistemlerini kontrol edebilmek için kullanılan yer kontrol istasyonu (YKİ), haberleşme imkânı sağlayan yer veri terminalinden oluşmaktadır (Akyürek ve ark., 2012).

İHA, önceden programlanmış uçuş planlarına veya daha karmaşık dinamik otomasyon sistemlerine bağlı şekilde uzaktan kumanda edilebilen ya da otonom olarak uçabilen pilot olmayan otonom insansız bir uçak olarak da tanımlanabilmektedir (Huang ve ark., 2013).

ABD Savunma Bakanlığı'na göre İHA, insan operatörü taşımayan, araç asansörü sağlamak için aerodinamik kuvvet kullanan, otonom olarak uçabilen veya uzaktan kumanda edilebilen, harcanabilir veya kurtarılabilir ve ölümcül ya da ölümcül olmayan bir taşıma yükü taşıyan araçtır (Akt. Gözkaydıran, 2019).

İHA’lar kesin çizgilerle oluşturulan bir sınırı bulunmamaktadır. Faydalı yükün çeşidine göre uçuş süresi ve irtifanın değiştiği kullanım alanına göre kanat çeşitleri değiştiği sınıflandırmalar olmaktadır. İHA’ların uçuş süreleri, meteorolojik etkenler vb. görev yaparken karşılaşılan kısıtlamalardır (Şener, 2019).

İHA terimi yaygın olarak kullanılmaktadır, ancak uzaktan kumandalı araç, uzaktan kumandalı uçak, uzaktan kumandalı helikopter, insansız araç sistemleri ve model helikopter gibi terimler de sıklıkla kullanılmaktadır. Boyut, ağırlık, dayanıklılık, menzil ve uçuş yüksekliğine bağlı olarak İHA, üç ana kategoriye ayrılmaktadır (Remondino ve ark., 2011):

- İlk kategori de mikro, mini, düşük irtifa derin penetrasyon, düşük irtifa uzun dayanıklılık, orta irtifa uzun dayanıklılık sistemleri içeren taktik İHA'lar bulunmaktadır. Bu İHA'lar birkaç kilogram ila 1.000 kg arasında, birkaç kilometre ila 500 km uçabilen, uçuş yüksekliği birkaç yüz metreden 5 km'ye ulaşan ve dayanıklılık olarak birkaç dakikadan 2-3 güne kadar değişen unsurları bulunmaktadır.

- İkinci kategori de 20.000 m yükseklikten daha yüksek uçan ve 2-4 günlük dayanıklılığa sahip, yüksek irtifaya uzun süre dayanıklılığı olan, stratosferik ve ekso-stratosferik sistemleri içeren stratejik İHA'lar bulunmaktadır.

- Üçüncü kategori de ölümcül ve tuzak sistemleri gibi özel görevler için tasarlanan İHA'lardır.

İHA'yı insanlı uçaklarla karşılaştırıldığında, iki sistem arasındaki temel farkın İHA'da fiziksel olarak pilot bulunmaması olduğu açıktır. Bu durum İHA'nın kendi başına bağımsız olarak uçuğu anlamına gelmemektedir. Çoğu durumda, bir İHA'dan sorumlu mürettebat (operatör, yedek pilot vb.) insanla uçan bir uçaktan daha fazla sayıda olabilmektedir (Everaerts, 2008).

İHA'ların insanlı uçak sistemlerine göre üstünlüğü insan yaşamını tehlikeye atmadan ve erişilemeyen alanlarda, yüksek riskli durumlarda, yüksek irtifada ve insanlı sistemlerin uçamayacağı cisimlere yakın uçuş profillerinde kullanılabilmesidir. Bu bölgelere doğal afet yerleri, kaza sahneleri, taşkın ovalar, dağlık ve volkanik alanlar, deprem ve çöl alanları olarak örnek verilebilmektedir. Erişimi zor bölgelerde, insanlı uçakların bulunmadığı ya da uçuşun riskli olması sebebiyle izni verilmediği yerlerde İHA'lar bazen tek pratik alternatiftir. İlaveten, bulutlu ve yağmurun çiselediği hava şartlarında, nesneye olan mesafe bulutların altında uçmaya izin verdiğinde İHA'larla veri elde etmek hala mümkündür. Bu tür hava koşulları, yer üstü hava yüksekliğinin gerekli olması nedeniyle insanlı uçaklara entegre edilmiş geniş format kameralarla veri edinilmesine izin vermemektedir. Ayrıca İHA'ları kullanmanın temel avantajlarından biri, insan pilotların fizyolojik sınırlamaları ve ekonomik harcamaları ile yükümlü olmamalarıdır. Dahası gerçek zamanlı kapasite ve hızlı veri elde etme kabiliyetidir.

Ayrıca görüntü, video ve yönlendirme verilerini gerçek zamanlı olarak YKİ'ye iletmektedir (Eisenbeiss, 2009).

İHA, bir yük, ağ sistemi ve pilottan oluşan monte edilmiş ekipmanla çeşitli görevleri gerçekleştiren bir sistemdir (Gözkaydırın, 2019). Bu sistem;

- Roket, bomba gibi yükler,
- İHA ile pilot arasında iletişim sağlayan sinyal sistemi,
- Komuta kontrol sistemi,
- Zemin sistemleri, donanım ve araçların birleştirildiği destek elemanı
- İHA'yı çalıştıran insan faktörü gibi bileşenlerine amaçlarına göre

sıralanmaktadır.



Şekil 2.16. Çeşitli İHA Türleri (Wikipedia, 2016).

İHA kullanıcısı olmayan, uzaktan komutayla kontrol edilebilen ve bu kontrolü komuta kontrol iletişimi ile yapabilen hava aracıdır. İHA temelde üç bileşenden oluşmaktadır (Torun, 2017). Bunlar ise

- Ana gövdeyi oluşturan iskelet, kanat, pervane, motor ve batarya,
- Kontrol birimini oluşturan elektronik algılayıcılar, haberleşme elektroniği ve Global Navigation Satellite System,
- Kullanım amacına dönük sensör, kamera vd. algılayıcılardır.

İHA'nın tanımı denince hava aracının göreve yönelik faydalı yükleri, komuta kontrol sisteminden oluşan yarı otonom ya da tam otonom bir sistem, iniş/kalkış gibi birçok alt sistem akla gelmektedir. İHA insanlı hava araçlarından ayıran önemli etken ise pilot yerine karar verebilen elektronik akıldır. Bu elektronik akıl için ise yapay zekâ, genetik algoritmalar yer almaktadır (Altınok T.2010).

İnsanlar İHA'ları tasarlamak istemiştir. Bu nedenleri aşağıda sıralayabiliriz. (Altınok T.2010).

- Uzun keşif ve gözetlemelerde insan dikkatinin azalması,

- İnsanlarını kimyasal ve nükleer tehditlere karşı sıkça maruz kalma durumu,
- Hava savunma ve taarruzlara karşı avantajlı olması,
- Araştırma ve geliştirmeler için insanlı uçaklara göre maliyetinin düşük olmasıdır.

Kendine ait hava savunma sistemleri olmayan terör örgütü ve oluşumlara karşı kullanılabilen bir platformdur. Gelişmiş teknolojiyi bünyesinde barındırması çoğu ülkenin anlık görüntü almak için başvurduğu bir hava aracıdır (Karakuş ve Katman, 2019).

2.5.2. İnsansız hava araçlarının tarihçesi

Uçma arzusu eski zamanlardan beri var olan ve daha sonra Leonardo da Vinci (1452-1519) ve Montgolfier (1783) gibi mucitler tarafından birçok projelerde test edilen makinelerdir. Uçuş rüyası bu tarihlerden itibaren hiç durmamış ve havada daha kolay şekilde uçabilen zeplin gibi makinelerle devam etmiş ve daha sonra havada daha ağır olan uçaklar ve nihayet I. ve II. Dünya Savaşı sırasında insan komutasında uçmayan ancak kontrol edilebilen uçaklar ortaya çıkmıştır (Prisacariu, 2017).

1920'lerin başından beri, hava gücünün savunucuları savaştaki uygun rolünü tartışmışlardır. Başlangıçta bu tartışma, taktiksel mi yoksa stratejik olarak mı uygulanacağına odaklanmıştır. Zamanın terminolojisinde, kara kuvvetlerini keşif, tedarik ve iletişim yoluyla desteklemek için taktik hava gücü olan ihtiyaç artmıştır. Stratejik olarak hava gücünü destekleyenler, hava gücünün şehirlerin ve sanayi merkezlerinin ağır bombalanmasıyla zafer sağlama potansiyeline sahip olduğuna inanmışlardır. Doğal olarak hem savaş anında hem de keşif uçaklarını geliştirilmesi ise İHA'ların yaratılmasına yol açtı (Blom, 2006).

Bugün bilinen İHA ilk defa New York'lu araştırmacı olan Charles Perley 19. yüzyıl sonlarında geliştirmiştir. Geliştirdiği mekanizma bomba bırakma işini yapacak bir balondur (Akt. Akyürek ve ark., 2012). İHA'ların tarihi, genellikle insansız balonların Avrupa'da ve daha sonra Amerikan iç savaşı sırasında bomba atmak için kullanıldığında onsekizinci yüzyılın ortalarına kadar uzanan askeri çatışmalara odaklanan döngüsel gelişmelerden biriydi. I. Dünya Savaşı sırasında ABD, "Kettering Bug" olarak bilinen pilotsuz bir uçak geliştirmiş ve önceden belirlenmiş bir süre boyunca uçmuştur. Bu uçak birkaç tane olacak şekilde üretilmiş ancak hiçbirisi savaş sırasında uçmamıştır. İlk gerçek uzaktan kumandalı uçak olan Sperry Messenger 1920 yılına kadar tasarlanmıştır. Ancak

barış zamanındaki İHA'lara olan ilgi, finansman sebebiyle azalmıştır. Daha sonra 1930'larda, savaş bir kez daha belirmeye başladığında İHA'lara olan ilgi yeniden canlanmıştır. 1930'larda İngilizler “Kraliçe Arılar” olarak bilinen 400'den fazla İHA'lar geliştirmiş ve üretmiştir. 1930'ların sonlarında, uzaktan kumanda uçak modeli meraklısı ünlü aktör Reginald Denny, ABD Ordusuna uzaktan kumandalı uçağın önemini konusunda ikna etmiş ve 1939'dan İkinci Dünya Savaşı'na kadar “Denny dronlarının” adlı verilen İHA'lar 15.000'inden fazla üretilmiştir (DeGarmo, 2004).

ABD ve İngiltere hacimce küçük insansız sistemler üretiliyor olsalar da II. Dünya Savaşı sırasında, otonom kontrole sahip bir uçak olan V-1 bombardıman uçağının ortaya çıkmasıyla insansız havacılık teknolojisinde en büyük gelişmeleri yapan Almanlar olmuştur. V-1, insansız bir uçağın ne kadar zorlu bir tehdit oluşturabileceğini göstermesi bakımından önemli bir gelişmedir. Savaşın ardından ABD, V-1'den teknolojisini öğrenmiş ve bunları kendi yeni İHA tasarımlarına uygulamıştır (DeGarmo, 2004).

İHA'lar çok çeşitli şekil, boyut, konfigürasyon ve özelliklerde geliştirilmiştir. İHA'ların gelişimi 1900'lerin başlarında başladı ve ilk aşamalarındaki teknoloji I. Dünya Savaşı nedeniyle ordu tarafından kullanılmış ve II. Dünya Savaşı ile genişletilmiştir. Ancak ileri teknolojik şekilde kullanılması 1970'li yıllara denk gelmektedir. Bu tarihte keşif ve gözetleme amacıyla İsrail tarafından kullanılmıştır. İsrail Arap Savaşı sırasında kullanılan İHA'lar elde ettiği istihbaratı bilgiyi belirlenen istasyona aktaran ve söz konusu istasyondan aldığı elektronik sinyallerle görevine devam edecek şekilde tasarlanmıştır (Pakkan ve Ermiş, 2010).

Yakın tarihlerde İHA'lar hem ABD hem de Avrupa tarafından sıklıkla kullanılmıştır. Amerika-Körfez Savaşı, Bosna Hersek Savaşı ve Kosova Harekâtında Avrupalı ülkeler tarafından Pioneer ve Pointer adlı ABD yapımı İHA kullanılmıştır (Pakkan ve Ermiş, 2010).

Kosova'da İHA'ların düşürülme riskine karşı savaş uçaklarına karşı görev almışlardır. 2000'li yıllarda İHA'lar Amerika tarafından mayın arama, sağlık malzemesi ikmali, koruma görevleri vb. görevlerde de kullanılmıştır. Silahlı İHA diyebileceğimiz ilk İHA Yemen'de kullanılmıştır. Silahlı İHA'lar daha sonra Körfez Harekâtında oldukça sık kullanılmıştır (Akt. Akyürek ve ark., 2012).

2.5.3. Türkiye’de insansız hava araçları

Türkiye'deki İHA'larla ilgili çalışmalar, Türk Havacılık ve Uzay Sanayii A.Ş. (TAI) tarafından İHA-X1 adlı bir proje ile Mart 1990'da başlamıştır. Uçuş testleri yapılmasına rağmen bu projenin bütçe sorunları nedeniyle devam edilememiştir. Türk Silahlı Kuvvetleri'nin (TSK) acil sınır güvenlik ihtiyaçları için, ABD Genel Atomik Havacılık Sistemleri A.Ş. tarafından üretilen GNAT-750'ler ve I-GNAT'ler sırasıyla 1994 ve 1998'de satın alınmıştır. 1980'lerin sonundan itibaren TSK, İngiltere Meggitt Savunma Sistemleri tarafından üretilen Banshee olarak bilinen hedef dronları da kullanmıştır. İlerleyen yıllarda bu dronların kullanımı ve satın alınması devam etmiş ve bu durum yabancı kaynaklara bağımlılığa yol açmıştır. 1995 yılında, Millî Savunma Bakanlığı ve TAI arasında imzalanan bir anlaşma sayesinde TSK'nın hava savunma eğitimi gereksinimlerini karşılamak için hedef İHA üzerine bir proje başlatılmıştır. Bu projelerin amacı, ileri teknolojiler kullanarak ulusal yetenekleri olan düşük riskli, düşük maliyetli ve operasyonel olarak esnek insansız uçaklar geliştirmektir. Bu çalışmalar sonucunda 2001 yılında TAF ile hedef drone Turna ve izleme drone Keklik üretilmiş ve hizmete girmiştir. Her iki drone da hala hava tehditlerinin simülasyonu ve takip tatbikatları için kullanılmaktadır. Bu çalışmalardan elde edilen deneyimler şu anda TAI tarafından yüksek hızlı hedef drone Şimşek gelişimi için kullanılmaktadır. Takip eden yıllarda TAI, pilot eğitim ve uçuş tesisinde Baykus, Pelikan ve Marty olarak adlandırılan çeşitli İHA türlerini geliştirmeye devam etmiştir (Kahvecioğlu ve Oktal, 2014).

2004 yılında Savunma Sanayi, İHA'ların araştırılması, geliştirilmesi ve satın alınması için bir otorite olarak belirlenmiştir. Savunma Sanayi öncelikle Türkiye'nin ihtiyaçlarını mini, taktik ve orta irtifa uzun menzilli (MALE) sınıfı İHA'lar olarak belirlemiş ve söz konusu sınıfların geliştirme projelerine başlatmıştır. Bu çerçevede Anka (MALE), Çaldıran (Taktik), Bayraktar (sabit kanatlı mini) ve Malazgirt (döner kanatlı mini) adlı İHA'lar geliştirilmiştir. Bu projelerin araştırma ve geliştirme süreçleri ilerlerken İHA'ların doğrudan tedarikleri devam etmektedir. Anka'nın uçuş testlerindeki gecikme nedeniyle, İsrail Havacılık ve Uzay Sanayii tarafından geliştirilen MALE sınıfı Kahramanlar sınır gözetimi için satın alınmıştır (Kahvecioğlu ve Oktal, 2014).

Türkiye’de 2006 yılı itibariyle birbirinden değişik firmaların ürettiği yerli mini İHA modelleri bulunmaktadır. (Akyürek ve ark., 2012). Türkiye’de üretilen mini İHA’lar ve üreten firmalar aşağıda verilmiştir.

- Kale /Baykar- Bayraktar ve Malazgirt,

- Vestel Savunma- Efe ve Arı,
- ODTÜ- Güventürk,
- Global Teknik – Globihadır.

12 Temmuz 2012 tarihli Savunma Sanayi İcra Komitesi toplantısında ANKA geliştirilmesi için TUSAŞ firması ile çalışmalara başlanmıştır. Son zamanlarda Türkiye İHA sistemlerini de içine bulunduğu havacılık sanayi hem platform bazında hem de alt sistem bazında önemli bir yol katetmiştir. Gelecek yıllarda, bugün konuşulan projeler daha da ilerleyerek savunma sanayisine katkı sağlamaya devam edeceği düşünülmektedir (Akyürek ve ark., 2012).

2.5.4. İnsansız hava araçlarının kullanım alanları

İHA askeri ve kamusal uygulamalar için kullanılabilecek yeni bir teknolojidir. İHA'ların askeri kullanımı, öncelikle sınır gözetimi ve keşiftir. Kamu kullanımı ise polis tarafından kamu güvenliği ve ulaşım yönetimi sağlamak için kullanılmaktadır. Felaket uyarılarında, kurtarma işlemlerini hızlandırmaya, erişilemez alanlara tıbbi malzeme taşımaktan, zehirli gaz sızmasına, yangınlara veya vahşi hayvan izlemeye kadar İHA'ları, ilgili personelin güvenliğini riske atmadan geniş bir alanı hızlı bir şekilde kullanılabilmektedir (Gupta ve ark., 2015).

İHA sistemleri sivil ve askeri kullanım alanları olarak ikiye ayrılmaktadır. Sivil İHA'lar, hobi amaçlı fotoğrafçılık, kargo vb. amacıyla da kullanıldığı gözlemlenmiştir. Çeşitli ölümcül ve ölümcül olmayan uçakları taşıyabilen İHA Silahlı ve Silahsız olarak adlandırılabilir (Çoban ve Oktay, 2018).

Askeri uygulamalarla karşılaştırıldığında İHA'ların askeri olmayan uygulamaları nispeten küçük olmakla birlikte hızla büyümektedir. İHA'ların tipik sivil uygulamaları arasında yangınla mücadele yardımı, sivil rahatsızlıkların ve suç mahallerinin polis tarafından izlenmesi, doğal afet müdahalesi için keşif, sınır güvenliği, trafik gözetimi ve yönetimi ve arkeolojik araştırmalarda bilimsel araştırma bulunmaktadır. Son on yılda, İHA'ların kullanımı, mahsul üretim yönetimine yardımcı olmak için hava fotoğrafçılığı ve ürün alanları üzerinde görüntüleme uygulamalarında da kullanılmaktadır (Akt. Huang ve ark., 2013).

