

T.C.  
BEYKENT ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI  
ENDÜSTRİ MÜHENDİSLİĞİ BİLİM DALI

**İNSANSIZ HAVA ARAÇLARI VE UYDU  
GÖRÜNTÜLERİNDEN ELDE EDİLEN VERİ SETİ İLE  
HAVAALANLARININ TESPİTİNİN YAPILMASINDA  
SSD VE FASTER R-CNN ALGORİTMALARININ  
KARŞILAŞTIRILMASI**

Yüksek Lisans Tezi

Tezi Hazırlayan:

**Muhammed Taha ZEREN**

İstanbul, 2020

T.C.  
BEYKENT ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI  
ENDÜSTRİ MÜHENDİSLİĞİ BİLİM DALI

**İNSANSIZ HAVA ARAÇLARI VE UYDU  
GÖRÜNTÜLERİNDEN ELDE EDİLEN VERİ SETİ İLE  
HAVAALANLARININ TESPİTİNİN YAPILMASINDA  
SSD VE FASTER R-CNN ALGORİTMALARININ  
KARŞILAŞTIRILMASI**

Yüksek Lisans Tezi

Tezi Hazırlayan:

**Muhammed Taha ZEREN**

Öğrenci No:

17080920016

Danışman:

Dr. Öğr. Üyesi Sabahattin Kerem AYTULUN

İstanbul, 2020

## YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “İnsansız Hava Araçları ve Uydu Görüntülerinden Elde Edilen Veri Seti İle Havaalanlarının Tespitinin Yapılmasında SSD ve Faster R-CNN Algoritmalarının Karşılaştırılması ” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmanın içinde kullandıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumda doğrularım. (16.01.2020)

**Muhammed Taha ZEREN**



T.C.  
BEYKENT ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ  
TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI

03.10.2020

Enstitümüz *Endüstri Mühendisliği* Anabilim Dalı *Endüstri Mühendisliği* Programı yüksek lisans öğrencilerinden 17080920016 numaralı *Muhammed Taha ZEREN*'in "*Beykent Üniversitesi Lisansüstü Eğitim – Öğretim Yönetmeliği*"nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği "*İnsansız Hava Araçları ve Uydu Görüntülerinden Elde Edilen Veri Seti İle Havaalanlarının Tespitinin Yapılmasında SSD ve Faster R-CNN Algoritmalarının Karşılaştırılması*" konulu tezini, Yönetim Kurulumuzun 21/01/2020 tarih ve 2020/03 sayılı toplantısında seçilen ve Taksim Yerleşkesinde toplanan biz jüri üyeleri huzurunda, Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 29. maddesinin 3. fıkrası gereğince (5.7) dakika süre ile aday tarafından sayunulmuş ve sonuçta adayın tezi hakkında *oyçokluğu/oybirliği* ile *Kabul/Red veya Düzeltme* kararı verilmiştir.

İşbu tutanak, 4 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü'ne sunulmak üzere tarafımızdan düzenlenmiştir.



**DANIŞMAN**  
Dr. Öğr. Üyesi Sabahattin Kerem AYTULUN  
(Beykent Üniversitesi)



**ÜYE**  
Dr. Öğr. Üyesi Ediz ŞAYKOL  
(Beykent Üniversitesi)



**ÜYE**  
Dr. Öğr. Üyesi Ömer ÇETİN  
(Milli Savunma Üniversitesi)

Adı ve Soyadı : Muhammed Taha ZEREN  
Danışmanı : Dr. Öğr. Üyesi Sabahattin Kerem AYTULUN  
Türü ve Tarihi : Yüksek Lisans, 2020  
Alanı : Endüstri Mühendisliği  
Anahtar Kelimeler : Makine Öğrenmesi, Derin Öğrenme, Bilgisayarlı Görü,  
Yapay Sinir Ağları, SSD - Single Shot Multibox Detector,  
Faster-R-CNN, İnsansız Hava Araçları (IHA), Uydu Sistemleri

## ÖZ

### İNSANSIZ HAVA ARAÇLARI VE UYDU GÖRÜNTÜLERİNDEN ELDE EDİLEN VERİ SETİ İLE HAVAALANLARININ TESPİTİNİN YAPILMASINDA SSD VE FASTER R-CNN ALGORİTMALARININ KARŞILAŞTIRILMASI

Günümüzde görüntü işleme çalışmalarında birçok farklı sektörde, özellikle sağlık, üretim ve askeri alanlarda, doğrudan insan yaşantısında çeşitli amaçlarla kullanılmaya başlanmıştır. Derin öğrenme algoritmalarının gelişmesi ve bilgisayarlı görüde kullanılmaya başlanması özellikle askeri alandaki kritik hedef, önemli konum ve stratejik bölge tespiti gibi çalışmalara hız kazandırmıştır. Bu çalışmada Airport olarak adlandırılan havaalanlarının, uçak iniş pistleri üzerinden tespiti gerçekleştirilmiştir. Hem orta ve yüksek irtifalı insanız hava araçlarından hem de uydu görüntüleri kullanılarak eğitim test ve değerlendirme veri setleri oluşturulmuştur. Tespit yapılması sürecinde SSD-Single Shot Multibox algoritması ve Faster R-CNN algoritması yeniden eğitilerek kullanılmıştır. Her iki algoritmanın sonuçları doğruluk oranı, duyarlılık, özgüllük, yanlış pozitif oranı, yanlış negatif oranı, doğru tahmin oranı, F puanı, hata oranı, sonuç ve eğitim süresi gibi değerlendirme kriterleri kapsamında değerlendirilmiştir. Değerlendirme veri seti üzerinde; SSD mimarisi ile %76,61 doğruluk oranıyla, Faster R-CNN mimarisinde ise %99,52 doğruluk oranı ile görüntü tespit sonucu elde edilmiştir. Söz konusu çalışma ile iki mimariden hangisinin insansız hava araçları ve uydu görüntülerinde kritik bölge tespitinde ne derece başarılı olduğu ortaya çıkarılmıştır.

Name and Surname : Muhammed Taha ZEREN  
Thesis Advisor : Asst. Prof. Sabahattin Kerem AYTULUN  
Thesis Type and Date : Master, 2020  
Study Field : Industrial Engineering  
Key Words : Machine Learning, Deep Learning, Computer Visioning,  
Artificial Neural Networks, SSD - Single Shot Multibox  
Detector, Faster-R-CNN, Unmanned Aerial Vehicles (UAV),  
Satellite Systems

## **ABSTRACT**

### **COMPARISON OF SSD AND FASTER R-CNN ALGORITHMS TO DETECT THE AIRPORTS WITH DATA SET WHICH OBTAINED FROM UNMANNED AERIAL VEHICLES AND SATELLITE IMAGES**

Today, image processing has been used in many different sectors, especially in health, production and military fields, for various purposes directly in human life. The development of deep learning algorithms and starting to use of computerized vision has accelerated the studies such as critical target, important location and strategic region determination especially in the military field. In this study, the airport has been determined on the landing runways. Training test and evaluation data sets were created by using both medium and high-altitude human aircraft and satellite images. SSD-Single Shot Multibox algorithm and Faster R-CNN algorithm were used by re-training during the determination process. The results of both algorithms were evaluated within the extend of evaluation criteria such as accuracy, sensitivity, specificity, false positive rate, false negative rate, positive pred value, F score, error rate, result and training time. The image detection accuracy with SSD algorithm was 76,61%, with Faster R-CNN algorithm the image detection accuracy was 99.52% according to valuation dataset. With this study, which of the two architectures has been revealed to be successful in determining critical areas in unmanned aerial vehicles and satellite images.

# İÇİNDEKİLER

Sayfa No.

<b>ÖZ</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>TABLolar LİSTESİ</b> .....	<b>vi</b>
<b>ŞEKİLLER LİSTESİ</b> .....	<b>vii</b>
<b>KISALTMALAR</b> .....	<b>x</b>
<b>GİRİŞ</b> .....	<b>1</b>

## BİRİNCİ BÖLÜM

### HAVACILIK, İNSANSIZ HAVA ARAÇLARI VE UYDU SİSTEMLERİ

1.1 Havacılık Tarihi ve Gelişimi .....	3
1.2 İnsansız Hava Araçları Gelişimi ve Özellikleri .....	4
1.3 Uydu Sistemleri Gelişimi ve Özellikleri .....	6

## İKİNCİ BÖLÜM

### DERİN ÖĞRENME VE BİLGİSAYARLI GÖRÜ

2.1 Yapay Zeka.....	9
2.2 Makine Öğrenmesi .....	10
2.3 Derin Öğrenme ve Bilgisayarlı Görüyü Anlama.....	11
2.4 Yapay Sinir Ağları.....	13
2.5 Aktivasyon Fonksiyonları .....	16
2.5.1 Sigmoid Fonksiyonu.....	17
2.5.2 Hiperbolik Tanjant Fonksiyonu (Tanh).....	19
2.5.3 ReLU (Rectified Linear Unit) Fonksiyonu.....	20
2.6 Konvolüsyonel (Evrişimli) Sinir Ağları (CNN).....	21
2.6.1 Giriş Katmanı (Input Layer).....	22
2.6.2 Konvolüsyon katmanı (Convulation Layer).....	22
2.6.3 Havuzlama Katmanı (Pooling Layer).....	26
2.6.4 Bırakma Katmanı (Dropout).....	27

2.6.5 Tam Bağlı Katman (Fully Connected Layer).....	28
2.6.6 Sınıflandırma Katmanı .....	28
2.7 Yapay Sinir Ağı Çeşitleri .....	28
2.7.1 R-CNN (Region Based Convolutional Neural Networks) .....	29
2.7.2 Fast R-CNN .....	30
2.7.3 Faster R-CNN .....	31
2.7.4 Single Shot Multibox Detector (SSD).....	33
2.8 Yapay Sinir Ağı Modelleri .....	34
2.8.1 LeNet-5.....	35
2.8.2 AlexNet.....	35
2.8.3 VGG .....	36
2.8.4 ResNet .....	36
2.8.5 GoogLeNet .....	37
2.8.6 MobileNet.....	37
2.8.7 Inception .....	38
2.9 Görüntü Sınıflandırma Eğitimi ve Kullanılan Veri Setleri.....	39
2.9.1 ImageNet Veri Seti .....	39
2.9.2 MS COCO Veri Seti .....	40
2.9.3 MNIST Veri Seti .....	41
2.9.4 CIFAR-10 / CIFAR-100.....	42
2.10 Sıfırdan Öğrenim ve Aktarımlı Öğrenme.....	43
2.10.1 Sıfırdan Eğitim (Training From Scratch) .....	43
2.10.2 Aktarımlı Öğrenme (Transfer Learning).....	44

## ÜÇÜNCÜ BÖLÜM

### UYGULAMA ÇALIŞMASI

3.1 Çalışmada Kullanılan Programlama Dili ve Kütüphaneler .....	45
3.2 Eğitim, Test, Değerlendirme (Validation) Veri Setlerinin Oluşturulması .....	46
3.3 Sonuç ve Performans Değerlendirme Kriterleri .....	50
3.4 SSD- Single Shot Multibox Algoritması İle Gerçekleştirilen Eğitim ve Değerlendirme Sonuçları.....	53
3.5 Faster R-CNN Algoritması İle Gerçekleştirilen Eğitim ve Değerlendirme Sonuçları.....	59



3.6 SSD ve Faster R-CNN İle Gerçekleştirilen Uygulamanın Sonuçlarının Kıyaslanması .....	66
<b>SONUÇLAR VE ÖNERİLER.....</b>	<b>72</b>
<b>KAYNAKÇA.....</b>	<b>73</b>
<b>ÖZGEÇMİŞ .....</b>	<b>78</b>



## TABLULAR LİSTESİ

	<b>Sayfa No.</b>
<b>Tablo 1.</b> İHA Sınıflandırması (Karaağaç 2012).....	6
<b>Tablo 2.</b> Çekirdek Matris (Kernel) Örnekleri ve Kullanım Alanları (Wikipedia2) .....	24
<b>Tablo 3.</b> Karmaşıklık Matrisi .....	50
<b>Tablo 4.</b> Derlenmiş Değerlendirme Kriterleri Tablosu .....	52
<b>Tablo 5.</b> Değerlendirme Kriterlerine Göre SSD Algoritmasının Sonuçları .....	55
<b>Tablo 6.</b> Değerlendirme Kriterlerine Göre Faster R-CNN Algoritmasının Sonuçları .....	61
<b>Tablo 7.</b> Değerlendirme Kriterlerine Göre SSD ve Faster R-CNN algoritmalarının Sonuçlarının Kıyaslanması .....	66
<b>Tablo 8.</b> Faster R-CNN ve SSD Algoritmalarının Tespit Sürelerinin Kıyaslanması.....	67

## ŞEKİLLER LİSTESİ

### Sayfa No.

<b>Şekil 1.</b> “Wright Flyer” İlk Sürdürülebilir, Kumanda Edilebilir ve Güçlendirilmiş (Motorlu) Uçak (Wikipedia1).....	3
<b>Şekil 2.</b> İHA Görüntüleme ve SAR Sistemi ile Görüntü Alma .....	5
<b>Şekil 3.</b> Uzaydan Çekilen İlk Uydu Görüntüsü (NASA, 1974).....	7
<b>Şekil 4.</b> Göktürk Uydu İlk Görüntüsü .....	8
<b>Şekil 5.</b> Yapay Zeka, Makine Öğrenimi ve Derin Öğrenmenin Tarihsel Değişimi.....	10
<b>Şekil 6.</b> Yeni Programlama Paradigması (Chollet, 2017).....	11
<b>Şekil 7.</b> Derin Öğrenme ve Makine Öğrenme Yaklaşımları Kıyaslaması (Karunakaran, 2018’ dan esinlenmiştir) .....	12
<b>Şekil 8.</b> Biyolojik Nöron (Zhang, Lipton, Lee ve Smola 2018) .....	13
<b>Şekil 9.</b> Perceptron (Algılayıcı) (Shanmugamani, 2018).....	14
<b>Şekil 10.</b> Tek Katmanlı Sinir Ağı Modeli (Nielsen, 2015’den esinlenmiştir.) .....	15
<b>Şekil 11.</b> Çok Katmanlı Sinir Ağı Modeli (Nielsen, 2015’den esinlenmiştir.).....	16
<b>Şekil 12.</b> Sigmoid Fonksiyonu Grafik Gösterimi .....	18
<b>Şekil 13.</b> Hiperbolik Tanjant Fonksiyonu Grafik Gösterimi .....	19
<b>Şekil 14.</b> ReLU (Rectified Linear Unit) Fonksiyonu Grafik Gösterimi .....	20
<b>Şekil 15.</b> Konvolüsyonel Sinir Ağlarını Oluşturan Katmanlar (Mathworks, 2019) ....	22
<b>Şekil 16.</b> Keskinleştirme Çekirdek Matrisi ile Kayma Miktarı 1 ve Kayma Miktarı 2 için Konvolüsyon İşleminin Uygulaması (Machinelearninguru.com, 2019) .....	25
<b>Şekil 17.</b> Maksimum Havuzlama (Firelord Phonix, 2018).....	26
<b>Şekil 18.</b> Ortalama Havuzlama (Firelord Phonix, 2018’den esinlenmiştir).....	27
<b>Şekil 19.</b> Standart Sinir Ağı ve Dropout (Bırakma Sonrası) Sinir Ağı (Budhiraja, 2016).....	27
<b>Şekil 20.</b> Düzleştirme işlemi (Flattening) .....	28
<b>Şekil 21</b> R-CNN Mimarisi .....	29
<b>Şekil 22.</b> R-CNN Çalışma Mantığı (Girshick, Donahue, Darrell ve Malik, 2014).....	30
<b>Şekil 23.</b> Fast R-CNN Mimarisi (Girshick, 2015) .....	31
<b>Şekil 24.</b> Faster R-CNN Mimarisi .....	32

<b>Şekil 25.</b> SSD (Single Shot Multibox Detector) Mimarisi )Liu, Anguelov, Erhan, Szegedy, Reed, Fu ve Berg 2016) .....	33
<b>Şekil 26.</b> Nesnelerin çerçeveleri için pozisyon (loc) ve doğruluk (conf) değerleri (Liu ve ark., 2016) .....	34
<b>Şekil 27.</b> LeNet Mimarisi (Lecun, Bottou, Bengio ve Haffner,1998) .....	35
<b>Şekil 28.</b> AlexNet Mimarisi (Krizhevsky, Sutskever ve Hinton, 2012) .....	35
<b>Şekil 29.</b> VGG Mimarisi (Heuritech,2016).....	36
<b>Şekil 30.</b> Klasik Model ile Res-Net Modeli Karşılaştırması (He, Zhang, Ren ve Sun, 2015).....	37
<b>Şekil 31.</b> MobileNet v1 ve MobileNet v2 Genel Mimarileri .....	38
<b>Şekil 32.</b> ImageNet Veri Setinden Örnekler (Massouh, Babiloni, Tommasi, Young, Hawes ve Caputo, 2017).....	40
<b>Şekil 33.</b> MS COCO Veri Setinden Örnek Görüntüler (Lin, Maire, Belongie, Bourdev, Girshick, Hays, Perona,Ramanan, Zitnick ve Dollar, 2014 ) .....	41
<b>Şekil 34.</b> MNIST Veri Setinden Örnekler.....	42
<b>Şekil 35.</b> CIFAR-10 Veri Setinden Örnekler (ImageNet, 2016) .....	43
<b>Şekil 36.</b> Eğitim Setinde Yer Alan Havaalanı İçeren Görüntü Örnekleri.....	48
<b>Şekil 37.</b> Test Setinde Yer Alan Havaalanı İçeren Görüntü Örnekleri.....	48
<b>Şekil 38.</b> Değerlendirme Setinde Yer Alan Havaalanı İçeren Görüntü Örnekleri.....	49
<b>Şekil 39.</b> Değerlendirme Setinde Yer Alan Havaalanı İçermeyen Görüntü Örnekleri .....	49
<b>Şekil 40.</b> ROC Eğrisi (Lane ve Gantley, 2017) .....	52
<b>Şekil 41.</b> SSD Algoritması Eğitimi Total Loss (Toplam Kayıp) Grafiği .....	54
<b>Şekil 42.</b> SDD Algoritmasının Değerlendirme Sonuçlarının ROC Eğrisinde gösterimi .....	56
<b>Şekil 43.</b> SDD Algoritması Görüntü Analiz Sonucu-1 .....	57
<b>Şekil 44.</b> SDD Algoritması Görüntü Analiz Sonucu-2.....	57
<b>Şekil 45.</b> SDD Algoritması Görüntü Analiz Sonucu-3 .....	58
<b>Şekil 46.</b> SDD Algoritması Görüntü Analiz Sonucu-4.....	58
<b>Şekil 47.</b> SDD Algoritması Havaalanı İçermeyen Görüntü Analiz Sonucu .....	59
<b>Şekil 48.</b> Faster R-CNN Algoritması Eğitimi Total Loss (Toplam Kayıp) Grafiği.....	60
<b>Şekil 49.</b> Faster R-CNN Algoritmasının Değerlendirme Sonuçlarının ROC Eğrisinde gösterimi.....	62

<b>Şekil 50.</b> Faster R-CNN Algoritması Görüntü Analiz Sonucu-1 .....	63
<b>Şekil 51.</b> Faster R-CNN Algoritması Görüntü Analiz Sonucu-2.....	63
<b>Şekil 52.</b> Faster R-CNN Algoritması Görüntü Analiz Sonucu-3.....	64
<b>Şekil 53.</b> Faster R-CNN Algoritması Görüntü Analiz Sonucu-4.....	64
<b>Şekil 54.</b> Faster R-CNN Algoritması Görüntü Analiz Sonucu-5.....	65
<b>Şekil 55.</b> Faster R-CNN Algoritması Havaalanı İçermeyen Görüntü Analiz Sonucu .	65
<b>Şekil 56.</b> SSD Algoritması Anderson-Darling Testi Sonuçları .....	68
<b>Şekil 57.</b> Faster R-CNN Algoritması Anderson-Darling Testi Sonuçları.....	68
<b>Şekil 58.</b> SSD Algoritması ve Faster R-CNN Algoritması İki Örneklem t Testi Sonuçları .....	69
<b>Şekil 59.</b> SSD ve Faster R-CNN Algoritmalarının Değerlendirme Sonuçlarının ROC Eğrisinde Kıyaslamaları .....	70
<b>Şekil 60.</b> SSD ve Faster R-CNN Algoritmaları Loss Grafikleri Kıyaslaması .....	71

## KISALTMALAR

<b>ANN</b>	: Artificial Neural Network
<b>ACC</b>	: Accuracy – Doğruluk Oranı
<b>CNN</b>	: Convolution Neural Network
<b>ERR</b>	: Error Rate – Hata Oranı
<b>FPR</b>	: False Positive Rate – Yanlış Pozitif Oranı
<b>FNR</b>	: False Negative Rate – Yanlış Negatif Oranı
<b>IHA</b>	: İnsansız Hava Aracı
<b>MNIST</b>	: Modified National Institute of Standards and Technology
<b>MS COCO</b>	: Microsoft Common Objects in Context
<b>NASA</b>	: National Aeronautics and Space Administration
<b>PPV</b>	: Positive Pred Value – Presicion – Doğru Tahmin Oranı
<b>R-CNN</b>	: Regional Convolution Neural Network
<b>ReLU</b>	: Rectified Linear Unit
<b>RPN</b>	: Region Proposal Network
<b>ROC</b>	: Receiver Operating Characteristics
<b>SAR</b>	: Sentetik Açıklıklı Radar
<b>SSD</b>	: Single Shot Multibox Detector
<b>SVM</b>	: Support Vektor Mission
<b>TPR</b>	: True Positive Rate – Doğru Pozitif Oranı
<b>TNR</b>	: True Negative Rate – Doğru Negatif Oranı
<b>UAV</b>	: Unmanned Aerial Vehicle

## GİRİŞ

Günümüzde görüntü işleme çalışmalarında birçok farklı sektörde çeşitli amaçlarla kullanılmaya başlanmıştır. Bunun en belirleyici örnekleri: iş yerlerinde yüz tanıma sistemleri, güvenlik kameraları için hareket algılama sistemleri, plaka tanımlama sistemleri, banka atmlerinde banknot tanıma sistemleri, imalat proseslerinde ürün tespiti sistemleri ve askeri güvenlik sistemleri verilebilir.

İnsansız hava araçları ve uydu sistemlerinde gerçekleşen son gelişmeler neticesinde anlık görüntü almak mümkün olmaktadır. Bu görüntüleri sürekli inceleyecek yetkililer bulunmasına karşın, hataya imkân vermeyen otomatik bir sistem ile gerçekleştirmek, doğru tespitin sürekliliği için önemlidir. Görüntü tespiti alanındaki iyileşmelerin ortaya çıkması konunun seçiminde belirleyici rol oynamaktadır.

Derin öğrenme yöntemleri, 2013 yılına kadar PASCAL VOC veri setinde yapılan nesne tespiti çalışmalarında tahmin yüzdeleri ortalama %40'ı aşamamıştır (Girshick, Donahue, Darrell ve Malik, 2014). Bu yüzdeler, derin sinir ağlarının bu alanda kullanımı ile yıllar içinde artış göstererek %80'ler seviyesini yakalamış ve üzerine çıkmıştır (Girshick, 2015). Makine öğrenmesinin spesifik bir parçası olan derin öğrenme, bu anlamda başarılı sonuçlar için, girdi dağılımındaki bilinmeyi açığa çıkarmaya çalışmaktadır.

Son dönemlerde derin sinir ağı mimarisi ile nesnelerin tanınmasında ve eylemlerin belirlenmesinde önemli başarılar elde edilmektedir. Bu mimarilerin güçlü ve ayırt edici özellik çıkarma ve temsil etme becerisi sayesinde belli bir nesne üzerinde derin özellikli bir tespit, harita çıkarımı, ağ yapısı kurgusunun oluşturulması mümkün hale gelmiştir (Kamran, Shahzad ve Shafait, 2018). Bu anlamda öne çıkan üç ana mimari, R-CNN (Regional Convolution Neural Network), Fast R-CNN ve Faster R-CNN şeklindedir.

R-CNN bölge çıkarımlı çalışan bir evrişimli sinir ağı mimarisine sahiptir. En temel biçimde nesnenin, nesne olma olasılığı olan bölge önerileri üzerinden belirlenmektedir. Evrişimli sinir ağı mimarisi işlemi öncesi boyutlarının eşitlenmesi sağlanmaktadır. Boyutları eşitlenmiş görüntüler evrişimli sinir ağından geçirilir. Elde

edilen deęerler bir destek vektör makinası üzerinden nesne sınırları belirlenir ve tahminleme işlemi gerçekleştirilir. (Xiaozhu, 2017) (Hsu, Chang ve Lin, 2016). Ürettięi başarılı sonuçlara karşın R-CNN mimarisinin en büyük dezavantajı eğitim ve test aşamalarına ayrılan sürenin oldukça uzun olmasıdır. Bu yöntemin sürelerinin optimize edilmiş farklı bir versiyonu Fast R-CNN mimarisidir.

Fast R-CNN mimarisi, R-CNN'e göre temel farkı tespit edilecek nesneye ilişkin bir bölge önerisi yapmak yerine, görüntüyü evrişimli sinir aęına doğrudan göndermektedir. Böylece orijinal resimle uyumlu, yüksek çözünürlüklü bir özellik haritası çıkarılmaktadır. Bu harita üzerinden seçici arama (selective search) yöntemi ile bölge haritaları çıkarılmış olmaktadır. Böylece bölge çıkarımı, orijinal resim üzerinden değil, özellik haritası üzerinden belirlenmektedir.

Derin öğrenme tekniklerinden Faster-CNN olarak tanımlanan bir diğer yöntem, Fast-CNN'deki gibi seçici arama ile bölge önerisi oluşturmak yerine bir bölge aęı önerisi kurgulamaktadır. Bölge Önerimli Aę (Region Proposal Network- RPN) olarak da adlandırılan bu yöntem, aę önerisi kurgusundan sonra Fast-CNN ile aynı işlemleri gerçekleştirmektedir. Kısaca bu yöntemde RPN sınıflandırıcı, RPN sınırlayıcı, son sınıf skorları ve son sınırlar gibi eğitilmesi gereken dört farklı aę mimarisi ile çalışılmaktadır (Hsu, Chang ve Lin, 2016).

Derin öğrenme tekniklerinden SSD (Single Shot Multibox Detector) farklı bir yaklaşım ile tek seferde nesne tanıma yapılmaktadır. Faster R- CNN de bölge önerisi ve bölge sınıflandırma 2 aşamada yapılmasına karşın, SSD tekniğinde bu ikisini tek seferde bir evrişimli sinir aęı içinde gerçekleştirmektedir.

Yapılan çalışmada kritik bölge olarak tanımlanan havaalanlarının, uydu görüntüleri ve insansız hava araçları görüntülerinden; belirlenmesi ve işaretlenmesi amaçlanmıştır. Bu belirleme işlemi gerçekleştirilirken aynı zamanda yapay sinir aęı mimarilerinin performanslarını kıyaslayarak, tercih edilme sürecine yön göstermek için hem SSD (Single Shot Multibox) algoritması ve Faster R-CNN algoritması kullanılmış sonuçlar değerlendirmelere tabi tutulmuştur.



