

**DERİN ÖĞRENME YÖNTEMİ İLE GOOGLE VE YANDEX
GÖRÜNTÜLERİNDEN STADYUM TESPİTİ**

Emre Batuhan SAMUR

Yüksek Lisans Tezi

Uzaktan Algılama ve Coğrafi Bilgi Sistemleri ABD Anabilim Dalı

Danışman: Doç. Dr. Uğur AVDAN

Eskişehir

Eskişehir Teknik Üniversitesi

Lisansüstü Eğitim Enstitüsü

Haziran 2022

27/06/2022

DANIřMAN ONAYI

Daniřmanlıęını yrttęm Yksek Lisans ęrencisi Emre Batuhan SAMUR, DERİN ÖĞRENME YÖNTEMİ İLE GOOGLE VE YANDEX GÖRNTLERİNDEN STADYUM TESPİTİ bařlıklı tez alıřmasını tamamlamıřtır. Hazırlamıř olduęu tez tarafımca incelenmiř ve ęrencinin tez savunma sınavına alınması bilimsel ve etik aıdan uygun grlmřtir.

Tez Daniřmanı

Do. Dr. Uęur AVDAN

ÖZET

DERİN ÖĞRENME YÖNTEMİ İLE GOOGLE VE YANDEX GÖRÜNTÜLERİNDEN STADYUM TESPİTİ

Emre Batuhan SAMUR

Uzaktan Algılama ve Coğrafi Bilgi Sistemleri ABD Anabilim Dalı
Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Haziran 2022

Danışman: Doç. Dr. Uğur AVDAN

Teknolojinin hızla gelişmesi ve bilgisayar donanımlarında görülen ilerleme sonucunda derin öğrenmenin önemi gün geçtikçe artmaktadır. Son dönemlerde birçok kurum/kuruluş insan gücü yerine, otonom sistemleri kullanmayı tercih etmektedir. Derin öğrenme algoritmalarının sürekli geliştirildiği alanlardan biri de nesne tespiti yapabilen modellerdir. Tez çalışmasında amaç literatürde bulunan çalışmalardan farklı olarak iki ayrı şirket tarafından aynı köşe koordinat ve ölçekte temin edilmiş uydu görüntüleri ile bir modelin eğitilmesi sonucu şirketlerin paylaştığı uydu görüntülerinin modelin başarımına olan etkisinin analiz edilmesidir. Tez çalışmasında yazılım dili olarak Python 3.6, yazılım geliştirici olarak ise Anaconda kullanılmıştır. Derin öğrenme kütüphanesi olarak Tensorflow kütüphanesi, bilgisayarlı görü kütüphanesi olarak OpenCV ve nesne tespit modeli olarak Tensorflow Object Detection Application Programming Interface (API) kullanılmıştır. Bu kapsamda COCO veri seti ile eğitilmiş ‘Faster R-CNN Inception v2 COCO’ ve ‘Faster R-CNN Resnet101 COCO’ modelleri kullanıcı sistemine aktarılmış, açık kaynak programı olan SAS Planet uygulaması üzerinden Google ve Yandex şirketlerinin sunduğu uydu görüntülerinden ‘stadyum’ kapsayan görüntüler temin edilmiş ve her iki model hem Google hem de Yandex şirketlerine ait uydu görüntüleri ile eğitilmiştir. Çalışmanın doğruluk analizi F1-Skoru yöntemiyle yapılmıştır. F1-Skoru doğruluk metriğinde en başarılı sonucu veren model % 91 doğruluk oranı ile Yandex şirketinin uydu görüntüleriyle eğitilen Faster R-CNN Inception v2 COCO’dur.

Anahtar Sözcükler: Derin öğrenme, Nesne tespiti, Uydu görüntüsü, Faster R-CNN, Tensorflow.

ABSTRACT

STADIUM DETECTION FROM GOOGLE AND YANDEX IMAGES WITH DEEP LEARNING METHOD

Emre Batuhan SAMUR

Department of Remote Sensing and Geographical Information Systems

Eskişehir Technical University, Institute of Graduate Programs, June 2022

Supervisor: Assoc. Prof. Dr. Uğur AVDAN

As a result of the rapid development of technology and the progress in computer hardware, the importance of deep learning is increasing day by day. Recently, many institutions/organizations prefer to use autonomous systems instead of manpower. One of the areas where deep learning algorithms are constantly being developed is models that can detect objects. The aim of the thesis study is to develop a model with satellite images provided by two different companies at the same corner coordinate and scale, and to analyze the effect of the satellite images shared by the companies on the success of the model. In the thesis study, Python 3.6 was used as the software language and Anaconda was used as the software developer. Tensorflow library was used as deep learning library, OpenCV library was used as computer vision library and Tensorflow Object Detection Application Programming Interface (API) was used as object detection model. In this context, the 'Faster R-CNN Inception v2 COCO' and 'Faster R-CNN Resnet101 COCO' models trained with the COCO data set were transferred to the user system, and the 'stadium' images from the satellite images shared by Google and Yandex companies via the open source program SAS Planet application were provided and both models were trained with satellite images belonging to both Google and Yandex companies. Accuracy analysis of the study was made using the F1-Score method. The model that gives the most successful result in the F1-Score accuracy metric is the Faster R-CNN Inception v2 COCO, which is trained with the satellite images of the Yandex company with an accuracy rate of 91%.

Keywords: Deep Learning, Object Detection, Satellite Image, Faster R-CNN, Tensorflow.

TEŞEKKÜR

Tez dönemine başladığım gün itibariyle böylesine güncel ve yazılım ile uzaktan algılamanın entegre olduğu derin öğrenme konusunda tez yapmama vesile olan, bu süreçte bilgisini, tecrübesini ve desteğini hiçbir şekilde esirgemeyen tez danışmanım Sayın Doç. Dr. Uğur AVDAN hocama müteşekkirim. Tez çalışmamın başından sonuna kadar fedakârlığını hiç esirgemeyen, bana sonsuz güvenen ve ‘her başarılı erkeğin arkasında güçlü bir kadın vardır’ söylemini bana derinden hissettiren çok değerli eşim Ayşe SAMUR’a teşekkür ederim. Son olarak çalışmamın başından itibaren hayatımıza ortak olan canım kızım Lavinya SAMUR’a tez çalışmamı armağan ediyorum.