İHA'lar nitelik ve nicelikler açısından ikiye ayrılmaktadır. Uçuş yüksekliği, uçuş süreleri, otomatik ya da yarı otomatik kontrol sistemleri, yakıt tipi, yükselme-alçalma süreleri vb. birçok özellik ve farklılıklar nicel sınıflandırmada temel faktörlerdir. Askeri

kullanım amaçları savunma, taarruz, gözlem ve keşif vb. farklı amaçlara göre ayrılmaktadır. Bunun yanında lidar entegrasyonu ile birlikte İHA'nın sivil kullanımı birçok alt dal için tercih edilmektedir. Jeolojik, meteorolojik, topografik gözlemler amacı ile çevresel ve çevreyle ilgili takip amaçlı, 3B katı modelleme, kentsel modelleme, belediyeçilik ve coğrafi bilgi sistemleri altlıkları için, endüstriyel ve tehlikeli bölgelerin haritalaması amacıyla kullanılan İHA'lar bulunmaktadır (Makineci, 2016).

İHA tehlikeli ve güvenli olmayan yerlerde çalışabilmektedir. Ayrıca İHA hızlı bir bakış veya derin bir bakış için kullanılabilir. Bunun için başarılı bir faaliyet olması iyi bir planlamaya bağlıdır. İHA görüntüleri yardımı ile 3B model çıkarma, ya da arama kurtarma, felaketler, yangın ile mücadele, acil müdahale gibi durumlarda kullanılmaktadır. Diğer bir kullanım ise kültürel mirası korumak için ve enerji ve boru hatları güvenliği, kadastro ve ortofoto harita üretiminde kullanılmaktadır (Akt. Toprak, 2014).

Fotogrametri haritacılık alanında oldukça önemli bir yere sahip olan farklı platformlardan harita üretimi için görüntü alımı önemli yere sahiptir. Bu açıdan, İHA teknolojisi görüntü alım platformu olarak yeni bir yöntem olarak ortaya çıkmaktadır. İHA sistemlerinin gerekli desteklerle kolayca görüntü alımı yapılmaktadır. Mini İHA'lar kolay ve hızlı bir şekilde kaliteli görüntüler çekebilmektedir (Akt. Mahmod ve Yılmaz, 2018).

Askeri amaçlarla kullanılan İHA 2000 yılından sonra silahlı modelleri bulunmakla birlikte tüm ordularda yoğunlukla keşif, gözetleme ve hedef tespiti gibi istihbarat amaçları için kullanılmaktadır. Uzaktan kumandalı bu araçların teknolojinin gelişmesi sayesinde daha da uzaktan uydu haberleşmesiyle kontrolü sağlanabilmektedir (Akyürek ve ark., 2012). İHA'larda var olan ya da ilerleyen zamanlarda kazandırılacağı tahmin edilen özellikler şunlardır:

- Keşif, taktik keşif ve gözetleme,
- Bomba veya füzeli hava taarruzları,
- Endirekt (görmeyerek) atışlar için ileri gözetleyicilik,
- Özel operasyonlar ve psikolojik harekât,
- Sınırların kontrolü ve korunması,
- Mayın arama ve imha,
- Sağlık ve askeri malzeme bütünleme ikmali,
- Kaçakçılık ile mücadele,
- Kimyasal, biyolojik ve radyolojik tarama,

- Denizcilikte gemi tanıma ve tecrit,
- Muharebe arama ve kurtarma,
- Hava radyolink ve role görevi,
- Noktadan noktaya kargo teslimi,
- Hava durumu veri toplamadır.

2.5.5. İnsansız hava araçlarının nesne tanımadaki rolü

İHA, kuşlar, normal insansız uçaklar kadar küçük veya insansız pilotu olmayan özel uçaklar kadar büyük olabilen İHA anlamına gelmektedir. Uçakta İHA olmadığı için uzaktan kontrol edilmektedir. İHA şu anda çok farklı amaçlar için kullanılmaktadır. Bu kullanımlara örnek olarak casus çekimleri, gökyüzü görüntüsü, keşif, saldıran roller, hava gözetimi, sinema filmi yapımı, felaket kurtarma, parsel teslimatı, depo yönetimi ve diğer kullanımlar gösterilmektedir. İHA çok işlevliliği ve taşınabilirliği nedeniyle, özellikle İHA talebi daha hızlı büyümektedir. Bu nedenle insanların askeri, güvenlik keşif ve gözetim için nesneleri gerçek zamanlı olarak tespit etmek için İHA (dronlar) ile çalışan sistemlere ihtiyacı bulunmaktadır. Makine öğrenme algoritmaları kullanarak kamera ve İHA görüntülerinden nesneleri algılamaya yönelik yaklaşımlar bulunmaktadır. Bilgisayarlı görme sistemlerinin büyümesi, büyük eğitim ve test veri kümeleri, daha hızlı GPU ve CPU kullanarak daha iyi algoritmalar başlamışlardır. Böylece sistemler, makine öğrenme yaklaşımını kullanarak verileri eğiterek ve sınıflandırarak en son nesne algılama sistemini elde etmektedirler. Bir görüntüde farklı yönlendirme, arka plan gürültülü olduğundan, nesne algılama kolay bir iş değildir. Nesne algılamanın amacı, İHA'daki görüntüleri ve video yayınlarını ortak kategorilere ayırmaktır (Ayalew ve Pooja, 2019).

İHA'lar son yıllarda dünya çapında kullanımının artmasıyla birlikte derin öğrenme algoritmaları birçok alanda olduğu gibi savunma ve güvenlik alanlarında da kullanımı artmıştır. Güvenlik ve savunma alanlarında bulunan birçok nesne ve bu nesnelerin tanımlanması, günümüzde buna yönelik birçok derin öğrenme algoritmalarının kullanılması hızla yaygınlaşmıştır. İHA'ların kullanılmasının yaygınlaşması bu hava araçlarının yapay zekâ algoritmalarıyla farklı ve çeşitli görev niteliklerinin ortaya çıkmasına sebep olmuştur.

Sivil ve askeri insansız uçaklar, otonom görevlerin yerine getirilmesi için sistem içi yapay zekâ kullanımı artmıştır. Birçok görevde Global Konumlandırma Sistemi ve

engel tespiti ve kaçınma için sensörler vb. kullanılmaya başlanmıştır. İHA'ların nesne tespitinde tam bağımsız olabilmeleri için, insan kaynaklı verilerin girdi olarak sunulması gerekmektedir (Radovic ve ark., 2017).

İHA'lardaki kameralar ortamdaki fotoğraf ve video çekecek ve nesne algılama ve sınıflandırma metodolojisi yöntemiyle işleme için kullanılacak ve istenen uygulama için kullanılabilir. Bir nesne algılama uygulamasının verimli bir şekilde çalışması için önce modelde, nesnelerin sınıflandırılmasını, yani modelin veri kümesinin görüntüsünün veya videosunun bir araç, insanlar, ağaç veya başka bir şey olup olmadığını tanımlaması gerektiği anlamına gelmesi gerekmektedir. Model sınıflandırma ile eğitildikten sonra, tüm görüntünün hangi piksellerinin hedef nesne olduğunu ve nesnenin sınır çerçevesini bulmasını içeren model nesne algılamasını eğitilmesi gerekmektedir (Ayalew ve Pooja, 2019).

İHA'ların yüzlerce nesne kategorisini algılamasına izin vermek için CNN kullanılmakta ve nesne tespit algoritmalarını kullanmasını sağlamaktadır. Veri güdümlü teknolojiler geliştikçe, farklı geleneksel bilgisayar görme araştırma problemlerini çözmek için makine öğrenme algoritması, özellikle derin öğrenme yaklaşımı yoğun bir şekilde kullanılmıştır (Wang ve ark., 2018).

3. MATERYAL VE YÖNTEM

3.1. Nesne Tespitinde Kullanılan İnsansız Hava Aracının Kamera Özellikleri

Yapılan uygulamada kullanılan Aden E58 drone ile nesne tespit için görüntüler farklı yükseklik ve farklı açılardan elde edilmiştir. Bu görüntüleri elde ederken düşük maliyetli bir İHA olarak (drone) ile nesne tespit yapılmıştır. Aden E58 drone ürünün tercih edilmesinde kamera özelliğinin 720p HD olması ve düşük özellikli görüntülerde nesne tespit uygulamasında başarılı olabildiğini görmek olmuştur.



Şekil 3.1. Aden E58 Drone (Dronenet, 2020)

Yukarıda Şekil 3.1.'de gösterilen Aden E58, İHA (Drone) uygun maliyeti ve kamera özelliğinin düşük pikseli olması nesne tespit uygulamasında tespitte etkisi incelenmiştir.

3.2. Nesne Tespitteki Değerlendirme İçin Yöntemler

3.2.1. Nesne tespit değerlendirmesi için ortalama hassasiyet

Ortalama hassasiyet (AP), Fast R-CNN, SSD vb. nesne tespitlerinin doğruluğunu ölçmede kullanılan popüler bir metriktir. AP 0'dan 1'e kadar R değeri için AP değerini hesaplamaktadır (Hui, 2018).

AP, mevcut algoritmalar göz önüne alındığında bir verinin ilgili olma olasılığını tahmin eden bir model öğrenmektedir. Bununla birlikte, yüksek AP elde etmek için sadece verilerin iyi bir şekilde sıralanması gerekmektedir. İyi olasılıkları bulmak için gerekenden daha zor bir problemin çözülmesini gerektirmekte ve muhtemelen aynı AP performansını elde etmek için daha fazla eğitim verisi gerekmektedir (Yue ve ark, 2007).

3.2.2. Hassasiyet, hatırlama ve F1

Bir modelden tahmin edilen veriler göz önüne alındığında, hassiyeti (P) hızlı bir şekilde hesaplamak ve hatırlamak için karışıklık matrisi kullanışlıdır. İkili sınıflandırma için bir karışıklık matrisi dört farklı sonucu göstermektedir. Bunlar gerçek pozitif (TP), yanlış pozitif (FP), gerçek negatif (TN) ve yanlış negatif (FN)'lerdir. Gerçek değerler sütunları, tahmin edilen değerler satırları oluşturmaktadır. Satır ve sütunların kesişimi dört sonuçtan birini göstermektedir (Koehrsen, 2018). Çizelge 2.3'de bu satır ve sütunları gösterebiliriz.

Tablo 3.1. Karışıklık matrisi

Karışıklık matrisi		Gerçek değer	
		POZİTİF	NEGATİF
Tahmin değer	POZİTİF	TP	FP
	NEGATİF	FN	TN

TP, pozitif örneğin doğru sınıflandırılması,

TN, negatif örneğin doğru sınıflandırılması,

FP, pozitif örneğin yanlış sınıflandırılması,

FN, negatif örneğin yanlış sınıflandırılması olarak açıklanmaktadır.

Güvenilir ve kullanışlı bilgiler elde etmek için her görüntüde kaç TP, kaç FP ve kaç FN görüldüğünü bilmek gerekmektedir. Bunun için görüntü başına karışıklık matrislerini hesaplamak gerekmektedir. Sınıflandırma görevinde karışıklık matrisi her öngörü satırına (orijinal sınıf) ve öngörülen sınıfın sütununa yerleştirilerek kolayca oluşturulmaktadır. Görüntü başına karışıklık matrislerini hesapladıktan sonra bilgileri elde etmek için bu görüntülerin her sınıfı TP, FP ve FN sayısı elde edilmektedir. Azalan görüntüleri toplam FP veya FN sayısına göre sıralayarak modelin en kötü performansı gösteren görüntüleri bulmak için Tensorboard kullanılmaktadır (Malla C. U. 2019).

Sınıfların çok dengesiz olduğu durumda ve bilginin alınması esnasında P, alaka düzeyinin ölçüsünü, hatırlama (R) ise alakalı kaç sonucun döndürüldüğünü göstermektedir. P ve R eğrisi, farklı eşik için P ve R arasındaki dengeyi göstermektedir. Eğrinin altındaki yüksek bir alan hem yüksek R hem de yüksek P temsil etmektedir. Yüksek P düşük TN oranı ile yüksek R ise düşük FN oranı ile ilgilidir. Her ikisi için de yüksek puanlar, sınıflandırıcının doğru sonuçlarını (yüksek P) ve tüm olumlu sonuçların çoğunu (yüksek R) döndürdüğünü göstermektedir. Yüksek R ancak düşük P'e sahip

birçok sonuç döndürmektedir. Yüksek P ancak düşük R'ye sahip bir sistem çok az sonuç vermektedir. Öngörülen etiketlerin çoğu eğitim etiketlerine kıyasla doğru olduğundan yüksek P ve yüksek R özelliğine sahip ideal bir sistem, tüm sonuçları doğru şekilde etiketlenmiş olarak getirmektedir (Scikit Learn, 2019).

P ve R, olumlu tahmin edilen TP vakaların oranıdır. Bu + P (Tahmini Pozitif) kuralıyla TP vakaların kapsamını ölçmektedir. P kuralının kaç tane ilgili veriyi aldığını yansıtmaktadır. Bir başka deyişle ilgili birçok verinin olduğu hangi alt kümeyi bulunduğu gerçekten önemli olmadığı ve olmayan verilerin alaka düzeyi hakkında hiçbir şey bilinmediği varsayımlarına dayanmaktadır (David, 2011).

P, aşağıdaki gibi TP sayısı artı FP sayısı üzerinden TP sayısı olarak tanımlanmaktadır (Scikit, 2019).

$$P_{Tp}/(Tp + Fp) \quad (3.1)$$

R, aşağıdaki gibi TP sayısı üzerindeki artıların TP yanı sıra FN olarak tanımlanmaktadır (Scikit, 2019).

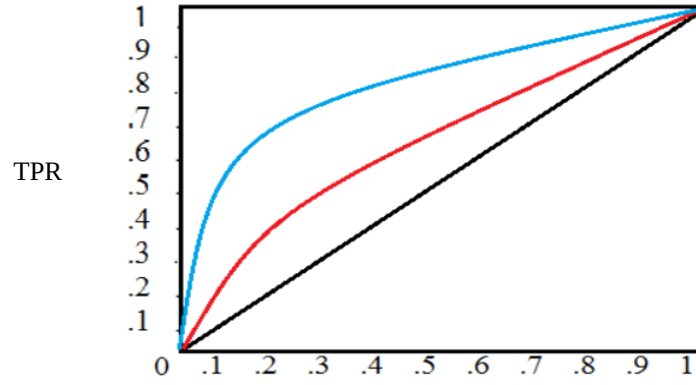
$$R = Tp/(Tp + Fn) \quad (3.2)$$

Aynı zamanda P ve R'nin harmonik ortalaması alınarak (F1) puanı elde edilmektedir (Scikit Learn, 2019).

$$F1 = 2. (P. R/(P + R)) \quad (3.3)$$

P bir sınıflandırıcının eşiğini düşürmenin, döndürülen sonuç sayısını artırarak paydayı artırabileceğini göstermektedir. Eşik daha önce çok yüksek ayarlanmışsa, yeni sonuçların tümü TP olabilmektedir. P artmış olacaktır. Önceki eşik çok düşükse, eşik daha da düşürmek FP yaratarak P'si azaltmış olacaktır. R sınıflandırıcı eşiklerine bağlı olmadığı için sınıflandırıcı eşiklerinin düşürülmesinin, TP sonuçların sayısını artırarak hatırlamayı artırabileceği anlamına gelmektedir. P dalgalanırken eşik düşürmenin hatırlamayı R değiştirmeden bırakması da mümkün olacaktır. R ve P arasındaki ilişki, eşikteki küçük bir değişiklik hatırlamayı sadece küçük bir kazançla, P'si de önemli ölçüde azaltmaktadır (Scikit Learn, 2019).

Bir sınıflandırma modelinin performansını göstermek için diğer ana görselleştirme tekniği ise Alıcı Çalışma Karakteristiği (ROC) eğrisidir. ROC eğrisi, bir pozitif tanımlamak için eşiği değiştirdikçe, P ile R hassas ilişkisinin nasıl değiştiğini göstermektedir. Eşik ise bir veri noktasının pozitif sınıfta dikkate alındığı değeri temsil etmektedir. Eşiği değiştirerek, hatırlama dengesine karşı doğru P elde etmek mümkün olmaktadır. Bir ROC eğrisi, y eksenindeki gerçek pozitif oranı (TPR), x eksenindeki yanlış pozitif oranı (FPR) çizmektedir. TPR hatırlamayı ve FPR yanlış alarm olasılığını göstermektedir. Bunların her ikisini de karışıklık matrisinden hesaplamak mümkün olmaktadır. Tipik bir ROC eğrisi aşağıda Şekil 3.2’ de gösterilmiştir (Koehrsen W. 2018).



Şekil 3.2. ROC eğrisi

P ve R eğrileri sınıf dağılımında büyük bir eğriye sahip görevler için ROC eğrilerine alternatif olarak görülmektedir. ROC alanı ile PR alanı arasındaki önemli bir fark eğrilerin görselliğinde yatmaktadır. PR eğrilerine bakmak, ROC alanında görünmeyen algoritmalar arasındaki farkları ortaya koymaktadır. Algoritmaların performansları ROC alanında karşılaştırılabilir gibi görünmektedir (Davis ve Goadrich, 2006).

3.2.3. Kesişim birliği

IoU, belirli bir veri kümesindeki bir nesne tespit doğruluğunu ölçmek için kullanılan bir değerlendirme metriği olarak adlandırılmaktadır. IoU sadece nesne tespit performansını değerlendirmek için kullanılan bir değerlendirme ölçütüdür. İncelendiğinde IoU’nun basitçe bir oran olduğu görülmektedir. Payda, öngörülen sınırlayıcı kutu ile yer-doğruluk sınırlayıcı kutu arasındaki çakışma alanını

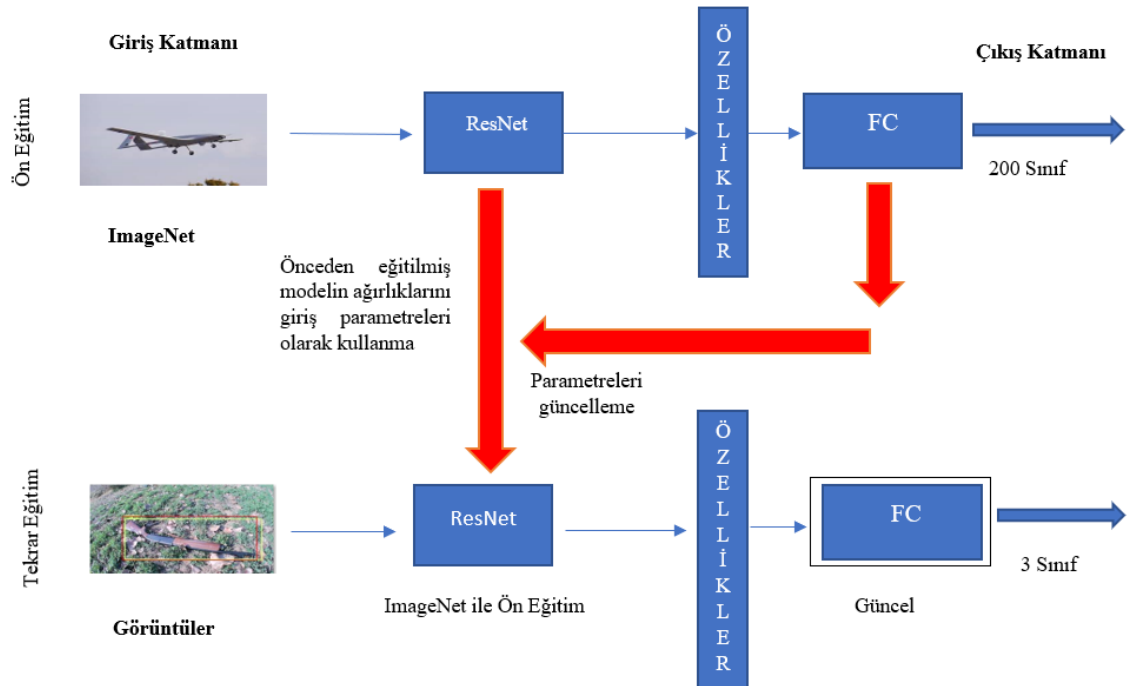
hesaplanmaktadır. Payda, birleşme alanıdır veya daha basit olarak hem öngörülen sınırlayıcı kutunun hem de gerçek-doğruluk sınırlayıcı kutunun kapsadığı alandır. Örtüşme alanını birleşme alanına bölmek nihai sonuç elde edilmesine yardımcı olmaktadır. IoU, nesne tespitlerini değerlendirmek için sınırlama kutularının mümkün olduğunca yakın eşleşmesini dikkate almaktadır (Rosebrock, 2016).

$$IoU = \text{Örtüşme Alanı} / \text{Birleşme Alanı}$$

Etkili bir CNN tabanlı nesne algılama ağı nesne sınırlarını ve doğrudan özellik haritalarındaki piksel tabanlı sınıflandırma puanlarını tahmin etmek için tamamen kıvrımlı bir ağ mimarisini benimsemektedir. Özellikle, sınırlayıcı kutu tahmini için yeni bir IoU kaybı işlevinden yararlanmaktadır. IoU kaybı, doğrudan öngörülen sınırlama kutusu ile zemin gerçeği arasındaki maksimum çakışmayı zorlar ve tüm bağlı değişkenleri bir bütün olarak birlikte geri almaktadır (Yu, Jiang ve ark., 2016).