# BİRİNCİ BÖLÜM

## HAVACILIK, İNSANSIZ HAVA ARAÇLARI VE UYDU SİSTEMLERİ

### 1.1 Havacılık Tarihi ve Gelişimi

Havacılık tarihi 9. yy da Abbas Ibn Firnas'a dayansa da, ilk uçağı uçurarak modern havacılık tarihinde yer edinen Wilbur ve Oliver Wright kardeşlerdir. (Lienhard, 2003). Wright kardeşler "The Flyer" ismini verdikleri planörleri ile 17 Aralık 1903 yılında beş görgü tanıkları ilk uçuşu gerçekleştirmiş ve toplamda elli dokuz saniye havada kalmışlardır.



**Şekil 1. "Wright Flyer" İlk Sürdürülebilir, Kumanda Edilebilir ve Güçlendirilmiş (Motorlu) Uçak (Wikipedia1)**

Sonrasında havacılık konusuna özellikle askeri alanda kullanmak üzere ülkeler yatırım yapmaya başlanmış 1909 yılında Amerika Birleşik Devletleri, Fransa ve İngiltere askeri teşkilatlarında hava sınıflarını oluşturmuştur. Sırası ile Osmanlı devleti 1911, Almanya ve Avusturya 1912 yılında ordu teşkilatlarına dahil etmiştir. (Yalçın, 2009)

1914-1918 yılları arasında gerçekleşen 1. Dünya Savaşı'nda uçaklar saldırı savunma ve keşif amaçlı olarak gerçek anlamda kullanılmıştır. 1918-1939 yılları arası uçakların Altın Çağı olarak adlandırılmış su ve hava soğutmalı motorlardan, komple alüminyum gövdeye sahip uçakların üretilmesi gibi ciddi anlamda büyük gelişmeler bu yıllarda meydana gelmiştir. 1939-1945 yılları arasında gerçekleşen 2. Dünya Savaşı'nda yeni bir sıçrama yaşanmış, ilk uzun mesafeli bombardıman uçağı ve ilk jet avcı uçakları bu dönemde üretilmiştir. 1945 yılından günümüze kadar olan sürede özellikle askeri havacılığın yanı sıra, ticari havacılık alanında ciddi atılımlar gerçekleştirilmiştir. Kuzey Amerika, Avrupa ve dünyanın diğer bölgelerini kapsayan uçuşlar yapan birçok şirket kurulmuştur. 2017 yılında dünya genelinde 4.1 Milyar yolcu taşınmıştır (ICAO, 2018).

## **1.2 İnsansız Hava Araçları Gelişimi ve Özellikleri**

Tarihte bilinen ilk İnsansız Hava Araçları A. M. Low tarafından 1916 yılında geliştirilmiştir. Sonrasında I. Dünya Savaşı sırasında sınırlı sayıda üretilen Hewitt-Sperry otomatik uçak kullanılmıştır. 1935 yılında Reginald Denny ilk ölçekli RPV (Remote Piloted Vehicle -Uzaktan Komutalı Araç) modelini geliştirmiş bu uçak özellikle II. Dünya Savaşı süresince çok fazla miktarda uçak üretilmiştir. Bu uçaklar trenleri korumak amacıyla uçaksavar ve saldırı görevlerinde kullanılmıştır.

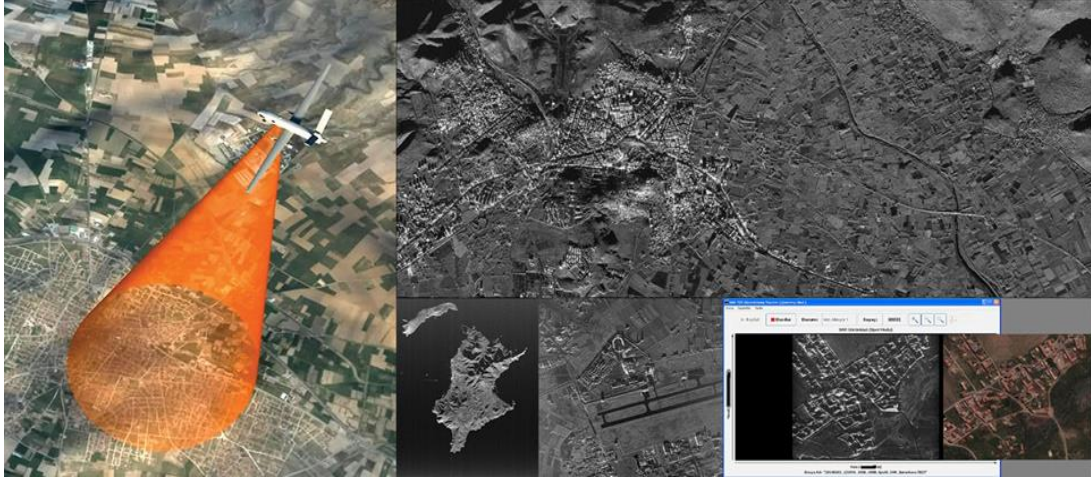
Jet motoruna sahip olan ilk insansız hava aracı 1951 yılında Teledyne Ryan firması tarafından geliştirilen Firebee I'dir. 1955 yılında ise Beechcraft tarafından ABD deniz kuvvetleri için Model 1001 isimli İnsansız Hava Araçları üretilmiştir (Taylor, 1977).

1970 yılına gelindiğinde, İHA geliştiricisi olarak İsrail ön plana çıkmıştır. İsrail'in Mısır'da yürüttüğü Yom Kippur savaşında İHA'ları görüntü almak için kullanmıştır.

Yine 2000'li yıllarda gelişen teknoloji ile faydalı yük kapasitesi artan insansız hava araçları orta ve yüksek irtifadan görüntü almaya başlamıştır.

Günümüzde İnsansız hava araçları; uzaktan algılama, taşıma, bilimsel araştırma, askeri amaçlar, arama ve kurtarma faaliyetlerini gerçekleştirmektedir. Bu faaliyetleri taşıdıkları faydalı yükler ile mümkün olmaktadır.

Faydalı yüklerin en önemlisi görüntüleme sistemleridir. Görüntüleme sistemlerinden yaygın olarak kullanılanlardan biri Sentetik Açıklıklı Radar (SAR) Sistemleridir. İnsanlı/İnsansız hava araçlarında, Uydu platformlarında keşif/gözetleme amaçlı gece/gündüz, her türlü hava koşulunda kullanılabilen, uzun menzillerden yüksek çözünürlüklü görüntü oluşturabilen gelişmiş radar sistemleridir. Keşif/Gözetlemenin yanı sıra SAR sistemleri, hassas konumlandırma ile havadan yere hedefleme/atış kontrolü, değişiklik tespiti, sınır güvenliği, yeryüzündeki değişimlerin izlenmesi ve yüksek çözünürlüklü sayısal yükseklik haritalarının oluşturulması amaçlarıyla da etkin olarak kullanılmaktadır. SAR sistemlerinin askeri amaçlı kullanımlarının yanında, afet risklerinin izlenmesi, afet sonrası hasar tespiti, gemi trafiğinin izlenmesi, harita ve kadastro, kaçak yapılaşmanın tespiti, tarım alanlarının izlenmesi, arama-kurtarma çalışmaları ve benzeri pek çok sivil uygulama alanı da bulunmaktadır. (SDT, 2016)



**Şekil 2. İHA Yer Gözleme ve SAR Sistemi ile Görüntü Alma (SDT, 2016)**

2016 yılında Türkiye ilk milli insansız hava aracını hizmete almıştır. Bununla birlikte insansız hava aracı üretici ve kullanıcı ilk 6 ülkeden biri konumuna gelmiştir.

İnsansız Hava Araçları Sınıfları aşağıdaki tabloda yer verilmiştir. Buna göre: Kategorileri, Görev Yüksekliği ve Görev Yarı Çaplarına göre aşağıdaki gibi sınıflandırılmıştır. (Karaağaç, 2012)

**Tablo 1. İHA Sınıflandırması (Karaağaç 2012)**

Sınıfı	Kategorisi	Görev Yüksekliği (ft)	Görev Yarıçapı (km)	Sivil Kategori
Sınıf I (150 kg'dan Hafif)	Mikro (<2 kg)	<200 (AGL)	5 (LOS)	Ağırlık Sınıfı Grup 1 Küçük İHA (<20 kg)
	Mini (2-20 kg)	<3.000 (AGL)	25 (LOS)	
	Küçük (>20 kg)	<5.000 (AGL)	50 (LOS)	Ağırlık Sınıfı Grup 2 Hafif İHA (20-150kg)
Sınıf II (150-600 kg)	Taktik	<10.000 (AGL)	200 (LOS)	
Sınıf III (600 kg'dan Ağır)	Orta İrtifa Uzun Havada Kalış (MALE)	<45.000 (MSL)	Limitsiz (BLOS)	Ağırlık Sınıfı Grup 3 Hafif İHA (>150kg)
	Yüksek İrtifa Uzun Havada Kalış (MALE)	<65.000	Limitsiz (BLOS)	
	Saldırı / Muharebe	<65.000	Limitsiz (BLOS)	

### 1.3 Uydu Sistemleri Gelişimi ve Özellikleri

1945 yılında uydu teknolojisinin temel fikri olan haberleşme kısmını ilk olarak Arthur C. Clarke ortaya atmıştır. “Sputnik 1” 4 Ekim 1957 tarihinde Sovyetler Birliği tarafından uzaya fırlatılan ve ilk kez yörüngeye yerleşmiş uydudur. Ardından Amerika Birleşik Devletleri tarafından 31 Ocak 1958 tarihinde Explorer-1 uydusu uzaya fırlatılmıştır.

1974 yılında NASA tarafından ilk uzaydan uydu görüntüsü çekilmiştir. Yeryüzünden 900 km çekilen bu görüntü aynı uydunun bulutsuz havada çektiği 595 resmin işlenmesi ile oluşturulmuştur.



**Şekil 3. Uzaydan Çekilen İlk Uydu Görüntüsü (NASA, 1974)**

Türkiye Cumhuriyeti, 2012 Yılında ilk Milli Gözlem Uydusu GÖKTÜRK-2'yi kendi milli tesislerinde üreterek gökyüzüne fırlatmıştır. Aşağıda GÖKTÜRK-2 uydusunun çekmiş olduğu ilk görüntülerden biri yer almaktadır.



**Şekil 4. Göktürk-2 Uydusu İlk Görüntülerinden**

Günümüzde dünya yörüngesinde konumlanan uydular, haberleşmeden, iletişime, meteorolojiden, navigasyona ve yer gözleme kadar birçok faaliyeti kesintisiz olarak gerçekleştirmektedir. Özellikle yer gözlem uyduları doğru konumlandırıldıklarında istenildiği irtifadan sürekli olarak görüntü alma imkânı sunmaktadır.

## İKİNCİ BÖLÜM

### DERİN ÖĞRENME VE BİLGİSAYARLI GÖRÜ

#### 2.1 Yapay Zekâ

İnsanoğlunun düşünen, tepki veren ve üreten bir varlık olduğu, daha anne karnındaki bir bebeğin ortamda bulunan seslere verdiği tepkiden anlaşılmaktadır. Gelişen birey ile birlikte düşünme, nesnelere algılama, akıl yürütme yetkinlikleri daha belirgin olarak ortaya çıkmaktadır. Bu durum binlerce nöronun yer aldığı insan beyninin özelliklerini ortaya koymaktadır.

Makinelerin düşünüp düşünemeyeceği sorusu ilk defa Alan Turing tarafından 1950 yılında yayınladığı “Computing Machinery and Intelligence” makalesinde sorulmuştur. İleride Turing testi olarak anılacak bu çalışma, makalede taklit oyunu (Imitation Game) olarak yer verilmiştir. Turing Testine göre bir dizi sorulan sorulara alınan cevaplara göre bir makinenin zeki olup olmadığı ortaya çıkmaktadır (Turing, 1950).

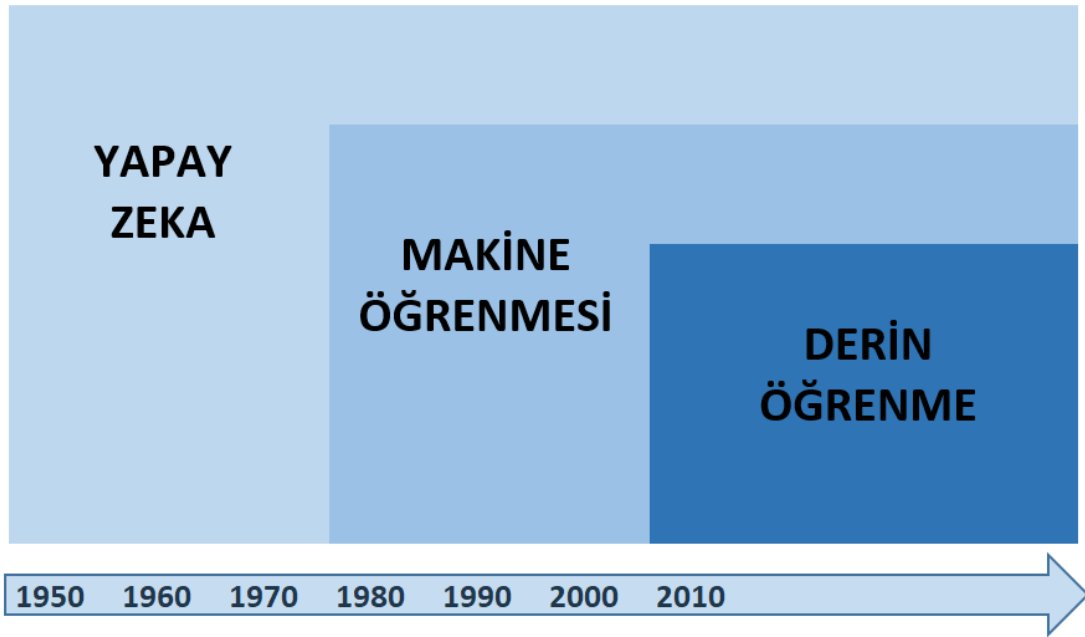
Alan Turing’in yapmış olduğu bu çalışmalara karşın, yapay zekâ terimini ilk kez 1956 yılında kullanan John McCarty’dır. Yine aynı yıl Dartmouth College’de düzenlenen konferans yapay zekânın doğuşu olarak kabul edilmektedir (Russel ve Norvig, 2009).

İnsan zekâsına özgü niteliklerin bir bilgisayar sisteminde gözlemlenmesi durumu yapay zekâ olarak nitelendirilmektedir. Farklı bir deęiş ile bir etkileşimin arkasında bir makinenin olup olmadığını ayırt edemiyorsak, karşıdaki yapı insanlara özel yeteneklerle tepki veriyorsa bir yapay zekâ ile karşılı olduğumuzu kabul edebiliriz.

Yapay zekânın hayatımıza nasıl yavaşça girdiğini irdeleyecek olursak karşımıza ilk olarak Dünya Satranç Şampiyonum Garri Kasparov ile Deep Blue isimli bilgisayarın 1996 yılındaki karşılaşması çıkmaktadır. Bu karşılaşmayı Kasparov’un kazanmasına karşın, 1997 yılında Deep Blue isimli bilgisayar insan müdahaleleri iddiaları da olsa 2,5’a karşı 3,5 puan ile Kasparov’u yenmiştir. 20 sene sonra yıl 2017 ye geldiğinde DeepMind’in geliştirdiği AlphaGo adlı yapay zekâ; satrançtan daha

karmaşık olan Go Oyununda dünya şampiyonunu şaibesiz olarak yenmiştir. Bunun dışında sürücüsüz otomobillerden cep telefonlarımıza kadar birçok alanda artık yapay zekâyla iç içe yaşamaktayız.

Yapay zekâ her ne kadar iyi çerçevesi çizilmiş mantıksal sorunlara çözmek için uygun olsa da, görüntüyü gruplama, dil çevrisi ve daha karmaşık olan problemlerde, programlanmadığı sonuçları öğrenebilme kabiliyeti olmadığı için, yetersiz kalmıştır. Bunun neticesinde Makine Öğrenmesi ortaya çıkmıştır.



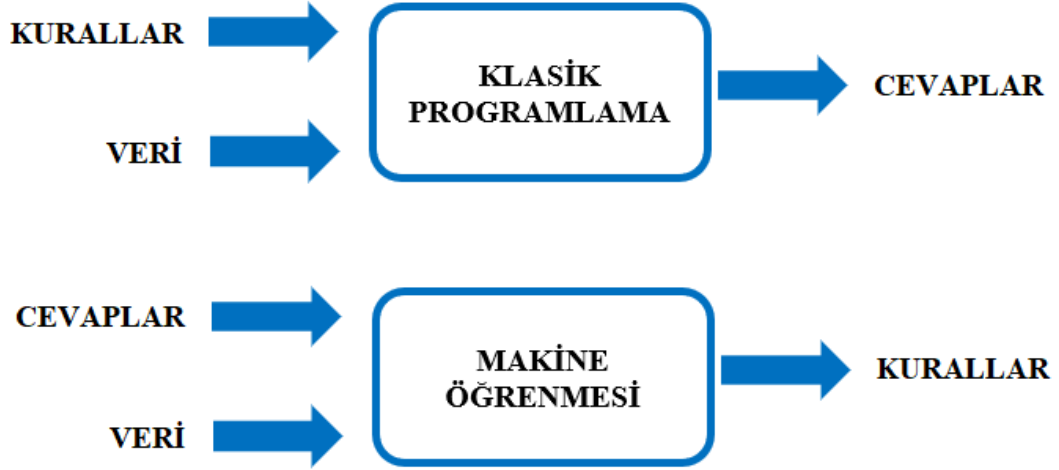
**Şekil 5. Yapay Zekâ, Makine Öğrenimi ve Derin Öğrenmenin Tarihsel Değişimi**

## 2.2 Makine Öğrenmesi

1959 yılında Arthur Samuel makine öğrenimini: “makinelere bilhassa programlanmadığı sonuçları öğrenebilme kabiliyeti” olarak tanımlamıştır. Buna göre makine öğrenimi, apaçık bir şekilde programlama gereksinimi duyulmadan belirlenmiş olan bir görevi bilgisayarların nasıl tamamlayacaklarını öğrendikleri durum olarak nitelendirilir. (Karunakaran, 2018)



Makine öğrenmesinin geliştirilmesinin altında, bir görevi kendi başına nasıl gerçekleştirebileceğini öğrenebilmesi mümkün olur mu? bilgisayar insanlığı şaşırta bilir mi? soruları yer almaktadır (Chollet, 2017).



**Şekil 6. Yeni Programlama Paradigması (Chollet, 2017)**

Şekil 6’da gözlemlendiği üzere makine öğrenimi temelinde sisteme kurallar ile verileri girdi olarak sağlayıp cevapları elde etmek yerine, cevaplar ve veriler yardımı ile eğitim gerçekleştirilmesi amaçlanmaktadır. Bunun sonucunda kurallar elde edilmiş olacak ve yeni cevapları üretmek için bambaşka verilere tekrardan uygulayabilecektir.

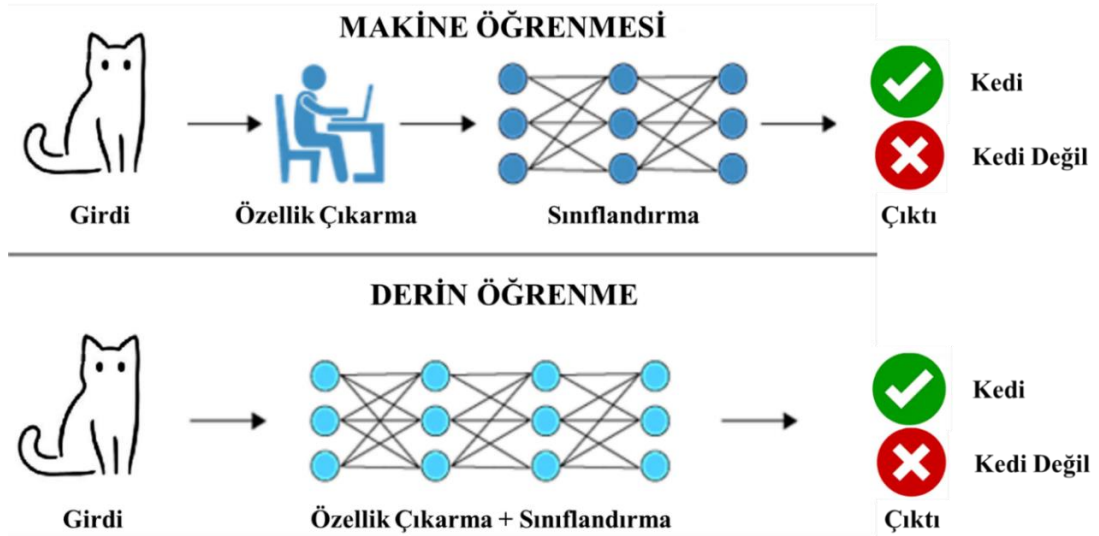
Makine öğrenmesi kuralları kendi oluşturması ve yeni cevapları bulması sayesinde tanıma ve robotik gibi alanlarda çokça soruna çözüm bulmamıza yardımcıdır.

### **2.3 Derin Öğrenme ve Bilgisayarlı Görüyü Anlama**

Derin öğrenme, veriyi ele alarak, özellik ve görevleri doğrudan öğrenerek ortaya çıkaran bir makine öğrenmesi tekniğidir. Ses, görüntü ve metin olarak birçok veri sisteme tanımlatılabilir. Derin öğrenme önceki bölümlerdeki şekilde gösterildiği üzere yapay zekâ ve makine öğrenmesi temellerine dayanmaktadır.

Makine öğrenimi tek katmanda işlem yaparken, derin öğrenme birçok katmanda aynı anda işlem yapmaktadır. Grup halinde makine öğrenim algoritmalarını aynı anda kullanarak tek işlemde sonuca ulaşmaya çalışmaktadır. Makine öğreniminde, insanoğlunun bugüne kadar edinmiş olduğu tecrübeleri, parametreler vasıtasıyla makineye tanıtılmaya çalışılmaktaydı, fakat derin öğrenme de bu farklılıkları makine kendi başına öğrenebilmektedir. Derin kavramı bir modelin katman sayıları ile kaç katmanın bulunduğu gibi derinliği ifade eder.

Bilgisayarlı görü ise derin öğrenmenin, görüntülerin işlenmesinde, algılanmasında ve çözümlenmesinde kullanılan yan bir kolu olarak nitelendirilebilir. Adından da anlaşılacağı gibi, bilgisayarlı görme bilgisayar aracılığıyla görme eylemini gerçekleştirmektedir. Bilgisayarlı görü uzun bir geçmişe dayalı çalışmalar sayesinde ortaya çıkmış araştırma sahasıdır. Derin Öğrenmenin ortaya çıkması ile bilgisayarlı görünün birçok uygulama için kullanışlı olduğu kanıtlanmıştır. Derin öğrenme aslında makine öğrenmesinin bir parçası olan Yapay Sinir Ağları (Artificial Neural Network - ANN) tekniklerinin bir araya getirilmesidir. Yapay Sinir Ağları (Artificial Neural Network - ANN) insan beynindeki gibi birbirine bağlı ve bilgiyi birbirine ileten düğümler gibi modellenmişlerdir (Shanmugamani, 2018).



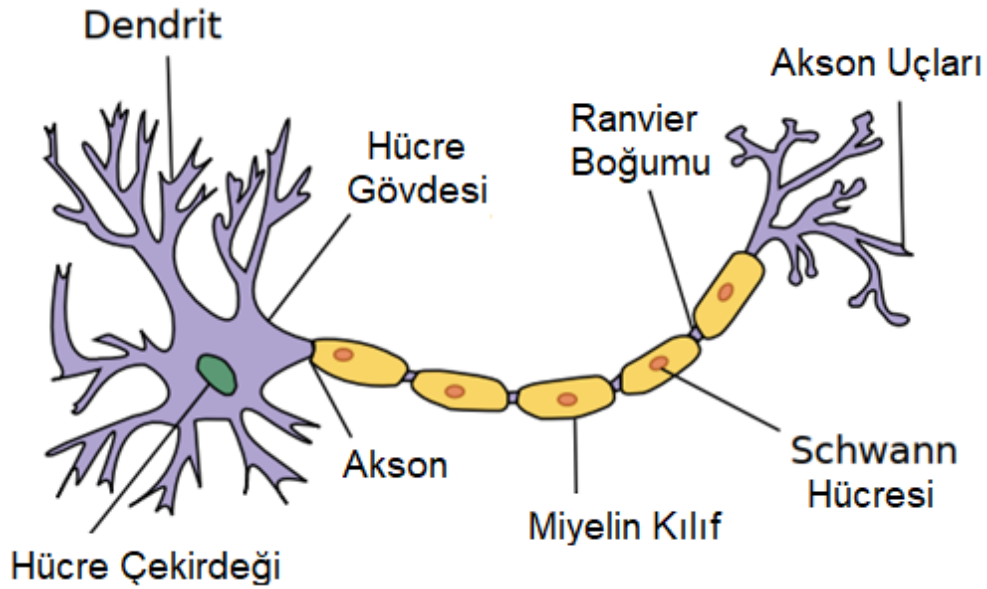
**Şekil 7. Derin Öğrenme ve Makine Öğrenme Yaklaşımları Kıyaslaması**  
(Karunakaran, 2018' dan esinlenmiştir)

Şekil 7’de gösterildiği üzere makine öğrenmesi ve derin öğrenme modelleri arasındaki temel fark özellik çıkarma alanı olup, makine öğreniminde insan vasıtası ile gerçekleştirilirken, derin öğrenme modellerinde özellikleri modelin kendisi belirlenmektedir.

## 2.4 Yapay Sinir Ağları

Yapay Sinir Ağlarının ortaya çıkışı insan beyninin hızlı ve doğru karar verişinin örnek alınmasına dayandığı söylenebilir. Temelde insan beynindeki sinir sisteminin matematiksel olarak modellenmesi denebilir.

Aşağıda yer alan Şekil 8’de biyolojik bir nöronun temsili resmine yer verilmiştir. Bu biyolojik nöronun dendritleri bilginin alındığı giriş uçları, çekirdeği ilk aşamada bilginin işlem görüp iletilip iletilmemesinin kararının verildiği kısım, akson uçları ise diğer nöronlara bağlanan terminaller olup çıkış kısmı olarak nitelendirilebilir (Zhang, Lipton, Lee ve Smola 2018).