Emre Batuhan SAMUR

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Eskişehir Teknik Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Emre Batuhan SAMUR

İÇİNDEKİLER

Sayfa

BAŞLIK SAYFASI	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
DANIŞMAN ONAYI	iii
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR	vi
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	vii
İÇİNDEKİLER	viii
TABLolar DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
SİMGELER VE KISALTMALAR DİZİNİ	xiii
1. GİRİŞ	1
1.1. Literatür Taraması	3
1.2. Problemin Tanımlanması	5
1.3. Tez Çalışmasının Özgün Değeri.....	5
1.4. Tez Çalışmasının Yapısı	6
1.5. Tez Çalışmasının Akışı	6
2. YAPAY ZEKA	7
2.1. Yapay Zekanın Tarihçesi	8
2.2. Makine Öğrenmesi	11
2.2.1. Gözetimli makine öğrenme stratejileri.....	11
2.2.1.1. Doğrusal regresyon analizi.....	11
2.2.1.2. Lojistik regresyon analizi	12
2.2.1.3. Sınıflandırma ve regresyon ağaçları tekniği (SRAT).....	13
2.2.1.4. Naive Bayes sınıflandırıcı.....	13
2.2.1.5. En yakın komşu algoritmaları (KNN)	14
2.2.1.6. Destek Vektör Makineleri (DVM)	15

2.2.2. Gözetimsiz makine öğrenme stratejileri.....	15
2.2.2.1. K-ortalamlar (K-Means) kümeleme algoritması	15
2.2.2.2. Hiyerarşik kümeleme.....	16
2.3. Yapay Sinir Ağları	17
2.4. Derin Öğrenme	17
3. EVRİŞİMLİ SİNİR AĞLARI.....	20
3.1. Evrişimli Sinir Ağları'nın Katmanları.....	21
3.1.1. Evrişim Katmanı	21
3.1.2. Havuzlama Katmanı	21
3.1.3. Tam Bağlı Katman	22
3.2. Evrişimli Sinir Ağı Modelleri.....	22
3.2.1. Lenet-5.....	23
3.2.2. AlexNet	23
3.2.3. VGGNet.....	24
3.2.4. Microsoft ResNet	24
3.2.5. GoogleNet.....	25
3.3. Derin Öğrenme ile Obje Tespiti.....	25
4. STADYUM TESPİT METODU	28
4.1. Kullanılan Programlar ve Kütüphaneler	28
4.2. Model Seçimi	28
4.3. Eğitim Veri Setleri	30
4.4. Modelin Eğitilme Prensibi.....	33
4.5. Hiper Parametre Seçimi	34
4.6. Model Performanslarının Karşılaştırılması	34
4.7. Kayıp (Loss) Grafikleri ve Karşılaştırılması	36
5. EĞİTİLEN MODELLERİN TEST EDİLMESİ.....	40
6. SONUÇ	50
KAYNAKÇA.....	55
ÖZGEÇMİŞ	

TABLÖLAR DİZİNİ

Sayfa

Tablo 4.1. COCO veri seti ile eğitilmiş modeller	29
Tablo 4.2. Modelin eğitilmesi ve test edilmesinde kullanılan bilgisayarın teknik özellikleri	29
Tablo 4.3. Google ve Yandex görüntüleriyle eğitilmiş model performanslarının karşılaştırılması	36
Tablo 4.4. Eğitilen modellerin kayıp değerleri	39



ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Yapay zeka, makine öğrenmesi ve derin öğrenmenin kronolojik süreci	7
Şekil 2.2. Rosenblatt'ın perceptron algoritması	8
Şekil 2.3. Ivakhnenko tarafından eğitilen, bilinen ilk derin ağ mimarisi (Şeker & Dirı & Balık, 2017)	9
Şekil 2.4. Fukushima tarafından geliştirilen Neokognitron mimarisi	9
Şekil 2.5. Doğrusal regresyon grafiğı	12
Şekil 2.6. İki düzeyli kategorik verilerden oluşan lojistik regresyon grafiğı (Kavuncu, 2018)	12
Şekil 2.7. Sınıflandırma ve regresyon ağaçları tekniğı şeması (Kavzoğılu & Çölkesen, 2010)	13
Şekil 2.8. En yakın komşu algoritması sınıf belirleme işlemi	14
Şekil 2.9. DVM yapısı (Ayhan & Erdoğan, 2014)	15
Şekil 2.10. K-ortalamlar (Means) algoritması	16
Şekil 2.11. Yapay sinir ağlarının mimarisi	17
Şekil 2.12. Derin öğrenme mimarisi	18
Şekil 3.1. Evrişimli sinir ağı yapısı (Zhang vd., 2019)	20
Şekil 3.2. Evrişim işlemi (Liu vd, 2016)	21
Şekil 3.3. Havuzlama işlemi (Türkoğılu vd., 2020)	22
Şekil 3.4. Tam bağı katman yapısı	22
Şekil 3.5. LeNet ağıının mimarisi (Şeker & Dirı & Balık, 2017)	23
Şekil 3.6. AlexNet mimarisi (Krizhevsky, Sutskever et al. 2012)	23
Şekil 3.7. VGG mimarisi	24
Şekil 3.8. Microsoft ResNet ilk 34 katmanının ağ mimarisi (He vd. 2016)	24
Şekil 3.9. GoogLeNet ağ mimarisi (Szegedy vd., 2015)	25
Şekil 3.10. R-CNN mimarisi	26
Şekil 3.11. Fast R-CNN mimarisi (Girshick, 2015)	26
Şekil 3.12. Faster R-CNN mimarisi (Ren vd., 2017)	27
Şekil 4.1. Tez kapsamında kullanılan Google veri setinden örnekler	32
Şekil 4.2. Tez kapsamında kullanılan Yandex veri setinden örnekler	32
Şekil 4.3. LabelImg_v1.8.0 programı ile veriyi etiketleme işlemi	33