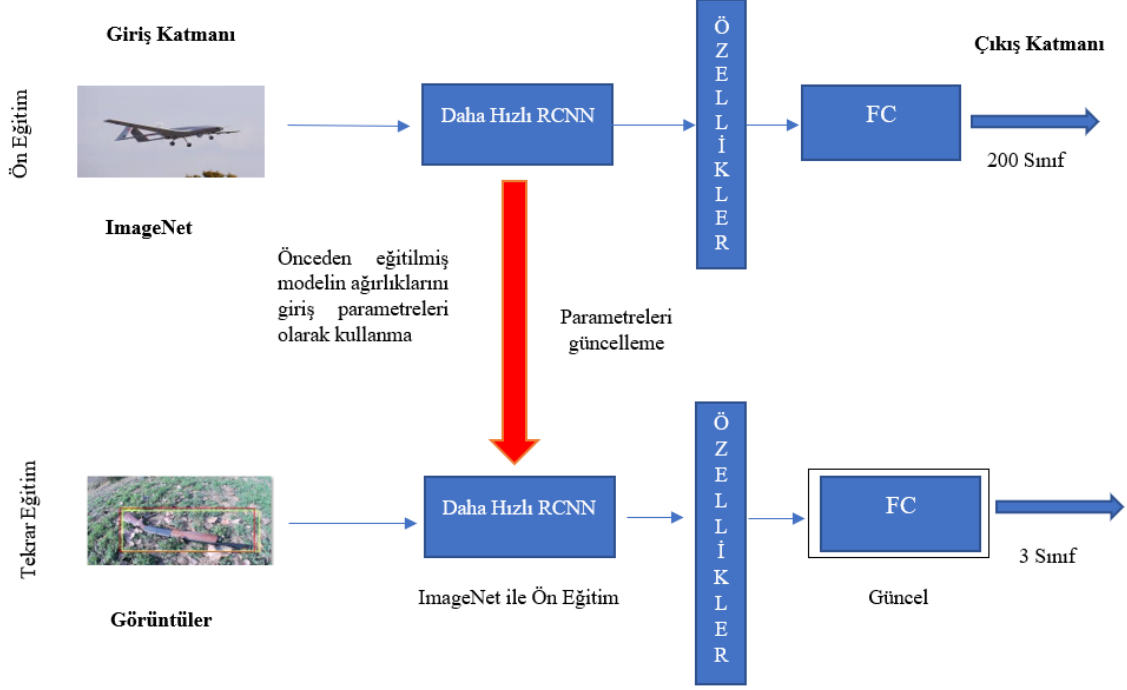
3.3. İnsansız Hava Araçları ile Nesne Tespit Aşamaları

Çalışmada kullanılan R-CNN algoritmaları (Faster R-CNN ve ResNet) nesne tespit aşamasında İHA'larla elde edilen görüntüleri eğitmek için eğitime olumsuz etki sağlayacak ağırlıkları dikkate almaktadır. Nesne tespit mimarisi Şekil 3.3 ve Şekil 3.4'de Faster R-CNN ile ResNet algoritmaları gösterilmektedir.



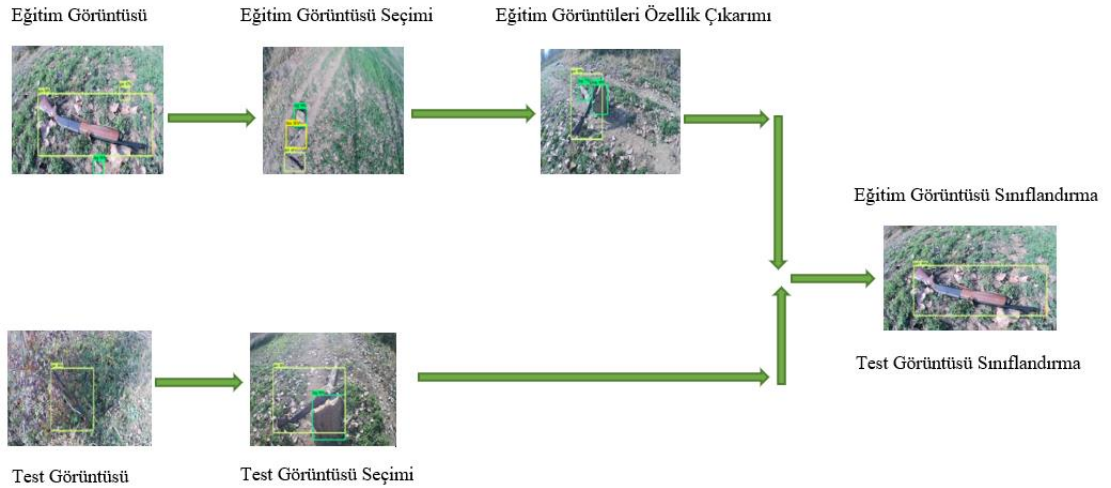
Şekil 3.3. ResNet nesne tespit mimarisi

Faster R-CNN ile ResNet nesne tespit mimarileri arasında görülen fark ise özellik haritaları elde edildikten sonra tekrar eğitime test görüntüleri ile birlikte girmesidir.



Şekil 3.4. Faster R-CNN nesne tespit mimarisi

Çalışmada İHA'larla elde edilen görüntüler test ve eğitim olarak ayrılmaktadır. Kullanılan Faster R-CNN ve ResNet algoritmaları her iki grubunda içinden görüntüler seçerek özellik haritası çıkarmaktadır. Özellik haritası çıkarılan görüntüler SVM ile sınıflandırılarak görüntüler içerisinde nesne (silah) tespiti yapmaktadır. Bu işleyiş Şekil 3.5'de gösterilmektedir.



Şekil 3.5. Nesne tespit işleyişi

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Eğitim ve Test Sonuçları ve Değerlendirme Adımları

İHA'dan elde edilen resimlerde derin öğrenme ile nesne tespiti (silah tespiti) kullanılan programlama dili Python'tur. Nesne tespiti esnasında Tensorflow kütüphanesini de kullanarak çalışmada başarılı sonuçlar elde edilmiştir. Bu çalışmada nesne tespitine kadar izlenen basamaklar sırasıyla şöyledir.

4.1.1. Tensorflow nesne tespit kurulumu ve anaconda virtual environment oluşturma

Tensorflow nesne tespit API'yı uygulaması yapılacak bilgisayara yüklemesini tamamladıktan sonra C klasöründe tensorflow1 isminde klasör oluşturup indirilen models klasörünün içine koyulmuştur. Uygulamada nesne tespit yolunu (C:\tensorflow1\models\research\object_detection) olarak belirlenmiştir. Oluşturulan object detection klasörünün içine Model zoo'dan Faster R-CNN Inception v2 COCO'yu indirip buraya bıraktan sonra tensorflow1 klasörün içinde bulunan dosyalara eklenmiştir.

Klasör oluşturduktan sonra komut penceresini yönetici olarak açtıktan sonra (C:\> conda create -n tensorflow1 pip) komutu yazılmaktadır. Virtual environmenti aktifleştirmek için ise (C:\> activate tensorflow1) komutu yazılmaktadır. Komut yazıldıktan sonra virtual environment aktifleştikten sonra sırayla aşağıdaki kütüphaneler kurulmaktadır.

- (tensorflow1) C:\> pip install --ignore-installed --upgrade tensorflow-gpu
- (tensorflow1) C:\> conda install -c anaconda protobuf
- (tensorflow1) C:\> pip install pillow
- (tensorflow1) C:\> pip install lxml
- (tensorflow1) C:\> pip install jupyter
- (tensorflow1) C:\> pip install matplotlib
- (tensorflow1) C:\> pip install pandas
- (tensorflow1) C:\> pip install opencv-python

Kütüphaneler yüklendikten sonra pythonpath ayarlama ve protobufları derleme için aşağıdaki komutlar yazılacaktır.

- (Tensorflow1) C:\> set PYTHONPATH=C:\tensorflow1\models;
C:\tensorflow1\models\research; C:\tensorflow1\models\research\slim
- (tensorflow1) C:\> cd C:\tensorflow1\models\research protoc --python_out=.
 .\object_detection\protos\anchor_generator.proto
 .\object_detection\protos\argmax_matcher.proto
 .\object_detection\protos\bipartite_matcher.proto
 .\object_detection\protos\box_coder.proto
 .\object_detection\protos\box_predictor.proto
 .\object_detection\protos\eval.proto
 .\object_detection\protos\faster_rcnn.proto
 .\object_detection\protos\faster_rcnn_box_coder.proto
 .\object_detection\protos\grid_anchor_generator.proto
 .\object_detection\protos\hyperparams.proto
 .\object_detection\protos\image_resizer.proto
 .\object_detection\protos\input_reader.proto
 .\object_detection\protos\losses.proto
 .\object_detection\protos\matcher.proto
 .\object_detection\protos\mean_stddev_box_coder.proto
 .\object_detection\protos\model.proto
 .\object_detection\protos\optimizer.proto
 .\object_detection\protos\pipeline.proto
 .\object_detection\protos\post_processing.proto
 .\object_detection\protos\preprocessor.proto
 .\object_detection\protos\region_similarity_calculator.proto
 .\object_detection\protos\square_box_coder.proto
 .\object_detection\protos\ssd.proto
 .\object_detection\protos\ssd_anchor_generator.proto
 .\object_detection\protos\string_int_label_map.proto
 .\object_detection\protos\train.proto
 .\object_detection\protos\keypoint_box_coder.proto
 .\object_detection\protos\multiscale_anchor_generator.proto
 .\object_detection\protos\graph_rewriter.proto
 .\object_detection\protos\calibration.proto

Aktifleşen Tensorflow1 klasöründe protobuflar derlemeyi müteakip aşağıdaki satırlar ile önce build ve install satırını çalıştırılması gerecektir. Daha sonra Jupyter notebooku açabilmek için sıradaki komutu yazmamış gerecektir.

- (Tensorflow1) C:\tensorflow1\models\research> python setup.py build
- (Tensorflow1) C:\tensorflow1\models\research> python setup.py install
- (Tensorflow1) C:\tensorflow1\models\research\object_detection> jupyter notebook object_detection_tutorial. Ipynb

4.1.1. Veri setini oluşturma ve etiketleme

Yapılan çalışmada İHA'dan elde edilen görüntülerle veri seti oluşturulmuştur. Veri seti için farklı ortamlarda olmasına dikkat ederek 200 adet resmin 160 adet resim train klasörüne 40 adet resim test klasörüne ayrılmıştır.

Toplanan resimlerin boyutunu eğitimin uzun sürmemesi için küçülterek resimleri tek tek etiketlenmesi gerekmektedir. Görüntü (veri seti) toplama ve etiketleme, çalışma da aşağıdaki Şekil 4.1, Şekil 4.2 ve Şekil 4.3'de gösterilmektedir.



Şekil 4. 1. İHA'lardan elde edilen görüntüler



Şekil 4. 2. Resimlerde etiketlenen nesneler-1



Şekil 4. 3. Resimlerde etiketlenen nesneler-2

Label Map Oluşturma, tüm görüntüleri etiketlemek için görüntü üzerinde bulunması gereken nesneleri tek tek seçerek LabelImg ile etiketleme işlemi yapılmıştır. Etiketlenen her görüntü .xml dosyasına çevirilerek bu dosyaları tfrecords'a dönüştürülmüştür. Oluşturulan generate_tfrecord.py dosyası herhangi bir metin editörü ile sınıf bilgilerini aşağıdaki gibi Şekil 4.4'te gösterildiği şekilde bag, fake ve gun şeklinde oluşturulmuştur.

```
# TO-DO replace this with label map
def class_text_to_int(row_label):
    if row_label == 'bag':
        return 1
    elif row_label == 'fake':
        return 2
    elif row_label == 'gun':
```

```

return 3
else:
    None

```

4.1.2. Eğitim ayarları ve eğitimi gerçekleştirme

Eğitim ayarlarının yapılması aşamasında hangi modelleri ve hangi parametrelerin kullanılacağını kullandığımız programlama dili üzerinde önce çalışmak istenilen modelin yolu C:\tensorflow1\models\research\object_detection\samples\configs klasörüne kullandığımız modelleri yapılandırıldıktan sonra aşağıdaki komut satırları çalıştırılmıştır.

Kullanılan modeller;

- Faster_rcnn_inception_v2_coco_2018
- Rfcn_resnet101_coco_2018

Çalıştırılan komut satırları;

- input_path : "C:/tensorflow1/models/research/object_detection/test.record"
- label_map_path:
- "C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"

Eğitimin Gerçekleştirilmesi komut penceresinden TensorFlow aktif edildikten sonra Python path ile aşağıdaki komutlar çalıştırılarak eğitim başlatılmıştır.

- python train.py --logtostderr --train_dir=training/ --
pipeline_config_path=training/faster_rcnn_inception_v2_pets.config
- python train.py --logtostderr --train_dir=training/ --
pipeline_config_path=training/rfcn_resnet101_inception_v2_pets.config.

Nesne (silah) tespiti için özet şeklinde yer alan adımlar aşağıda anlatılacaktır.

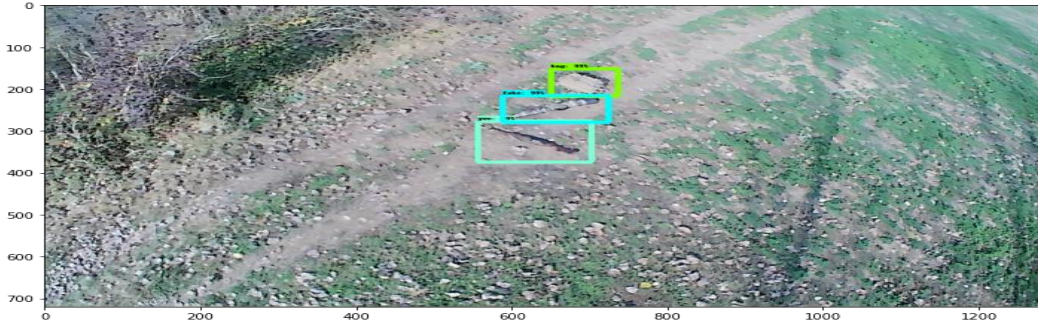
- Önce os, cv2, numpy, sys ve tensorflow kütüphaneleri programa yüklenmiş,
- Yol tanımı (path) oluşturulmuş,
- Nesne tespiti içeren dizinin adı konsola yazılmış,
- Görüntü yolu belirtilmiş,
- Nesne tespitinde tanımlayabileceği sınıf sayısı belirtilmiş,
- Etiket haritası ve Label harita indekslerini kategori isimlerine göre eşleştirilmiş,
- TensorFlow modeli belleğe yüklenmiş,

- Nesne sınıflandırıcısı için giriş ve çıkış (veri) tanımlanmış böylece giriş görüntü olacak şekilde çıktı verileri tespit edilen nesneler olacaktır,
- Giriş olarak model görüntüyle çalıştırma ve gerçek algılama işlemini gerçekleştirmiş,
- Sonuçlara eğitim tamamlandıktan sonra görüntüyü göstererek Nesne Tespiti yapılmış olacaktır.

Eğitim başladıktan sonra kullanılan Faster_rcnn_inception_v2_coco_2018, Rfcn_resnet101_coco_2018) model ile 50.000 iterasyon olacak şekilde devam edilmiştir. Şekil 4.4-Şekil 4.25’de her iki model ile yapılan eğitimin sonucunda elde edilen görüntüler gösterilmektedir.