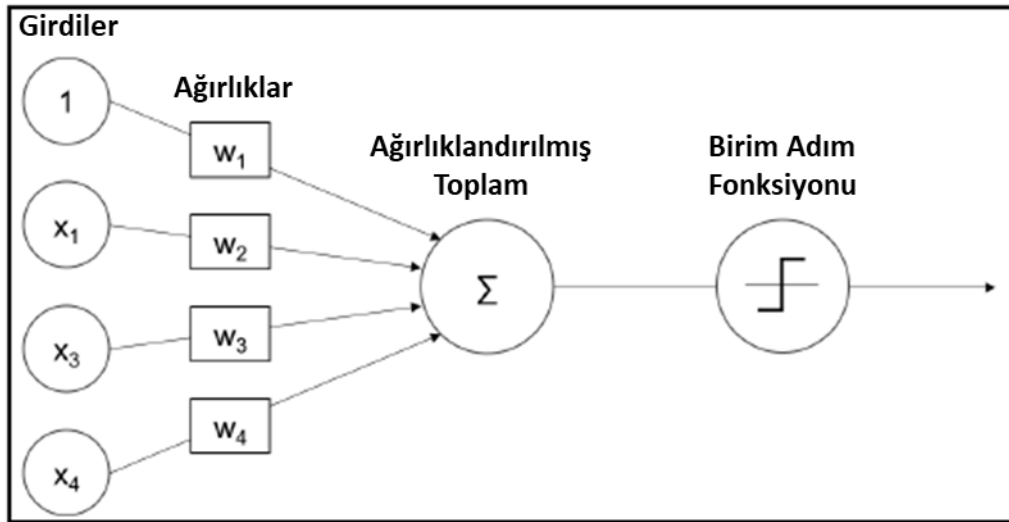
Şekil 8. Biyolojik Nöron (Zhang, Lipton, Lee ve Smola 2018)

Diğer nöronlardan ya da çevreden gelen  $x_i$  uyarı, etki ya da bilgisi dentrit uçları tarafından girdi kabul edilerek alınır. Sonrasında  $x$  bilgisinin girdilere cevap vermesi, girdilerin ağırlık ve önem oranının değiştirilmesi için  $w_i$  sinaptik ağırlıkları ile ağırlıklandırılmaktadır. Bütün bu ağırlıklandırılmış bilgiler toplanır akabinde önyargı, sapma ve yanılma olarak nitelendirilen bias  $b$  değeri eklenir.

$$z = b + \sum_{i=1}^n x_i w_i$$

İşlemin sonucu ortaya çıkması sonrasında ya hedefi olan kas grubuna ya da bir sonraki sinir hücresine daha fazla işlem yapılabilmesi için  $z$  aksonuna iletilir.

Perceptron (algılayıcı) tek katmanlı sinir ağı olarak adlandırılabilir. Bunun çok katmanlı hali ise Yapay Sinir Ağlarını ortaya çıkarmaktadır.



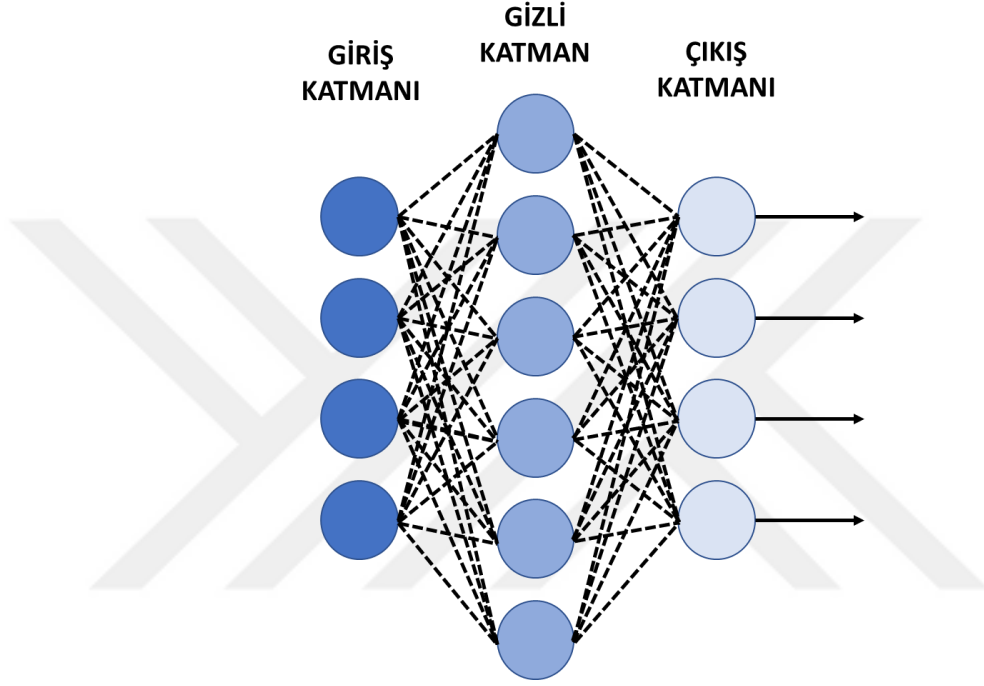
Şekil 9. Perceptron (Algılayıcı) (Shanmugamani, 2018)

Girdiler yukarıda görüldüğü Şekil 9'deki gibi ağırlıklandırılır ve toplanır. Toplam, herhangi bir ikili sınıflandırma problemine karşı, birim adım fonksiyonu boyunca ilerletilir. Bir algılayıcı örneklerden ağırlık öğrenmesi ile sadece basit fonksiyonları öğrenebilir. Ağırlıkların öğrenilmesi sürecine eğitim denir. Bir algılayıcı üzerindeki eğitim sonraki bölümlerde açıklanacağı üzere gradyan tabanlı metotlarca

gerçekleştirilebilir. Algılayıcının çıktısı aktivasyon fonksiyonu ya da transfer fonksiyonundan geçirilebilmektedir.

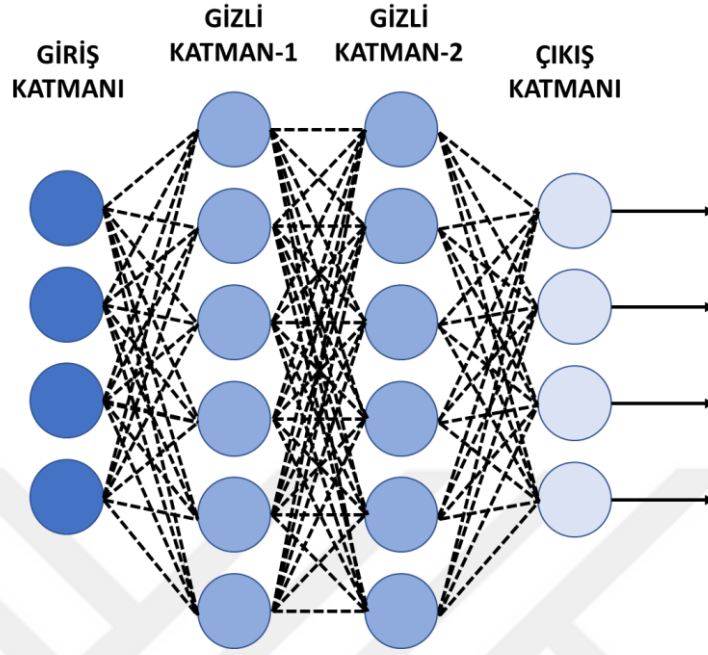
Yapay Sinir Ağları algılayıcılardaki nöronların ve katmanların artması ile meydana gelmektedir.

### TEK KATMANLI YAPAY SİNİR AĞI



Şekil 10. Tek Katmanlı Sinir Ağı Modeli (Nielsen, 2015'den esinlenmiştir.)

## ÇOK KATMANLI YAPAY SİNİR AĞI



Şekil 11. Çok Katmanlı Sinir Ağı Modeli (Nielsen, 2015'den esinlenmiştir.)

Şekil 10 ve Şekil 11'de belirtildiği gibi her bir katmanın modele etkisi, her bir nöronun da katmana dolaylı olarak etkisi bulunmaktadır. Bir yapay sinir ağında aynı katmanda bulunan nöronların birbiri arasında ilişki yoktur. Bilgi ve veri akışları sadece birbiri ardında gelen katmanlar arasında gerçekleştirilmesinin ardından çıkış katmanına doğru iletilir.

### 2.5 Aktivasyon Fonksiyonları

Çıktının performansını doğrudan etkilediği için aktivasyon fonksiyonları yapay sinir ağlarında kritik ve önemli bir rolü vardır. Bir aktivasyon fonksiyonu algılayıcının ateşlenip ateşlenmemesinin kararını vermektedir. Eğitim boyunca aktivasyon fonksiyonu, gradientlerin uyarlanmasıyla önemli rol oynamaktadır.

Aktivasyon fonksiyonunun yapay sinir ağlarında kullanılmaması halinde çıkış sinyalleri sadece doğrusal fonksiyonlardan ibaret olacaktır. Bu durum neticesinde

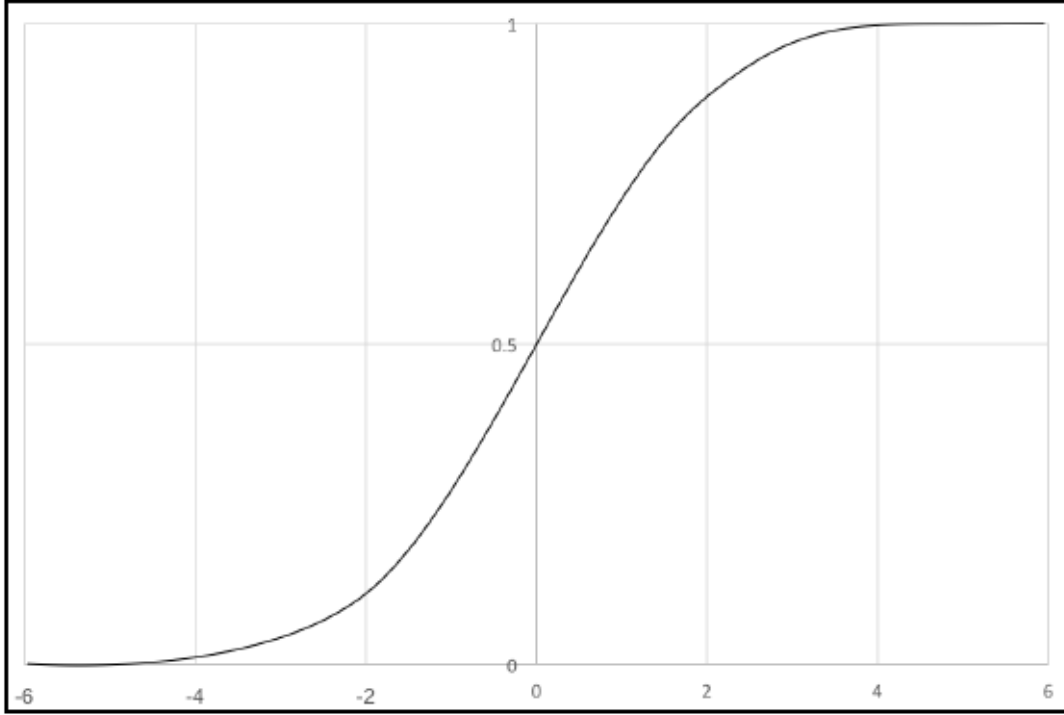
sinir ađının öğrenme gücü kısıtlı olacak ve lineer bađlanım davranışı gözlemlenecektir. Sinir ađında görüntü ses vb. gibi karmaşık bilgiler geldiđinde doğrusal yapı yetersiz kalacaktır. Doğrusal olmayan durumlarda bile ađın öğrenmesi kritiktir. Bir yapay sinir ađına aktivasyon fonksiyonları sayesinde doğrusal olmayan özellik kazandırılmakta ve bu da daha güçlü bir öğrenme imkânı sunmaktadır. Gizli katmanlarda hesaplanan matris çarpanları sonucunda ortaya çıkan nöron ađırlıkları bir sonraki katmana doğrusal olmayan bir yapıda giriş olarak hesaplanmaktadır (Stanford University, 2019).

Kullanılmakta olan 3 aktivasyon fonksiyonu ise sigmoid, hiperbolik tanjant fonksiyonu (Tanh) ve ReLU (Rectified Linear Unit) fonksiyonlarıdır.

### **2.5.1 Sigmoid Fonksiyonu**

Sigmoid düzgünleştirilmiş basamak fonksiyonu ya da Hence Türevi olarak düşünülebilir. Sigmoid, herhangi bir deđerin olasılıđa dönüştürmede çok kullanışlıdır. Bu sebepten ikili sistem sınırlandırmasını gerçekleştirebilir.

Sigmoid grafiklerinde girdi 0 ile 1 aralığında deđer alır. Örneđi aşıđıdaki gibidir:



**Şekil 12. Sigmoid Fonksiyonu Grafik Gösterimi**

Sigmoid fonksiyonu basitçe:

$$(0,1) \text{ aralığı için } f(x) = 1/(1 + e^{-x})$$

Sigmoid fonksiyonunun ismi de fonksiyonun kartezyen uzayda çizilmiş halinin andırdığı S harfinden (sigma) gelmektedir.

Bu fonksiyon özellikle yapay sinir ağındaki sinir hücrelerinin aktivasyonu (ateşlenmesi, fire) oldukça kullanışlıdır.

Y değerleri x deki değişikliklere özellikle grafiğin uçlarına doğru çok az tepki vermektedir. Bu nedenden gradyanların ölmesi/kaybolması (vanishing gradient) gerçekleşir. Yavaş bir öğrenme olayı gerçekleştirildiğinde hatayı minimize eden optimizasyon algoritması yerel (lokal) minimum değerlere takılabilir Yapay sinir ağından alınacak performans bu sebepten düşecektir.

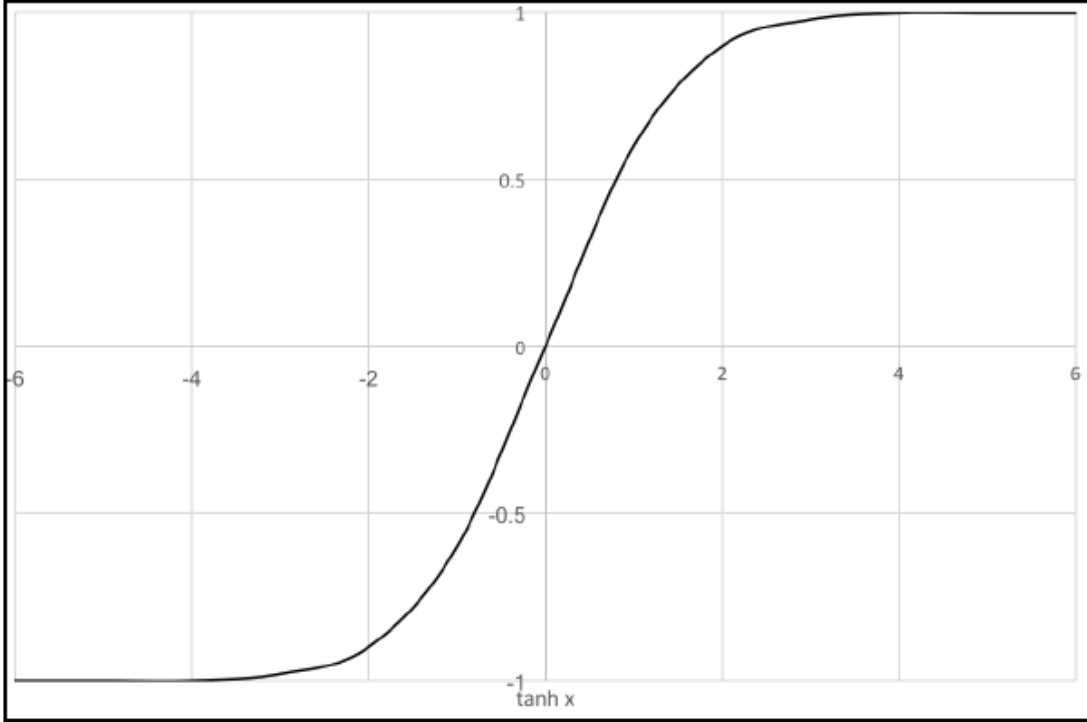


## 2.5.2 Hiperbolik Tanjant Fonksiyonu (Tanh)

Diğer bir aktivasyon fonksiyonu olan hiperbolik tanjant fonksiyonu ya da tanh fonksiyonu, sigmoid fonksiyonunun ölçeklendirilmiş halidir. Hiperbolik tanjant fonksiyonu incelendiğinde fonksiyonun aralığı (-1, +1) olarak değer almaktadır.

Hiperbolik Tanjant Fonksiyonu (Tanh) fonksiyonunun grafiği aşağıdaki gibidir.

$$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$



**Şekil 13. Hiperbolik Tanjant Fonksiyonu Grafik Gösterimi**

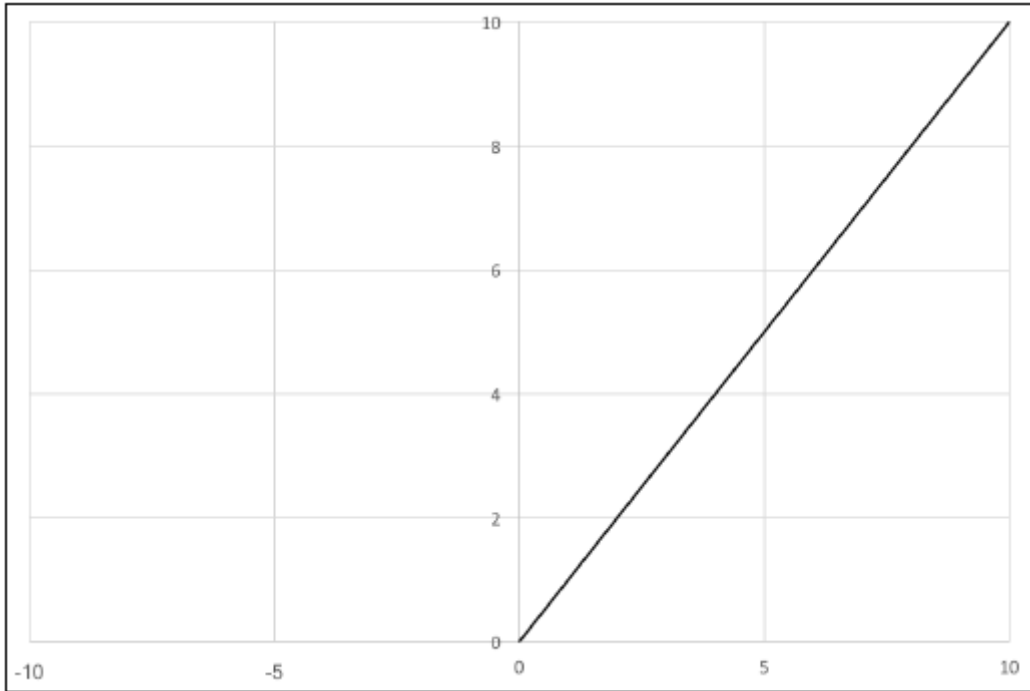
Sigmoid fonksiyonuna göre dik olmasından kaynaklı daha fazla değer alabilmekte ve daha karardır. Bu nedenden dolayı gradyanların ölmesi /kaybolması (vanishing gradient) daha az gözlemlenir. Fonksiyonun uçlarında gradyanların ölmesi/kaybolması problemi devam etmektedir.

### 2.5.3 ReLU (Rectified Linear Unit) Fonksiyonu

İlk bakışta pozitif ekseninde doğrusal fonksiyon ile aynı özelliklere sahip gibi görünmekle birlikte, ReLU fonksiyonu büyük sayıların geçmesine izin vermektedir. Bu birkaç nöronu geçersiz kılmakta ve aktive olmasını (ateşlenmesi, fire) engellemektedir. Bu durum seyrekliği arttıracak olup bundan dolayı iyidir. ReLU grafiği için x değerlerinden maksimumuma (0, x) değerler alır, negatif girdi değerleri “0” alırken pozitif girdiler bir değişikliğe uğramaksızın çıktı olarak yansır.

ReLU fonksiyonunun grafiği aşağıdaki gibidir.

$$[0, \infty) \text{ aralığı için } f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ x & \text{ için } x \geq 0 \end{cases}$$



**Şekil 14. ReLU (Rectified Linear Unit) Fonksiyonu Grafik Gösterimi**

ReLU daha hızlı eğitim sağlar çünkü her zaman ateşleme yapmamaktadır. ReLU büyük bir alandaki problemlerin çözümlerinde kullanılabilir. ReLU fonksiyonu ile ilgili tek problem işlem hızı kazandıran sıfır bölgesinde eğitimin gerçekleşmiyor olmasıdır.

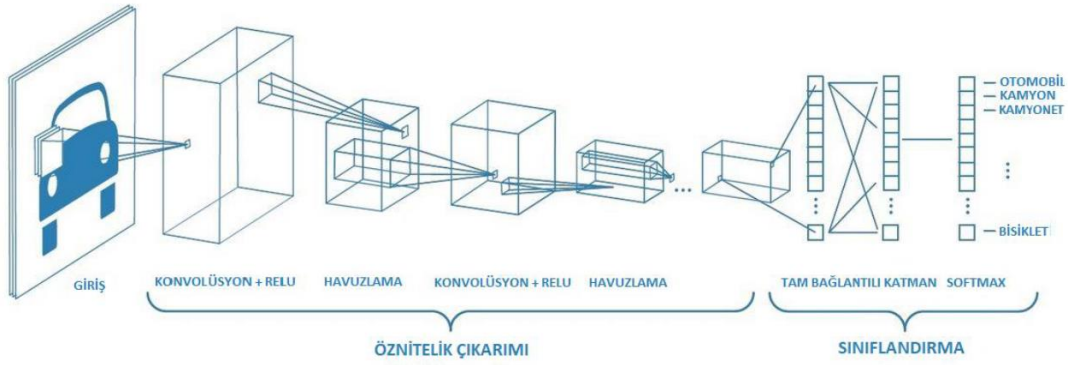
## 2.6 Konvolüsyonel (Evrışimli) Sinir Ağları (CNN)

Klasik makine öğreniminin en zorlu işlemlerinden birinin özellik çıkarımı olduğunu bahsetmiştik. Söz konusu özelliklerin belirlenmesi konusunu makinenin kendisinin gerçekleştirmeyip hazır halde girdi olarak verilmesi nedeniyle alanında yetkin kişilerin bir araya getirilmesi gerekmektedir. Derin öğrenmede ise girdi olarak özellik çıkarımına gerek duyulmamakta, özellik çıkarımı veri üzerinde karmaşık doğrusal olmayan ilişkilerin öğrenimi gerçekleştirilmektedir (İnik ve Ülker, 2017).

Derin öğrenme kavramını eskiye dayanmasına karşın, uzunca bir süre uykuda kalmasının sebebi geçmişte yeterli veri elde edilememesi, büyük veri kavramı olmaması ve güçlü donanımların bulunmamasından kaynaklıdır. Ana işlemcilerin ve paralel işlemci teknolojisinin gelişmesi ile birçok gizli katmana sahip ağların eğitimi ve karmaşık hesapların gerçekleştirilmesine imkân sağlamıştır.

Bir insan uzaktaki bir şekle baktığında o şeklin özelliklerini sorun çekmeden ayırt edebilir. Özelliklerine göre ne olduğunu bilebilir. Aynı görüntülere bilgisayarlar baktığında bu resimleri matris olarak değerlendirilmektedir. Resimdeki her bir piksel için matrise bir değer karşılık gelecektir. Evrışimli sinir ağları birçok gizli katmanda işlem yaparak işler bunun neticesinde farklı özellikler algılanır. Ortaya çıkan özellikler ile nesnelere kolay ayırt edilebilir.

Önceki başlıklarda CNN (Konvolüsyonel Sinir Ağı – Convolutional Neural Network) birçok katmandan oluştuğundan bahsedilmiştir. Bu katmanlar; giriş katmanı, konvolüsyon katmanı, havuzlama katmanı, aktivasyon katmanı, dropout katmanı, tam bağlantılı katman ve sınıflandırma katmanıdır. Konvolüsyonel Sinir Ağları bu katmanın farklıca sıralanması ile oluşmaktadır.



**Şekil 15. Konvolüsyonel Sinir Ağlarını Oluşturan Katmanlar (Mathworks, 2019)**

### 2.6.1 Giriş Katmanı (Input Layer)

Konvolüsyonel Sinir Ağına verinin ilk giriş yaptığı noktadır. Bu katmandaki verinin boyutunun artışı çıkarılacak sonucun başarısını olumlu yönde arttırmasına karşın hem eğitim süresi hem de test süresini arttıracaktır. Aksi durum da geçerli olup düşük boyuttaki resim test ve eğitim süresini azaltmasına karşın eğitimin başarısını olumsuz yönde etkilemesi söz konusudur (İnik ve Ülker, 2017).

### 2.6.2 Konvolüsyon katmanı (Convolution Layer)

Bu katmanda görüntü yatay ve dikey olarak taranarak konvolüsyon işlemini gerçekleştirilmektedir. Bu işlemlerin neticesinde özellik haritaları oluşturulur. Her bölge için girdiler alınır ve bir ağırlık matrisi ile çarpılır sonrasında eşik değeri ilave edilir. Konvolüsyon işlemi; bir çekirdek şablonun (matrisin/kernel) resim üzerindeki piksellerle 'kaydırma ve çarpma' işlemi olarak tanımlanabilir. Bu işlemde çekirdek şablon resim üzerinde kaydırılır ve değeri resim üzerindeki uygun piksellerle çarpılır Konvolüsyon işlemi; bir çekirdek şablonun (matrisin/kernel) resim üzerindeki

piksellerle 'kaydırma ve çarpma' işlemi olarak tanımlanabilir. Bu işlemde çekirdek şablon resim üzerinde kaydırılır ve değeri resim üzerindeki uygun piksellerle çarpılır.

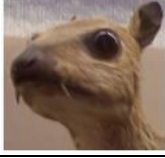





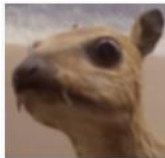

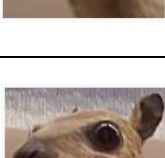
Konvolüsyonel işleminin denklemi aşağıdaki gibidir.

$$S(i, j) = (I * K)(i * j) = \sum_m \sum_n I(i + m, j + n)K(m + n)$$

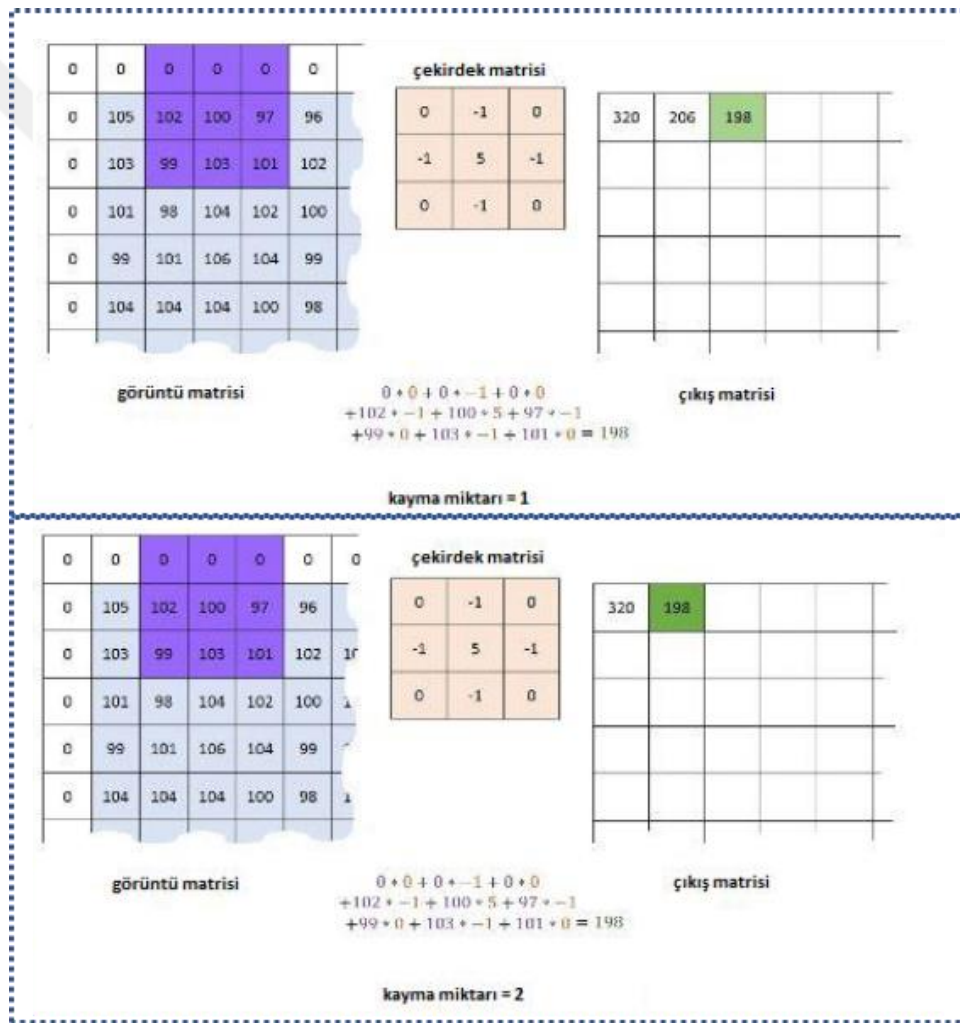
Denklemde  $I$  giriş görüntüsü,  $K$  çekirdeği (kernel);  $S$  ise konvolüsyon işlemi çıktısını göstermektedir. Konvolüsyon işlemi '\*' işareti ile gösterilmektedir (Googfellow, Bengio ve Courville, 2016).