Şekil 4.4. <i>Google uydu görüntüleri ile eğitilen Faster R-CNN Inception v2 COCO modelinin eğitim veri seti için kayıp grafiği</i>	36
Şekil 4.5. <i>Yandex uydu görüntüleri ile eğitilen Faster R-CNN Inception v2 COCO modelinin eğitim veri seti için kayıp grafiği</i>	37
Şekil 4.6. <i>Google uydu görüntüleri ile eğitilen Faster R-CNN ResNet101 COCO modelinin kayıp grafiği</i>	37
Şekil 4.7. <i>Yandex uydu görüntüleri ile eğitilen Faster R-CNN ResNet101 COCO modelinin kayıp grafiği</i>	38
Şekil 4.8. <i>Google ve Yandex veri setleri ile eğitilen Faster R-CNN inception v2 ve Faster R-CNN Resnet101 COCO modellerinin kayıp grafiği.....</i>	38
Şekil 5.1. <i>Birinci test görüntüsü</i>	40
Şekil 5.2. <i>İkinci test görüntüsü</i>	41
Şekil 5.3. <i>Üçüncü test görüntüsü.....</i>	42
Şekil 5.4. <i>Dördüncü test görüntüsü</i>	43
Şekil 5.5. <i>Beşinci test görüntüsü</i>	44
Şekil 5.6. <i>Altıncı test görüntüsü.....</i>	45
Şekil 5.7. <i>Yedinci test görüntüsü</i>	46
Şekil 5.8. <i>Sekizinci test görüntüsü</i>	47
Şekil 5.9. <i>Dokuzuncu test görüntüsü</i>	48
Şekil 5.10. <i>Onuncu test görüntüsü.....</i>	49
Şekil 6.1. <i>Google veri seti ile eğitilen Faster R-CNN Inception v2 COCO ve Faster R-CNN Resnet101 COCO modellerin grafiği</i>	51
Şekil 6.2. <i>Yandex veri seti ile eğitilen Faster R-CNN Inception v2 COCO ve Faster R-CNN Resnet101 COCO modellerin grafiği</i>	52
Şekil 6.3. <i>Google ve Yandex veri seti ile eğitilen Faster R-CNN Inception v2 COCO modelinin grafiği.....</i>	53
Şekil 6.4. <i>Google ve Yandex veri seti ile eğitilen Faster R-CNN Resnet101 COCO modelinin grafiği.....</i>	54

SİMGELER VE KISALTMALAR DİZİNİ

ILSVRC	: ImageNet Büyük Ölçekli Görsel Tanıma
ResNet	: Residual Networks
R-CNN	: Region-based Convolutional Neural Networks
KNN	: En Yakın Komşu Algoritmaları
DVM	: Destek Vektör Makineleri
YSA	: Yapay Sinir Ağları
ESA	: Evrişimli Sinir Ağları
CPU	: Merkezi İşlem Birimi
GPU	: Grafik İşlemci Birimi

1. GİRİŞ

Uzaktan algılamada kullanılan platformların teknik olarak gelişmesi ve paralelinde yapay zekâ, makine öğrenimi ve derin öğrenme gibi bilişimsel gelişmelerin de yaşanması insan ihtiyacını karşılamada ve yapılan çalışmalarda teknolojiler arası entegrasyonun önemini bir hayli artırmaktadır. Günümüzde sivil kişi/kurumların ihtiyacı ve askeri istihbarat kapsamında çeşitli uzaktan algılama platformlarından elde edilen görüntüler her geçen gün gelişmekte olan derin öğrenme algoritmaları ile işlenmekte olup kullanıcıların amaçları doğrultusunda ürün elde edilmektedir.

Günümüz teknolojisiyle birlikte daha hızlı büyüyen ve karmaşık hale gelen verilerin analiz edilmesinde kişi ve kurumlar, insanların kendi analiz çıkarımlarına göre daha süratli ve daha az hata yapan derin öğrenme algoritmalarını kullanmaktadırlar. Bilişimsel gelişmelerin yaşanması şirketlerin üretim / hizmet işlemlerini süreç ve maliyet açısından minimize etmekte, kâr elde etme açısından maksimize etmektedir. Dolayısıyla bilişimsel gelişmeler şirketler arası rekabeti de beraberinde getirmekte ve yapay zekanın günümüzde evrildiği derin öğrenme alanına yönelimini de sağlamaktadır.

Daha önce geliştirilmiş olan algoritmalar etiketlenmiş verileri öğrenip ardından bu öğrendiklerini daha fazla veri seti üzerinden yeni çıkarımlar sağlama yöntemi prensibinde tasarlanmıştır. Ancak istenilen sonuç elde edilmediğinde yeniden öğretilme işlemine gereksinim duymaktadırlar. Bu kapsamda geliştirilen derin öğrenme algoritmalarının daha önce geliştirilen algoritmalarından en önemli farkı insan müdahalesine ihtiyaç duymamasıdır. Çünkü algoritma, yapısında barındırdığı çoklu katmanlar ile veriyi çeşitli nosyonlardan meydana gelen bir aşamaya yerleştirmektedir. Bu doğrultuda algoritma kendi içerisinde nöronların sahip olduğu ağırlık değerlerini güncelleyerek kendi içerisinde hatalarını tekrar etmemeye yönelik sonuçlar çıkarmaktadır (Song ve Lee, 2013).

Günden güne büyüyen veri boyutlarına karşı bilgisayar donanımlarının yetersiz kalması yapay zekanın önemini yitirmesine sebep olmuştur ancak 2000’li yıllardan sonra başta grafik işlemci ünitesi olmak üzere donanımsal gelişmelerin yaşanması özellikle 2010’lu yılların başları itibariyle yapay zekanın çoklu gizli katmanlar ile kullanımını kolaylaştırmıştır.

Yapay zekanın önemini yeniden kazanmasıyla birlikte 2012 yılında düzenlenen ILSVRC (ImageNet Büyük Ölçekli Görsel Tanıma) yarışmasında Hinton ve Krizhevsky'nin 8 katmanlı bir evrişimli sinir ağı modeli olan AlexNet modeli yaklaşık %15,3 hata oranı ile yarışmayı kazanarak derin öğrenme konusunda önemli bir adım atmıştır.

Bu gelişmeden sonra özellikle Facebook, Microsoft ve Google gibi teknoloji firmaları başlıca olmak üzere birçok firma yatırımlarını derin öğrenme üzerine yapmıştır. Facebook şirketinin 2012 yılında yüz tanıma teknolojisi üzerine çalışan face.com'u alması, Google şirketinin 2014 yılında yapay zeka ile öğrenme ve sinir ağlarını harmanlayarak genel amaçlı öğrenme algoritmaları tasarlayan Deepmind şirketini satın alması, Microsoft firmasının klavye uygulaması SwiftKey'i satın alması gibi bir çok girişim teknoloji firmalarının yapay zeka konusuna verdiği önemi göstermektedir.