Şekil 4. 4. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-1)



Şekil 4. 5. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-2)



Şekil 4. 6. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-3)



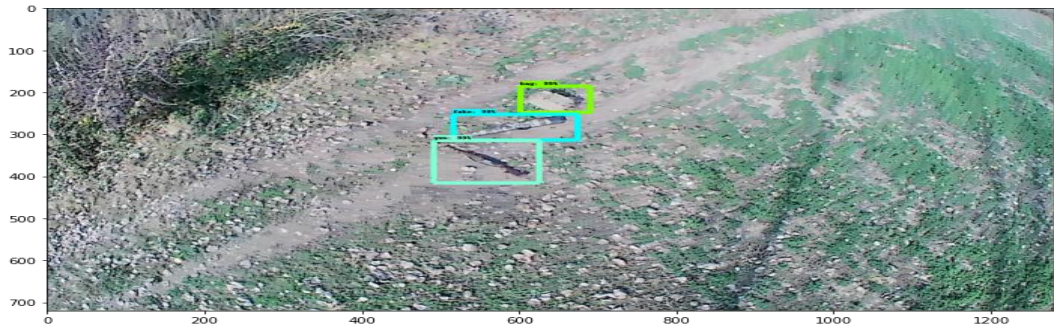
Şekil 4. 7. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-4)



Şekil 4. 8. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-5)



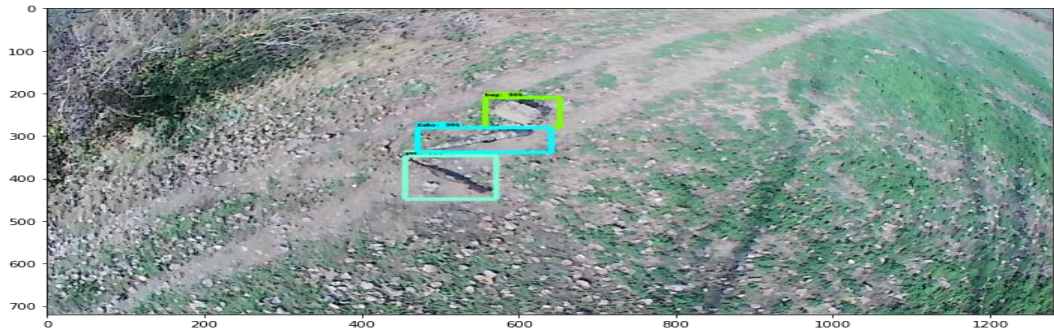
Şekil 4. 9. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-6)



Şekil 4. 10. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-7)



Şekil 4. 11. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-8)



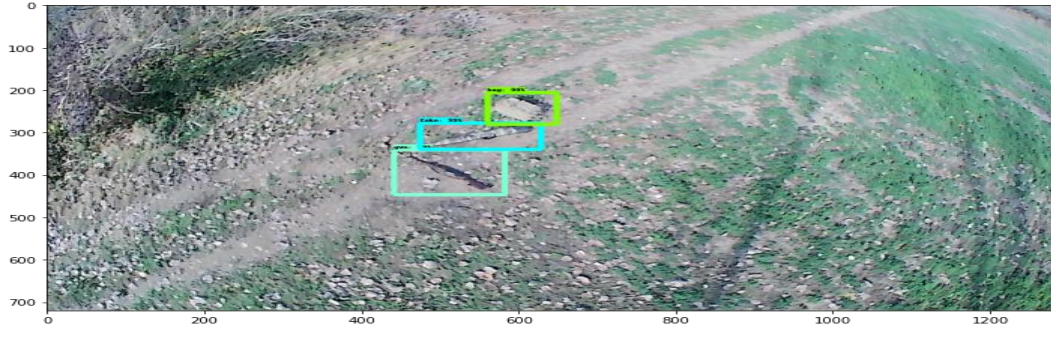
Şekil 4. 12. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-9)



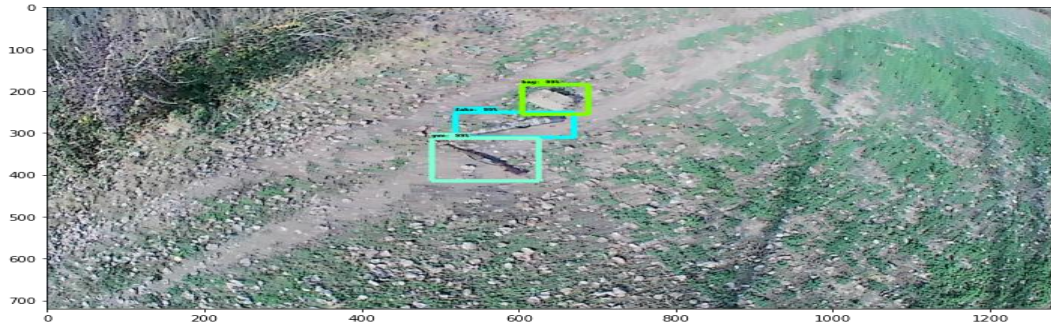
Şekil 4. 13. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-10)



Şekil 4. 14. Faster_rcnn_inception_v2_coco_2018 (Test Resmi-11)



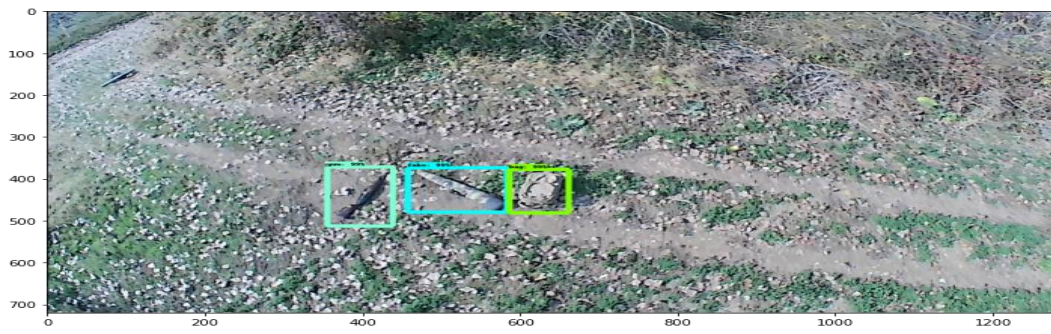
Şekil 4. 15. Rfcn_resnet101_coco_2018 (Test Resmi-1)



Şekil 4. 16. Rfcn_resnet101_coco_2018 (Test Resmi-2)



Şekil 4. 17. Rfcn_resnet101_coco_2018 (Test Resmi-3)



Şekil 4. 18. Rfcn_resnet101_coco_2018 (Test Resmi-4)



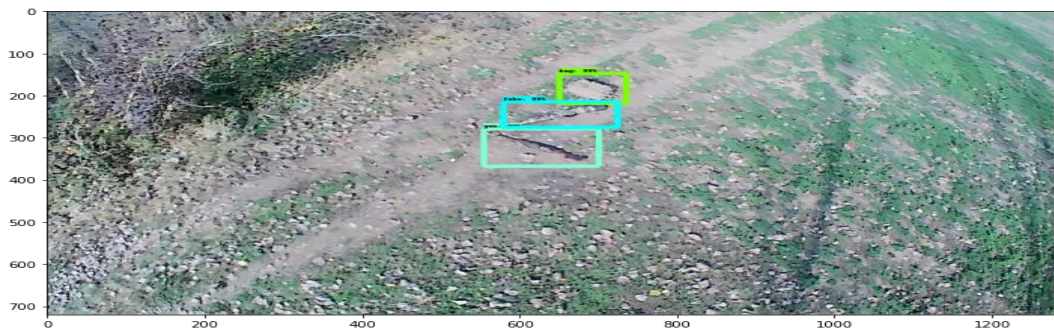
Şekil 4. 19. Rfcn_resnet101_coco_2018 (Test Resmi-5)



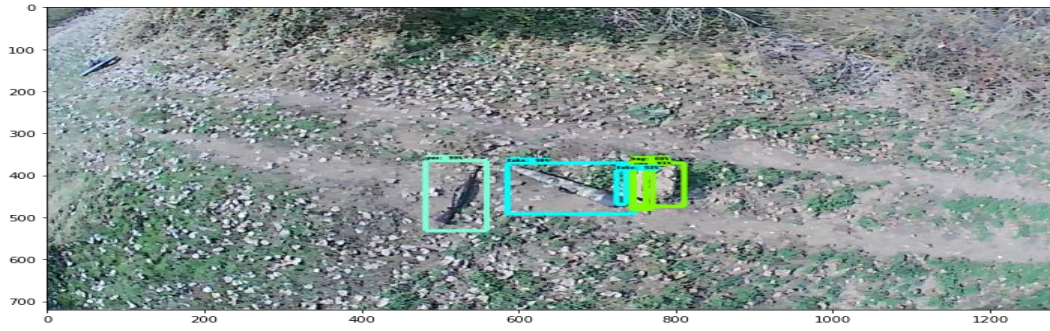
Şekil 4. 20. Rfcn_resnet101_coco_2018 (Test Resmi-6)



Şekil 4. 21. Rfcn_resnet101_coco_2018 (Test Resmi-7)



Şekil 4. 22. Rfcn_resnet101_coco_2018 (Test Resmi-8)



Şekil 4. 23. Rfcn_resnet101_coco_2018 (Test Resmi-9)



Şekil 4. 24. Rfcn_resnet101_coco_2018 (Test Resmi-10)



Şekil 4. 25. Rfcn_resnet101_coco_2018 (Test Resmi-11)

4.2. Eğitim Modellerinin Tensorboard Grafiklerinin Değerlendirilmesi

Tensorboard Tensorflow uygulamasına ait bir metriktir. Tensorboard verilerin görselleştirilmesinde kayıp ve doğruluk gibi deneme metriklerini izlemeyi, model grafiğini görselleştirmeyi, düğümleri daha düşük boyutlu bir alana yansıtmayı ve çok daha fazlasını sağlamaktadır. Eğitim sonucunda elde edilen DetectionBoxes_Precision_mAP (medium), DetectionBoxes_Recall_AR@100 (medium), global_norm_clipped_gradient_norm, global_norm_gradient_norm, global_step_sec, learning_rate, loss, loss1, loss2,

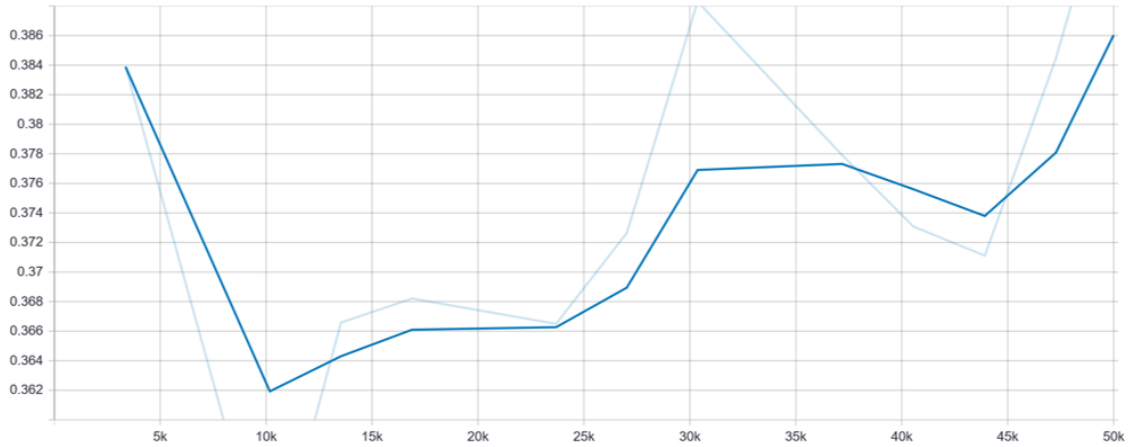
Loss_BoxClassifierLoss_classification_loss, Loss_RPNLoss_localization_loss, Loss_total_loss grafikleri eğitimde kullanılan modeller hakkında bilgiler verecektir.

Tensorboard, modelin nesne tespiti için kullandığı resimler üzerindeki gerçek doğruluk kıyaslamasında kullanılan modellerin ne kadar başarılı olduğunu göstermiştir.

4.2.1. Tensorboard grafiklerinin değerlendirilmesi

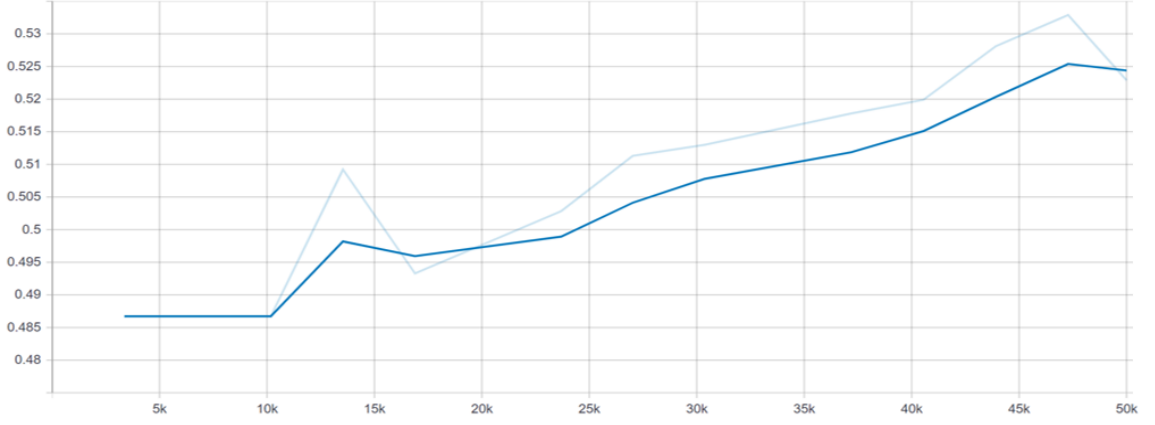
4.2.1.1. Faster_rcnn_inception_v2_coco_2018 modeli grafiklerinin değerlendirilmesi

Detectionboxes_Precision mAP grafiği Şekil 4.26'da gösterilmektedir. mAP puanının bu şekilde gösterilmesi, 32^2 ile 96^2 piksel arasında olan nesneleri yakalaması ile ilgilidir. Eğitimde yaklaşık değeri 0,38'dir. Bu değer düşük boyuttaki nesneleri tespit edebileceğini göstermektedir.



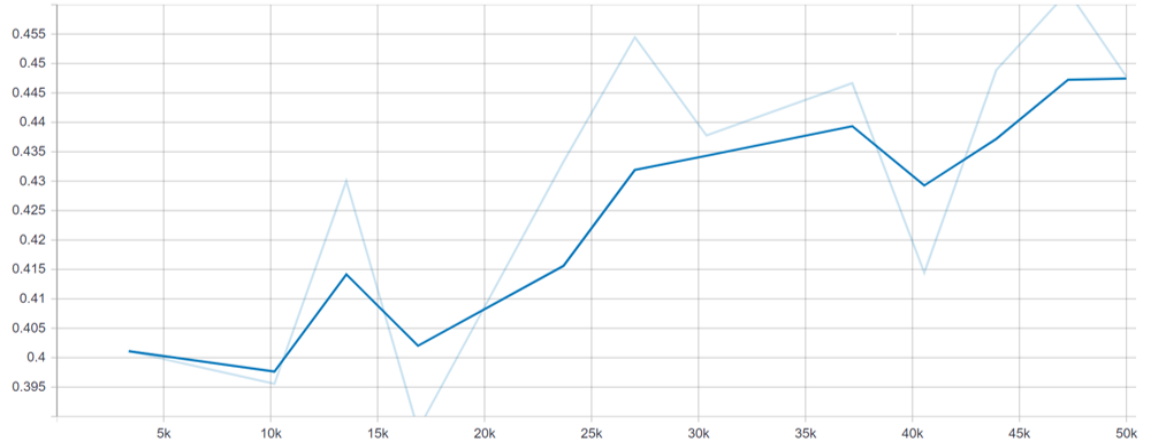
Şekil 4. 26. Faster R-CNN Detectionboxes_Precision mAP grafiği

Detectionboxes_Recall grafiği Şekil 4.27'da gösterilmiştir. Ortalama recall değerini 1 tanımlama ile gösteren bir grafikdir. Tanımlama değeri 0,525 'tir. Bu değer recall değerinin ortalamanın üstünde olduğunu göstermektedir.



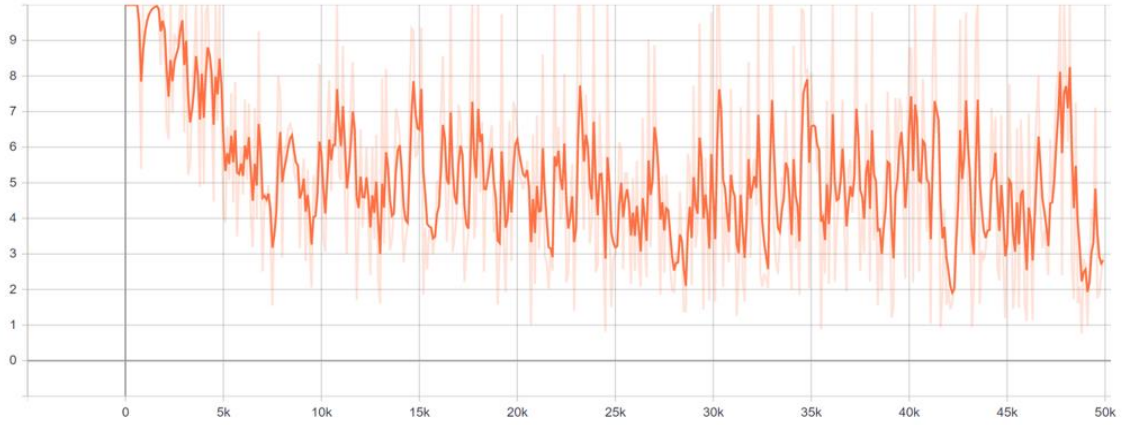
Şekil 4. 27. Detectionboxes_Recall grafiği

Detectionboxes_Recall (medium) grafiği Şekil 4.28’de gösterilmektedir. Bu grafik ortalama değerleri aldığı için recall değerini 0,445 değerine kadar düşerse bile tespit için geri aldıklarını ortalama değere yakın oldukları gözükmemektedir.



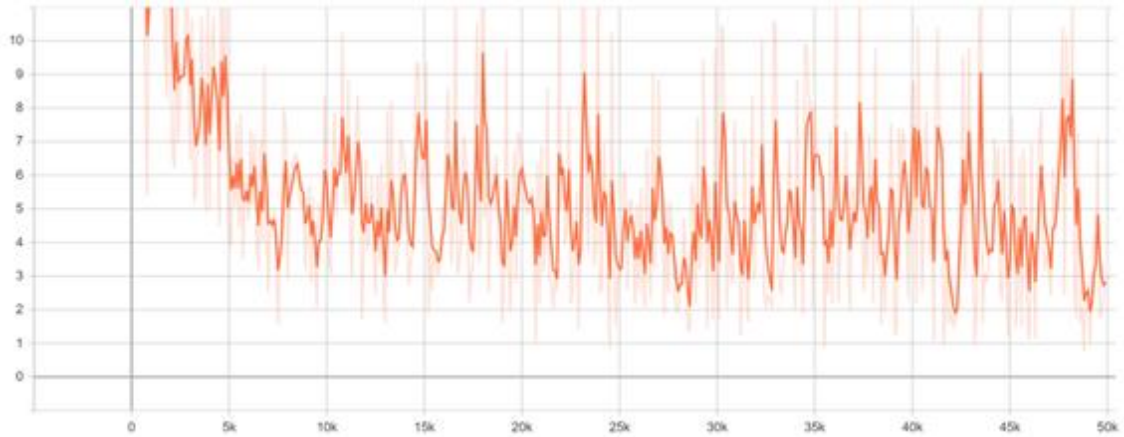
Şekil 4. 28. Detectionboxes_Recall (medium) grafiği

Clipped Gradient Norm ve Global Gradient Norm grafikleri Şekil 4.29 ve Şekil 4.30’da gösterilmektedir.



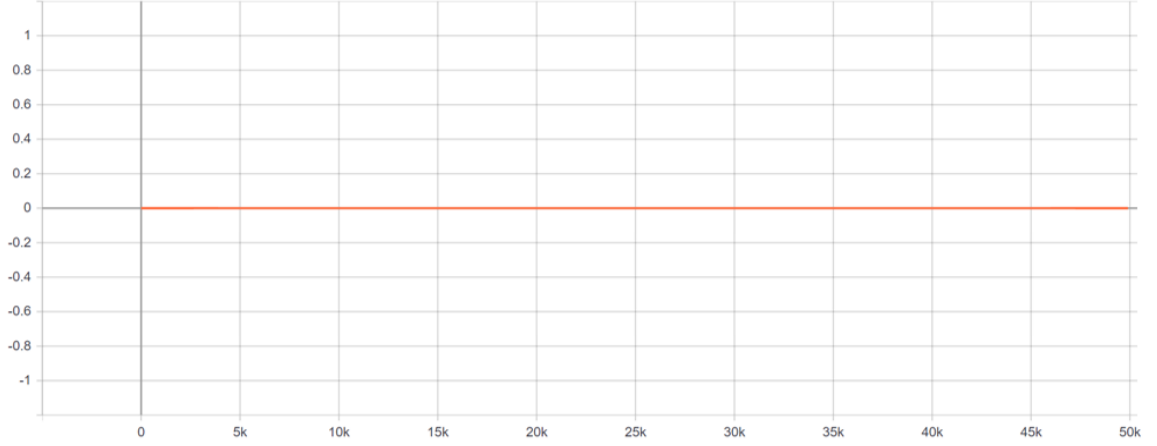
Şekil 4. 29. Clipped gradient norm grafiği

Grafikler eğitim sırasında ağırlıklara yapılan büyük güncellemeleri göstermektedir. Birden fazla eğitimde oluşan kaybın tahmin edilmesi için ağ üzerinden geriye doğru yayılan kayıp türevinin hesaplanmasını ifade etmektedir. Böylece ağırlıklar, öğrenme oranı tarafından kontrol edilmektedir.