Çekirdek matrisinin belli özel uygulamalar için çeşitli standart şablonları vardır. Burada şablonun büyüklüğü ve şekli, işlemin özelliklerini belirler. Konvolüsyon filtreleri çekirdek matrislerin (kernel) temeline dayanır. Aşağıda Tablo 2'de belli başlı çekirdek matrisleri ve kullanım amaçlarına yer verilmiştir.

**Tablo 2. Çekirdek Matris (Kernel) Örnekleri ve Kullanım Alanları (Wikipedia2)**

İşlem	Çekirdek Matrisi (Kernel - $\omega$ )	Görüntü Sonucu
Tanımlama	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Kenar Tespiti	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Keskinleştirme	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Bulanıklaştırma	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian Bulanıklaştırma 3 x 3	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian Bulanıklaştırma 5 x 5	$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Keskinlik Azaltım Maskesi 5 x 5	$\frac{-1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

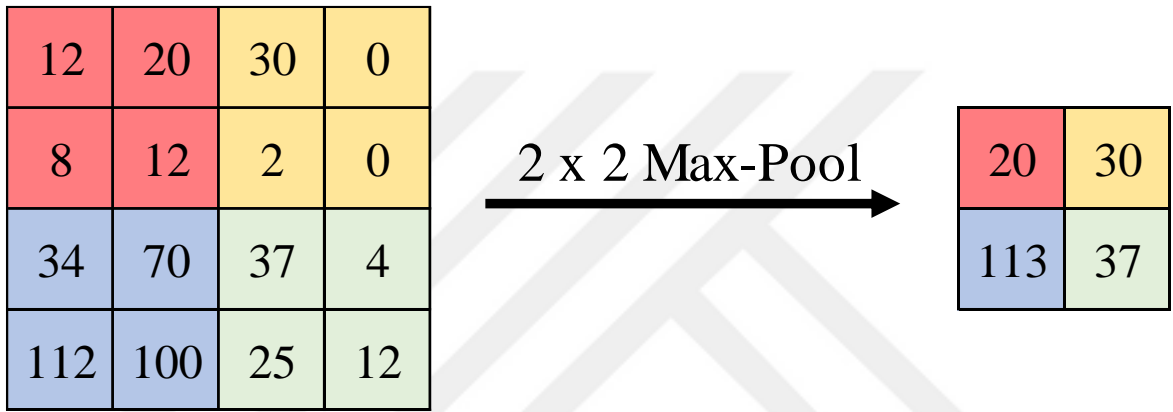
Konvolüsyon işleminde kullanılacak çekirdek çerçevesi giriş görüntüsünün üzerinde düşey ve yatay olarak dolaştırılarak görüntünün piksel değeri ile çarpılır. Çıktının toplamı alınarak ilgili piksel alanına yazılır. Yatay düzlem bittikten sonra bir birim aşağı kaydırılarak yatay düzlemde hesap yapılmaya devam edilir. Renkli görüntü işlenmesine karar verilirse her 3 renk kanalı için işlemin ayrı ayrı uygulanması gerekmektedir. Üçünün toplamı sonucunda aktivasyon haritasının ortaya çıkması sağlanır. Çekirdek çerçevesi dolandırılma işlemi yapılırken her bir işlem sonrasında yatay hareketinin kaç birim kayacağına kayma miktarı (stride) parametresi ile belirlenmektedir.



**Şekil 16. Keskinleştirme Çekirdek Matrisi ile Kayma Miktarı 1 ve Kayma Miktarı 2 için Konvolüsyon İşleminin Uygulanması (Machinelearninguru.com, 2019)**

### 2.6.3 Havuzlama Katmanı (Pooling Layer)

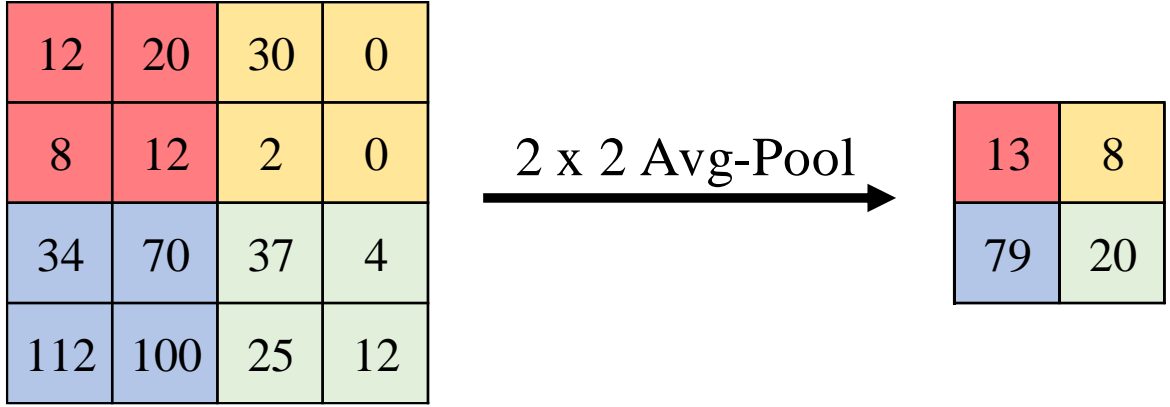
Havuzlama katmanı hesaplama alanı olarak kabul edilir. Bu katmanın amacı modelin başarısını arttırmak ve yeniden boyutlandırma yapmak için özellik çıkarmaktadır. Aşağıdaki Şekil 17’te yer aldığı üzere maksimum havuzlama modeldeki en yüksek piksel değerini alıp çıkarmaktadır. Bunu gerçekleştirirken 2\*2 filtre uygulanmış, kayma miktarı 2 olarak tutulmuştur (Salomon, 2018).



Şekil 17. Maksimum Havuzlama (Firelord Phonix, 2018)

Aşağıdaki Şekil 18’te yer aldığı üzere ortalama havuzlama modeldeki kullanılmış her grubun ortalaması alınarak işlenmiştir. Bunu gerçekleştirirken 2\*2 filtre uygulanmış, kayma miktarı 2 olarak tutulmuştur.

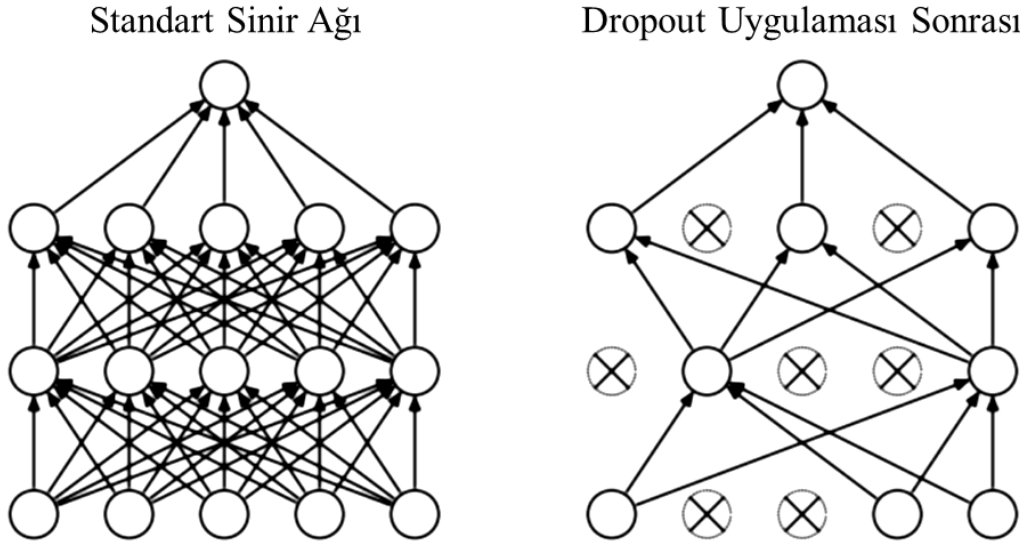




Şekil 18. Ortalama Havuzlama (Firelord Phonix, 2018'den esinlenmiştir)

#### 2.6.4 Bırakma Katmanı (Dropout)

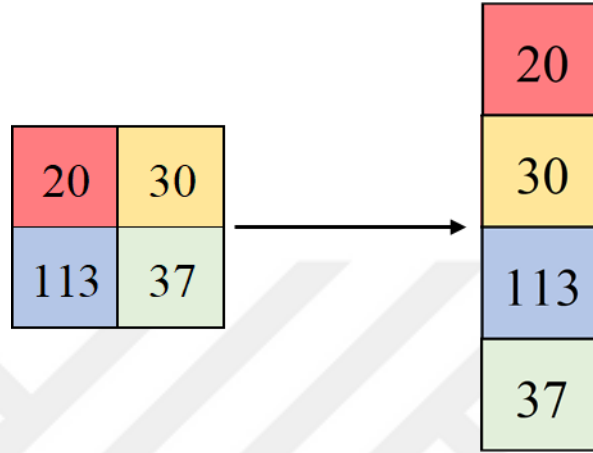
Bırakma katmanının asıl amacı konvolüsyonel sinir ağlarında ezberlemeyi önlemektedir (Verma ve Pal, 2016). Bu durum sayesinde konvolüsyon ağının çok fazla yüklemesi ya da tek bir nörona bağımlılığı önlenmiş olur. Bunu yapmak amacıyla eğitim aşamasında rastgele seçilen nöronların bazıları ya da tek tek ortadan kaldırılır. Dropout belirli bilgilerin yokluğunda bile modeli doğru bir şekilde öğrenme noktasında zorlar. Şekil 19'da standart sinir ağı ve bırakma katmanı sonrası sinir ağına yer verilmiştir.



Şekil 19. Standart Sinir Ağı ve Dropout (Bırakma Sonrası) Sinir Ağı  
(Budhiraja, 2016)

### 2.6.5 Tam Bağlı Katman (Fully Connected Layer)

Konvolüsyonel sinir ağları çoğunlukla bir ya da daha fazla tam bağlı katman ile neticelenir. Diğer katmanlardan elde edilen çıktılar Şekil 20'deki gibi düzleştirilmesini sağlar.



Şekil 20. Düzleştirme işlemi (Flattening)

### 2.6.6 Sınıflandırma Katmanı

Tam bağlı katman sonrasında geliyor olup, sınıflandırma işleminin yapıldığı son katmandır. Sınıf sayısı kadar çıkış değeri ortaya çıkarır. Örnek olarak, 5 farklı hayvan türünün sınıflandırıldığı bir yapıda 5 adet çıkış üretilmekte ve bu çıkışlar 0-1 aralığında olmaktadır. 1 değerine en yakın olan ağı tahmin ettiği sınıftır. Yüksek başarıyı göstermesi nedeniyle çoğunlukla softmax sınıflandırıcısı kullanılmaktadır (İnik ve Ülker, 2017).

## 2.7 Yapay Sinir Ağı Çeşitleri

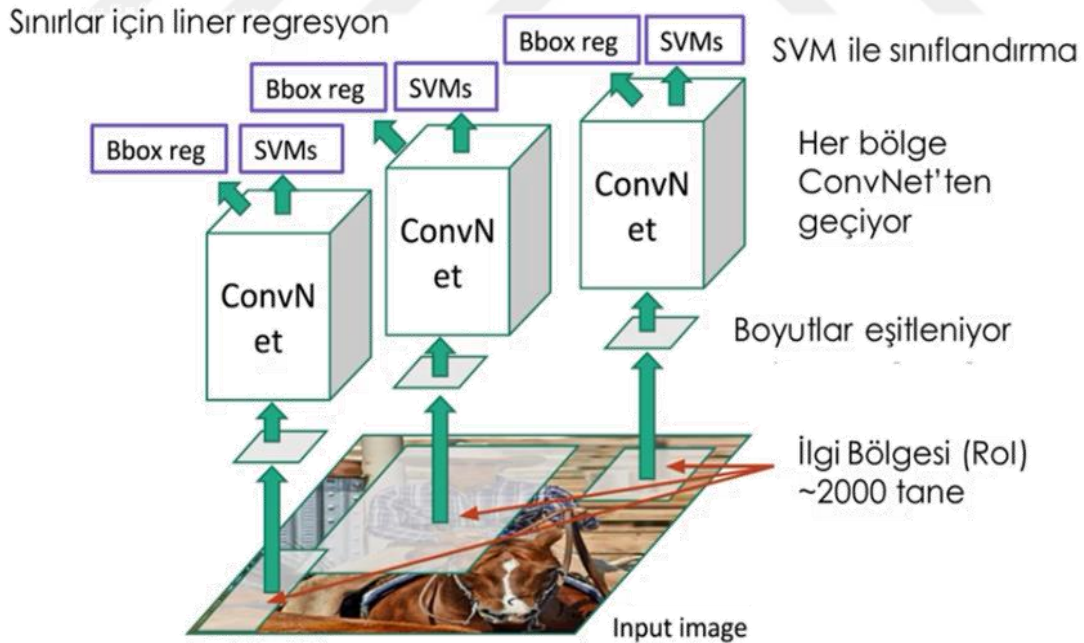
2006 yılı sonrası kullanılmaya başlayan yapay sinir ağları görüntü tespitlerinde etkin rol oynamaya başlamış özellikle 2013 yılı sonrasında doğru tahmin oranları %80'lere yaklaşmış, 2016 yılı sonrasında doğru tahmin oranları %90'ların

üzerine çıkmıştır. Aşağıda günümüzde sıklıkla karşımıza çıkan, R-CNN, Fast R-CNN, Faster R-CNN ve SSD – Single Shot Multibox, yapay sinir ağı çeşitlerine yer verilmiştir.

### 2.7.1 R-CNN (Region Based Convolutional Neural Networks)

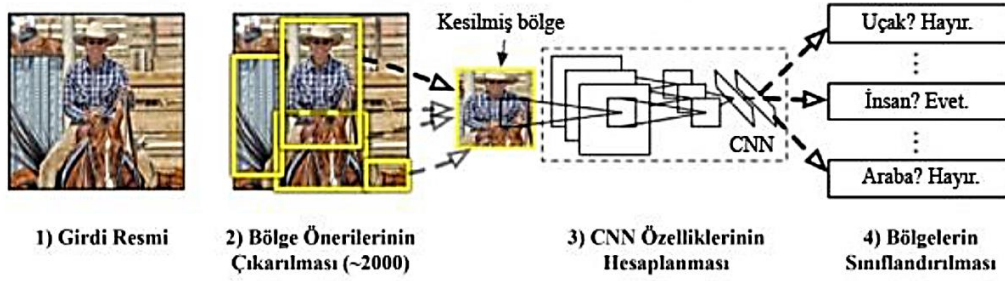
CNN'ler ile nesne tespiti üzerine öncü çalışmalardan birini Girshick, Donahue, Darrell, Malik, Berkeley (2014) gerçekleştirmişlerdir.

R-CNN (Bölge Tabanlı –Evrışimli Sinir Ağı)'de bölge önerisi kullanılmakta olup R-CNN'de önerilen bölgelere ilgi bölgesi (region of interest) denilmektedir. Önerilen farklı boyuttaki pencereler evrişimli sinir ağından geçirilmesi için boyutları eşitlenmektedir. Sınıflandırma yapan Evrişimli Sinir Ağları hep belli boyutta girdi almaktadır. Her bölge Evrişimli Sinir Ağlarında geçirilir sonrasında SVM (Support Vektor Mission – gözetimli öğrenimde kullanılan bir makine öğrenimi yöntemidir.) ile sınıflandırılır. Lineer regresyon ile nesnenin sınırların belirlenmesi ve tam oturtulması gerçekleştirilir.



Şekil 21 R-CNN Mimarisi

Özetlenirse R-CNN’de ilk olarak görüntü verilir ardından nesnelerin olabileceği 2000 üzerinde bölge önerisi oluşturulur. Sunulan bu bölge önerilerinin her biri sinir ağından geçirilir. Son olarak eski bilgilere dayanan SVM metodu ile sunulan bölgedeki nesnenin hangi nesne olduğunun bulunması ve bölgelerin lineer regresyon modelinden geçirilerek sınıflandırılmış olan nesnenin koordinatlarının daha kesin bir biçimde bulunması işlemleri gerçekleştirilir. Eğitim bu sebeple oldukça yavaş gerçekleşmektedir. Yavaş gerçekleşmesi gerçek zamanlı uygulamalarda kullanımındaki en büyük engellerden biridir.



**Şekil 22. R-CNN Çalışma Mantığı (Girshick, Donahue, Darrell ve Malik, 2014)**

### 2.7.2 Fast R-CNN

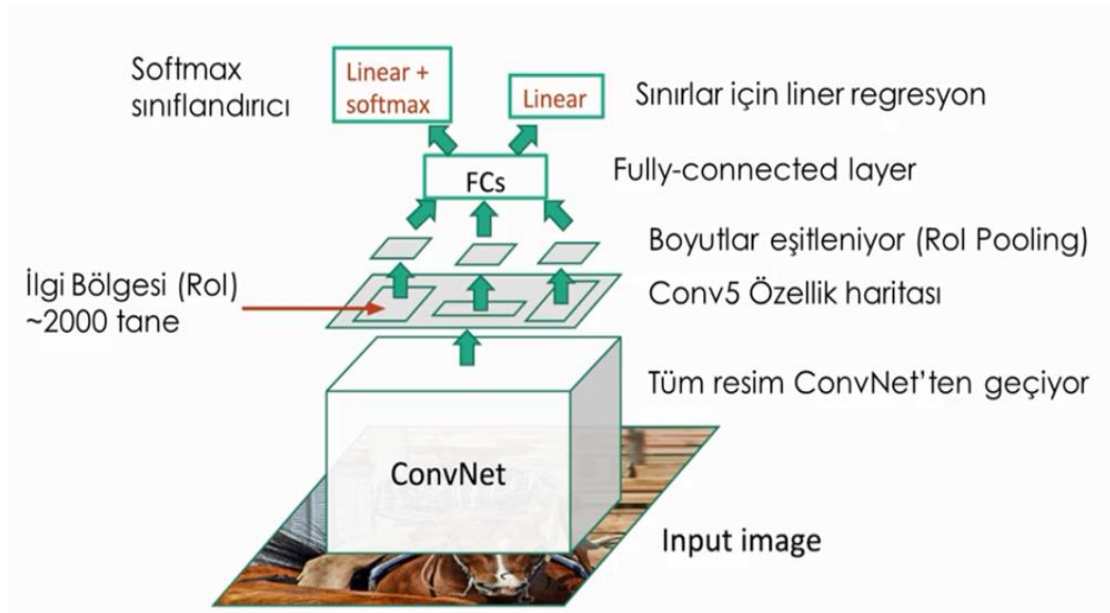
Fast R-CNN (Hızlı Bölge Tabanlı –Evrışimli Sinir Ağı)’de mantık R-CNN ile aynı olup, hızlandırmak için farklı teknikler kullanılmaktadır. Fast R-CNN de resim üzerinde doğrudan bölge önerisi yapmak yerine bütün resim doğrudan Evrişimli Sinir Ağından geçirilmektedir. Bunun sonucunda orijinal resme uyan yüksek çözünürlüklü bir özellik haritası oluşturulur.

Oluşturulan özellik haritasında selective search ile 2000 civarı ilgi bölgesi (Bölge Önerisi) oluşturulur. Bunun neticesinde her bölge Evrişimli Sinir ağından geçirmek yerine resim sadece 1 kez evrişimli sinir ağından geçirilmiş olacaktır. (Uzun süren işlem olan evrişimli sinir ağı işlemleri azaltılmış olmaktadır.)

Önerilen bölgelerin boyutları eşitlendikten sonra, bu bölgeler alınıp Fully Connected Layer’e bağlanmaktadır.

Lineer ve Softmax sınıflandırmak için (sınıflandırmak için yeni model oluşturmak yerine sinir ağı içinde sınıflandırma gerçekleştiriliyor), sınırlar için lineer regresyon kullanılmaktadır.

Fast R-CNN zamanın çoğunu bölge önerisi yapmak için kullanmakta bu durum modelin eğitimi ve testi aşamasında dar boğaza neden olmaktadır.



Şekil 23. Fast R-CNN Mimarisi (Girshick, 2015)

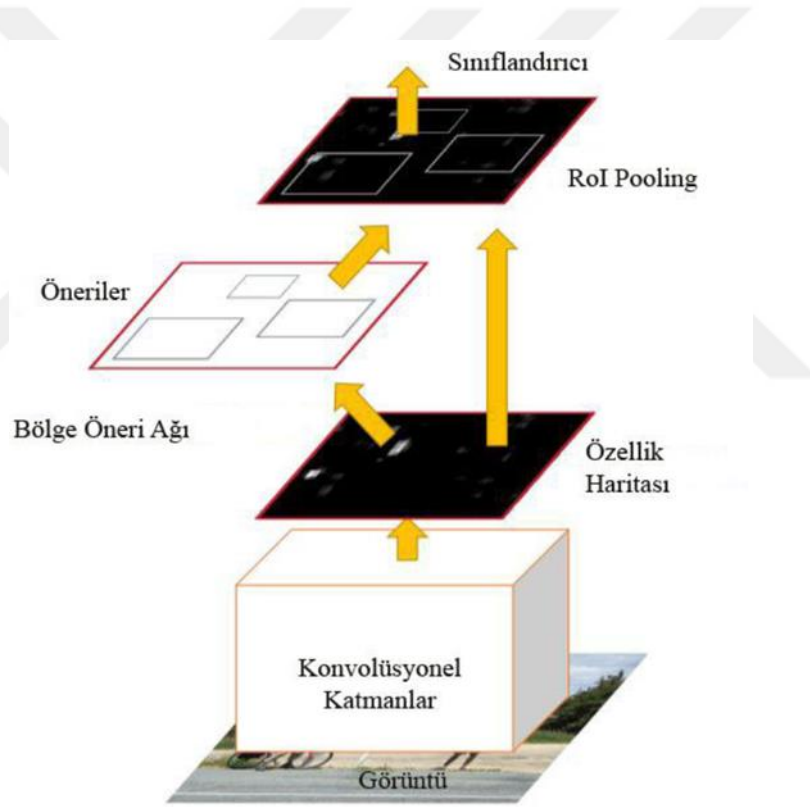
### 2.7.3 Faster R-CNN

Fast R-CNN’de yer alan dar boğazı fark eden Ren, He, Girshick ve Sun seçici tarama işleminden sistemi kurtaracak yeni bir ağ geliştirmişlerdir. Bunun bir neticesi olarak Faster R-CNN ortaya çıkmıştır.

Faster R-CNN ‘de girdi olarak resim alınır bu resim evrişimli sinir ağından geçirilip bir özellik haritası çıkarılır.

Faster R-CNN (Daha Hızlı Bölge Tabanlı –Evrışimli Sinir Ağı) Selective Search ile bölge önerisi yapmak yerine ayrı bir bölge önerisi ağı oluşturulmakta, bu önerileri network içinde yapılması sayesinde hız kazanmaktadır. Bölge önerileri bu ağ üzerinden yapılmaktadır.

Belirlenen bölgeler yeniden şekillendirildikten sonra fully connected layer'den geçirilir ve sınıflandırma yapılır. Bu durumda hem bölge önerisi veren ağı eğitmemiz hem de normal evrışimli işlemlerin yapıldığı ağı eğitmemiz gerekmektedir. Bölge ağı her bölge için karar vermesi ve pencere boyutunu bulması gerekir. Asıl sinir ağı ise sınıflandırarak baktığı bölge içinde nesne var mı tespit edecek bulduğu nesnenin sınıflarını belirleyecektir.



**Şekil 24. Faster R-CNN Mimarisi**

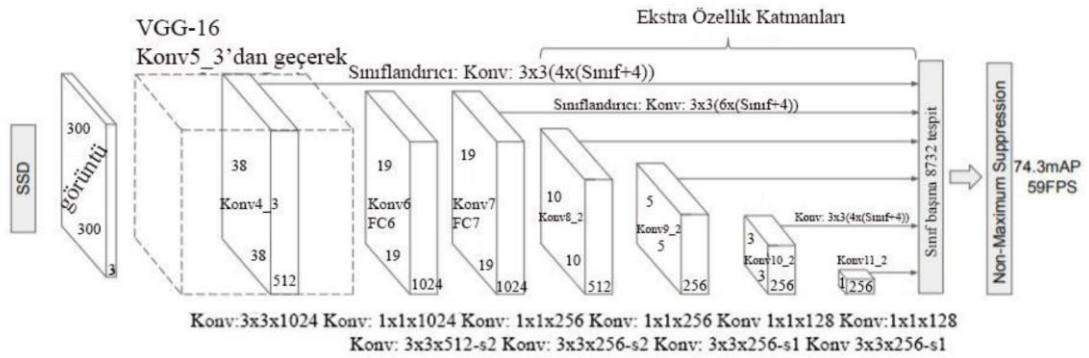
## 2.7.4 Single Shot Multibox Detector (SSD)

Liu, Anguelov, Erhan, Szegedy, Reed, Fu ve Berg 2016 yılında gerçek zamanlı nesne tanıma uygulamaları için Single Shot Multibox Detector (SSD) adını vermiş oldukları yeni bir derin öğrenme metodu gerçekleştirmişlerdir. Faster R-CNN, sınır kutuları oluşturmak için bir bölge öneri ağı kullanmaktadır ve bu kutuları nesnelere sınıflandırmak için kullanır. SSD- Tek seferde nesne tanıma yapılmaktadır. Faster R- CNN de bölge önerisi ve bölge sınıflandırma 2 aşamada yapılmaktadır. SSD bu ikisini tek seferde gerçekleştirilmektedir.

SSD Algoritmasında ilk olarak resim alınmakta ve bu resim Evrişimli Sinir Ağından geçirilmektedir. Bu şekilde farklı boyuttaki özellik haritaları çıkarılmaktadır. Bu özellik haritalarının tamamında 3\*3 Evrişimli filtre yardımı ile az miktarda sınırlayıcı dikkörtgenler elde edilmektedir.