Derin öğrenmenin yaygınlaşması ve gelişmesi özellikle bilgisayarlı görü alanında popülerlik kazanmıştır. Yapılan bilimsel çalışmalarda; parçalara ayırma (segmentation), sınıflandırma (classification), sahne sınıflandırma (scene classification) ve nesne tespiti (object detection) konularının ağırlıklı olduğu görülmektedir (Körez, 2020).

Parçalara ayırma işlemi bir görüntü üzerinde ilgilenilen alanların ayrıştırılması işlemidir. Hareket halindeki taşıtların çevreden ayırt edilebilmesi için parçalara ayırma işlemi uygulanabilmektedir. En temel haliyle parçalara ayırma, piksellerin radyometrik çözünürlük değerlerine göre etiketleme işleminin yapılmasıdır. Etiketleme esnasında etiketlenen objenin çoklu özelliklerinden yararlanılır ve veri setinde objeye ait fazla görüntünün olması sınıflandırma başarısını genellikle artırır. Uzaktan algılama vasıtaları kullanılarak çıkarım yapmak için görüntülerin spektral yansıma özellikleri analiz edilmektedir. Bu analizler geliştirilmiş birtakım algoritmalar ile denetimli veya denetimsiz olmak üzere iki ayrı şekilde yapılmaktadır. Sahne sınıflandırmada ise görüntüde yer alan objeler bir bütün olarak analiz edilerek görüntünün anlamlandırılması sağlanır. Nesne tespitinde ise geliştirilen algoritmalar vasıtasıyla görüntüdeki objelerin konumlarının belirlenmesine çalışılır.

Nesne tespiti konusunda son dönemlerde kullanılan en önemli derin öğrenme algoritması evrişimsel sinir ağlarıdır. Evrişimsel sinir ağlarıyla ilgili detaylı bilgi tez çalışmasının ilerleyen bölümlerinde ayrı bir başlık altında verilmiştir.

1.1. Literatür Taraması

Doğan (2021) SatNet: uydu görüntülerinden nesne tespiti için yeni bir derin öğrenme modeli adlı tez çalışmasında tarafınca geliştirilen SatNet modeline ilave on bir adet derin öğrenme modeli kullanarak tekli ve çoklu nesne tespiti yapmıştır. SatNet modeli tekli nesne tespitinde %86,6 başarı oranı ile ikinci, çoklu nesne tespitinde ise en iyi sonucu vermiştir.

Murat (2021) insansız hava aracı görüntülerinden derin öğrenme yöntemleriyle nesne tanıma adlı çalışmada insansız hava araçları ile elde edilen görüntüler üzerinden YOLOv5, YOLOv3, RetinaNet ve EfficientDet olmak üzere dört farklı modelin eğitilmesi ile modellerin doğruluk değerleri karşılaştırılmıştır. YOLOv5 modeli %69 doğruluk oranı ile nesne tespitinde en başarılı sonucu veren model olmuştur.

Duman (2019) derin öğrenme yöntemi ile optik uydu görüntülerinden gemi tespiti adlı çalışmada 2085 adet görüntü ile “Faster R-CNN Inception v2 COCO, Faster R-CNN Resnet101 COCO, SSD Inception v2 COCO” olmak üzere üç farklı modeli ‘eğitmiş, daha sonra aynı modeli çözünürlüğü yaklaşık 10 kat daha yüksek olan 1056 veri seti ile ayrıca eğitmiştir. Neticede yüksek çözünürlüklü veri seti ile eğitilen modelin doğruluk değerinin daha yüksek olduğu tespit edilmiştir.

Körez (2020) derin öğrenme kullanarak uzaktan algılama görüntülerindeki nesnelerin tespiti adlı çalışmada uzaktan algılama görüntülerinde nesne tespiti için iki farklı model önerilmiştir. İlk model Daha Hızlı Bölgesel Evrişimli Sinir Ağı (Daha Hızlı B-ESA), ikinci model ise üç farklı derin öğrenme nesne tespit modelinin optimize edilmiş katsayı ağırlıklı nesne tespiti yapmasına olanak sağlayan işbirlikçi bir nesne tespit modelidir. Bu iki modelin etkinliği ve başarısı Northwestern Polytechnical University Very High Resolution 10 (NWPU-VHR10) veri seti üzerinde yapılan testlerde değerlendirilmiştir. Literatürde aynı veri setleri üzerinden yapılan çalışmalarla mukayese edildiğinde birinci modelin tespit başarısının %1,7, ikinci modelin ise %3,64 oranında arttığı görülmüştür.

Tan (2019) derin öğrenme yardımıyla araç sınıflandırma adlı çalışmada 10 farklı modelde/markada aracın yaklaşık 20000 adet veri seti ile transfer öğrenme yöntemi aracılığıyla Faster R-CNN Resnet50, Faster RCNN ResNet101, R-FCN ResNet101 ve

SSD mobilenet modelleri eğitilmiş, Faster R-CNN Resnet50 modeli %94,4 araç tespit oranı ile en başarılı sonucu vermiştir.

Daş ve diğ. (2019) derin öğrenme ile resim ve videolarda nesnelerin tanınması ve takibi adlı çalışmada hareketli nesnelerin tespiti ve takibi için Google şirketine ait Tensorflow kütüphanesini kullanmış ve Faster R-CNN modelini eğiterek resim, video ve webcam üzerinden nesne tespitleri yapmıştır.

Dandil ve diğ. (2019) daha hızlı bölgesel-evrimsel sinir ağları ile sığır yüzlerinin tanınması adlı çalışmada derin öğrenme modeli olan Daha Hızlı Bölgesel-Evrimsel Sinir Ağları (DHB-ESA) modelini kullanılarak sığırların yüz görüntülerinin sınıflandırılarak tespit edilmesini sağlamıştır.

Bayram (2019) derin öğrenme tabanlı otomatik plaka tanıma adlı çalışmada Maskeli Bölgesel Evrimsel Sinir Ağları (M-BESA) tabanlı model kullanarak otomatik olarak plaka tanıyan sistem geliştirmiştir.