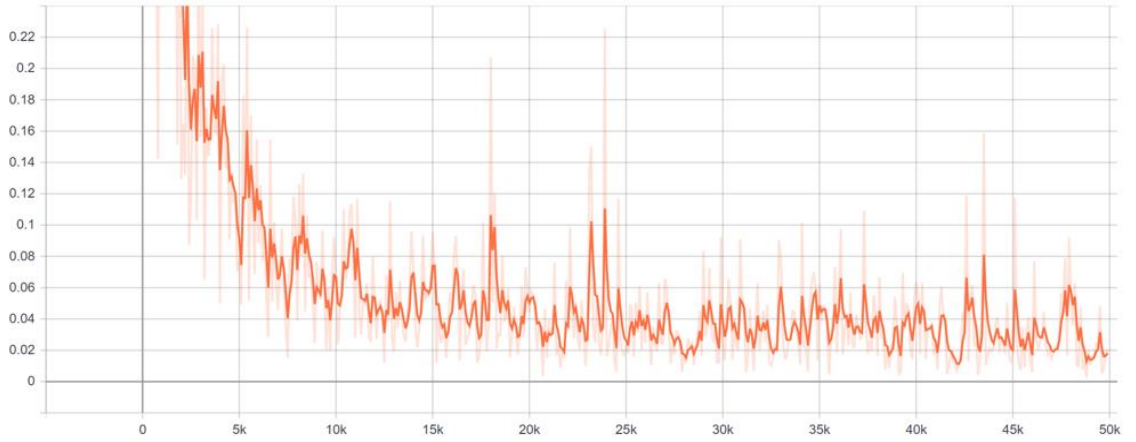
Şekil 4. 30. Global gradient norm grafiği

Learning rate grafiği Şekil 4.31’de gösterilmektedir. Öğrenme oranı olarak da adlandırabilen bu grafik ağıdaki ağırlıkların kayıp eğimini kontrol eden bir parametredir. Genellikle öğrenme oranları kullanıcı tarafından belirlenmiştir. Bu çalışmada learning rate (0,0003) gibi bir değer verdiği için Şekil 4.30’da gösterilmiş olan learning rate sabit bir şekilde öğrendiği ifade etmektedir. Bu oranı kullanıca olarak geçmişte kullanılmış deneyimlerden yararlanarak belirlenmiştir.



Şekil 4. 31. Learning rate grafiği

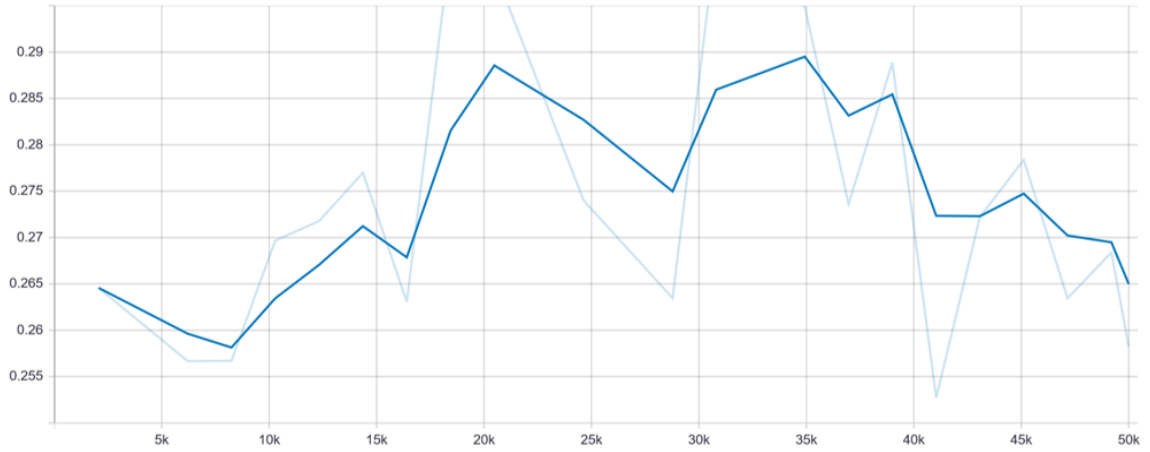
Loss fonksiyon grafiği Şekil 4.32’de gösterilmektedir. Loss fonksiyonu kullanılan modelin tahmini ile gerçek değerin arasındaki farkı ifade etmektedir. Bu yüzden modelin iyi bir model olduğunu loss değerinin az olmasıyla doğru orantılıdır. Kullanılan Faster_rcnn_inception_v2_coco_2018 modelinin tahminlerinin başarılı olduğunu loss fonksiyon grafiğinin (0,02)’ye yani sıfıra (0) yaklaşmış olduğundan anlaşılmaktadır.



Şekil 4. 32. Loss fonksiyon grafiği

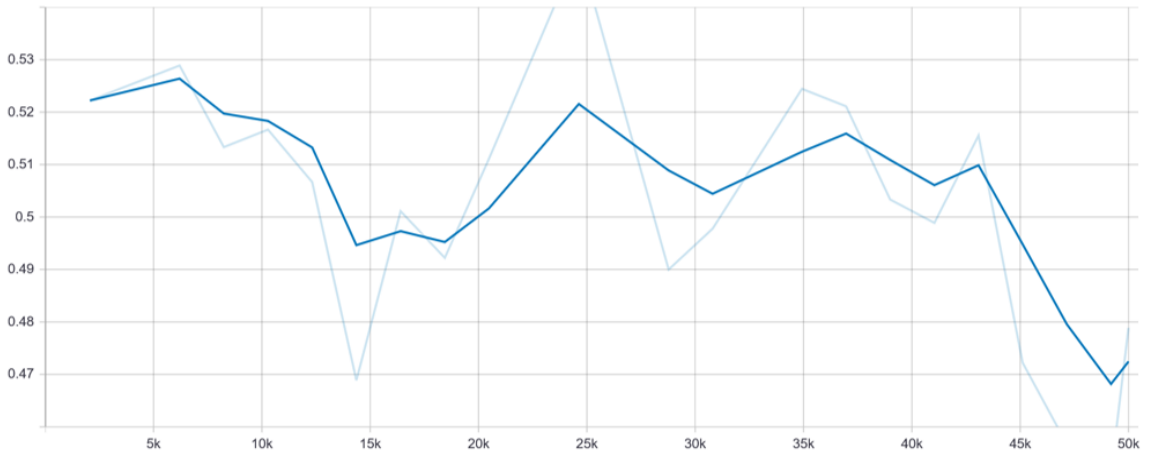
4.2.1.2. Rfcn_resnet101_coco_2018 modeli grafiklerinin değerlendirilmesi

Detectionboxes_Precision mAP grafiği Şekil 4.33’te gösterilmektedir. mAP puanının bu şekilde gösterilmesi, 32^2 daha düşük pikseli olan nesneleri yakalaması ile ilgilidir. Eğitimde yaklaşık değeri (0,29) olarak görülmektedir. Bu değer daha düşük boyuttaki nesneleri tespit edebileceğini göstermektedir.



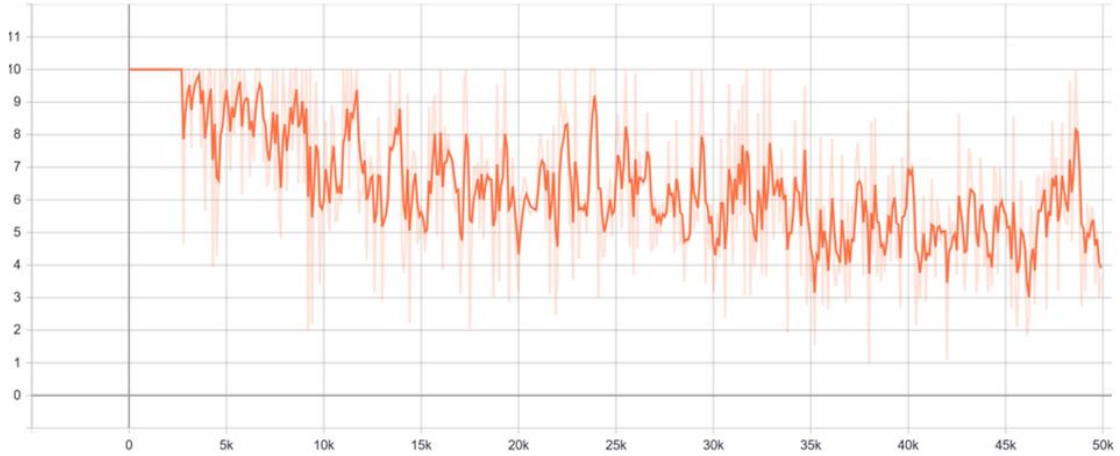
Şekil 4. 33. ResNet101 Detectionboxes_Precision mAP grafiği

Detectionboxes_Recall grafiği Şekil 4.34’de gösterilmiştir. Ortalama recall değeri tepiti yapılacak her nesne için hassasiyet ifade etmektedir. 1 tanımlama ile gösteren bir grafikdir. Kullanılan modelin P değeri (0,527) olarak ölçülmüştür. Bu değer bize seçilen modelin nesne tespitindeki hassasiyetini anlatmaktadır.



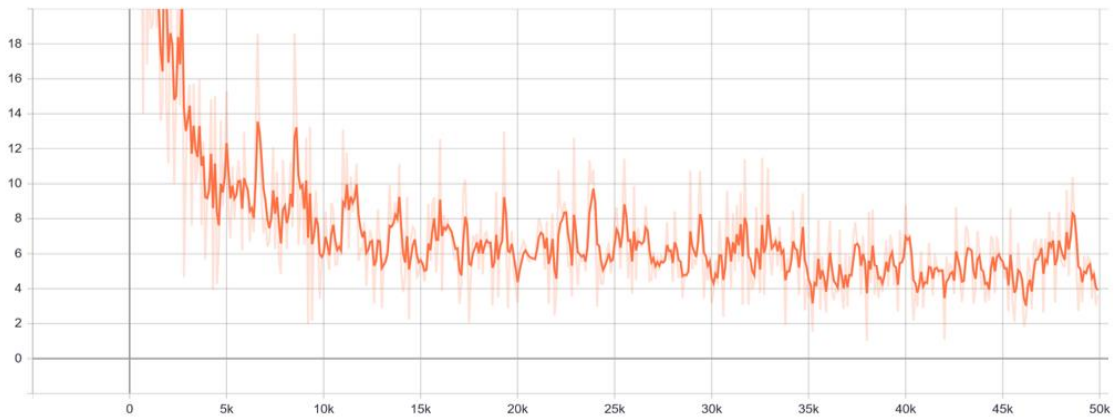
Şekil 4. 34. Detectionboxes_Recall grafiği

Clipped Gradient Norm ve Global Gradient Norm grafikleri Şekil 4.35 ve Şekil 4.36’de gösterilmektedir.



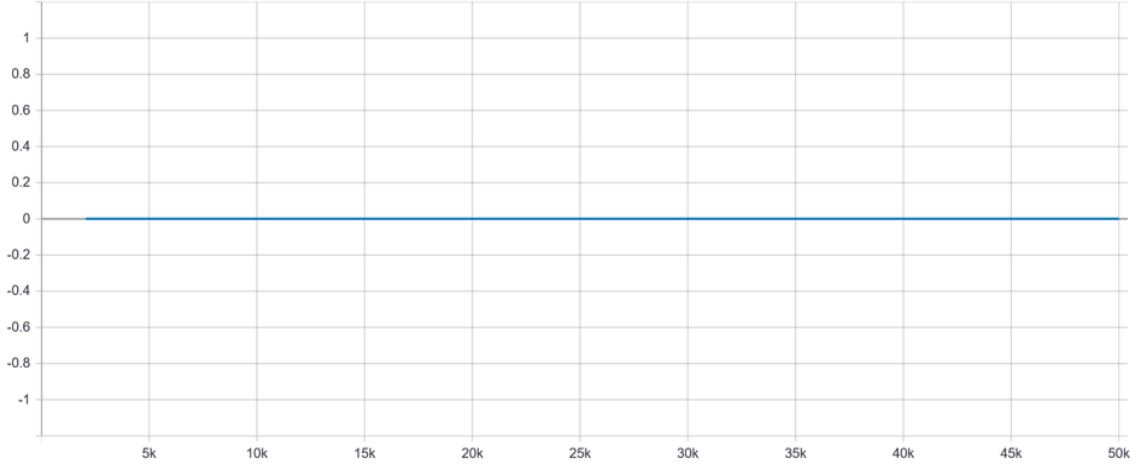
Şekil 4. 35. Clipped gradient norm grafiği

Grafikler eğitim sırasında model tarafından ağırlıklara yapılan büyük güncellemeleri göstermektedir. Birden fazla eğitimde oluşan kaybın tahmin edilmesi için ağ üzerinden geriye doğru yayılan kayıp türevinin hesaplanmasıyla ağırlıkların öğrenme oranındaki adımları ifade etmektedir.



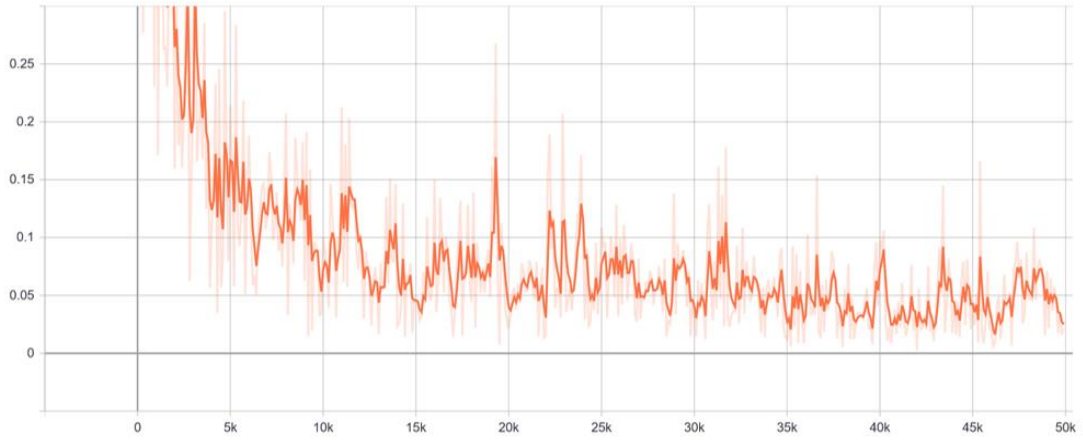
Şekil 4. 36. Global gradient norm grafiği

Learning Rate grafiği Şekil 4.37’de yer almaktadır. Çalışmada kullanılan Rfcn_resnet101_coco_2018 modelinin öğrenme oranını (0,0003) gibi bir değer verilmesi sebebiyle için Şekil 4.37’da gösterilmiş olan learning rate sabit bir şekilde olduğu görülmektedir.



Şekil 4. 37. Learning rate grafiği

Loss fonksiyon grafiği Şekil 4.38’de gösterimiştir. Loss fonksiyonu kullanılan modelin tahmini ile gerçek değerin arasındaki farkı ifade etmesi kullanılan Rfcn_resnet101_coco_2018 modelinin tahminlerinin başarılı olduğunu loss fonksiyon grafiğinin (0,05)’lerin altına yaklaşmış olduğundan anlaşılmaktadır.

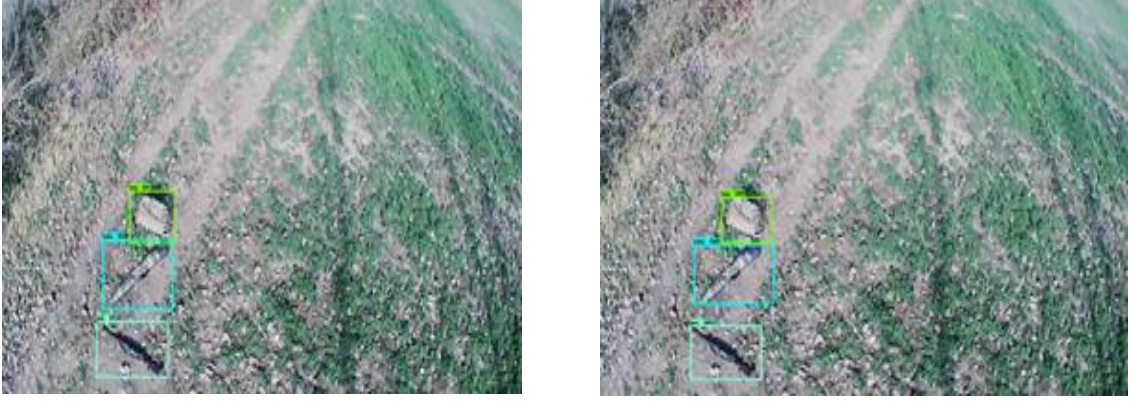


Şekil 4. 38. Loss fonksiyon grafiği

4.2.2. Kullanılan modellerin tahmini ile gerçek değerlerin karşılaştırılması

Çalışmada kullanılan birinci model olarak Faster_rcnn_inception_v2_coco_2018 ile birlikte ikinci model olarak Rfcn_resnet101_coco_2018 yer almıştır. Bu modellerin loss fonksiyon grafiklerinde görüldüğü üzere aldığı değerler yaklaşık (0,02) (0,05)’tir. Kullanılan modellerin kendi nesne tespiti için yaptığı tahminlerin kesin referans değerler arasındaki farkın neredeyse sıfıra (0) yaklaşması bu modellerin başarılı olduğunu

ispatlamaktadır. Eğitim sonunda nesne tespit için gösterilen resimlerde kesin referans değerler ile kendi tahmini gösteren görüntüler Şekil 4.39-Şekil 4.58’de gösterilmektedir. Test resimlerinde çıplak gözle bile zor görülebilecek nesnenin (silah) kolaylıkla tespit edildiği görülmektedir.