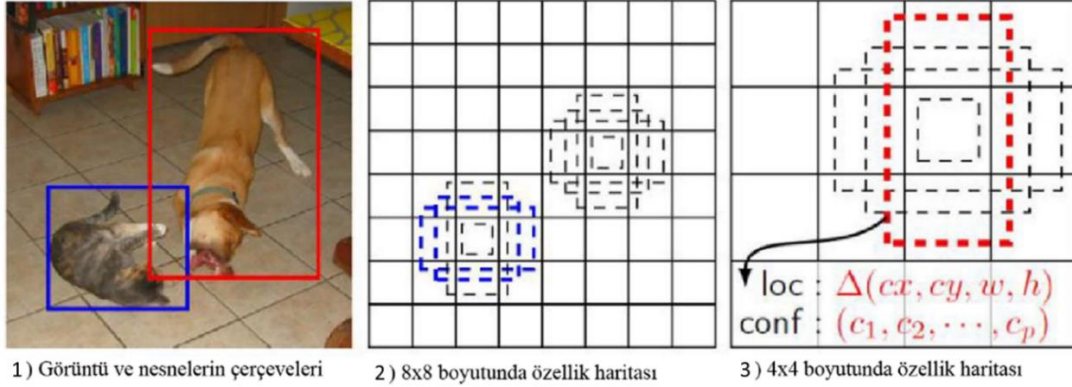
Her dikkörtgenler için aynı zamanda hem sınırlar hem sınıflandırmalar belirlenmektedir. Bu dikkörtgenler her aktivasyon haritasında olduğu için hem büyük hem küçük nesnelere tanımlanmaktadır.

Her bölge için farklı işlemler yapmak yerine bütün tahminleri tek seferde bir evrişimli sinir ağı içinde yapılmaktadır. Girdi resmi grid şeklinde ayrıldığı düşünülebilir.



Şekil 25. SSD (Single Shot Multibox Detector) Mimarisi (Liu, Anguelov, Erhan, Szegedy, Reed, Fu ve Berg 2016)

Cok kutulu (Multibox) Dedektörü incelendiğinde, özellik çıkarımı için gerekli adımlar tamamlandıktan sonra “ $m \times n \times p$ ” boyutunda bir özellik katmanı elde edilir. Bu katmanda “ $m \times n$ ” nesne bölgelerinin sayısı “ $p$ ” ise kanal sayısını ifade eder.



**Şekil 26. Nesnelerin çerçeveleri için pozisyon (loc) ve doğruluk (conf) değerleri (Liu, Anguelov, Erhan, Szegedy, Reed, Fu ve Berg, 2016)**

Her bölge için  $k$  sayıda çerçeve yer aldığı kabulü ile  $k$  sınırlayıcı kutular farklı en boy oranlarına sahiptir. Çeşitli nesnelere göre uygunluk gösteren çerçevelerde, belki dikey bir dikdörtgen insan için daha uygun olacakken yatay bir dikdörtgen araba ya da bir hayvan için daha uygun olması beklenmektedir. Bu çerçevelerin doğruluğu iki kayıp oranı ile doğrulanır. Bunlardan ilki ‘güven kaybı (confidence loss)’ ikincisi ise ‘konum kaybı (location loss)’ oranlarıdır. Güven kaybı, ağırlık hesaplanan sınırlama kutusunun içinde nesne olup olmadığına ne kadar emin olduğunu ölçer. Konum kaybı, ağırlık öngörülen sınırlayıcı kutuların test setindeki doğrulardan ne kadar uzakta olduğunu ölçmektedir.

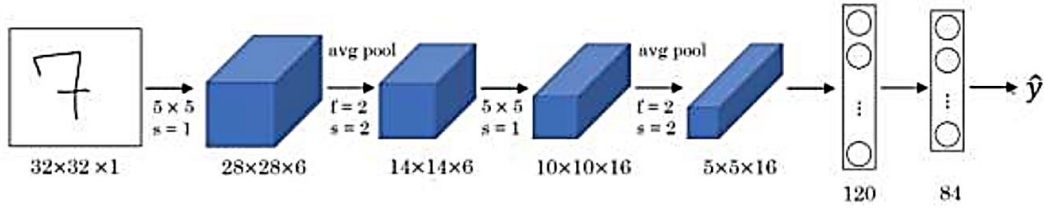
## 2.8 Yapay Sinir Ağı Modelleri

Yapay sinir ağı modelleri mimarinin belirlenmesinde kritik rol almaktadır. Aşağıda sıklıkla kullanılan yapay sinir ağı modellerinden olan LeNet, ALEXNet, VGG, ResNet GoogLeNet, MobileNet ve Inception modellerine yer verilmiştir.



### 2.8.1 LeNet-5

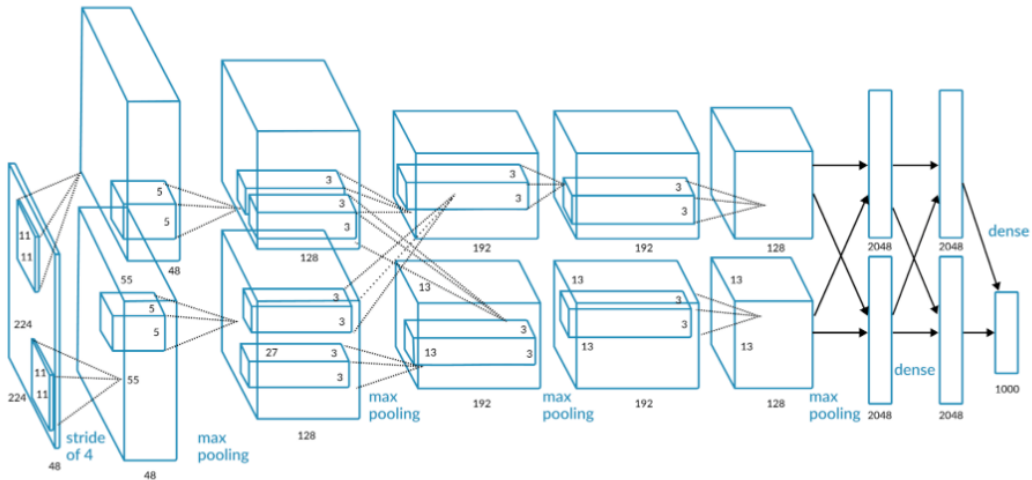
Le Cun ve ekininin 1998 yılında evrişimli sinir ağı modeli olarak ilk başarılı sonucu veren modeldir. Bu modeli diğerlerinden ayıran en önemli özelliklerden bir tanesi modelde maksimum havuzlama almak yerine, ortalama havuzlama işlemi kullanılmış olup, ilave olarak aktivasyon fonksiyonu sigmoid ve hiperbolik tanjant kullanılmıştır. LeNet modelinde 60 bin parametre hesaplanmaktadır (Lecun, Bottou, Bengio ve Haffner,1998).



Şekil 27. LeNet Mimarisi (Lecun, Bottou, Bengio ve Haffner,1998)

### 2.8.2 AlexNet

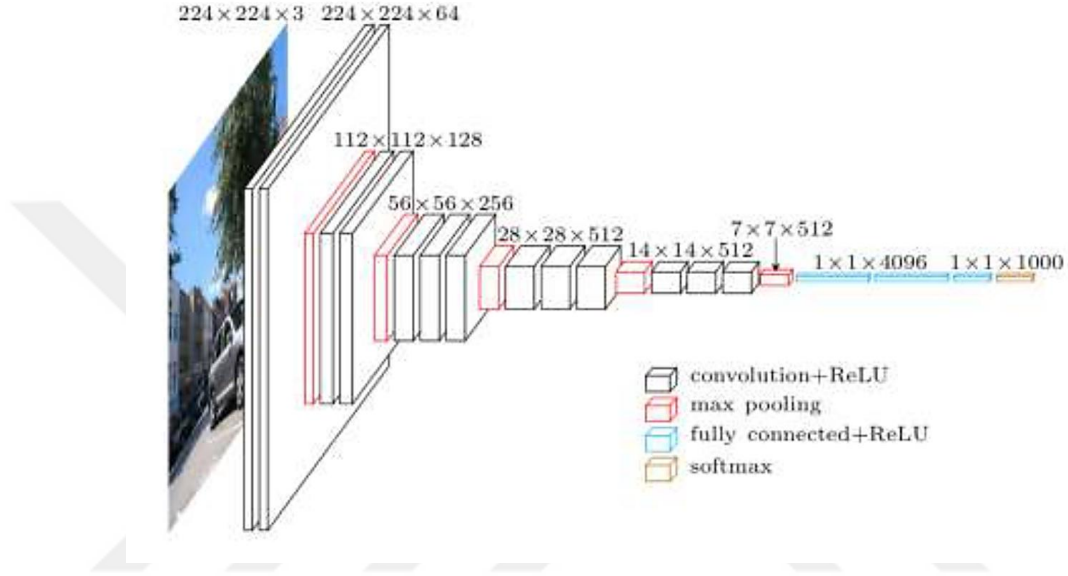
2012 yılındaki ImageNet yarışmasında derin öğrenme mimarisi ile tasarlanan AlexNet modeli galip gelmiştir. AlexNet modelinde aktivasyon fonksiyonu olarak ReLu, havuzlama katmanında da maksimum havuzlama kullanılmıştır. AlexNet 60 milyon parametre kullanmasıyla birlikte paralel çift işlemci üzerinde çalışan ilk modeldir (Krizhevsky, Sutskever ve Hinton, 2012).



Şekil 28. AlexNet Mimarisi (Krizhevsky, Sutskever ve Hinton, 2012)

### 2.8.3 VGG

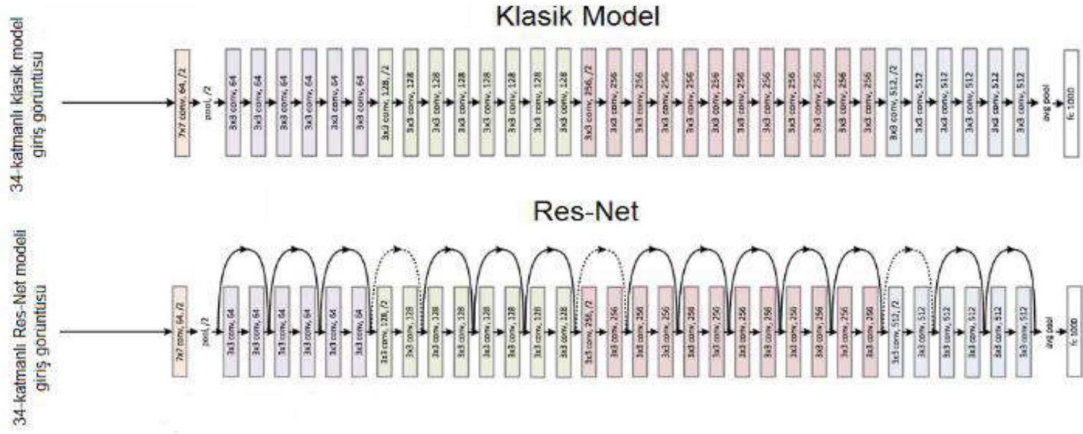
VGG basit bir model olarak nitelendirilmesine karşın önceki modellere göre başlıca farkı ikili ya da üçlü evrişim katmanlarının sonrasında havuzlama katmanlarının izlemesidir (Simonyan ve Zisserman, 2014). Önceki modellerde havuzlama ve evrişim katmanları birbirini takip etmekte olup, VGG modelinde 138 milyona yakın parametrenin hesabı yapılmaktadır.



Şekil 29. VGG Mimarisi (Heuritech,2016)

### 2.8.4 ResNet

Şimdiye kadar tasarlanan tüm mimarilerden daha derin olarak tasarlanan Resnet, 152 katmandan meydana gelmektedir. Diğer mimarilerden ayıran en büyük özelliği artık değerlerin sonraki katmanlara besleyen blokların modele eklenmesiyle oluşmasıdır. Doğrusal ve ReLu aktivasyon kodları arasında iki katmanda bir eklenen bu değer belirtildiği gibi sistem hesabını değiştirmektedir.



Şekil 30. Klasik Model ile Res-Net Modeli Karşılaştırması (He, Zhang, Ren ve Sun, 2015)

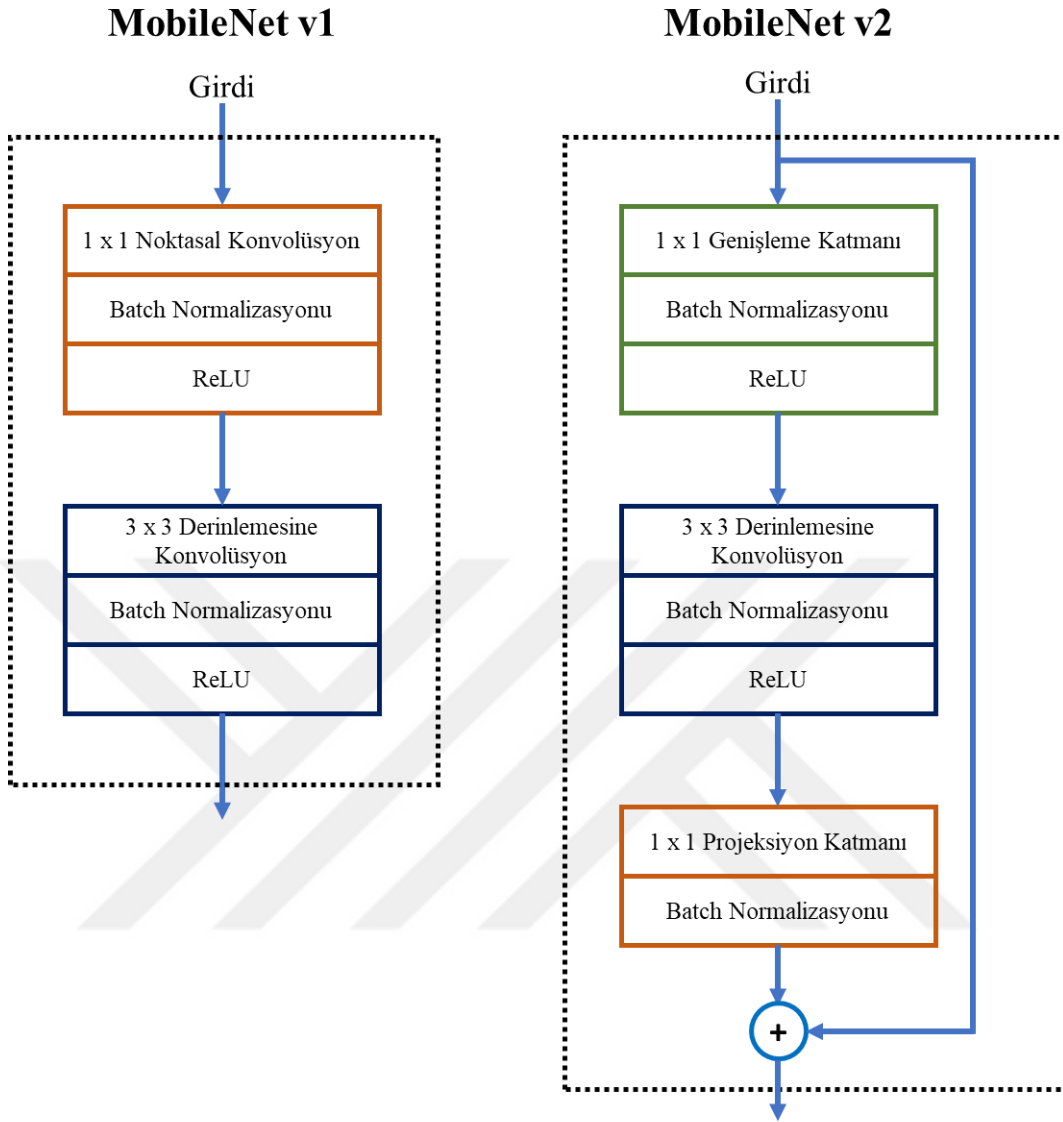
### 2.8.5 GoogLeNet

Katman sayısını arttırarak daha başarılı sonuç elde etmesi için tasarlanan bir ağ yapısıdır. Katman sayısının arttırılması yeterli olmayacağı için daha derine inilmiştir.

### 2.8.6 MobileNet

MobileNet incelendiğinde diğer modellerden boyutunun daha küçük ve daha hızlı olan bir derin konvolüsyonel sinir ağı modeli olduğu gözükmektedir. Derinlik açısından ayrılabilir konvolüsyonlarda her girdi kanalında tek bir filtre uygulandığı ve bazı konvolüsyonlarda 1 x 1 konvolüsyonların kullanıldığı görülmektedir. (Howard, Zhu, Chen, Kalenichenko, Wang, Weyand, Andreetto ve Adam 2017).

Derinlik açısından incelendiğinde ayrılabilir katmanların konvolüsyon katmanları ile benzeştiği, ancak filtreleme ve birleştirme işlemlerini iki katmana ayırarak yaptığı için konvolüsyonel katmanlara göre farklılaşmaktadır. Bunun dışında çok daha hızlı işlem yapmaktadır. Bütün bunların bir sonucu olarak hesaplama güç taleplerini azaltır ve modelin boyutu en azlanır. Çıktıyı sınıflandırmak için softmax katmanını besleyen tam bağlantılı katman (Fully Connected Layer) haricinde, her bir katman sonrasında bir batchnorm ve ReLU doğrusal olmayan aktivasyon katmanı gelir. Mobilenet, noktasal konvolüsyonlar haricinde 28 katmanı vardır (Howard, Zhu, Chen, Kalenichenko, Wang, Weyand, Andreetto ve Adam 2017).



**Şekil 31. MobileNet v1 ve MobileNet v2 Genel Mimarileri**

### 2.8.7 Inception

Inception Modeli incelendiğinde çok yüksek CPU'ya sahip bilgisayarlar ile ImageNet veri seti üzerinde önceden eğitilmiş milyonlarda parametreye sahip bir derin konvolüsyonel (Evrışimli) sinir ağıdır. Bu modelin en baştan eğitilmesi haftalarca sürebilir. Inception modelinin ilk sürümü olan Inception-v1 Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A. tarafından 2015 yılında GoogleLeNet adıyla tanıtılmıştır. 2016 yılına geldiğimizde söz konusu model batch normalizasyon tekniği ve ilave yöntemler ile

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. ve Wojna, Z. tarafından geliştirilerek Inception-v2 ortaya çıkmıştır.

Hesaplama hız artışı için çarpanlara ayırma tekniğine dayanarak 7 x7 konvolüsyonlar üç tane 3 x 3 konvolüsyona bölünmüştür. Ağın inception (başlangıç) bölümünde 3 geleneksel inception modülü vardır. Grid azaltma tekniğinin kullanımı ile 288 filtreli 35 x 35 grid, her biri 768 filtreli 17 x 17 grid'e indirgenmiştir. Grid azaltma tekniği kullanılarak 8 x8 x1280 grid'e düşürülmüştür. En merkezi 8 x 8 seviyesinde iki inception modülü vardır ve her desen için 2048 boyutunda bir çıktı filtre bankası ile bir araya getirilmiştir (Szegedy, Vanhoucke, Ioffe, Shlens ve Wojna, 2016).

## **2.9 Görüntü Sınıflandırma Eğitimi ve Kullanılan Veri Setleri**

Görüntü sınıflandırma eğitimlerinin gerçekleştirilebilmesi, yapay sinir ağlarının özellik tespiti ve çıkarım yapabilmesi için mutlaka veri setlerine ihtiyaç duyulmaktadır. Aşağıda derin öğrenme işlemlerinde kullanılan başlıca veri setlerinden olan, ImageNet, MS COCO (Microsoft Common Objects in Context), MNIST (Modified National Institute of Standards and Technology) ve CIFAR-10 / CIFAR-100 veri setlerine yer verilmiştir.

### **2.9.1 ImageNet Veri Seti**

Her yıl yapılan “ImageNet Large Scale Visual Recognition Challenge” yarışmasında kullanılan veri setidir. Bu yarışmanın temel amacı yarışmacıların resmi sınıflandırmada, tek nesnenin yerini belirlemede ve nesne tanımada farklı veri setleri kullanarak başarı sağlamaktır. Görüntü içindeki nesne sayısı birden fazla olabileceği gibi sıklığı da farklı olabilmektedir.

Bu veri setinde:

- 1000 sınıf,
- Yaklaşık 1.3 milyon eğitim verisi,

- 50000 doğrulama verisi,
- 100.000 test verisinden oluşmaktadır.



**Şekil 32. ImageNet Veri Setinden Örnekler (Massouh, Babiloni, Tommasi, Young, Hawes ve Caputo, 2017)**

PASCAL veri seti ile kıyaslanırsa ImageNet'in çok iyi bir sınıflandırmış veri içerdiği gözükmektedir. Örneğin köpek sınıfına her birinin türleri ve renkleri gibi ayrı sınıflarda eklenmiştir.

### 2.9.2 MS COCO Veri Seti

MS-COCO - Microsoft Common Objects in Context veri seti 91 sınıftan oluşmaktadır.

Bunlardan;

- 85 tanesi 5.000'den fazla etiketlenmiş veri
- 328.000 resim
- 2 Milyondan fazla etiketlenmiş nesne bulunmaktadır.

ImageNet data setine göre daha az sınıf içerse de her bir sınıfta daha fazla görüntü yer almaktadır.



**Şekil 33. MS COCO Veri Setinden Örnek Görüntüler (Lin, Maire, Belongie, Bourdev, Girshick, Hays, Perona, Ramanan, Zitnick ve Dollar, 2014 )**

### 2.9.3 MNIST Veri Seti

Görüntü işleme ve makine öğrenmesi ile ilgili araştırmacıların ilk kullandığı veri setlerinin başında MNIST (Modified National Institute of Standards and Technology database) veri seti gelmektedir. MNIST veri seti lise öğrencileri ve US Sayım Bürosu çalışanları tarafından el ile yazılmış rakamlardan oluşmaktadır (Lecun, Bottou, Bengio ve Haffner, 1998).

MNIST Veri setinde;

- 60000 adet eğitim verisi

- 10000 adet test verisi
- Boyutu 28x28 boyutlarında ve siyah arka plan üzerinde beyaz renklerle yazılmış rakamların resimleri yer almaktadır.



Şekil 34. MNIST Veri Setinden Örnekler

#### 2.9.4 CIFAR-10 / CIFAR-100

CIFAR-10 veri seti;

- 32 x 32 boyutlarında
- 60.000 adet görüntü (50.000 adedi eğitim, 10.000 adedi test)
- 10 farklı sınıftan oluşmaktadır (Krizhevsky, 2009).

CIFAR-100 veri seti;

- 32 x 32 boyutlarında
- 100 sınıf
- Her sınıfta 600 Adet Resim
- Sınıfların derlendiği 20 farklı üst sınıftan oluşmaktadır (Krizhevsky, 2009).





Şekil 35. CIFAR-10 Veri Setinden Örnekler (ImageNet, 2016)

## 2.10 Sıfırdan Öğrenim ve Aktarımlı Öğrenme

Derin Öğrenmede bir eğitim yapılacak olduğunda iki temel yol ile karşılaşılmaktadır. Bunlar modelin sıfırdan mı eğitileceği yoksa aktarımlı öğrenmeyi yapılıcağıdır. Aşağıda her ikisi de detaylı olarak incelenmiştir.

### 2.10.1 Sıfırdan Eğitim (Training From Scratch)

Derin bir ağı sıfırdan eğitmek meşakkatli olup, çok büyük etiketli veri seti toplamak ve özellikleri ve modeli öğrenecek bir ağ mimarisi tasarlanmak temeline oluşturmaktadır. Bu çok yaygın bir uygulama olmamakla birlikte, büyük miktarda veri, güçlü işlemcili bilgisayarlar ve öğrenme oranıyla, bu ağlarda tipik olarak eğitimi gerçekleştirmek haftalarca sürebilmektedir (MathWorks, 2018b).

### 2.10.2 Aktarımlı Öğrenme (Transfer Learning)

Çoğu derin öğrenme uygulaması, önceden eğitilmiş bir modelin ince ayarını içeren bir süreç olan aktarımlı öğrenme yaklaşımını kullanmaktadır. AlexNet veya GoogleLeNet gibi mevcut bir ağ ile başlamakta ve önceden bilinmeyen sınıfları içeren yeni verilerle beslenmesi gerekmektedir. Ağ ile ilgili ayarlamalar gerçekleştirildikten sonra 1000 farklı nesne yerine sadece birkaç nesnenin kategorize edilmesi ile yeni bir görev gerçekleştirilebilir. Bu sayede çok az veri ile eğitim gerçekleştirilebilir. Eğitim haftalardan, saatlere düşmektedir. (MathWorks, 2018b).

Aktarımlı öğrenme daha önceden var olan ağın içyapısında yeni bir ara yüz gereksinimi doğurur. Bu sebeple yeni işlem için parametre değişikliği ve parametre geliştirilmesi olmaktadır. (MathWorks, 2018b).

## ÜÇÜNCÜ BÖLÜM

### UYGULAMA ÇALIŞMASI

#### 3.1 Çalışmada Kullanılan Programlama Dili ve Kütüphaneler

2006 yılı ile 2012 yılları arasında PascalVOC tekniği nesne tanıma %40'a yaklaşırken, Konvolüsyonel (Evrışimli) Sinir Ağlarının kullanılmaya başlanması ile 2013 – 2015 yılları arasında bu oran %80'lere yaklaşmıştır. Günümüz algoritmaları olan R-CNN, Fast R-CNN, Faster R-CNN ve SSD (Single Shot Multibox) algoritmaları ile 2016 yılından sonra bu oran %90'ların da üzerine çıkmaktadır.

Söz konusu çalışma gerçekleştirilirken, derin öğrenme konusunda daha öncesinde birçok araştırma ve çalışmaların yer alması sebebiyle Python programlama dili tercih edilmiştir. Görüntü işlemede sağladığı kolaylıklar ve ara yüzler sebebiyle IDLE olarak Pycharm tercih edilmiştir.

Aşağıda çalışmada kullanılan kütüphaneler ile ilgili kısa bilgilere yer verilmiştir.

Keras: Keras Python'da derin öğrenme hesaplamaları için bir oluşturulmuş bir kütüphanedir. Keras Python programlama dili ile yazılmıştır. Keras aynı zamanda Tensorflow gibi yapılar ile entegre çalışabilen bir sinir ağı API'sidir. Geliştirilmesindeki temel amaç sonuçların daha hızlı elde edilmesidir. (Gulli ve Pal, 2017)

Numpy: N boyutuna sahip dizi ve matrisleri kullanmak ve üzerine hesaplamalar yapılmasını imkân tanıyan bir kütüphanedir. Bu kütüphane matematiksel işlemler için geliştirilmiştir (Salomon, 2018).

Matplotlib: 2 boyutlu çizim ve grafikler için kullanılan kütüphanedir. Büyük dizilerde iyi sonuç alabilmek adına Numpy ile birlikte kullanılmalıdır. Basit komut yapısı ile grafiklerin oluşturulmasına yardımcı olmaktadır (Hunter, Dale, Firing ve Droettboom, 2017)

OpenCV: Görü işlemede kullanılan, C++ dili ile geliştirilmiş ve içerisinde birçok bilgisayarlı görü algoritmasına sahip olan açık kaynak kodlu bir kütüphanedir. Matematiksel sınıflandırma, regresyon, veri kümeleme gibi makine öğrenmesi fonksiyonlarını kullanmaktadır (Doxygen, 2015).

Pandas: Veri çerçevelemesinde kullanılan Pandas kütüphanesi, aynı zamanda veriyi yeniden şekillendirme, dilimleme, birleştirme gibi çeşitli işlemler için kullanılır.