Ophoff ve diğ. (2020) tarafından icra edilen çalışmada temin edilen uydu görüntülerinden araç ve gemi gibi küçük nesnelerin tespiti için YOLOv2, YOLOv3, D-YOLO ve YOLT modellerini eğitmişlerdir. Çalışma neticesinde en başarılı model D-YOLO modeli olmuştur. Çalışmanın literatüre katkısı, büyük veri setlerinin daha hızlı ve doğruluk değeri yüksek yeni modellerin eğitilmesine olanak sağlamasıdır.

Carlet ve Abayowa (2017) tarafından icra edilen çalışmada hava fotoğraflarından elde edilen veri setleri ile Faster R-CNN ve YOLOv2 modellerini eğiterek modellerin başarımlarını karşılaştırmıştır. Sonuç olarak YOLOv2 modelinin Faster R-CNN modeline göre doğruluk ve hız olarak dört kat daha başarılı olduğu sonucuna varmışlardır.

Jin ve diğ. (2007) icra ettikleri çalışmada uydu görüntülerinden araç tespit etmek için CBS yol vektör haritası kullanarak ve pikselleri araç ve araç olmayan piksel biçiminde sınıflandırmıştır. Morfolojik paylaşılan sinir ağı modeli eğitilerek yanlış alarm sayısının düştüğünü ve başarı oranının %84 seviyesinde olduğunu tespit etmişlerdir. Çalışmanın literatüre katkısı yüksek çözünürlüklü uydu görüntülerinden küçük ebatlardaki nesnelerin tespit edilmesine olanak sağlamasıdır.

1.2. Problemin Tanımlanması

Literatürde derin öğrenme yöntemleriyle nesne tespitine yönelik birçok çalışma yer almaktadır. Ancak literatür incelemesinde genel olarak algoritmaların birbirleri ile karşılaştırılması üzerine çalışmalar yapıldığı tespit edilmiştir. Bu sebepten dolayı literatür incelemesinde karşılaşılmayan bir problem olarak bir derin öğrenme modelinin farklı şirketlerin açık kaynak üzerinden sunduğu uydu görüntülerinin aynı ölçek ve köşe koordinatlarında temin edilerek eğitilmesi sonucu veri setlerinin nesne tespitine yönelik başarı ve hata oranlarının karşılaştırmasını sağlamak bu tez çalışmasında temel problem olarak ele alınmıştır. Bir başka problem olarak; aynı veri seti ile farklı derin öğrenme modellerinin eğitilmesi sonucu modellerin başarı ve hata oranlarının ortaya konmasını sağlamaktır.

Bu doğrultuda tez çalışması iki temel amacı kapsamaktadır. Bunlardan ilki; hiper parametreleri sabit tutulan bir modelin açık kaynaklı yazılım olan SAS Planet programı vasıtasıyla temin edilen aynı köşe koordinatlarına ve aynı ölçeğe sahip Google ve Yandex uydu görüntülerinin kullanılarak eğitilmesi ve böylece model eğitiminde aynı özelliklere sahip iki farklı veri setinin modelin başarı ve hata oranına etkisini tespit etmektir. İkinci temel amaç; bir veri seti ile hiper parametreleri ve çalışma algoritması birbirinden farklı iki model ile eğitilmesi sonucu modeller arasında başarı ve hata oranlarının karşılaştırılmasını sağlamaktır.

Çalışma kapsamında kullanılan modeller “Faster R-CNN Inception v2 COCO” ve “Faster R-CNN Resnet101 COCO”dur. Kullanılan veri setleri ise Google ve Yandex şirketlerinin SAS Planet uygulaması üzerinde paylaşımına sundukları uydu görüntüleridir.

1.3. Tez Çalışmasının Özgün Değeri

Literatür araştırması kapsamında bir modelin farklı şirketlerin açık kaynak üzerinde sunduğu aynı ölçek ve köşe koordinatlara sahip uydu görüntüleri ile eğitilerek mukayese edilmemesi ve bu problemin tez çalışması kapsamında değerlendirilerek analiz edilmesi tez çalışmasının özgün değerini oluşturmaktadır.

1.4. Tez Çalışmasının Yapısı

Tez çalışmasının içeriğinin oluşturulması sürecinde Eskişehir Teknik Üniversitesi Lisansüstü Tez Yazım Kılavuzu (Temmuz, 2018) referans olarak alınmıştır.

1.5. Tez Çalışmasının Akışı

Çalışmanın birinci bölümünde giriş yapılmış olup, ardından literatür araştırmasına yer verilmiş, problemin tanımlaması yapılmış, tez çalışmasının özgün değerinden bahsedilmiş ve sonrasında çalışmanın yapısı ve akışı hakkında bilgi verilmiştir.

Çalışmanın ikinci bölümünde yapay zekâ hakkında temel düzeyde bilgi verilmiş, tarihçesinden bahsedilmiştir. Aynı başlık altında literatürde yapay zekanın alt dalı olarak gösterilen makine öğrenmesi, makine öğrenmesinin yöntemleri, yapay sinir ağları, derin öğrenme ve derin öğrenmenin tarihçesi hakkında temel düzeyde bilgilere yer verilmiştir.

Çalışmanın üçüncü bölümünde derin öğrenmenin nesne tespiti alanında en sık kullanılan ve halen geliştirilmekte olan evrimsel sinir ağlarının katmanları, mevcut nesne tespit modelleri ve derin öğrenme ile nesne tespiti hakkında bilgi verilmiştir.

Çalışmanın dördüncü bölümünde tez çalışması kapsamında kullanılan program ve kütüphanelerden, çalışmanın yapılacağı sisteme uygun derin öğrenme modelinin seçiminden, modelin eğitilmesi sürecinde kullanılacak olan veri setlerinin özelliklerinden, modelin eğitime prensiplerinden ve modelin hiper parametrelerinden bahsedilmiş olup eğitim sonunda model performanslarının ve kayıp grafiklerinin karşılaştırılmasına yer verilmiştir.

Çalışmanın beşinci bölümünde eğitilmiş modeller açık kaynak üzerinden temin edilen 10 adet hava fotoğrafları ile test edilmiştir. Test edilen görüntüler çalışmanın beşinci bölümünde görsel olarak sunulmuştur.