(a)

(b)

Şekil 4. 39. Birinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)

(b)

Şekil 4. 40. İkinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)



(b)

Şekil 4. 41. Üçüncü test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)



(b)

Şekil 4. 42. Dördüncü test resmi Faster_rcnn_inception_v2_coco_2018 modelinin
(a) kesin referans değeri, (b) tahmini değeri



(a)



(b)

Şekil 4. 43. Beşinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



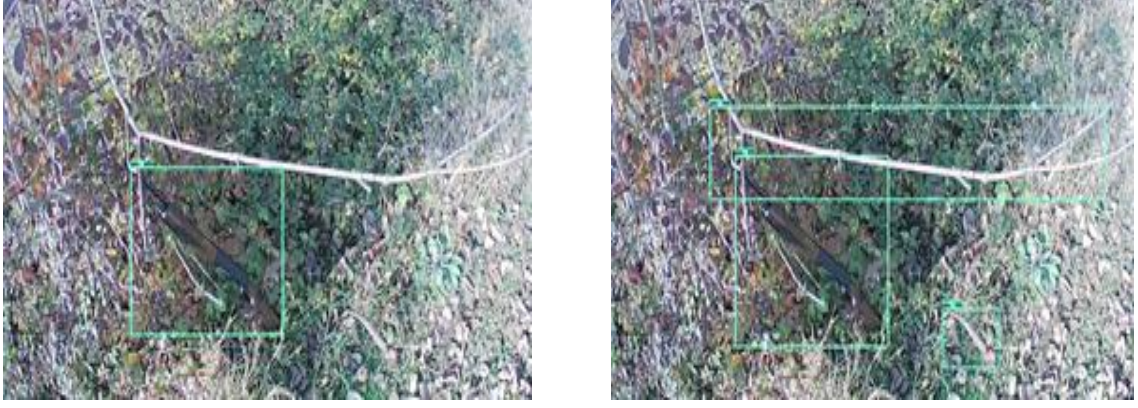
(a) (b)
Şekil 4. 44. Altıncı test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a) (b)
Şekil 4. 45. Yedinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



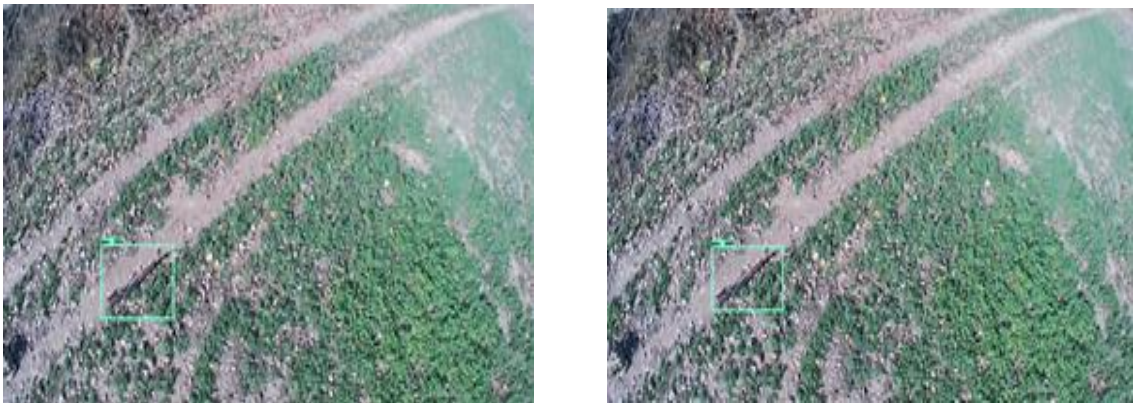
(a) (b)
Şekil 4. 46. Sekizinci test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



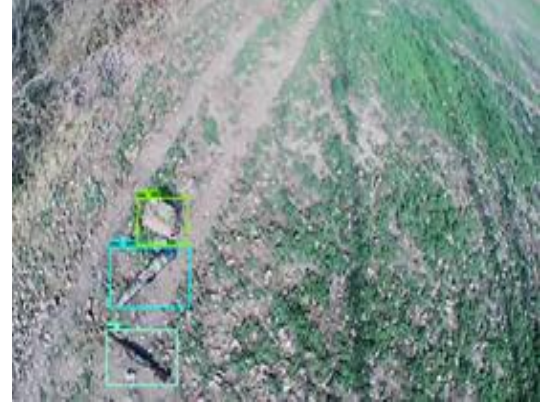
(a) (b)
Şekil 4. 47. Dokuzuncu test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a) (b)
Şekil 4. 48. Onuncu test resmi Faster_rcnn_inception_v2_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a) (b)
Şekil 4. 49. Birinci test resmi Rfcn_resnet101_coco_2018 modeli
(b) kesin referans değeri, (b) tahmini değeri



(a)

(b)

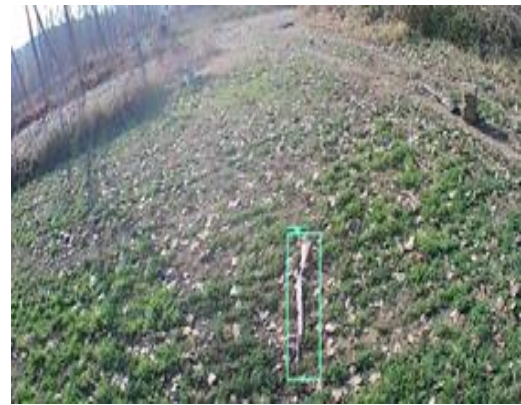
Şekil 4. 50. İkinci test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)

(b)

Şekil 4. 51. Üçüncü test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)

(b)

Şekil 4. 52. Dördüncü test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a) (b)
Şekil 4. 53. Beşinci test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a) (b)
Şekil 4. 54. Altıncı test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a) (b)
Şekil 4. 55. Yedinci test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)



(b)

Şekil 4. 56. Sekizinci test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)



(b)

Şekil 4.57. Dokuzuncu test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri



(a)



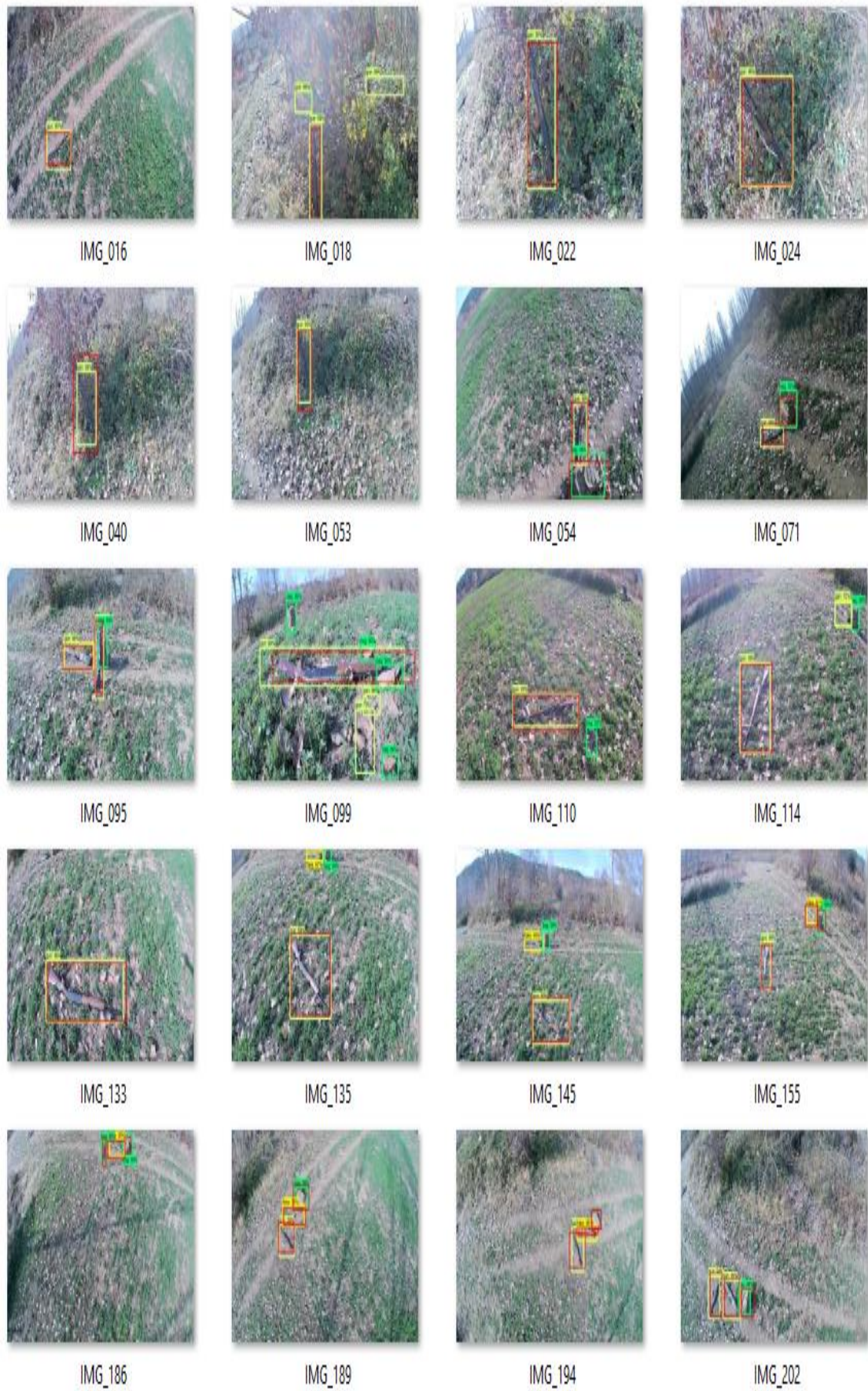
(b)

Şekil 4.58. Onuncu test resmi Rfcn_resnet101_coco_2018 modeli
(a) kesin referans değeri, (b) tahmini değeri

4.2.3. Kullanılan modellerin hassasiyet ve hatırlama eğrisi

Çalışmada kullanılan R-CNN mimarileri tespit alanında oldukça başarılı algoritmalar. Nesne tespitinde yapılan sınıflandırmada tespit edilecek nesnenin konum bilgileri oluşturulmaktadır. Kullanılan mimarilerden her birinin kendine özgü Loss (hata) grafikleri bulunmaktadır. Çalışmada kullanılan R-CNN mimarisinden Faster R-CNN ve ResNet mimarilerini Loss (Hata) grafikleri Faster R-CNN ve ResNet mimarisini Loss (Hata) grafiklerinin sıfıra (0) yaklaşmış olmaları iki mimarininde başarılı olduğunu göstermektedir. Faster R-CNN mimarisinin Loss değerinin (0.02) ResNet mimarisinin Loss değerinin de (0,05) olması iki mimarinin hata değerlerinin sıfıra yaklaşık olmaları nesne tespiti için dikkat edilecek bir husustur.

Nesne tespit için kullanılan Faster R-CNN ile ResNet mimarileri tespit için tahminlerde bulunarak veri setinde yer alan nesnelerin kesin referans değerlerine ground truth (kesin referans) denmektedir. Kullanılan Faster R-CNN ile ResNet mimarilerinin tahminlediği ve kesin referans değerlerin keştiği alanlar eğitim sonrası elde edilen görüntülerde Şekil 4.59 ve Şekil 4.60'da gösterilmektedir.



Şekil 4.59. Faster R-CNN kesin referans değerleri



IMG_001



IMG_005



IMG_007



IMG_011



IMG_025



IMG_031



IMG_034



IMG_037



IMG_073



IMG_076



IMG_085



IMG_092



IMG_117



IMG_122



IMG_126



IMG_130



IMG_161



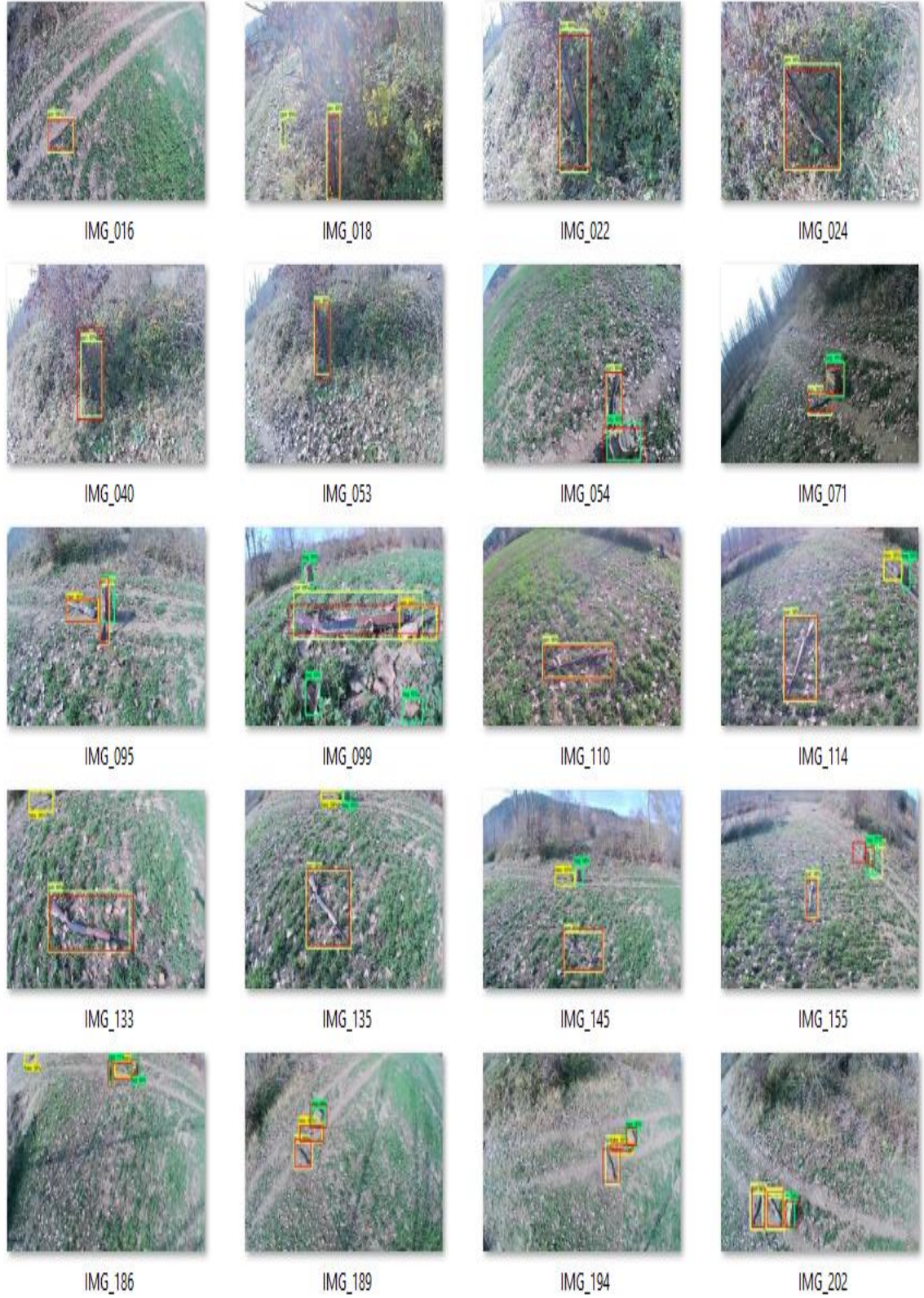
IMG_167



IMG_173



IMG_182



Şekil 4.60. ResNet kesin referans değerleri

Çalışmada kullanılan Faster R-CNN ve ResNet mimarisinin test verilerinde elde edilen TP, FP ve FN değerlerine bakıldığında doğru sınıflandırma değerlerinin yanlış

sınıflandırma değerlerindeki oldukça düşük olduğu görülmektedir. Çizelge 4.1 ve Çizelge 4.2’de TP, FP ve FN değerleri gösterilmektedir.

Tablo 4.1. Faster R-CNN mimarisinin TP, FP VE FN değerleri

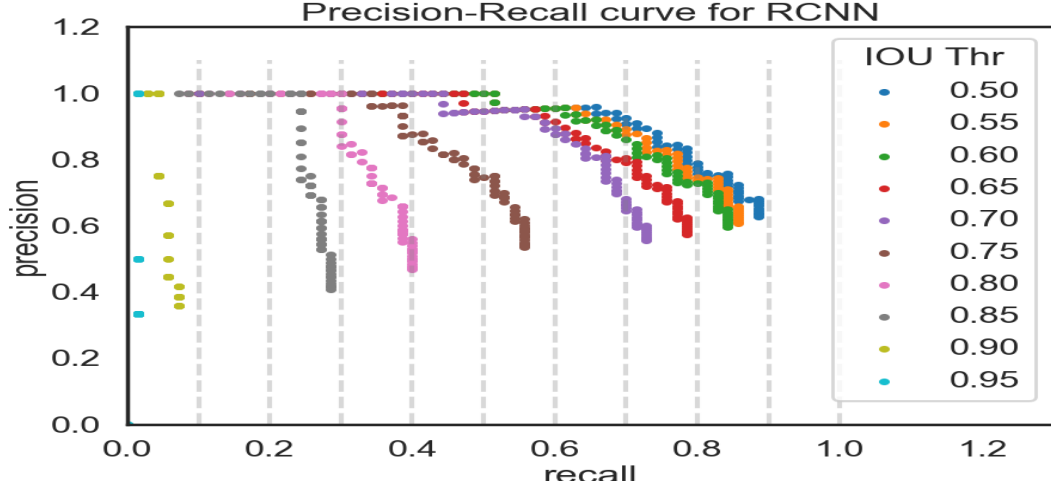
	TP	FP	FN	P1	P	R1	R	P.R	P+R	F1
IMG_001.jpg	3	0	0	3	1	3	1	1	2	1
IMG_007.jpg	1	2	0	3	0,33	1	1	0,33	1,33	0,49
IMG_016.jpg	1	0	0	1	1	1	1	1	2	1
IMG_022.jpg	1	0	0	1	1	1	1	1	2	1
IMG_025.jpg	1	2	0	3	0,33	1	1	0,33	1,33	0,49
.....										
.....										
.....										
IMG_135.jpg	1	2	0	3	0,33	1	1	0,33	1,33	0,49
IMG_155.jpg	3	0	0	3	1	3	1	1	2	1
IMG_167.jpg	3	0	0	3	1	3	1	1	2	1
IMG_182.jpg	3	0	0	3	1	3	1	1	2	1
IMG_189.jpg	3	0	0	3	1	3	1	1	2	1

P ve R eğrisinde hassasiyetin yüksek olması için FP değerinin düşük olmasıyla ve hatırlamanın da yüksek olması için FN değerinin düşük olması doğrudan eğriden anlaşılmaktadır.

Tablo 4.2. ResNet mimarisinin TP, FP ve FN değerleri

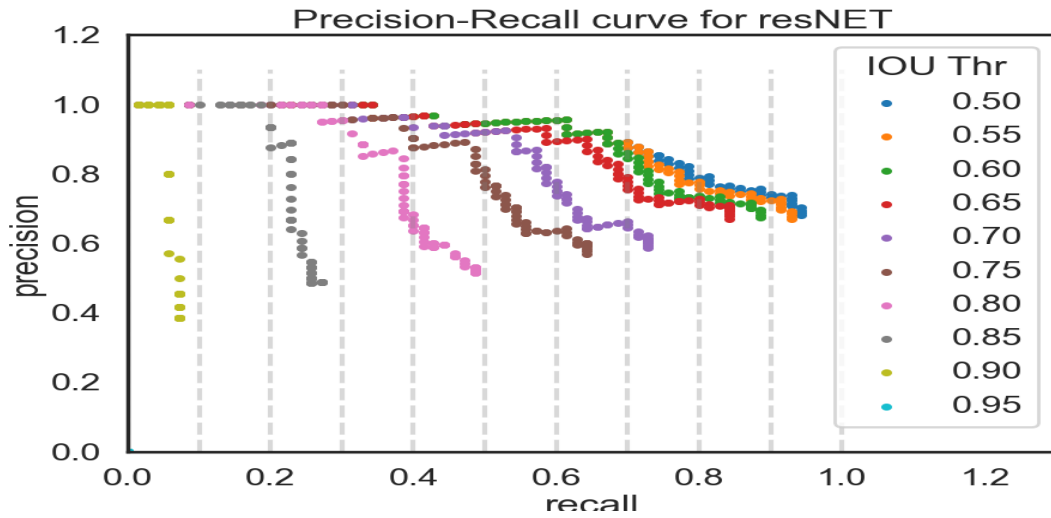
	TP	FP	FN	P1	P	R1	R	P.R	P+R	F1
IMG_001.jpg	3	0	0	3	1	3	1	1	2	1
IMG_007.jpg	1	2	0	3	0,33	1	1	0,33	1,33	0,49
IMG_016.jpg	1	0	0	1	1	1	1	1	2	1
IMG_022.jpg	1	0	0	1	1	1	1	1	2	1
IMG_025.jpg	1	2	0	3	0,33	1	1	0,33	1,33	0,49
.....										
.....										
.....										
IMG_155.jpg	2	2	1	4	0,5	3	0,66	0,33	1,16	0,56
IMG_167.jpg	3	0	0	3	1	3	1	1	2	1
IMG_182.jpg	3	0	0	3	1	3	1	1	2	1
IMG_189.jpg	3	0	0	3	1	3	1	1	2	1
IMG_202.jpg	3	2	0	5	0,6	3	1	0,6	1,6	0,75

Eğrideki P ve R değerlerinin yüksek olması nesne tespiti esnasında yapılan sınıflandırmada doğru sonuçlara ve sonuçların çoğunun doğru tespit ettiğini göstermektedir. Şekil 4.61 ve Şekil 4.62’de Faster R-CNN ile ResNet mimarisinin P ve R eğrisi gösterilmektedir.