Tensorflow: Google'ın sunmuş olduğu açık kaynaklı bir derin öğrenme kütüphanesidir. Tensorflow bilgisayarın ekran kartının uyumluluğuna göre CUDA aracılığı ile GPU üzerinden ya da bilgisayarın doğrudan işlemcisi üzerinden (CPU) performans göstermesi mümkündür (Salomon, 2018).

### **3.2 Eğitim, Test, Değerlendirme (Validation) Veri Setlerinin Oluşturulması**

Avantajlı olmaları sebebiyle SSD- Single Shot Multibox MobileNet\_v2 ve Faster R-CNN Inception\_v2 mimarileri seçilmiştir.

Söz konusu mimarilerde eğitime başlamadan önce veri setlerinin hazırlanması önemlidir. Çalışmanın temel amacında uydu ve orta ve yüksek irtifalı insansız hava araçlarından, havaalanı pistleri üzerinden havaalanlarının tespiti amaçlandığından veri setleri uydu ve İHA (İnsansız Hava Aracı)'ların çektiği görüntülerden oluşturulmuştur.

İlk olarak içerisinde bir ve birden fazla havaalanının yer aldığı 310 adet görüntü Yandex Maps, Google Earth ve internet üzerinden alınmıştır. Hepsinde yer alan havaalanı görüntüleri bir ya da birden fazla olmak üzere ilave bir arayüz programı kullanılarak etiketlenmiştir.

Veriler derlenirken Dünya'nın dört bir yanından farklı mevsim özelliklerine sahip bölgeleri seçilmiştir. Bunun dışında özellikle modelin zorlanması için ormanlık bölge, deniz ve okyanus kenarı, ada yüzeyi, çöl ve şehir merkezi gibi birçok farklı arazi koşulundan görüntüler derlenmiştir. Özellikle şehir merkezi ve ana yolların yakınında

yer alan havaalanları, anayolların havaalanlarına benzemesi sebebiyle, modelin daha da zorlanması için tercih edilmiştir.

Yine modelin zorlanması için alınmış irtifa yüksekliği 2 km- 14 km'ye kadar olan görüntüler tercih edilmiş. Bu sayede tespit edilmesi istenen bölgenin ölçeğinin değiştirilmesi sağlanmıştır.

İçinde renkli ve renksiz görüntülerin de yer aldığı 310 adet görüntü %80'i eğitim veri seti, %20'si test kümesi olacak şekilde ikiye ayrılmıştır. Buna göre eğitim veri setinde 248 adet görüntüye yer verilmiş, test veri setinde ise 62 adet görüntüye yer verilmiştir. Söz konusu görüntülerde özellikle birden fazla havaalanı içeren görüntülerde konulmuştur. Her iki veri setinde de görüntülerde havaalanları işaretlenmiş ve üzerinde eğitim gerçekleştirilebilmesi için koordinatlar XML formatına dönüştürülmüştür. XML dosyaları şeklindeki bu kümeler Tensorflow kütüphanesinin eğitilebilmesi için TFRecord dosyalarına dönüştürülmüş, eğitim için önemli parametre değişiklikleri yapılmıştır.

Eğitim ve test veri setlerinin haricinde sonuç ve performansların objektif bir şekilde değerlendirilmesi için eğitim ve test veri setlerinden tamamıyla bağımsız ve farklı olan değerlendirme veri seti oluşturulmuştur. Söz konusu veri seti yine farklı arazi koşullarından, farklı coğrafi iklim tipine ait alanlarda ve farklı irtifalardan tercih edilmiştir. Değerlendirme veri setine 50 adet bir ve birden fazla havaalanı pisti içeren görüntüye ve 50 adet havaalanı içermeyen görüntüye yer verilmiştir. Aşağıda eğitim verisinde kullanılan birkaç havaalanı görüntüsüne yer verilmiştir.



**Şekil 36. Eğitim Setinde Yer Alan Havaalanı İçeren Görüntü Örnekleri**



**Şekil 37. Test Setinde Yer Alan Havaalanı İçeren Görüntü Örnekleri**



**Şekil 38. Değerlendirme Setinde Yer Alan Havaalanı İçeren Görüntü Örnekleri**



**Şekil 39. Değerlendirme Setinde Yer Alan Havaalanı İçermeyen Görüntü Örnekleri**

### 3.3 Sonuç ve Performans Değerlendirme Kriterleri

SSD- Single Shot Multibox ve Faster R-CNN mimarilerinin sonuçlarını ve performanslarını değerlendirmek amacıyla aşağıdaki tabloda yer verilmiş kriterler, değerlendirme veri seti ölçümlerine göre hesaplanmıştır. Bunun dışında hem eğitim hem de sonuç sunma süreleri her iki algoritma için de hesaplanıp kıyaslanmıştır. Söz konusu hesaplamalar yapılırken her iki mimari içinde Intel Core i53230M 2.60 Ghz işlemcili aynı iş istasyonu kullanılarak sonuçlar elde edilmiştir.

Genel olarak bu algoritmaların performans değerlendirilmesi için sınıfların gerçek ve tahmin değerleri karmaşıklık matrisi ile karşılaştırılmaktadır. (Polat, Mehr ve Cetin, 2017). Alıcı İşletim Karakteristiği (ROC- Receiver Operating Characteristics) sınıflandırma performansını ölçmek adına kullanılan yöntemlerden biridir (Lasko, Bhagwat, Zou, ve Ohno-Machado, 2005).

Yapılan çalışmada ilgili algoritmaların performansının kıyaslanması için değerlendirme veri seti ile üretilen sonuçların olası dört performans sonucuna aşağıdaki tabloda yer verilmiştir.

**Tablo 3. Karmaşıklık Matrisi**

KARMAŞIKLIK MATRİSİ		GERÇEK	
		POZİTİF	NEGATİF
TAHMİN	POZİTİF	TP Doğru Pozitif	FP Yanlış Pozitif
	NEGATİF	FN Yanlış Negatif	TN Doğru Negatif

TP: Pozitif Örneğin Doğru Sınıflandırılması İşlemi olarak tanımlanır.

TN: Negatif Örneğin Doğru Sınıflandırılması İşlemi olarak tanımlanır.

FP: Negatif Örneğin Yanlış Sınıflandırılması İşlemi olarak tanımlanır.



FN: Pozitif Örneğin Yanlış Sınıflandırılması İşlemi olarak tanımlanır.

Doğruluk Oranı: veri setinde her bir örneğin sınıf etiketlerinin sınıflandırılmasında algoritmaların performansının değerlendirilmesinde doğruluğu ile hesaplanır (Polat, Mehr ve Cetin, 2017).

Doğru pozitif oranı: sınıflayıcıya göre gerçek pozitif olanların ne kadarının pozitif sınıf olarak etiketlendiğini göstermektedir (Polat, Mehr ve Cetin, 2017).

Yanlış pozitif oranı: sınıflayıcıya göre gerçek pozitif olanların ne kadarının negatif sınıf olarak etiketlendiğini göstermektedir (Polat, Mehr ve Cetin, 2017).

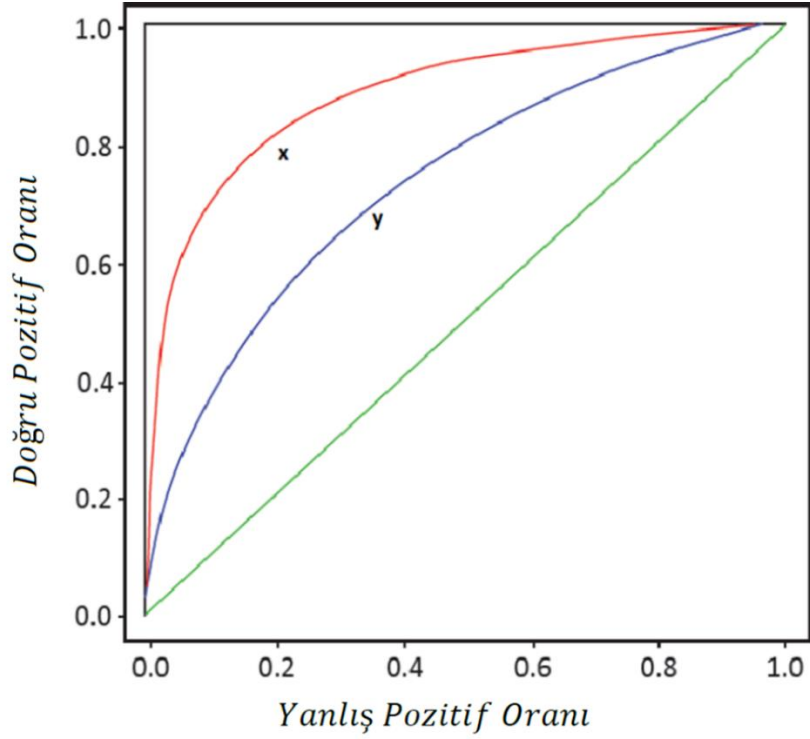
Doğru negatif oranı: sınıflayıcıya göre gerçek negatif olanların ne kadar negatif sınıf olarak etiketlendiğini göstermektedir (Polat, Mehr ve Cetin, 2017).

Yanlış negatif oranı: sınıflayıcıya göre gerçek negatif olanların ne kadarının pozitif sınıf olarak etiketlendiğini göstermektedir.

Doğru pozitif oranı ve yanlış negatif oranının toplamı 1'e eşittir. Ayrıca doğru negatif oranı ile yanlış pozitif oranının toplamı da 1'e eşittir.

Doğru tahmin oranı: pozitif olarak etiketlenen sınıfların ne kadarının gerçekte pozitif olduğunu göstermektedir.

Bir diğer değerlendirme kriterlerinden olan ROC eğrisi, farklı eşik değerleri için doğru ve yanlış pozitiflik oranlarının, yatay ve dik eksenlerde yer aldığı eğridir (Metz, 2006). Sol üst köşeye en yakın olan ROC eğrisini veren test en kullanışlı olarak bilinmektedir (Dirican, 2001). Diagonal eğrinin üstünde kalan ve sol üst köşeye yaklaşan ROC eğrisinde x ve y sınıflandırıcılarından, sınıflandırma performansı başarısı olarak x daha başarılı kabul edilmektedir.



Şekil 40. ROC Eğrisi (Lane ve Gantley, 2017)

Değerlendirme kriterleri derlendiğinde aşağıdaki tablo ortaya çıkmıştır.

Tablo 4. Derlenmiş Değerlendirme Kriterleri Tablosu

KRİTERLER	KISALTMALAR	FORMÜLLER
DOĞRULUK ORANI (ACCURACY)	<i>ACC</i>	$(TP+TN)/(N+P)$
DUYARLILIK (DOĞRU POZİTİF ORANI) (SENSITIVITY)	<i>TPR</i>	$TP/(TP+FN)$
ÖZGÜLLÜK (DOĞRU NEGATİF ORANI) (SPECIFICITY)	<i>TNR</i>	$TN/(TN+FP)$
YANLIŞ POZİTİF ORANI (FALSE POSITIVE RATE)	<i>FPR</i>	$FP/(FP+TN)$
YANLIŞ NEGATİF ORANI (FALSE NEGATIVE RATE)	<i>FNR</i>	$FN/(FN+TP)$
DOĞRU TAHMİN ORANI (POSITIVE PRED VALUE -PRECISION)	<i>PPV</i>	$TP/(TP+FP)$
F PUANI (F SCORE)	<i>F</i>	$2*((PPV*TPR)/(PPV+TPR))$
HATA ORANI (ERROR RATE)	<i>ERR</i>	$(FP+FN)/(N+P)$

Bunun dışında modellerin eğitim süreleri ve sonuç üretme süreleri de önemli olup, ortalama, standart sapma, minimum ve maksimum değerleri hesaplanarak analizler gerçekleştirilmiştir.

**Kayıp Oranı:** Kayıp oranı her eğitim set grubundaki kayıpların ortalamasıdır. Derin öğrenme modeli zaman içinde öğrenmesi nedeniyle bir adımın ilk zaman üzerindeki kaybı genellikle son zamansa göre daha yüksektir. Kayıp oranı oluşturulmuş olan modelin eğitimi aşamasındaki her yinelemesinden sonra güçlü veya zayıf davrandığını ortaya koymaktadır. Yinelemelerin ardından kayıp oranının azalması hedeflenmektedir (Chen, 2017).

Kayıp (Loss) fonksiyonu temelde modelin yaptığı tahminin, gerçek değerden (ground truth) ne kadar farklı olduğunu hesaplamaktadır. Bu nedenle iyi tahmin eden bir model oluşturmamışsak, gerçek değer (ground truth) ile tahmin edilen değer arasındaki fark yüksek olacak dolayısıyla loss değeri yüksek olacak, iyi modele sahipsek loss değeri az olacaktır. Birebir aynı olduğu durumda loss 0 olacaktır.

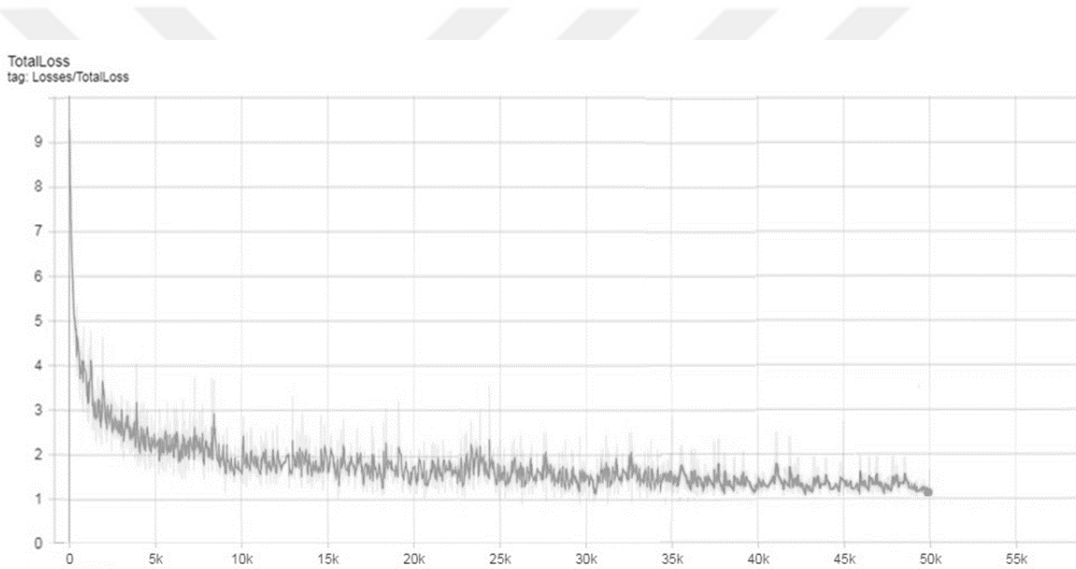
Loss değeri hesaplayan birden çok lineer loss fonksiyon türü mevcuttur. Bu loss fonksiyonları yapay sinir ağının ürettiği skorları normalize ederek loss değeri hesaplamaktadır. Bunlardan en sık kullanılanları Sigmoid, Multiclass Support Vector Machine (SVM), Softmax vb.dir.

### **3.4 SSD- Single Shot Multibox Algoritması İle Gerçekleştirilen Eğitim ve Değerlendirme Sonuçları**

SSD-Single Shot Multibox Algoritması kullanılırken daha öncesinde 91 sınıftan oluşan MS- COCO Microsoft Common Objects in Context veri seti ile eğitilmiş ve MobileNet v2 yapay sinir ağı modeline sahip versiyonu ele alınmıştır (Github). Söz konusu algoritma sonradan 310 adet uydu ve orta ve yüksek irtifalı insansız hava araçlarınca çekilip sonrasında işaretlenmiş havaalanı görüntülerinden oluşan 248 eğitim veri seti ve 62 test veri seti ile aktarımlı öğrenme ile havaalanlarının tespiti için eğitime başlanmıştır.

SSD – Single Shot Multibox MobileNet\_v2 mimarisi ile 50.011 itaresyon boyunca eğitilmiştir. Eğitim Intel Core i53230M 2.60 Ghz İşlemcili iş istasyonu ile toplam 763 saat 53 dakika sonunda tamamlanmıştır.

Yapılan eğitim sonucunda ortaya çıkan SSD Algoritması Total Loss (Toplam Kayıp) Grafiği aşağıdaki şekildedir. Kayıp oranı her eğitim set grubundaki kayıpların ortalaması olduğu bilindiğinde göre, ilk adımlarda kayıp oranları çok yüksek olan modelin zayıf davrandığı gözükmektedir. Söz konusu model 20.000 iterasyonun ardından güçlenmiş kayıp oranları azalmış kayıpların ortalaması 2'nin altına inmiştir. 50.000 iterasyona ulaştığında model güçlenmesine ve kayıp oranının azalmasına karşın 1'in altına yeni inemediği gözükmektedir.



**Şekil 41. SSD Algoritması Eğitimi Total Loss (Toplam Kayıp) Grafiği**

SSD – Single Shot Multibox MobileNet\_v2 mimarisi ile 50.011 adımda gerçekleştirilen eğitim sonrasında değerlendirme veri setinde yer alan içerisinde havaalanı içeren 50 görüntü ve içermeyen 50 görüntü ile test edilmiştir.

Değerlendirme kriterlerine göre sonuçlar aşağıdaki gibidir.

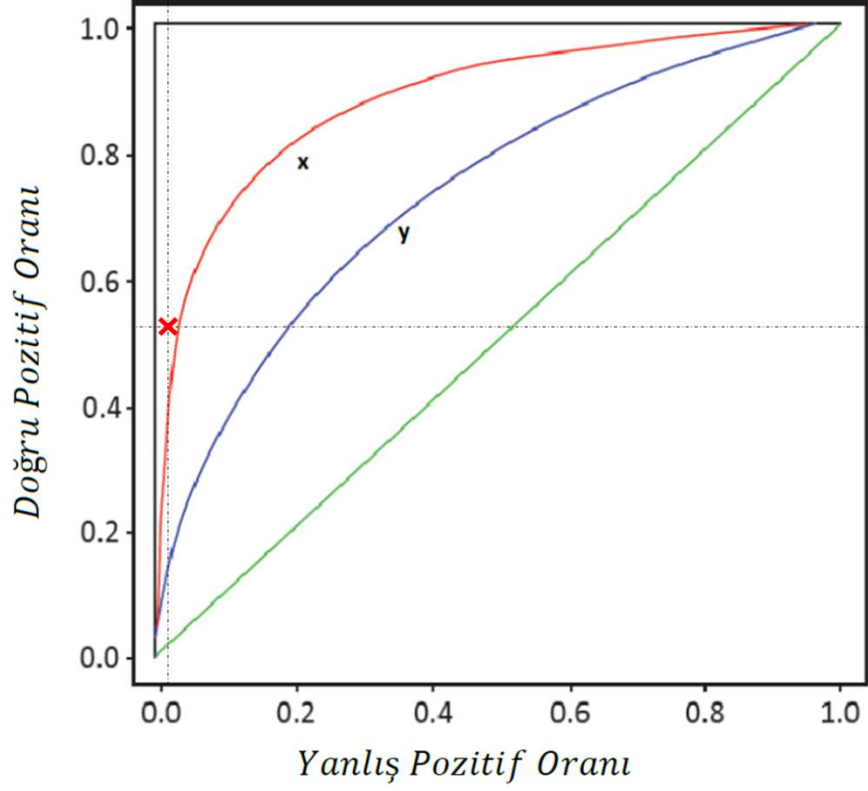
**Tablo 5.Değerlendirme Kriterlerine Göre SSD Algoritmasının Sonuçları**

KRİTERLER	KISALTMALAR	FORMÜLLER	SSD - Değerlendirme Veri Setine Göre Sonuçlar
DOĞRULUK ORANI (ACCURACY)	<i>ACC</i>	$(TP+TN)/(N+P)$	76,61%
DUYARLILIK (DOĞRU POZİTİF ORANI) (SENSITİVİTİY)	<i>TPR</i>	$TP/(TP+FN)$	55,07%
ÖZGÜLLÜK (DOĞRU NEGATİF ORANI) (SPECİFİTY)	<i>TNR</i>	$TN/(TN+FP)$	98,14%
YANLIŞ POZİTİF ORANI (FALSE POSİTİVE RATE)	<i>FPR</i>	$FP/(FP+TN)$	1,86%
YANLIŞ NEGATİF ORANI (FALSE NEGATİVE RATE)	<i>FNR</i>	$FN/(FN+TP)$	44,93%
DOĞRU TAHMİN ORANI (POSİTİVE PRED VALUE -PRESİCİON)	<i>PPV</i>	$TP/(TP+FP)$	96,73%
F PUANI (F SCORE)	<i>F</i>	$2*((PPV*TPR)/(PPV+TPR))$	70,18%
HATA ORANI (ERROR RATE)	<i>ERR</i>	$(FP+FN)/(N+P)$	23,40%

Değerlendirme kriterlerinin en başında gelen doğruluk oranı %76,61 olarak gerçekleşmiş SSD algoritması ile bu oranda havaalanlarının tespiti yapılabilmektedir.

Bunun dışında değerlendirme veri setinde bulunan görüntülerin her birinin analiz edilerek sonuç üretilme süreleri değerlendirildiğinde 100 değerlendirme veri seti görüntülerinin her birinin ortalama 4,06 saniyede sonuç üretildiği ölçülmüştür. Aynı kümenin standart sapması 0,61 saniye minimum sonuç üretim süresi 2,8 saniye, maksimum sonuç üretim süresi 5,5 saniyedir.

İlave olarak ROC eğrisine göre doğru pozitif oranı ve yanlış pozitif oranı yerleştirilirse aşağıdaki sonuç elde edilmektedir.

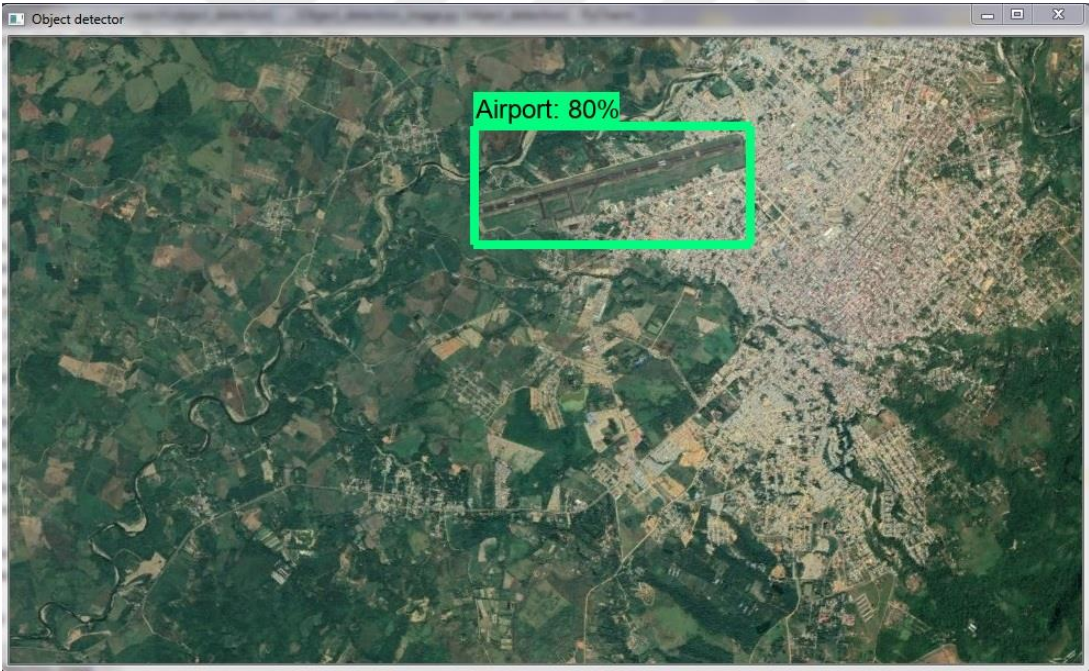


**Şekil 42. SDD Algoritmasının Değerlendirme Sonuçlarının ROC Eğrisinde gösterimi**

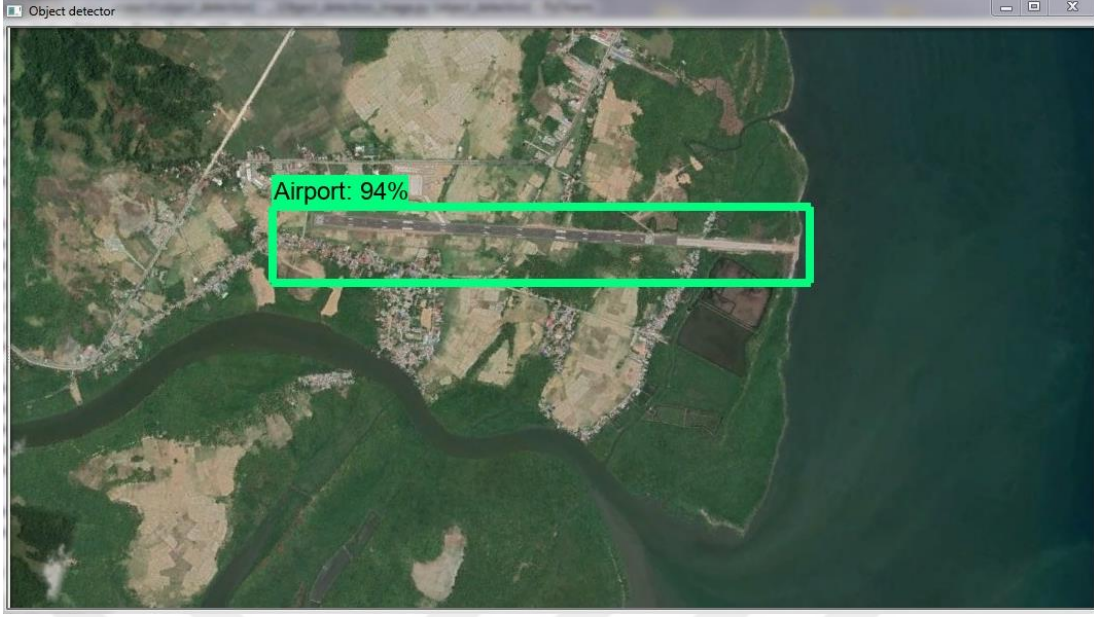
SSD – Single Shot Multibox MobileNet\_v2 mimarisi ile değerlendirme veri setindeki gerçekleştirilen, havaalanlarının tespit sonuçları aşağıdaki şekillerde yer verilmiştir.