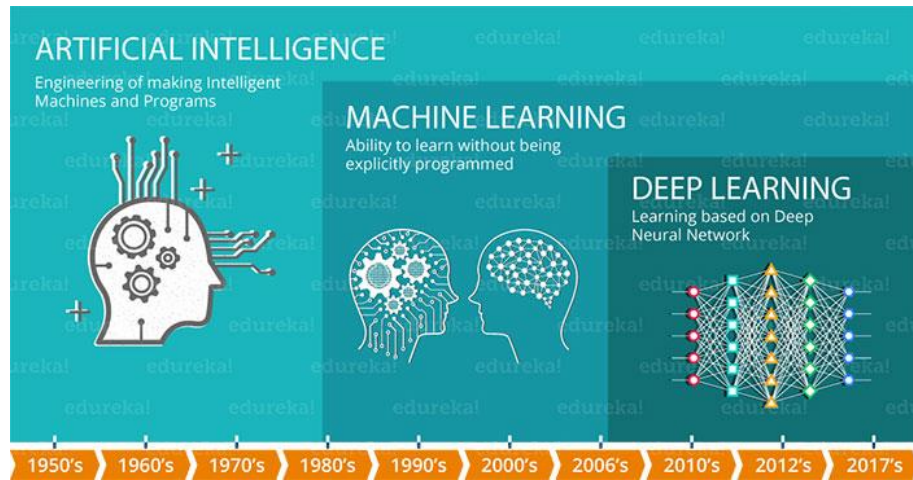
Çalışmanın son bölümü olan altıncı bölümde aynı veri seti ile eğitilen modellerin başarı oranları ve aynı modelin farklı veri setleriyle eğitilmesi sonucu veri setlerinin modelin başarı oranına etkisi incelenmiş olup sayısal veriler ve grafikler ile sonuçlar ortaya koyulmuştur.

2. YAPAY ZEKA

Yapay zeka, insan beynindeki nöronların etkileşimlerinden esinlenilerek öğrenme işleminin makinelere aktarılabilceği düşüncesiyle ortaya çıkmıştır. Birçok araştırmacı ve tarihçiye göre modern yapay zekanın başlangıcını 1956 yılında gerçekleştirilen Darmouth Konferansı olduğunu belirtmektedir. John McCarthy, Marvin Minsky, Nathaniel Rochester ve Claude Shannon, konferansta yapay zekayı, "akıllı makineler, özellikle de akıllı hesap programları yapmanın bilimi" olarak tanımlamışlardır (McCarthy vd., 1955). 1997 yılında IBM (International Business Machines) şirketinin Deep Blue isimli satranç bilgisayarı, dünyanın en iyi satranç oyuncusu Garry Kasparov'u yenmesi yapay zekayı daha popüler hale getirmiştir (King, 1997).

Özellikle son 10 yılda, ev otomasyonu (aydınlatma, ev eğlencesi, priz ve iklimlendirme sistemleri kontrolü), otonom araçlar, Google Asistan, Siri veya Alexa gibi sesli asistanlar, görüntü tanıma yazılımları, güvenlik ve dolandırıcılık karşıtı (anti-fraud) yazılımları, analitik araçlar, e-ticaretteki kişisel öneri/tavsiye araçları yapay zekanın gündelik hayatta ne kadar sık kullanıldığını göstermektedir.

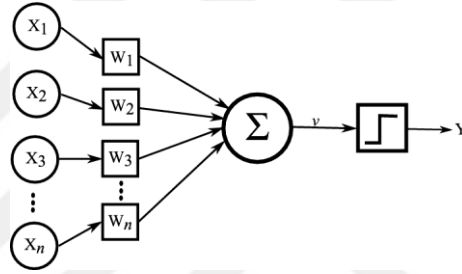
Yapay zekanın ortaya çıktığı süreçten itibaren makine öğrenmesinin gelişmesi daha sonra derin öğrenmenin ortaya çıkması kronolojik bir süreci kapsamaktadır. Bu bağlamda yapay zeka, makine öğrenmesi ve derin öğrenmenin kronolojik süreci Şekil 2.1'de gösterilmiştir.



Şekil 2.1. Yapay zeka, makine öğrenmesi ve derin öğrenmenin kronolojik süreci

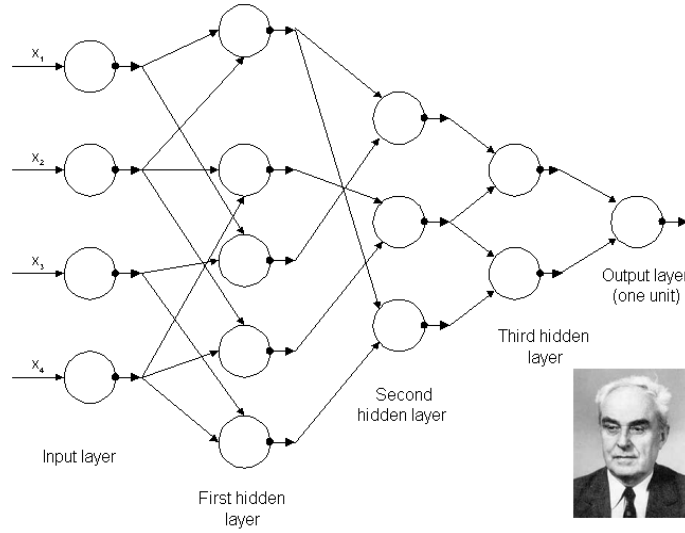
2.1. Yapay Zekanın Tarihçesi

Yapay zekanın tarihi, Warren McCulloch ve Walter Pitts'in 1943 yılında düşünce sürecini taklit etmek için matematiğe ve sinir mantığı olarak adlandırılan algoritmalara dayalı sinir ağları için bir hesaplama modeli oluşturmalarına uzanmaktadır (McCulloch ve Pitts, 1943). Bu model Yapay Sinir Ağları'nın (YSA) temelini oluşturmuştur. 1958 yılında Frank Rosenblatt basit toplama ve çıkarma işlemleri kullanarak iki katmanlı bir bilgisayar yapay sinir ağına dayalı, denetimli öğretimli bir desen tanıma olan perceptron algoritmasını oluşturmuştur (Rosenblatt, 1958). Rosenblatt'ın perceptron algoritması Şekil 2.2.'de gösterilmiştir.