Şekil 4.61. Faster R-CNN mimarisinin P ve R eğrisi

Faster R-CN mimarisinin eğitimdeki mAP değeri (50,84) ile ResNet mimarisinin mAP değerinin (52,17) olması iki mimarinin de P değerlerinin birbirine yakın olduğu görülmektedir. Böylece eşik değerlerinin farklı değerler olması P ve R eğrisine yansımaktadır.



Şekil 4.62. ResNet mimarisinin P ve R eğrisi

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Bu çalışma ile İHA'dan elde edilen görüntülerde bulunan nesneleri derin öğrenme algoritmaları ile tespit edilmiştir. Nesne tespitinde Python program dili kullanılmış ve Tensorflow kütüphanesinden faydalanılmıştır. Oluşturulan veri setinde bulunan nesneleri derin öğrenme algoritmaları tarafından yeterince öğrenebilmesi için nesnelerin İHA'dan farklı açılardan ve mesafelerden görüntüler alınmıştır. Nesneler içerisindeki silah (gun), silah kılıfı (fake) ve çantanın (bag) tespitini yapmak için yaklaşık 200 adet görüntü hazırlanmış ve bu görüntüleri test ve train klasörlerine ayrılmıştır. Nesne tespitinde doğruluk ve hız bakımından daha başarılı olan Faster R-CNN yöntemi tercih edilmiştir. Bununla birlikte görüntü sınıflandırma alanında oldukça etkili olduğu ispatlanan ResNet mimarisi de çalışmada kullanılmıştır. Nesne tespitinde kullanılan söz konusu derin öğrenme algoritmalarının eğitimleri Tensorboard üzerinde değerlendirilmiştir. Kullanılan bu algoritmalar, görüntü üzerindeki nesneleri tespit etmek için nesneleri bölgesel olarak özellik haritasını çıkartmayı sağlamaktadır.

Çalışmada nesne tespitinde kullanılan derin öğrenme algoritmaları sayesinde R-CNN ve ResNet mimarilerinin tahminleri ile gerçek değerler karşılaştırıldığında %99 doğruluk oranıyla başarılı bir nesne (silah) tespit uygulaması yapılmıştır. Özellikle insanoğlunun çıplak gözle bile göremediği nesneleri bu iki model oldukça başarılı bir şekilde tespit etmiştir. Eğitim aşamasında nesne (silah) tespit için elde edilen verilerin önceden istenilen nesnenin görüntü içerisinde belirlenmesi eğitimin başarılı olmasına katkı sağlamıştır.

Günümüzdeki olayların birçoğunda İHA'nın etkin bir şekilde kullanılması bu olaylarda karar vericilere yardım açısından ciddi bir yere sahiptir. Yapılan bu çalışmada derin öğrenme algoritmalarının gelecekte birçok alanda karar vericilere destek olacağı düşünülmektedir. Özellikle düşük maliyetli bir İHA ile nesne (silah) tespit uygulamalarının başarılı olacağı kesindir. Bu başarının arkasında günümüzde kullanım alanlarının yaygınlaştığı derin öğrenme mimarileri bulunmaktadır.

Her türlü göreve yönelik tasarlanabilen İHA'lar günümüzde olduğu kadar gelecekte de yerini koruyacaktır. Yapay zekâ sayesinde İHA'lar derin öğrenme algoritmalarıyla desteklenerek birçok alanda zaman, mekân ve insan kaynağından tasarruf sağlanacağı kesindir. Savunma, doğal afet ve deprem sağlık gibi daha birçok

alandaki kullanılacak olan yapay zekâ, derin öğrenme algoritmalarıyla İHA ile birlikte kullanılarak çoğu alanda öne çıkmasını sağlayacaktır. Derin öğrenme ile birlikte karmaşık görüntüler arasından istenilen nesnenin kısa sürede tespit imkânı gelecekte birçok alanda kullanılacaktır.

Çalışmada ulaşılan sonucun oldukça başarılı olması savunma ve güvenlik alanlarında yeryüzünün en değerli varlığı olan insanın her türlü olumsuz etkenlerden korunmasını sağlayacaktır. Silahlı ve Silahsız İHA'nın keşif, gözetleme ve tespit görevlerine destek sağlayacaktır. Özellikle Silahlı İHA'nın derin öğrenme algoritmaları ile nesne (silah) tespitlerini anlık olarak oluşabilecek tehditleri bertaraf etmek için iyi bir fırsat olacağını düşünülmektedir.

Sonuç olarak savunma sanayi başta olmak üzere birçok uygulama alanlarında derin öğrenme mimarilerinin kullanımı insanoğlunun hayatını kolaylaştıracaktır. Derin öğrenme sayesinde teknolojiye yön vermek isteyen ülkelere bir sene önceki teknolojinin bile bugün eski kaldığı günümüzde oldukça avantaj sağlaacağı düşünülmektedir.

5.2. Öneriler

İHA kullanım alanları gün geçtikçe yaygınlaşması göz önüne alındığında savunma sanayiinde kullanımının artacağı düşünüldüğünde derin öğrenme algoritmalarıyla birlikte nesne tespit ve nesne algılama görevlerinde kullanılmasında yararlı olacaktır.

Yapay zekâ alanının gelişmesi ile birlikte nesne tespitinde bulunan görüntülerin eş zamanlı olarak metin işleme uygulamalarıyla birlikte gelecek yıllarda kullanımı yaygınlaştırılmalıdır.

Kullanılan İHA'ların yüksek özelliklere sahip olması, farklı yükseklik ve açılardan görüntü elde edebilmesi sayesinde nesne tespitindeki başarısı gelecek yıllarda devam edeceği düşünülmektedir.

Derin öğrenme algoritmalarının karşılaşılan problemlere yönelik tasarlanabilir olmasından dolayı bu algoritmaların kullanım alanları yaygınlaştırılmalıdır.

KAYNAKLAR

- Aalami N., 2020, Derin öğrenme yöntemlerini kullanarak görüntülerin analizi, *ESTUDAM bilişim dergisi*, 1(1), 17-20.
- Acharya, T., Ray, A. K., 2005, Image processing: principles and applications, *WILEY*, USA, 1-4.
- Akademi 4.0 Ekibi, 2019, Makine öğrenimi ve derin öğrenmenin tarihi, <https://akademi40.org/makine-ogrenimi-tarihi>, [Ziyaret Tarihi: 10 Aralık 2019].
- Akgül M., Yurtseven, H., Demir, M., Akay, M. E., Gulci, S., Öztürk, T., 2016, İnsansız hava araçları ile yüksek hassasiyette sayısal yükseklik modeli üretimi ve ormancılıkta kullanım olanakları, *İstanbul Üniversitesi Orman Fakültesi Dergisi*, 66(1), 104-118.
- Akıncı İ.B., 2017, Gerçek zamanlı olarak meyveleri görüntü işleme ile sınıflandıran otomasyon sistemi, Yüksek Lisans Tezi, *Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı*, Karabük, 28-29.
- Akyürek S., Yılmaz M., Taşkiran M., 2012, İnsansız hava araçları muharebe alanında ve terörle mücadelede devrimsel dönüşüm, *Bilge Adamlar Stratejik Araştırma Merkezi*, Rapor No: 53, Ankara, 1-36.
- Albera S.A.A., 2016, Vehicle logo recognition using image processing methods, Yüksek Lisans Tezi, *Atılım Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 5-8.
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Essen, B. V., Awwal, A. S., Asari, V. K., 2018, The history began from AlexNet: a comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164. [Ziyaret Tarihi: 11 Şubat 2019].
- Altaş Z., 2017, Drone kullanılarak görüntü işleme tekniği ile şeker pancarı yaprak lekesi hastalığı (*cercospora beticola*) düzeyinin belirlenmesi, Yüksek Lisans Tezi, *Gaziosmanpaşa Üniversitesi Fen Bilimleri Enstitüsü*, Tokat, 16-18.
- Altınok T., 2010, Türkiye'nin İHA serüveni, *TÜBİTAK Bilim ve teknik dergisi*, 44(517), 39-41.
- Asht, S., Dass, R., 2012, Pattern recognition techniques: a review, *International journal of computer science and telecommunications*, 3(8), 25-29.
- Ayalew, A., Pooja, D., 2019, a review on object detection from unmanned aerial vehicle using CNN, *International journal of advance research, ideas and innovations in technology*, 5(4), 241-243.
- Balık H., Diri B., Şeker A., 2017, Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme, *Gazi mühendislik bilimleri dergisi*, 3(3), 47-64.

- Benenson, R, Omran, M., Hosang, J., Schiele, B., 2014, Ten years of pedestrian detection, what have we learned?, <https://arxiv.org/pdf/1411.4304.pdf>, [Ziyaret Tarihi: 11 Şubat 2019].
- Blom, J. D., 2006, Unmanned aerial systems: a historical perspective, *Combat studies institute press*, USA, Occasional Paper 37, 2.
- Boyacı, E., 2017, Sayısal video ve çizim verilerinde anlamsal kavram tanıma, Yüksek Lisans Tezi, *Başkent Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 17.
- Bradski G., Kaehler A., 2008, Learning open CV, *O'Reilly*, USA, 2-3.
- Brownlee J., 2016, Popular deep learning libraries, <http://machinelearningmastery.com/popular-deep-learning-libraries>, [Ziyaret Tarihi: 27 Temmuz 2019].
- Brownlee, J., 2017, Deep learning for natural language processing. *Machine learning mastery*, Vermont, Avustralya, 2,16.
- Chen, C. Y., Huang, J. J., 2019, . Double deep autoencoder for heterogeneous distributed clustering, *MDPI Informartion*, 10(144) 2-15, https://www.researchgate.net/publication/332488703_Double_Deep_Autoencoder_for_Heterogeneous_Distributed_Clustering/link/5cb7f40092851c8d22f2e9aa/download, [Ziyaret Tarihi: 27 Aralık 2019].
- Chowdhury, G., 2003, Natural language processing, *Annual review of information science and technology*, 37, 51-89.
- Costea A. D., Nedeveschi, S., 2014, Word channel based multiscale pedestrian detection without image resizing and using only one classifier, *IEEE conference on computer vision and pattern recognition*, 2393-2400.
- Çağlayan A., 2018, Derin öğrenme tekniklerini kullanarak RGB-D nesne tanıma, Doktora Tezi, *Hacettepe Üniversitesi Fen Bilimleri*, Ankara, 30-32.
- Çakar U.F., Tarcan A., 2008, Bilgisayarlı dil tanımlamada dilbilimsel yaklaşımlar ve bir yazılım denemesi, *Elektronik sosyal bilimler dergisi*, 7(26), 64-70.
- Çoban, S., Oktay, T., 2018, Unmanned aerial vehicles (UAVs) according to engine type, *Journal of aviation*, 2 (2), 177-184.
- Çömert R., Avdan U. ve Şenkal E., 2012, İnsansız hava araçlarının kullanım alanları ve gelecekteki beklentiler, *IV. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu*, Zonguldak, <http://dergipark.org.tr/en/download/article-file/70057>, [Ziyaret Tarihi: 3 Ekim 2019].
- Daş R., Polat B., Tuna G., 2019, Derin öğrenme ile resim ve videolarda nesnelerin tanınması ve takibi, *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 31(2), 571-581.

- David M.W., 2011, Evaluation: from precision, recall and F-measure to roci informedness, markednes correlation, *Journal of machine learning technologies*, 2(1), 37-63.
- Davis J. ve Goadrich M. 2006, The relationship between precision-recall and ROC curves, *University of Wisconsin-Madison*, 233-240.
- DeGarmo, M. T., 2004, Issues concerning integration of unmanned aerial vehicles in civil airspace, *MITRE*, 1-59.
- Demiroğ E, 2019, Bilgisayar görüşü ile yüz ve nesne tanıma, <https://www.udemy.com/course/bilgisayar-gorusu/learn/>, [Ziyaret Tarihi: 24 Kasım 2019].
- Demirtaş M.A., 2018, Derin öğrenme yöntemleri ile duygu analizi ve aktivasyon fonksiyonlarının karşılaştırılması, Yüksek Lisans Tezi, *Atatürk Üniversitesi Fen Bilimleri Enstitüsü*, Erzurum, 20-21.
- Deperlioğlu Ö., Köse U., 2018, Görüntü işleme ve evrimsel sinir ağları kullanarak diyabetik retinopati teşhisi, *TIPTEKNO18*, KKTC, 70-73.
- Doğan F.ve Türkoğlu İ., 2019, Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme, *DÜMF Mühendislik Dergisi*, 10(2), 409-445.
- Dronet, 2020, <https://www.drone.net.tr/drone-modelleri/kamerali-drone/aden-e58-fly-more-combo-drone.html>, [Ziyaret Tarihi: 12 Temmuz 2019].
- Eck D., Schmidhuber J., 2002, Learning the long-term structure of the blues, *Springer*, Heidelberg-Berlin, 284–289.
- Eisenbeiss, H., 2009, UAV photogrammetry, Doktora Tezi, *ETH ZURICH*, Almanya, 2-4.
- Elman, J. L., (1990), Finding structure in time, *Cognitive science*, 14(2), 179–211.
- Everaerts, J., 2008, The use of unmanned aerial vehicles (uavs) for remote sensing and mapping, *The international archives of the photogrammetry, remote sensing and spatial information*, 37, 1187-1192.
- Fernández S., Graves A., Schmidhuber J., 2007, An application of recurrent neural networks to discriminative keyword spotting, *International conference on artificial neural networks*, 220–229.
- Fogg, A., 2018, A history of deep learning, <https://www.import.io/post/history-of-deep-learning/>, [Ziyaret Tarihi: 11 Temmuz 2019].
- Fukushima K., 1980 neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological cybernetics*, (36), 193-202.

- Gandi R. ,2018 R-CNN, Fast R-CNN, Faster R-CNN, YOLO-Object detection algorithms,<https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>, [Ziyaret Tarihi:2 Mayıs 2020].
- Galvez, R. L., Bandala, A. A., Dadios, E. P., Vicerra, R. R. P., Maningo, J. M. Z., 2018, Object detection using convolutional neural networks. *TENCON 2018- 2018 IEEE region 10 conference*, Korea, 2012-2016.
- Ghoury S., 2019, Konvolüsyonel sinir ağlarını kullanarak üzüm ve üzüm yapraklarının hastalıklarının tespit edilmesi, Yüksek lisans tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya, 30,31*.
- Girshick R., Donahue J., Darrell, T., Malik, J., 2014, Region based convolutional networks for accurate object detection and segmentation, *IEEE*, <https://ieeexplore.ieee.org/document/7112511>, [Ziyaret Tarihi:5 Mayıs 2019].
- Girshick R. Donahue J., Darrell, T., Malik, J., 2014, Rich feature hierarchies for accurate object detection and semantic segmentation, Computer vision foundation, <https://arxiv.org/abs/1311.2524>, [Ziyaret Tarihi:12 Temmuz 2019].
- Girshick R. 2014, Fast R-CNN, Computer vision foundation, *IEEE*, <https://ieeexplore.ieee.org/document/7410526>, [Ziyaret Tarihi:25 Eylül 2019], 1440-1448
- Gözkaydırın S., 2019, Impletmenting cyber countermeasures against terrorist drone in military service, Yüksek Lisans Tezi, *The Republic Of Turkey Bahçeşehir University İstanbul*, 21-24.
- Greff K., Srivastava R., Koutník J., Steunebrink B. R., Schmidhuber J., 2017, LSTM: A search space odyssey, *IEEE transactions on neural networks and learning systems*, 28(10), 2222 – 2232.
- Gopal S. S., Dessai A., 2019, Automatic classification of cervical magnetic resonance images using ResNet-101, *International journal of scientific research and review*, 7(6), 366.
- Gupta, L., Jain, R., Vaszkun, G., 2015, Survey of important issues in UAV communication networks, *IEEE communications surveys and tutorials*, 99, 1-32.
- Gülcü, A., Kuş, Z., 2019, Konvolüsyonel sinir ağlarında hiper-parametre optimizasyonu yöntemlerinin incelenmesi, *Gazi Üniversitesi Fen Bilimleri Dergisi*, 7(2), 503-522.
- Gürel, C., Erden, A., 2012, Design of a face recognition system, *The 15th international conference on machine design and production*, Denizli, https://www.researchgate.net/publication/262875649_Design_of_a_Face_Recognition_System/link/00b7d5390cd5560a1a000000/download, [Ziyaret Tarihi: 1 Eylül 2019].
- Hinton, G. E., Osindero S., Teh, Y. W., 2006, A fast learning algorithm for deep belief nets, *Neural computation*, 18, 1527–1554.

- Hinton, G. E., Salakhutdinov, R. R., 2006, Reducing the dimensionality of data with neural networks, *Science*, 313, 504-507.
- Hinton, G. (2014). Boltzmann machines, 1,7.
<http://people.stat.sfu.ca/~dac5/BoltzmannMachines.pdf>, [Ziyaret Tarihi: 3 Nisan 2019].
- He K., Zhang X., Ren S., Sun, J., 2016, Deep residual learning for image recognition, Computer Vision Foundation, 770-778,
https://www.cvfoundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf, [Ziyaret Tarihi: 21 Haziran 2019].
- He K., Zhang X., Ren S., Sun J., 2014, Spatial pyramid pooling in deep convolutional networks for visual recognition, ECCV, <https://arxiv.org/abs/1406.4729>, [Ziyaret Tarihi: 10 Aralık 2018].
- Huang, J., Thomson, S. J., Hoffmann, W. C., Lan, Y., Fritz, B. K., 2013, Development and prospect of unmanned aerial vehicle technologies for agricultural production management, *Int J Agric & Biol Eng*, 6(3), 1-10.
- Hui J. 2018, mAP (mean Average Precision) for object detection,
https://medium.com/@jonathan_hui/map-mean-average, [Ziyaret Tarihi: 11 Mayıs 2020].
- Hung C, Sukkarieh S. ve Xu Z., 2014, Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a UAV, *Remote Sensing*, 6(12), 12037-12054.
- Ivakhnenko, A. G., Lapa, V. G., 1966, Cybernetic Predicting Devices.
<https://www.worldcat.org/title/cybernetic-predicting-devices/oclc/608778762>, [Ziyaret Tarihi: 7 Kasım 2018].
- Goodfellow I., Bengio, Y., Courville, A., 2016, Deep learning, MIT press, Cambridge-Birleşik Krallık,
<http://faculty.neu.edu.cn/yury/AAI/Textbook/DeepLearningBook.pdf>, [Ziyaret Tarihi: 7 Kasım 2018].
- İnik Ö. ve Ülker E., 2017, Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri, *Gaziosmanpaşa bilimsel araştırma dergisi (GBAD)*, 6(3), 85-104.
- Jain, A., Kulkarni, G., Shah, V., 2018, Natural language processing, *International journal of computer sciences and engineering*, 6(1), 161-167.
- Jay P. 2018, Understanding and Implementing Architectures of ResNet and ResNeXt for state-of-the-art Image Classification: From Microsoft to Facebook
<https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624>, [Ziyaret Tarihi: 16 Mayıs 2020].

- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014, Caffe: convolutional architecture for fast feature embedding, <https://arxiv.org/pdf/1408.5093.pdf>, [Ziyaret Tarihi: 2 Ağustos 2019].
- Kahvecioglu, S., Oktal, H., 2014, Turkish UAV capabilities as a new competitor in the marke, *International journal of intelligent unmanned systems*, 2(3), 183-191.
- Kang, K., Ouyang, W., Li, H., Wang, X., 2016, Object detection from video tubelets with convolutional neural networks, IEEE conference on computer vision and pattern recognition, <https://arxiv.org/pdf/1604.04053.pdf>, [Ziyaret Tarihi: 24 Eylül 2019].
- Karakoç M., 2011, Görüntü işleme teknikleri ve yapay zekâ yöntemlerde kullanarak görüntü içinde görüntü arama, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli, 34-35.
- Karakuş C., Katman, F., 2019, Male sınıfı insansız hava aracı (İHA) teknolojisi ve konvansiyonel (geleneksel) savaştaki yeri, *Akademik tarih ve düşünce dergisi*, 6(2), 882-897.
- Karathanasopoulos, A., 2017, Modelling and trading commodities with a new deep belief network, *Economics and business letter*, 6(2), 28-34.
- Karpathy, A., Fei-Fei, L., 2015, Deep visual-semantic alignments for generating image descriptions, *CVPR*, 3128–3137, <https://cs.stanford.edu/people/karpathy/cvpr2015.pdf>, [Ziyaret Tarihi: 25 Temmuz 2019].
- Kayaalp K., Süzen A.A., 2018, Derin öğrenme ve Türkiye’deki uygulamaları, *IKSAD yayınevi*, Ankara, 5-45.
- Khurana, K., Awasthi, R., 2013, Techniques for object recognition in images and multi-object detection, *International journal of advanced research in computer engineering & technology*, 2(4), 1383-1388.
- Kılınç H., 2018, Derin öğrenmeli konvolüsyonell sinir ağları (deep learning convolutional neural network) kullanarak fotoğraftan trizomi 21 (down sendromu) tespiti, *Çukurova Üniversitesi Fen Bilimleri Enstitüsü*, Adana, 39-43.
- Kın, B. Z., 2019, Türk işaret dili alfabesinin derin öğrenme yöntemi ile sınıflandırması, Yüksek Lisans Tezi, *Başkent Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 35.
- Koehrsen W. 2018, Beyond accuracy: precision and recall, <https://towardsdatascience.com/beyond-accuracy-precision-and-recall> [Ziyaret Tarihi: 11 Mayıs 2020].
- Krishna, R. (2017). Computer vision: foundations and applications, *Stanford University*, 17-25, http://vision.stanford.edu/teaching/cs131_fall1718/files/cs131-class-notes.pdf, [Ziyaret Tarihi: 12 Kasım 2019].
- Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012., ImageNet classification with deep convolutional neural networks, *Advances in neural information processing*

- systems,25(2), <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>, [Ziyaret Tarihi: 13 Haziran 2019].
- Kuş Z., 2019, Mikrokanonikal optimizasyon algoritması ile konvolüsyonel sınır ağlarında hiper parametrelerin optimize edilmesi, Yüksek Lisans Tezi, *Fatih Sultan Mehmet Vakıf Üniversitesi Lisansüstü Eğitim Enstitüsü*, İstanbul,7-12.
- Küçük D., Arıcı N., 2018, Doğal dil işlemede derin öğrenme uygulamaları üzerine bir literatür çalışması, *Uluslararası yönetim bilişim sistemleri ve bilgisayar bilimleri dergisi*, 2, 76-86.
- Lazebnik S. Schmid C. ve Ponce J. 2006, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, *CVPR*, 2169-2178, <https://ieeexplore.ieee.org/document/1641019>, [Ziyaret Tarihi: 11 Haziran 2019].
- LeCun, Y., Bengio, Y., Hinton, G., 2015, Deep learning, *Nature*, 521, 436-444.
- Lee, H., Grosse, R., Ranganath, R., Ng., A. Y., 2009, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, *ICML '09: Proceedings of the 26th annual international conference on machine learning*, 60-616. <https://ai.stanford.edu/~ang/papers/icml09ConvolutionalDeepBeliefNetworks.pdf>, [Ziyaret Tarihi: 6 Mayıs 2019].
- Leibe B., Seeman, E., Scenes, B., 2005, Pedestrian detection in crowded scenes, *IEEE computer society conference on*, <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/leibe-cvpr-05.pdf>, [Ziyaret Tarihi: 11 Kasım 2019].
- LISA lab ,2017, Theano documentation, 1, <http://deeplearning.net/software/theano/theano.pdf> [Ziyaret Tarihi: 15 Ocak 2020].
- Li, Y., Cha, S., 2019, Face recognition system, <https://arxiv.org/ftp/arxiv/papers/1901/1901.02452.pdf>, [Ziyaret Tarihi: 12 Nisan 2019].
- Mahmod A.A., Yılmaz H.M., 2018, Generating to three dimensional models from taken photos in vertical position with unmanned aerial vehicles: Aksaray University Campus Mosque, *Aksaray University journal of science and engineering*, 2(2), 144-160.
- Makineci, H.B., 2016, İnsansız hava araçları lidar etkileşimi, *Geomatik dergisi*, 1(1), 19-23.
- Malla C. U. 2019, How to Improve Object Detection Evaluation, <https://medium.com/moonvision/smart-object-detection-evaluation-with-confusion-matrices->, [Ziyaret Tarihi:11 Mayıs 2020].
- Marques, I., 2010, Face recognition algorithms, <http://www.ehu.eus/ccwintco/uploads/e/eb/PFC-IonMarques.pdf>, [Ziyaret Tarihi: 14 Nisan 2019].

- MathWorks, 2018, Convolutional neural network, <https://www.mathworks.com/solutions/deep-learning/convolutional-neuralnetwork.html>, [Ziyaret Tarihi: 15 Haziran 2019].
- MathWorks, 2020, Getting started with R-CNN, Fast R-CNN, and Faster R-CNN, <https://www.mathworks.com/help/vision/ug/getting-started>, [Ziyaret Tarihi: 3 Mayıs 2020]
- MathWorks, 2020, Resnet101, <https://www.resnet101mathworks.com/help/deeplearning>, [Ziyaret Tarihi: 16 Mayıs 2020].
- Mishra, M., 2019, Convolutional neural networks, explained, <https://www.datascience.com/blog/convolutional-neural-network>, [Ziyaret Tarihi: 14 Nisan 2020].
- Mutlu İpek B., 2018, EEG sinyallerinin epileptik rahatsızlıkların teşhisi için konvolüsyonel sinir ağları ve destek vektör makineleri ile tasnif edilmesi, Yüksek Lisans Tezi, *KTO Karatay Üniversitesi Fen Bilimleri Enstitüsü*, Konya, 20.
- Nandy, D., Padariya, R. J., 2016, An overview of pattern recognition, *Ternational journal for innovative research in science & technology*, 2(9), 132-135.
- Nayak, J., Routray, A. R., Adhikary, C., 2011, Classification of trees by pattern recognition, *International journal of computer applications*, 19(3), 10-14.
- NVIDIA DIGITS, <https://developer.nvidia.com/digits>.
- Nielsen M., 2015, Neural networks and deep learning, 169-171, <http://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf>, [Ziyaret Tarihi: 10 Nisan 2020].
- Nilson, N. J., 1998, Introduction to machine learning, *Stanford University*, USA, <https://ai.stanford.edu/~nilsson/MLBOOK.pdf>, [Ziyaret Tarihi: 10 Mart 2019].
- Özdağ, M. E., 2019, Derin öğrenme teknikleri kullanılarak anayol trafik analizi, Yüksek Lisans Tezi, *Karabük Üniversitesi Lisansüstü Eğitim Enstitüsü*, Karabük, 46-47.
- Özgören, M., 2006, Flow structure in the downstream of square and circular cylinders, *Flow measurement and instrumentation*, 17 (4), 225-235.
- Pakkan, B., Ermiş, M., 2010, İnsansız hava araçlarının genetik algoritma yöntemiyle çoklu hedeflere planlanması, *Havacılık ve uzay teknolojileri dergisi*, 4(3), 77-84.
- Parmar, D. N., Mehta, B. B., 2013, Face recognition methods & applications, *Int. J. computer technology & applications*, 4(1), 84-86.
- Pervan, N., 2019, Derin öğrenme yaklaşımları kullanarak Türkçe metinlerden anlamsal çıkarım yapma, Yüksek Lisans Tezi, *Ankara Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 28-29.

- Prisacariu, V., 2017, The history and the evolution of uavs from the beginning till the 70s. *Journal of defense resources management*, 8(1), 181-189.
- Radovic M, Adarkwa O ve Wang Q., 2017, Object recognition in aerial images using convolutional neural networks, *Journal of imaging*, 3(21), 1-9.
- Remondino, F., Barazzetti, L., Nex, F., Scaioni, M., Sarazzi, D., 2011, UAV photogrammetry for mapping and 3d modeling – current status and future perspectives, *International archives of the photogrammetry, remote sensing and spatial information sciences*. 38-1(22), 25-31.
- Ren., S., He, K., Girshick, R., Sun, J., 2016, Faster R-CNN: Towards real-time object detection with region proposal networks, <https://arxiv.org/pdf/1506.01497.pdf>, [Ziyaret Tarihi: 9 Eylül 2019].
- Rosebrock A., 2016, Intersection over Union (IoU) for object detection, <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>, [Ziyaret Tarihi:12 Mayıs 2020]
- Santhi, V., Acharjya, D. P., Ezhilarasan, M., 2016, Emerging technologies in intelligent applications for image and video processing, *Information science reference, an imprint of igt global*, USA, 402.
- Scikit Learn, 2019, Precison-recall, https://scikitlearn.org/stable/auto_examples/model_selection/plot_precision_recall.html, [Ziyaret Tarihi:11 Mayıs 2020].
- Serdoğan, U., 2020, Yüz parametreleri yardımı ile insanların etnik kökenlerine göre sınıflandırılması, Yüksek Lisans Tezi, *İstanbul T.C. Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü*, İstanbul, 18.
- Sert E., 2010, Görüntü işleme teknikleri ile şeftali ve elma sınıflandırma, Yüksek Lisans Tezi, *Trakya Üniversitesi Fen Bilimleri Enstitüsü*, Tekirdağ, 22-25.
- Sertkaya, M. E. (2018). Derin öğrenme tekniklerinin biyomedikal imgeler üzerine uygulamaları, Yüksek Lisans Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elâzığ, 17.
- Solem, J. E., 2012, Programming computer vision with python, *O' Reilly Media Inc.* http://programmingcomputervision.com/downloads/ProgrammingComputerVision_CCdraft.pdf, [Ziyaret Tarihi: 5 Kasım 2019].
- Stipaničev, D., 1994, Introduction to digital image processing and analysis. https://bib.irb.hr/datoteka/525713.Stipanicev_Introduction_to_Digital_Image_Processing_Analysis.pdf, [Ziyaret Tarihi: 13 Mayıs 2019].
- Süberk N.T., 2019, Deep learning techniques for bulding density estimation from remotely sensed imagery, Yüksek Lisans Tezi, *Işık Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 19-203.

- Szeliski, R., 2010, Computer vision: algorithms and applications, *Springer*, London-Birleşik Krallık, 3-5.
- Szegedy, C., Toshev A., Erhan, D., 2013, Deep neural networks for object detection, NIPS, <https://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf>, [Ziyaret Tarihi: 17 Kasım 2018].
- Şeker A., Yüksek A.G., 2017, Stacked autoencoder method for fabric defect detection, *Cumhuriyet Üniversitesi Fen Fakültesi Fen Bilimleri Dergisi*, 38(2), 342-354.
- Şeker, A., Yüksek, A. G., 2017, Stacked autoencoder method for fabric defect detection. *Cumhuriyet Üniversitesi Fen Fakültesi fen bilimleri dergisi*, 38(2), 342-354.
- Şener E., 2019, İnsansız hava araçları kullanılarak süleyman demirel üniversitesi yerleşkesinin yüksek çözünürlüklü ortofoto haritasının hazırlanması, *Mühendislik bilimleri ve tasarım dergisi*, 7(2), 393 – 402.
- Taşkıran A., 2019, Aktivasyon fonksiyonları https://www.akademianinbahcesi.com/aktivasyon_fonksiyonlari/, [Ziyaret Tarihi: 11 Nisan 2020].
- Taşova, O., 2011, Yapay sinir ağları ile yüz tanıma, Yüksek Lisans Tezi, *Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü*, İzmir, 24.
- Taylor G. W., Hinton G. E., Roweis S. T., 2006, Modeling human motion using binary latent variables, *Advances in neural information processing systems*, 19, 1345–1352.
- Tensorflow, 2020, https://www.tensorflow.org/tensorboard/get_started [Ziyaret Tarihi:11 Mayıs 2020].
- Teodoridis, S., Koutroumbas, K., 2003, Pattern recognition, *Elsiver Academic Press*, İkinci Baskı, Amerika, 1.
- Theis F. J., Meyer-Base, A., 2010, Biomedical signal analysis- contemporary methods and applications, *The MIT Press*, London, 3.
- Tobias, L., Ducournau, A., Rousseau, F., Mercier, G., Fable, R., 2016, Convolutional neural networks for object recognition on mobile devices: a case study, *23rd international conference on pattern recognition*, México, 3519-3524.
- Tobias, L., Ducournau, A., Rousseau, F., Mercier, G., Fablet, R., 2016, Convolutional neural networks for object recognition on mobile devices: a case study. *23rd international conference on pattern recognition 2016*, 3519-3524.
- Tomas M, Martin K, Lukas B, Jan H. C, Sanjeev K.2010, Recurrent neural network based language model, Department of Electrical and Computer Engineering, Johns Hopkins University, USA, 26-30

- Toprak A.S., 2014, Fotogrametrik tekniklerin insansız hava araçları ile mühendislik projelerinde kullanılabilirliğinin araştırılması, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya, 20-24.
- Torun A., 2017, İnsansız hava aracı (İHA) sektörü ve İHA fotogrametrisinin ölçme bağlamında konumlandırılması, 16. *Türkiye Harita Bilimsel ve Teknik Kurultayı*, https://www.hkmo.org.tr/resimler/ekler/ee7bb21c93fa52e_ek.pdf, [Ziyaret Tarihi: 11 Mart 2019].
- Türkoğlu İ., 1996, Yapay sinir ağları ile nesne tanıma, Yüksek Lisans Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elâzığ, 26-36.
- Wanda, P., Jin Jie, H., 2020, RunPool: a dynamic pooling layer for convolution neural network, *International journal of computational intelligence systems*, 13(1), 66-76.
- Wikipedia, 2016, Unmanned aerial vehicle, https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle, [Ziyaret Tarihi:24 Nisan 2020].
- Wang, X., Cheng, P., Liu, X., Uzochukwu, B., 2018, Fast and accurate, convolutional neural network based approach for object detection from UAV, *44th annual conference of the ieee industrial electronics society*, <https://arxiv.org/ftp/arxiv/papers/1808/1808.05756.pdf>, [Ziyaret Tarihi: 5 Mart 2019].
- Wikipedia Guide, Introduction to Machine Learning, <http://www.datascienceassn.org/sites/default/files/Introduction%20to%20Machine%20Learning.pdf>, [Ziyaret Tarihi: 3 Mart 2019].
- Wikipedia, 2020,Residual Neural Network, https://en.wikipedia.org/wiki/Residual_neural_network [Ziyaret Tarihi:16 Mayıs 2020].
- Wu, J. (2017). Introduction to convolutional neural networks, 11, 22, https://pdfs.semanticscholar.org/450c/a19932fcef1ca6d0442cbf52fec38fb9d1e5.pdf?_ga=2.221445279.607784203.1587377991-1416605480.1558866067, [Ziyaret Tarihi: 19 Kasım 2019].
- Vapnik VN., 1999, An Overview of Statistical Learning Theory, *IEEE, Transactions on Neural Networks*, 988-999.
- Yaman, K., Sarucan, A., Atak, M., Aktürk, N., 2001, Dinamik çizelgeleme için görüntü işleme ve arama modelleri yardımıyla veri hazırlama, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 16(1), 19-40.
- Yetkin, M., 2014, Tanker şamandıra bağlama sistemlerinin yapay sinir ağları tekniğiyle optimizasyonu, Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 10.

- Young, T., Poria, S., Hazarika, D., Cambria, E., 2018, Recent trends in deep learning based natural language processing, *IEEE computational intelligence magazine*, 13(3), 55-75.
- Yu J., Jiang Y., Wang Z., Cao Z., Huang T., 2016, UnitBox: an advanced object detection network, *University of Illinois at Urbana–Champaign*, 516-520,
- Yue Y., Finley T., Radlinski F., Joachims T., 2007, A support vector method for optimizing average precision, *SIGIR 2007 Proceedings*, 271-278, https://www.cs.cornell.edu/people/tj/publications/yue_etal_07a.pdf, [Ziyaret Tarihi: 15 Haziran 2020].
- Zaccone, G. (2016). Getting started with TensorFlow. *Kindle edition*, <http://read.pudn.com/downloads779/ebook/3085493/Getting%20Started%20with%20TensorFlow.pdf>, [Ziyaret Tarihi: 20 Nisan 2020].
- Zhou. C., Paffenrot, R. C., 2017, Anomaly detection with robust deep autoencoders, *KDD '17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 665-674. <https://dl.acm.org/doi/10.1145/3097983.3098052>, [Ziyaret Tarihi: 10 Temmuz 2019].
- Zhao X. Hu C. 2018, Research on pedestrian detection technology based on msr and Faster R-CNN, *Journal of computer and communications*, 6, 54-63

ÖZGEÇMİŞ**KİŞİSEL BİLGİLER**

Adı Soyadı : Mustafa BURGAZ
Uyruğu : T.C.
Doğum Yeri ve Tarihi: Eskişehir-16.04.1986
e-mail : atamizindeyiz3113@gmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Işıklar Askeri Lisesi	2005
Üniversite	: Kara Harp Okulu	2009
Yüksek Lisans:	Batman Üniversitesi	Devam

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2009-Devam	Türk Silahlı Kuvvetleri	Subay

YABANCI DİLLER

İngilizce, Farsça