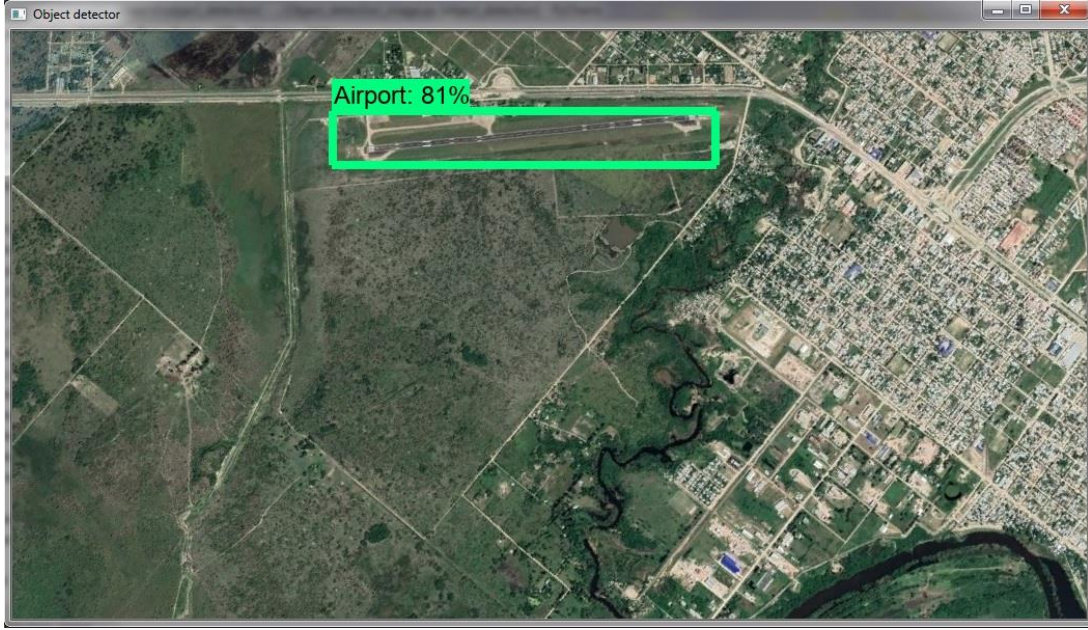
**Şekil 43. SDD Algoritması Görüntü Analiz Sonucu-1**



**Şekil 44. SDD Algoritması Görüntü Analiz Sonucu-2**

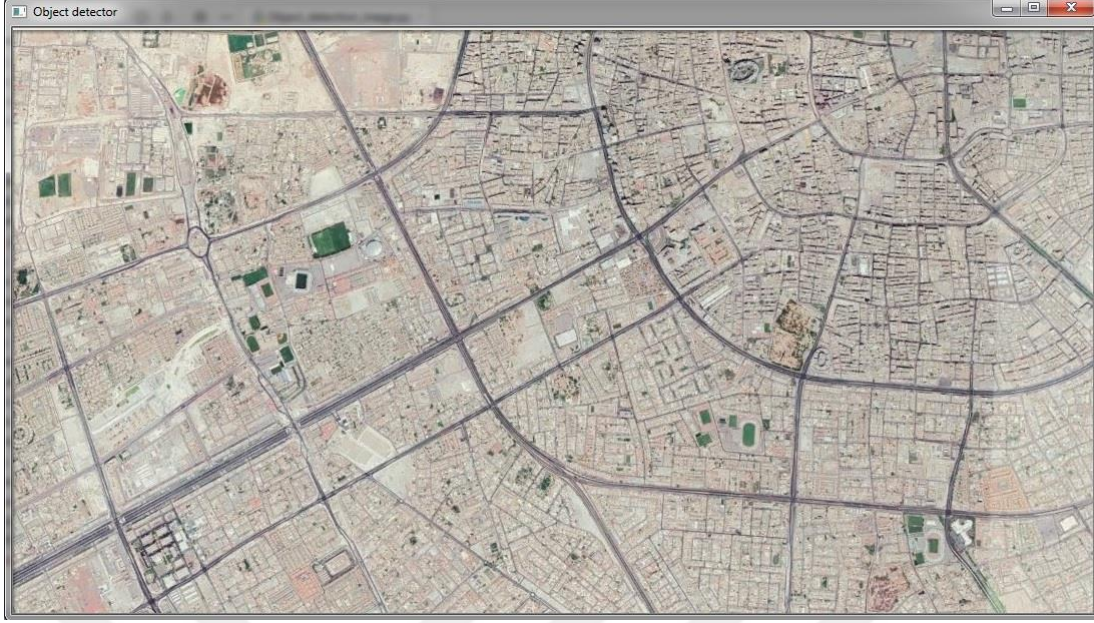


**Şekil 45. SDD Algoritması Görüntü Analiz Sonucu-3**



**Şekil 46. SDD Algoritması Görüntü Analiz Sonucu-4**





**Şekil 47. SDD Algoritması Havaalanı İçermeyen Görüntü Analiz Sonucu**

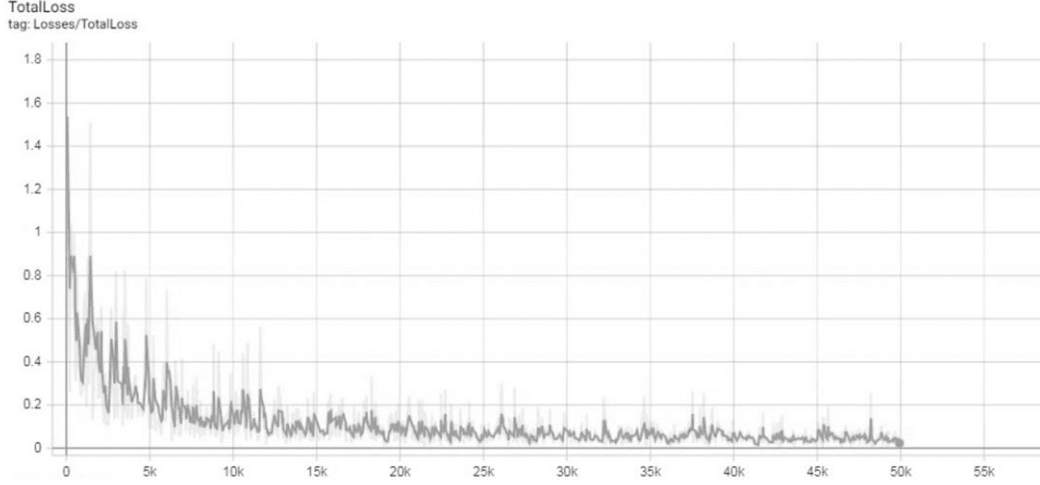
### **3.5 Faster R-CNN Algoritması İle Gerçekleştirilen Eğitim ve Değerlendirme Sonuçları**

Faster R-CNN Algoritması daha öncesinde 91 sınıftan oluşan MS- COCO Microsoft Common Objects in Context veri seti ile eğitilmiş ve Inception\_v2 yapay sinir ağı modeline sahip versiyonu yeniden eğitilmek üzere ele alınmıştır (Github). Algoritma sonradan 310 adet uydu ve orta ve yüksek irtifalı insansız hava araçlarınınca çekilip sonrasında işaretlenmiş havaalanı görüntülerinden oluşan, 248 eğitim veri seti ve 62 test veri seti aktarımlı öğrenme ile havaalanlarının tespiti için eğitime başlanmıştır.

Söz konusu Faster R-CNN Inception\_v2 mimarisi ile 50.003 itaresyon boyunca eğitilmiştir. Eğitim Intel Core i53230M 2.60 Ghz İşlemcili iş istasyonu ile toplam 165 saat 37 dakika sonunda tamamlanmıştır.

Yapılan eğitim sonucunda ortaya çıkan Faster R-CNN Algoritması Total Loss (Toplam Kayıp) Grafiği aşağıdaki şekildedir. Kayıp oranı grafiklerinde her eğitim set grubundaki kayıpların ortalamasını gösterdiğini belirtmiştik. Faster R-CNN ile ortaya çıkan Total Loss grafiğinde ilk adımlarda kayıp oranları çok yüksek olmadığı 1,4 arasında başladığı buna karşın 5.000 iterasyona varılmadan 0,5 değerinin altına

inilerek modelin çok güçlü davrandığı gözükmemektedir. 50.000'inci iterasyona yaklaşıldığında ise model çok güçlü davranmış kayıp oranları 0,05'in bile çok altına inmiştir. Bu durum modelin eğitim ve test setleri arasında doğruya çok yakın tahminleme yapabildiğini göstermiştir.



**Şekil 48. Faster R-CNN Algoritması Eğitimi Total Loss (Toplam Kayıp) Grafiği**

Faster R-CNN Inception\_v2 mimarisi ile 50.003 adımda gerçekleştirilen eğitim değerlendirme veri setinde yer alan içerisinde havaalanı içeren 50 görüntü ve içermeyen 50 görüntü ile test edilmiştir.

Değerlendirme kriterlerine göre sonuçlar aşağıdaki gibidir.

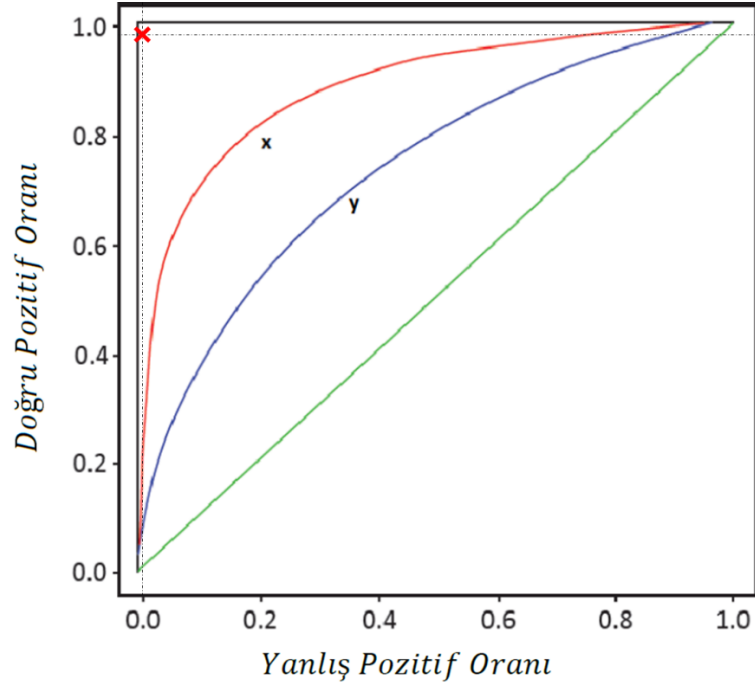
**Tablo 6. Değerlendirme Kriterlerine Göre Faster R-CNN Algoritmasının Sonuçları**

KRİTERLER	KISALTMALAR	FORMÜLLER	Faster R-CNN Değerlendirme Veri Setine Göre Sonuçlar
DOĞRULUK ORANI (ACCURACY)	<i>ACC</i>	$(TP+TN)/(N+P)$	99,52%
DUYARLILIK (DOĞRU POZİTİF ORANI) (SENSITİVİTİY)	<i>TPR</i>	$TP/(TP+FN)$	99,04%
ÖZGÜLLÜK (DOĞRU NEGATİF ORANI) (SPECİFİTY)	<i>TNR</i>	$TN/(TN+FP)$	100,00%
YANLIŞ POZİTİF ORANI (FALSE POSİTİVE RATE)	<i>FPR</i>	$FP/(FP+TN)$	0,00%
YANLIŞ NEGATİF ORANI (FALSE NEGATİVE RATE)	<i>FNR</i>	$FN/(FN+TP)$	0,96%
DOĞRU TAHMİN ORANI (POSİTİVE PRED VALUE -PRESİCİON)	<i>PPV</i>	$TP/(TP+FP)$	100,00%
F PUANI (F SCORE)	<i>F</i>	$2*((PPV*TPR)/(PPV+TPR))$	99,52%
HATA ORANI (ERROR RATE)	<i>ERR</i>	$(FP+FN)/(N+P)$	0,48%

Değerlendirme kriterlerinin en başında gelen doğruluk oranı %99,52 olarak başarı ile gerçekleşmiş Faster R-CNN algoritması ile havaalanlarının tespiti bu oranda yapılmıştır.

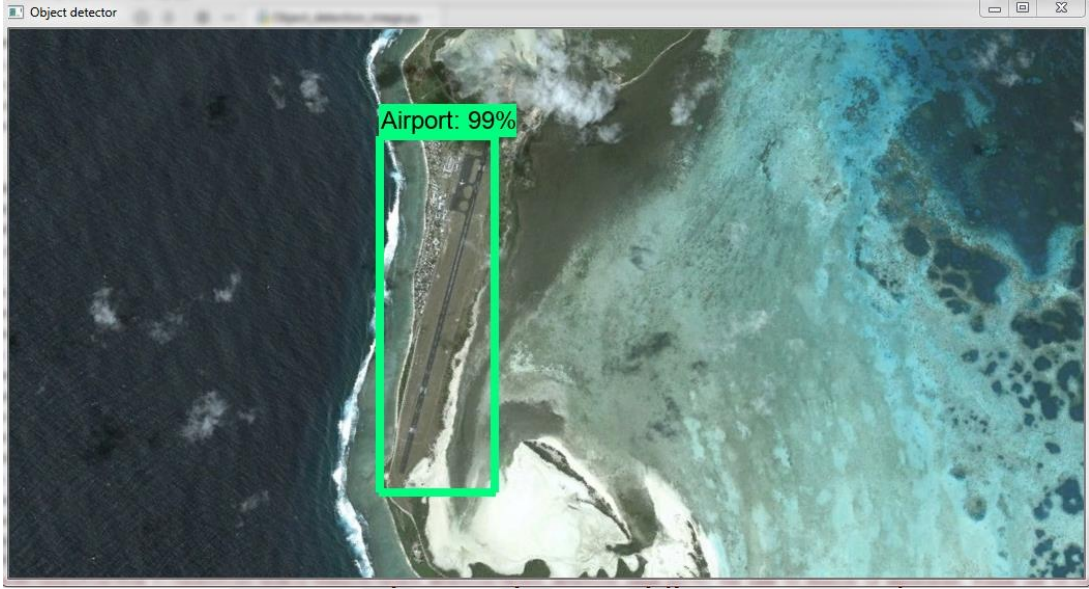
Bunun dışında değerlendirme veri setinde bulunan görüntülerin her birinin analiz edilerek sonuç üretilme süreleri değerlendirildiğinde 100 değerlendirme veri seti görüntülerinin her birinin ortalama 10,37 saniyede sonuç üretildiği ölçülmüştür. Aynı kümenin standart sapması 1,18 saniye, minimum sonuç üretim süresi 7,4 saniye, maksimum sonuç üretim süresi 14,2 saniyedir.

İlave olarak ROC eğrisine göre doğru pozitif oranı ve yanlış pozitif oranı yerleştirilirse aşağıdaki sonuç elde edilmektedir.

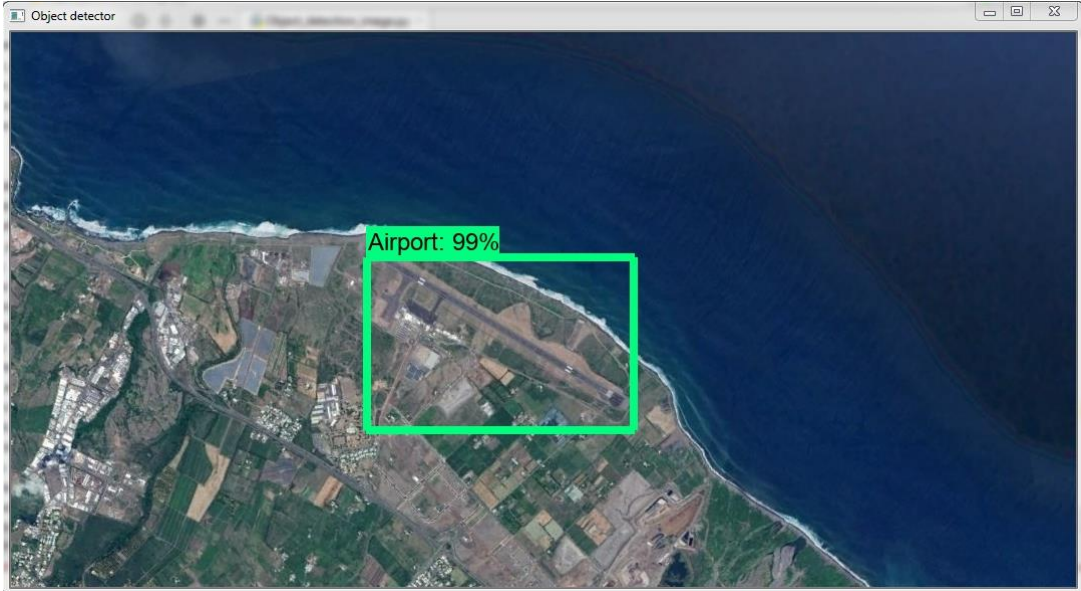


**Şekil 49. Faster R-CNN Algoritmasının Değerlendirme Sonuçlarının ROC Eğrisinde gösterimi**

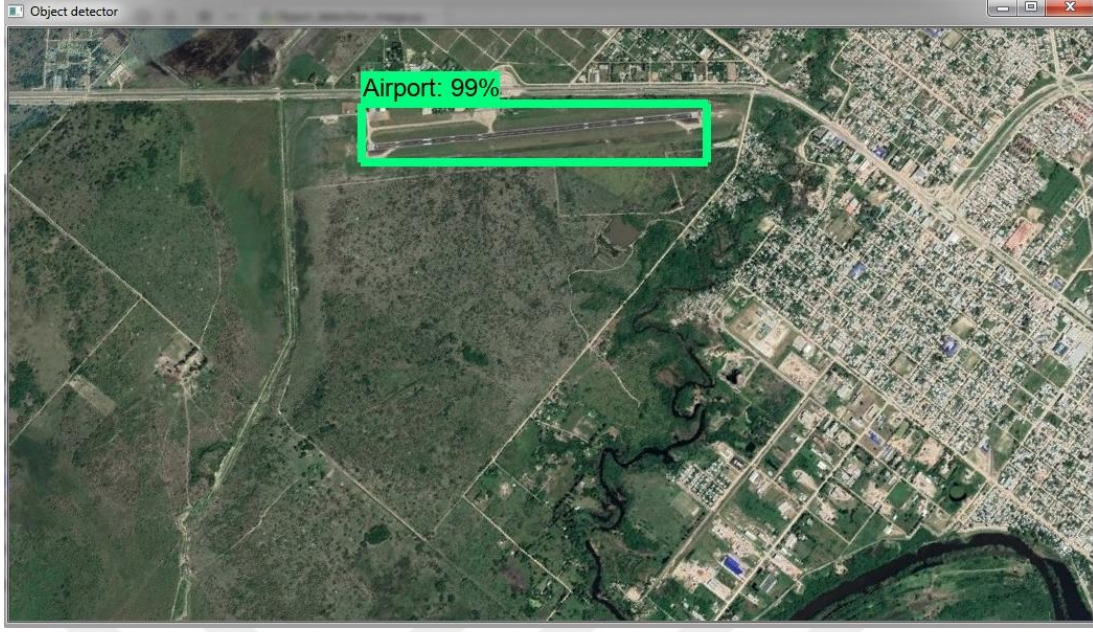
Faster R-CNN Inception\_v2 mimarisi ile deęerlendirme veri setindeki gerekleřtirilen, havaalanlarının tespit sonuları ařađıdaki řekillerde yer verilmiřtir.



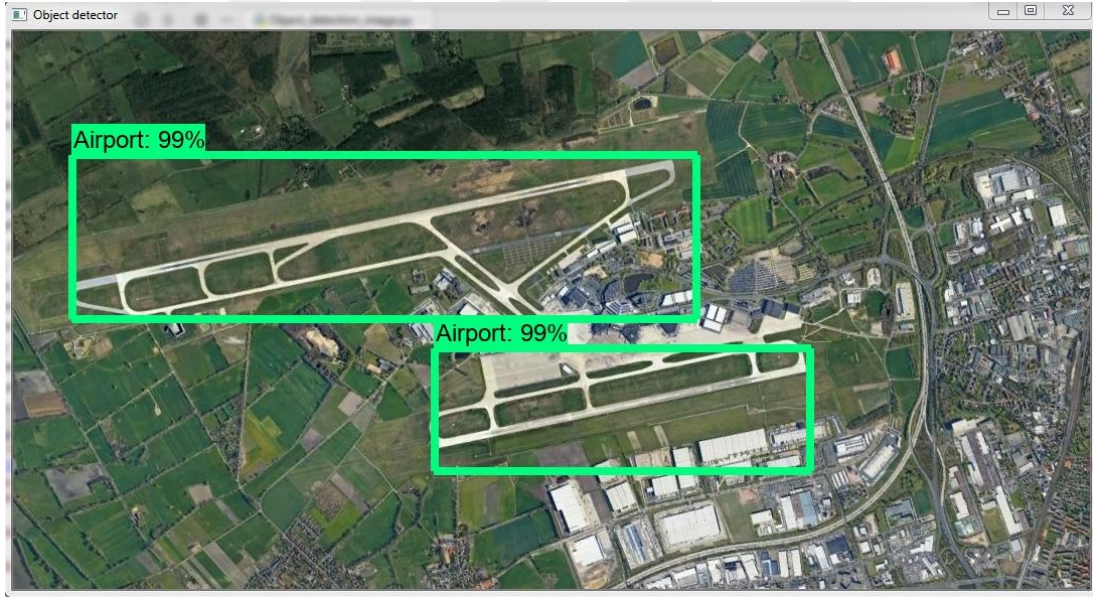
**řekil 50. Faster R-CNN Algoritması Grnt Analiz Sonucu-1**



**řekil 51. Faster R-CNN Algoritması Grnt Analiz Sonucu-2**



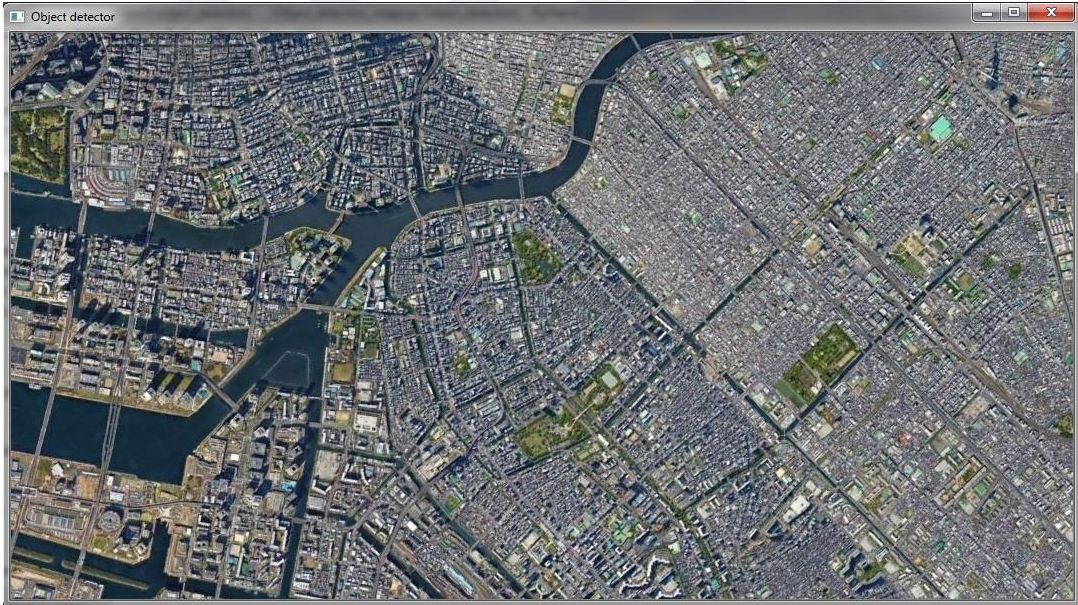
**Şekil 52. Faster R-CNN Algoritması Görüntü Analiz Sonucu-3**



**Şekil 53. Faster R-CNN Algoritması Görüntü Analiz Sonucu-4**



**Şekil 54. Faster R-CNN Algoritması Görüntü Analiz Sonucu-5**



**Şekil 55. Faster R-CNN Algoritması Havaalanı İçermeyen Görüntü Analiz Sonucu**

### 3.6 SSD ve Faster R-CNN ile Gerçekleştirilen Uygulamanın Sonuçlarının Kıyaslanması

Her iki algoritmanın yeniden eğitilmesi sonuçları eğitim ve test veri setinden tamamen bağımsız olarak oluşturulmuş olan değerlendirme veri seti ile değerlendirilmiş ve sonuçları elde edilmiştir.

Öncelikle doğruluk oranları kıyaslandığında Faster R-CNN algoritması ile %99,52 oranında tespit ile oldukça başarılı bir sonuç elde edilmiştir. SSD algoritması ile elde edilen sonuç ise sadece %76,61 oranında kalmıştır.

İki algoritma için eğitim süreleri kıyaslandığında Faster R-CNN algoritması ile gerçekleşen eğitim Intel Core i53230M 2.60 Ghz İşlemcili iş istasyonu ile 165,62 saat sürmesine karşın, SSD algoritması ile eğitim 763,88 saat sürmüştür. Bu durumda SSD mobilenet\_v2 mimarinin eğitimi Faster R-CNN inception\_v2 mimarisine göre 4,6 kat daha uzun sürmüştür. Söz konusu işlem eğer CPU yerine GPU kullanılması halinde yahut daha işlem hızı yüksek bir CPU kullanılması durumunda daha hızlı sonuç alınması beklenmektedir.

Değerlendirme kriterlerine göre her iki algoritma için sonuçların kıyaslanması aşağıdaki gibidir.

**Tablo 7. Değerlendirme Kriterlerine Göre SSD ve Faster R-CNN algoritmalarının Sonuçlarının Kıyaslanması**

KRİTERLER	KISALTMALAR	FORMÜLLER	SSD - Değerlendirme Veri Setine Göre Sonuçlar	Faster R-CNN Değerlendirme Veri Setine Göre Sonuçlar
DOĞRULUK ORANI (ACCURACY)	ACC	$(TP+TN)/(N+P)$	76,61%	99,52%
DUYARLILIK (DOĞRU POZİTİF ORANI) (SENSITIVITY)	TPR	$TP/(TP+FN)$	55,07%	99,04%
ÖZGÜLLÜK (DOĞRU NEGATİF ORANI) (SPECIFYTY)	TNR	$TN/(TN+FP)$	98,14%	100,00%
YANLIŞ POZİTİF ORANI (FALSE POSITIVE RATE)	FPR	$FP/(FP+TN)$	1,86%	0,00%
YANLIŞ NEGATİF ORANI (FALSE NEGATIVE RATE)	FNR	$FN/(FN+TP)$	44,93%	0,96%
DOĞRU TAHMİN ORANI (POSITIVE PRED VALUE -PRESICION)	PPV	$TP/(TP+FP)$	96,73%	100,00%
F PUANI (F SCORE)	F	$2*((PPV*TPR)/(PPV+TPR))$	70,18%	99,52%
HATA ORANI (ERROR RATE)	ERR	$(FP+FN)/(N+P)$	23,40%	0,48%



Görüldüğü üzere Faster R-CNN havaalanlarının tespiti SSD algoritmasına göre oldukça başarılı ve hata oranı çok düşüktür.

Faster R-CNN ve SSD algoritmalarının sonuçları üretme ya da görüntü tespit süreleri kıyaslanırsa SSD Mobilenet v2 mimarisinin Faster R-CNN Inception v2 mimarisine göre 2,5 kattan daha hızlı görüntü tespit edebildiği sonucu elde edilmiştir. Tespit sürelerinin ortalama süreleri, standart sapması, minimum ve maksimum sonuç süreleri aşağıdaki tabloda kıyaslanabilmeleri için özetlenmiştir.