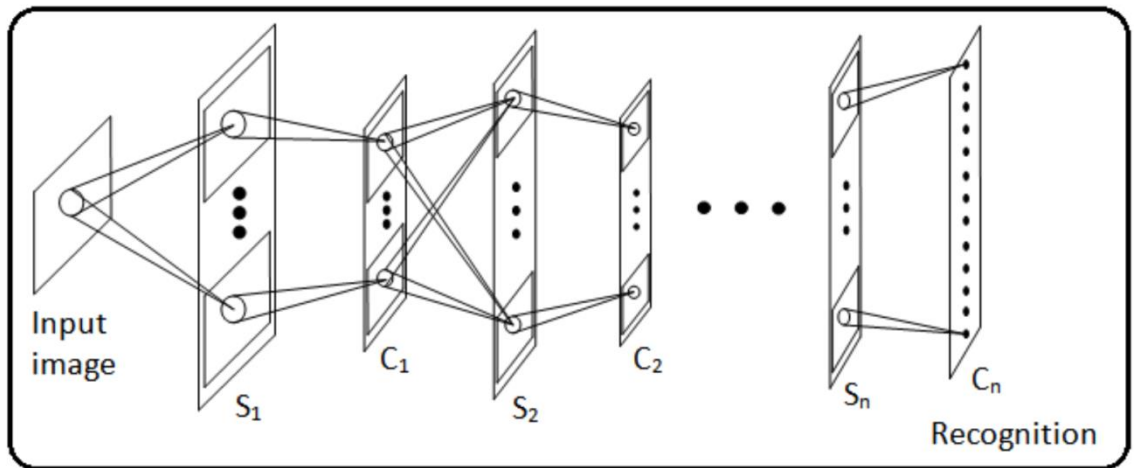
Şekil 2.2. Rosenblatt'ın perceptron algoritması

1966 yılında Ivakhnenko ve Lapa yapay zeka algoritmalarını geliştirmeye yönelik çalışmalarında karmaşık aktivasyon fonksiyonlarına sahip modelleri kullanmıştır (Ivakhnenko ve Lapa, 1966). Bu çalışmada her bir katman için istatistiksel yöntemler ile en iyi özellikler belirlenip ardından gelen katmana iletilmektedir. Ağlarını uçtan uca eğitmek için geri yayılım (backpropagation) kullanılmamış, önceki katmanlardan sonraki katmanlara en küçük kareler yöntemi kullanılmıştır (Şeker vd., 2017). Ivakhnenko tarafından eğitilen, bilinen ilk derin ağ mimarisi Şekil 2.3'te gösterilmiştir.



Şekil 2.3. *Ivakhnenko tarafından eğitilen, bilinen ilk derin ağ mimarisi (Şeker & Diri & Balık, 2017)*

1970’li yılların sonlarına doğru Fukushima görüntü işleme üzerine geliştirdiği “Neokognitron” modelini önermiştir. (Fukushima, 1979). Bu model, sonuçların kullanılabilirliği konusuna ağırlık veren pragmatist yaklaşım içeriyordu (Fukushima vd., 1983). Fukushima tarafından geliştirilen Neokognitron mimarisi Şekil 2.4’te gösterilmiştir.



Şekil 2.4. *Fukushima tarafından geliştirilen Neokognitron mimarisi*

1992 yılında Weng, Cohen ve Herniou karma sahnelerden otomatik olarak üç boyutlu nesne tanıma işlemini gerçekleştiren “Cresceptron” yöntemini yayımlamıştır (Weng vd., 1992).

1995 yılında Cortes ve Vapnik tarafından benzer verilere sahip iki grubun sınıflandırılması için destek vektör ağları kullanılmıştır (Cortes ve Vapnik, 1995).

1997 yılında Hochreiter ve Schmidhuber tarafından bilgiyi tekrarlayan geri yayılımla uzun süre boyunca saklamayı öğrenme için “Uzun Kısa-Sürekli Bellek (LSTM)” adlı bir model önerilmiştir (Hochreiter ve Schmidhuber, 1997). Bu model değerleri rastgele aralıklar ile hatırlayan bir sinir ağı mimarisidir.

Derin Öğrenme 2000’li yıllar itibariyle popülerlik kazanmaya başlamıştır. 2012 yılında Google’ın araştırma ekibi tarafından, 16000 işlemciden ve bir milyardan fazla bağlantıdan oluşan yapay desen tanıma algoritmalarının performansı, insan düzeyine ulaşmıştır (Nytimes, 2012).

2014 yılında Facebook, fotoğraflarda kullanıcılarını otomatik olarak etiketlemek için DeepFace adlı derin öğrenme teknolojisini kullanmıştır. Bu etiketleme için derin öğrenme algoritmaları, 120 milyon parametreyi hCNNba katarak yüz tanıma görevlerini gerçekleştirmiştir (Taigman vd., 2014).

2016 yılında Hinton ve Salakhutdinov çok katmanlı sinir ağının, bir kerede bir katmanın nasıl önceden eğitilebileceğini gösteren çalışmalar yapmıştır (Hinton ve Salakhutdinov, 2016).

Google DeepMind şirketi tarafından geliştirilen CNN tabanlı AlphaGo algoritması ile Güney Koreli Lee Sedol arasında 2016 yılında oynanan Go oyununda, AlphaGo rakibi Lee Sedol’u 4-1 yenerek ilk kez profesyonel oyuncuyu yenebilen program olmuştur (Lee vd., 2016).

2017 yılında Facebook çeviri üzerine inşa ettiği Caffe2 derin öğrenme kütüphanesini kamuoyunun kullanımına açmıştır. Caffe2'nin yinelemeli yapay sinir ağları sayesinde milyarlarca veri anlaşılmıştır. Bu model ile makineler tarafından çevrilen insan dilinin performansını ölçen bir metrik olan BLEU'nun performansı yaklaşık yüzde 11 artmıştır.