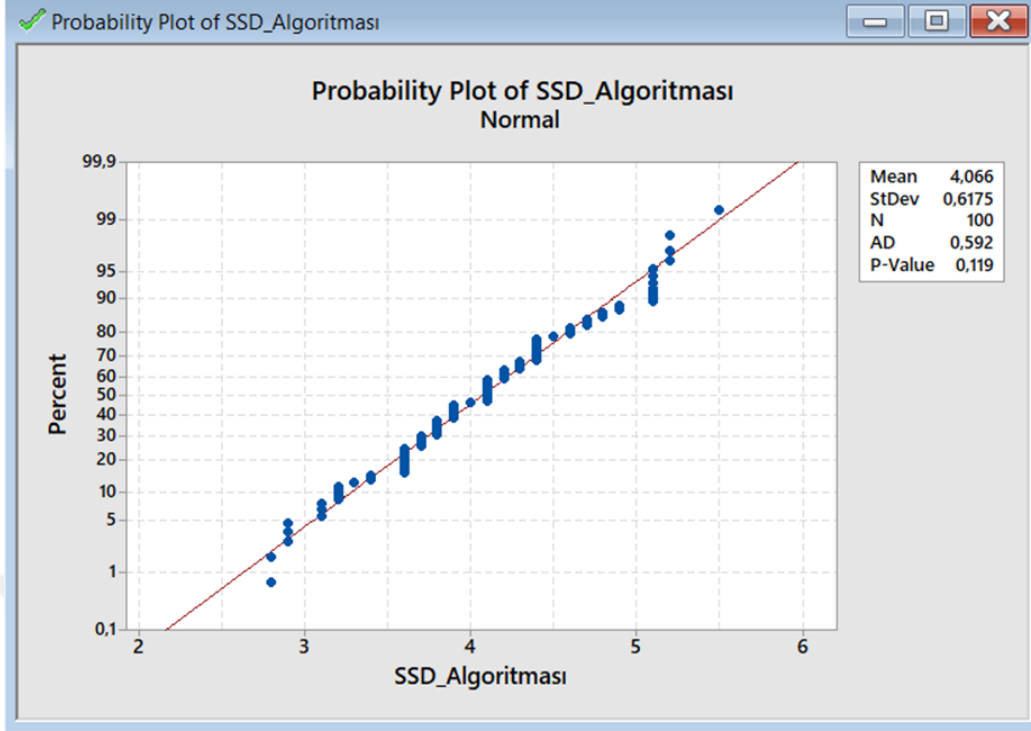
**Tablo 8. Faster R-CNN ve SSD Algoritmalarının Tespit Sürelerinin Kıyaslanması**

<b>100 Değerlendirme Veri Seti İçin Süreler</b>	<b>Faster R-CNN Inception_v2</b>	<b>SSD Mobilenet_v2</b>
Ortalama Tespit Süresi	<i>10,37 sn</i>	<i>4,06 sn</i>
Tespit Süresi Standart Sapması	<i>1,18 sn</i>	<i>0,62 sn</i>
Minimum Sonuç Süresi	<i>7,4 sn</i>	<i>2,8 sn</i>
Maksimum Sonuç Süresi	<i>14,2 sn</i>	<i>5,5 sn</i>

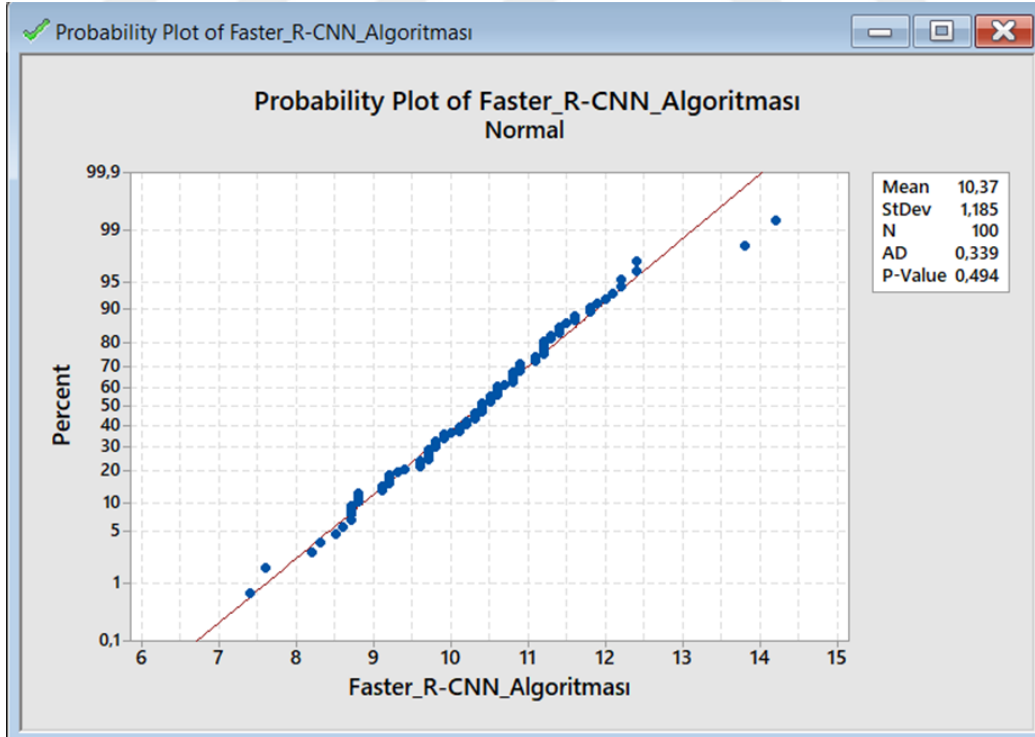
İlave olarak Minitab programı ele alınarak iki serinin sonuç üretim süreleri için 2 örneklem *t* testi yapılmıştır. Öncelikle SSD algoritma ve Faster R-CNN algoritması için ayrı ayrı Normal dağılıma uygunluk testi yapılmıştır.

$H_0$ : Veriler normal dağılmıştır.

$H_1$ : Veriler normal dağılıma uygun değildir.



Şekil 56. SSD Algoritması Anderson-Darling Testi Sonuçları



Şekil 57. Faster R-CNN Algoritması Anderson-Darling Testi Sonuçları

Her iki algoritmanın sonuç seri verileri Anderson-Darling Testi ile değerlendirilmesi sonucu her iki P değeri 0,05'den büyük olduğu için  $H_0$  Kabul edilir. Buna göre her iki seri de Normal Dağılıma uygundur.

İki serinin de Normal Dağılıma uyması neticesinde %95 anlamlılık düzeyi ile iki örneklem  $t$  testine geçilmiştir.

$H_0$ : Her iki algoritmanın görüntü tespit süreleri açısından anlamlı bir fark yoktur.

$H_1$ : Her iki algoritmanın görüntü tespit süreleri açısından anlamlı bir fark vardır.

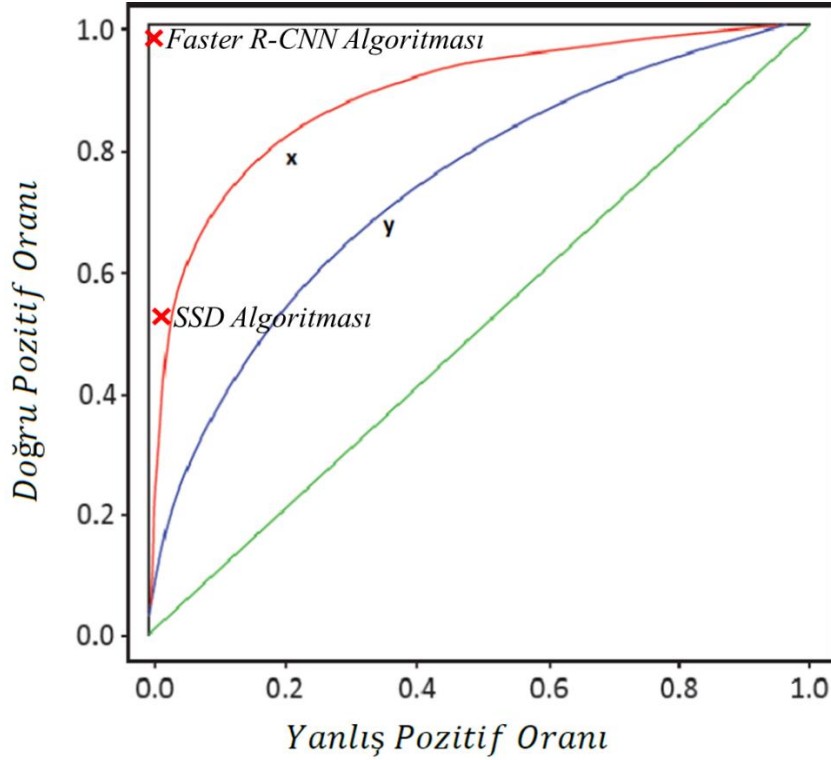
Two-Sample T-Test and CI: SSD_Algoritması; Faster_R-CNN_Algoritması				
Two-sample T for SSD_Algoritması vs Faster_R-CNN_Algoritması				
	N	Mean	StDev	SE Mean
SSD_Algoritması	100	4,066	0,617	0,062
Faster_R-CNN_Algoritması	100	10,37	1,18	0,12

Difference =  $\mu$  (SSD\_Algoritması) -  $\mu$  (Faster\_R-CNN\_Algoritması)  
Estimate for difference: -6,301  
95% CI for difference: (-6,565; -6,037)  
T-Test of difference = 0 (vs  $\neq$ ): T-Value = -47,16 P-Value = 0,000 DF = 149

**Şekil 58. SSD Algoritması ve Faster R-CNN Algoritması İki Örneklem  $t$  Testi Sonuçları**

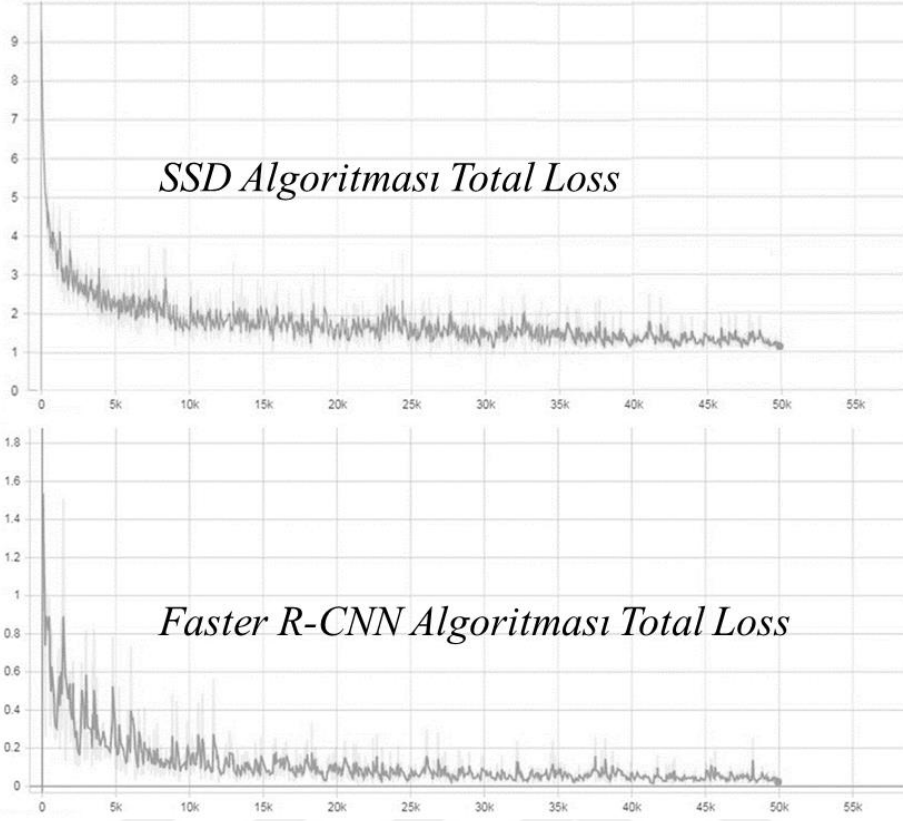
P değeri 0,05'den küçük olduğu için olduğundan  $H_0$  reddedilir. Buna göre SSD Algoritması ve Faster R-CNN Algoritmaları için görüntü tespit süreleri açısından anlamlı bir fark vardır. "Estimate for difference" ise ortalamalar arasındaki farkı belirtir ve bu fark 6,301 saniye olarak belirtilmiştir.

ROC Eğrisine göre SSD mobilenet\_v2 mimarinin eğitimi Faster R-CNN inception\_v2 mimarisi sonuçları yerleştirilir ve kıyaslanırsa, Faster R-CNN algoritmasının sınıflandırma performansı başarısının SSD algoritmasına göre daha yüksek olduğu gözükmemektedir.



**Şekil 59. SSD ve Faster R-CNN Algoritmalarının Değerlendirme Sonuçlarının ROC Eğrisinde Kıyaslamaları**

Toplam kayıp (Total Loss) grafikleri kıyaslandığında 50.000 adımda SSD algoritmasının kayıp oranının 9 üzerinden başlayarak 1 seviyesine yaklaştığı, Faster R-CNN algoritmasında ise 1,55 seviyesinden 0,1'in altına yakınsadığı görülmektedir. Bu durum R-CNN algoritmasının SSD algoritmasına kıyasla ne kadar güçlü ve doğru karar verdiğini göstermekte ilave olarak hata oranlarının çok hızlı bir şekilde azaldığını ifade etmektedir. Grafikler incelendiğinde Faster R-CNN algoritmasının 1.000 iterasyona varmadan Total Loss'un 1'in altına indiğini gösterirken SSD algoritmasının 50.000 iterasyona vardığında yeni inebildiğini göstermektedir.



**Şekil 60. SSD ve Faster R-CNN Algoritmaları Loss Grafikleri Kıyaslaması**

## SONUÇLAR VE ÖNERİLER

Görüntü işleme çalışmalarının birçok sektörde kullanıldığı ve gelişen teknoloji, işlemciler, algoritmalar ve mimariler ile doğruluk oranı yüksek sonuçlar elde edildiği yadsınamayan bir gerçek olarak hayatımıza girmiştir.

Görüntü işleme alanında gerçekleştirilen bu çalışma ile insansız hava araçları ve uydu görüntülerinden nesne tespitinde; hangi algoritmanın daha başarılı bir sonuç üreteceği, hangisinin hangi amaçla tercih edilmesi gerektiği net bir şekilde ortaya çıkarılmıştır. Yapılan çalışma ile uydu ya da İnsansız Hava Aracı görüntülerini inceleyen ve analiz eden operatörlere, Faster R-CNN algoritması ile havaalanlarının belirlenmesinde %99,52 oranında doğru sonuç üreten, SSD algoritması ile %76,61 oranında doğru sonuç üreten yardımcı bir karar destek sistemi geliştirilmiş olmuştur.

Görüntünün başarılı tespiti ile; havadan ve uzaydan çekilmiş farklı görüntüler için tanıtılabilecek birden fazla farklı nesnenin veri seti oluşturularak modelin yeniden eğitilmesinin ardından başarı ile tespitinin de sağlanabileceğinin kapısı aralanmıştır. Bunun sonucunda geliştirilebilecek söz konusu sistemler; hem operatörlere destek verebilecek hem de gözden kaçırma ya da hata yapma riskini minimize edecektir.

Intel Core i53230M 2.60 Ghz İşlemcili iş istasyonu ile gerçekleştirilen bu çalışmada SSD algoritması ile değerlendirme veri seti ile elde edilen sonuçların üretim süresi ortalama 4,06 saniye ve %76,61 oranında başarılı tespiti, 763,88 saatlik eğitimin sonunda gerçekleştirmiştir. Faster R-CNN algoritması ile 10,37 saniye ve %99,52 oranında başarılı tespiti, 165,62 saatlik eğitim sonunda gerçekleştirmiştir. Söz konusu çalışma daha iyi hıza sahip bir işlemci ya da ekran kartı üzerinden (GPU) ile gerçekleştirilirse, hem eğitimin süresi hem de nesne tespit etme süresinde ciddi oranda kısalma beklemektedir. Sürenin kısalması ile görüntü tespiti işlemi anlık görüntülerde canlı olarak mobilize araçlarda da kullanılabilir.

Oluşturulan çalışma kritik bölge tespitinde kullanılmasının yanı sıra; yeni görüntülerin tanımlanması halinde, arama ve kurtarma faaliyetlerinde, enkaz-kaza alanlarının tespiti çalışmalarında da kullanılabilir, birebir insanın görüntü analiz etmesinden bağımsız olarak hızlı bir şekilde geniş alanlar, herhangi bir duraksama, dinlenme ya da mola verilmesine ihtiyaç duymadan, tespiti gerçekleştirilen potansiyel alanlar ilgililere sunulacaktır.

## KAYNAKÇA

- Budhiraja, A., «Dropout in (Deep) Machine Learning.», <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhirajalearning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>, 2016, [Erişim Tarihi: 03.05.2019]
- Chen, K., «How to interpret “loss” and “accuracy” for a machine learning model.», ErişimAdresi:<https://stackoverflow.com/questions/34518656/how-to-interpret-loss-and-accuracy-for-a-machine-learning-model>, 2017, [Erişim Tarihi: 20.12.2019].
- Chollet, F., «Deep Learning with Python, Manning Publications Company.» 2017.
- Dirican, A., «Tanı testi performanslarının değerlendirilmesi ve kıyaslanması.» Cerrahpaşa Tıp Dergisi, 32(1), 2001, ss. 25-30.
- Firelord Phoenix, «Max-pooling / Pooling», [https://computersciencewiki.org/index.php/Max-pooling/\\_Pooling](https://computersciencewiki.org/index.php/Max-pooling/_Pooling), 2018, [Erişim Tarihi: 20.05.2019].
- Girshick, R., Donahue, J., Darrell, Malik, T. J., Berkeley, U. C. «Rich feature hierarchies for accurate object detection and semantic segmentation.» 2014.
- Girshick, R., «Fast R-CNN». Proceedings of the IEEE international conference on computer vision, 2015, ss 1440-1448.
- Github, «Tensorflow detection model zoo.», [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/detection\\_model\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md), [Erişim Tarihi: 18 Kasım 2019].
- Goodfellow, I., Bengio, Y. ve Courville, A. «Deep learning». MIT Press, 2016.
- Gulli, A., & Pal, S. «Deep Learning with Keras.» Packt Publishing Ltd., 2017
- He, K., Zhang, X., Ren, S. ve Sun, J. «Deep Residual Learning for Image Recognition,» In IEEE Conference on Computer Vision and Pattern Recognition, 2015, ss. 770-778.
- Heuritech, «A BRIEF REPORT OF THE HEURITECH DEEP LEARNING MEETUP,» [Çevrimiçi]. Available: <https://blog.heuritech.com/2016/02/29/a-briefreport-of-the-heuritech-deep-learning-meetup-5/>. [Erişim Tarihi: 20.12. 2019].

- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. ve Adam, H., «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.», <https://arxiv.org/abs/1704.04861>, 2017, [Eriřim Tarihi: 22.12.2019].
- Hsu, C., Chang, C., Lin, C. «A Practical Guide to Support Vector Classification.» vol. 1, no. 1. 2016. ss 1–16.
- Hunter, J., Dale, D., Firing, E., & Droettboom, M., «Matplotlib-development-team. » matplotlib., Chicago, 2017
- ImageNet, «ImageNet Large Scale Visual Recognition Challenge 2016.», Available: <http://image-net.org/challenges/LSVRC/2016/results#team>, 2016, [Eriřim Tarihi: 11.11.2019].
- ICAO; « Continued passenger traffic growth and robust air Cargo demand in 2017», <https://www.icao.int/Newsroom/Pages/Continuedpassenger-traffic-growth-and-robust-air-cargo-demand-in-2017.aspx>, 2018, [Eriřim Tarihi: 25.11.2019].
- İnik, Ö. ve Ülker, E. «Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri.», Gaziosmanpasa J. Sci. Res., vol. 6, no. 3, 2017, ss. 85–104.
- Kamran, F., Shahzad, M., Shafait, F. «Automated Military Vehicle Detection From Low-Altitude Aerial Images.» 2018 Digit. Image Comput. Tech. Appl. 2018. ss 1–8.
- Karaağaç, C. «İnsansız hava aracı sistemleri: Gökyüzünün yeni yırtıcı kuřları.», Academia.edu. 11 Ekim 2014'de Savunma Bilimleri Dergisi, 2015, 14 (1), 2012, ss. 145-176.
- Karunakaran, D., «Deep learning series 1: Intro to deep learning [online].» <https://medium.com/intro-to-artificial-intelligence/deep-learning-series-1-intro-to-deep-learning-abb1780ee20>, [Ziyaret Tarihi: 20 Aralık 2019]. 2018.
- Krizhevsky, A. , Sutskever, I. ve Hinton, G. E., «ImageNet Classification with Deep Convolutional Neural Networks,» NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, cilt 1, 2012, ss. 1097-1105.



- Krizhevsky, A., «Learning Multiple Layers of Features from Tiny Images.» Tech Report, 2009.
- Lasko, T. A., Bhagwat, J. G., Zou, K. H., & Ohno-Machado, L., «The use of receiver operating characteristic curves in biomedical informatics. Journal of biomedical informatics.» 38(5), 2005, ss.404-415.
- Lane, J. E., & Gantley, M. J., «Utilizing Complex Systems Statistics for Historical and Archaeological Data». Journal of Cognitive Historiography, 3(1-2), 2017, ss 68-92.
- Lecun, Y., Bottou, L., Bengio Y. ve Haffner, P. «Gradient-based learning applied to document recognition,» proceeding of the IEEE, cilt 86, no. 11, 1998, ss. 2278-2324.
- Lienhard, J. H., «The Engines of Our Ingenuity.» , University of Houston, <https://www.uh.edu/engines/epi1910.htm> , 2003, [Erişim Tarihi: 29.11.2019].
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L. ve Dollar, P., «Microsoft coco: Common objects in context,» in European conference on computer vision, 2014 ss. 740-755.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. ve Berg, A. C., «Ssd: Single shot multibox detector, European conference on computer vision.», 2016, ss. 21-37.
- Machinelearningguru, [http://machinelearningguru.com/computer\\_vision/basics/convolution/convolution\\_layer.html](http://machinelearningguru.com/computer_vision/basics/convolution/convolution_layer.html), [Erişim Tarihi: 22.11.2019].
- Massouh, N., Babilon, F., Tommasi, T., Young, J., Hawes, N. and Caputo, «Learning Deep Visual Object Models From Noisy Web Data: How to Make it Work.» B IROS At Vancouver, 2017.
- Mathworks, <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>, [Erişim Tarihi: 22.12.2019].
- MathWorks, «What Is Deep Learning.», <https://www.mathworks.com/discovery/deep-learning.html>, 2018b [Erişim Tarihi: 22.12.2019].

- Metz, C. E., «Receiver operating characteristic analysis: a tool for the quantitative evaluation of observer performance and imaging systems.» *Journal of the American College of Radiology*, 3(6), 2006, ss 413-422.
- Nielsen, M. A., «Neural networks and deep learning (Vol. 25).» San Francisco, CA, USA: Determination press, 2015.
- Polat, H., Mehr, H. D., & Cetin, A. «Diagnosis of chronic kidney disease based on support vector machine by feature selection methods.» *Journal of medical systems*, 41(4), 2017, ss. 55.
- Ren, S., He, K., Girshick, R. ve Sun, J., «Faster r-cnn: Towards real-time object detection with region proposal networks.», *Advances in neural information processing systems*, 2015, ss. 91-99.
- Russel, S. ve Norvig, P., «Artificial intelligence: A modern approach.» Prentice Hall., 2009. ss 17.
- Shanmugamani, R. «Deep Learning for Computer Vision» Pactc. 2018. ss 6-7
- Simonyan, K. ve Zisserman, A. «Very deep convolutional networks for large-scale image recognition,» *ArXiv Prepr. ArXiv*, 2014, ss. 1409-1556.
- Stanford University, «CS231n Convolutional Neural Networks for Visual Recognition.», <http://cs231n.github.io/neural-networks-2/>. [Erişim Tarihi: 20.12.2019].
- Salomon, J., «Lung Cancer Detection using Deep Learning.», Dublin Institute of Technology, 2018
- SDT, «Uzaktan algılama Sistemleri/Radar sistemleri.», <http://www.sdt.com.tr/tr/uzmanlik-alanlari/uzaktan-algilama-sistemleri/radar-sistemleri/>, 2016, [Erişim Tarihi: 09 Aralık 2019].
- Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A., «Going deeper with convolutions.», 2015 IEEE, 2015.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., «Rethinking the Inception Architecture for Computer Vision.», *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818 - 2826, doi: 10.1109/CVPR.2016.308, 2016.

- Taylor, A. J. P. «Jane's Book of Remotely Piloted Vehicles.», Collier Books, 1977
- Turing, A.M., «Computing Machinery and Intelligence.» Mind 49. 1950. ss 433-460.
- Wikipedia1, [https://tr.wikipedia.org/wiki/Dosya:First\\_flight3.jpg](https://tr.wikipedia.org/wiki/Dosya:First_flight3.jpg), [Eriřim Tarihi: 25.12.2019].
- Wikipedia2, [https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)), [Eriřim Tarihi: 25.12.2019].
- Verma, H. K. ve Pal, S., «Modified sigmoid function based gray scale image contrast enhancement using particle swarm optimization.» J. Inst. Eng. Ser. B, vol. 97, no. 2, 2016. Ss. 243–251
- Xiaozhu, X., «Object Detection of Armored Vehicles Based on Deep Learning in Battlefield Environment.» 2017. ss 1569–1571.
- Yalçın, O., «Türk Hava Harp Sanayi Tarihi (1913-2009).», Hv.K.K.lığı Tarihçe Őube M¼d¼rl¼ę¼ 45 Sayılı Yayını, Hava Basımevi ve Neřriyat M¼d¼rl¼ę¼, Etimesgut-Ankara, 2009, s.15.
- Zhang, A., Lipton, Z. C., Li, M. and Smola, A. J., «Dive into Deep Learning. » <http://en.diveintodeeplearning.org>, 2018, [Eriřim Tarihi: 21 Aralık 2019].

## ÖZGEÇMİŞ

07 Eylül 1990 tarihi, İstanbul İli Fatih ilçesi doğumluyum. İlk ve Orta Öğrenimimi Kocatepe İlk Öğretim Okulunda tamamlamamın ardından Liseyi Fahrettin Kerim Gökay Anadolu Lisesinde tamamladım. Lisans öğrenimimi Kocaeli Üniversitesi Endüstri Mühendisliği Bölümünden 2013 yılında tamamlayarak mezun oldum. Aynı yıl ticari otomotiv sektörünün öncülerinden olan özel bir firmada, askerlik hizmetime kadar olan sürede, Endüstri Mühendisi olarak görev aldım. Askerlik görevimi Van İl Jandarma Komutanlığında 2016 yılında tamamladım. Aynı yıl ülkemizin Dünya platformlarına yayın yapan yayın organı olan öncü kurumda 2,5 yıl süre ile Proje Yöneticisi olarak görev aldım. 2018 yılında ülkemizin havacılık ve uzay sektöründe öncü olan kurumunda Kaynak (Resource) Yönetimi Mühendisi olarak başladığım görevime, Strateji Yönetimi Mühendisi olarak devam etmekteyim. 2018 yılı Ocak ayında Beykent Üniversitesi, Endüstri Mühendisliği Ana Bilim Dalında yüksek lisans eğitimine başladım.

Özel ilgi alanlarım, problem çözme, süreç planlama ve optimizasyonu, kaynakların etkin yönetimi, program yönetimi, proje yönetimi ve strateji yönetimidir.

Yabancı dilim İngilizce ve Almancadır.

**Muhammed Taha ZEREN**