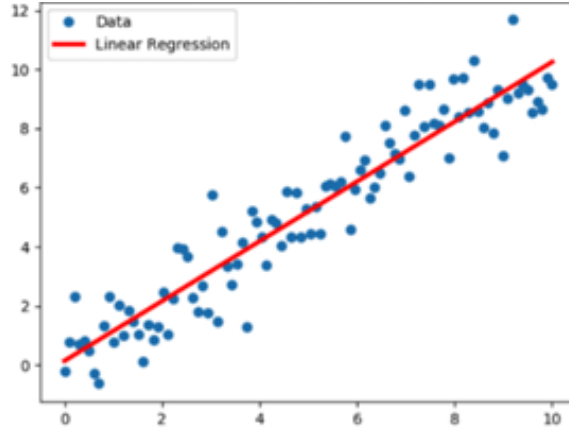
2.2. Makine Öğrenmesi

Makine öğrenmesi, insan beyinde yer alan nöronların biyolojik algoritmalarının işlevlerinden yola çıkarak bir probleme ait verinin matematiksel işlemlerle oluşturulduğu bilgisayar algoritmalarıdır. Makine öğrenmesinde öğrenim teknikleri üç ayrı grupta toplanmaktadır. Bu gruplar ‘gözetimli (supervised) öğrenme’, ‘gözetimsiz (unsupervised) öğrenme’ ve ‘takviyeli (reinforcement) öğrenmedir. Gözetimli öğrenme ile geliştirilen model, etiketlenen veri ile ulaşılması istenen değer arasındaki etkileşimin öğrenilmesi prensibine dayanmaktadır. Model eğitiminin sonucunun istenilen değere yakın olması amaçlanmaktadır. Gözetimsiz öğrenme tekniğinde etiketlenmemiş veri girdisi yoluyla bilinmeyen bir objeyi tespit edebilmek temel amaçtır. Bu yöntemde girdi verileri arasında değer bazında birbirine en yakın olanlar kendi içlerinde sınıflandırılır. Sonrasında gelen veriler ise kendi değerine en yakın olan sınıfın içerisinde yer alır. Takviyeli öğrenme tekniği ise çevresiyle etkileşim içerisinde ve karar mekanizmasına sahip bir modelin, amacına ulaşmasında doğru ve hatasız nasıl karar alabileceğini geliştiren bir yöntemdir. Bu yöntem, çıktı verisinin girdi verisine karşı doğruluğunu veya yanlışlığını sorgulayan kritere sahiptir (Atalay ve Çelik, 2017).

2.2.1. Gözetimli makine öğrenme stratejileri

2.2.1.1. Doğrusal regresyon analizi

Basit Doğrusal Regresyon, bir bağımsız değişkenle bir bağımlı değişken (tahmin edilen) arasındaki ilişkinin doğrusal olarak açıklanmasıdır. Eğer ilişkisi ölçülen değişkenler birden fazlaysa Çoklu Doğrusal Regresyon, değişkenlerin aralarındaki ilişki bir doğru değil de bir polinomsa Polinomal Regresyon denir (Kılıç, 2013). Doğrusal regresyon grafiği Şekil 2.5’te gösterilmiştir.

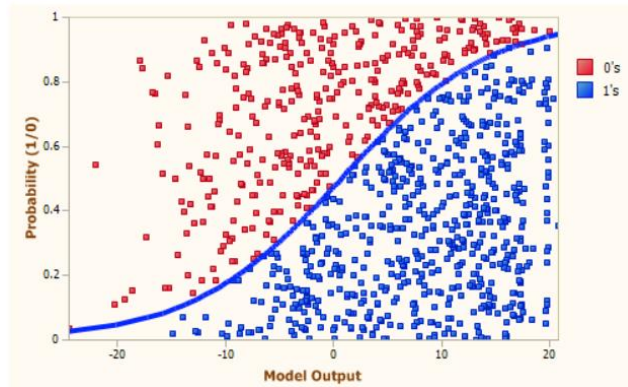


Şekil 2.5. Doğrusal regresyon grafiği

Doğrusal regresyon analizinde girdi verileri k değeri ile, çıktı verileri m değeri ile ifade edildiğinde denklem; $m=a * k + b$ şeklinde ortaya çıkmaktadır.

2.2.1.2. Lojistik regresyon analizi

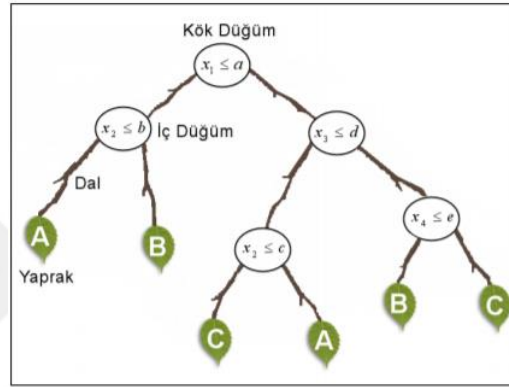
Lojistik regresyon analizi, probleme ait herhangi bir sonuç belirlemek üzere ikili (binary) ya da çoklu karakteristiği mevcut bağımlı değişken ile bir ya da birden fazla bağımsız değişkenin arasındaki ilişkiyi belli olasılık hesaplarına dayandırarak sınıflandırma yapmaya olanak sağlayan bir yöntemdir. Bu analizin kullanım amacı, değişkenleri minimum kullanarak maksimum uyum sağlayacak şekilde bağımlı değişken ve bağımsız değişkenlerin birbirleri ile olan ilişkisini araştırmaya yönelik modeli geliştirmektir (Bircan, 2004). İki seviyeli sınıflandırılmış verilerden meydana gelen lojistik regresyon grafiği Şekil 2.6.'da gösterilmiştir.



Şekil 2.6. İki düzeyli kategorik verilerden oluşan lojistik regresyon grafiği (Kavuncu, 2018)

2.2.1.3. Sınıflandırma ve regresyon ağaçları tekniği (SRAT)

Sınıflandırma ve Regresyon Ağaçları Tekniği (SRAT), Karar Ağaçlarının farklı bir versiyonudur. Her bir karar düğümünün iki dala ayrılması ilkesine dayanmaktadır. Bu tip karar ağaçlarında ikili dallanmalar söz konusudur (Kavzoğlu ve Çölkesen, 2010). SRAT şeması Şekil 2.7.'de gösterilmiştir.



Şekil 2.7. Sınıflandırma ve regresyon ağaçları tekniği şeması (Kavzoğlu & Çölkesen, 2010)

Şekil 2.7.'de görüldüğü üzere en tepedeki düğüm bölünmenin başladığı kök düğüm, alt bağlantıda yer alan düğümler iç düğüm ve yeşil renk ile gösterilenler ise yaprak (terminal) düğümlerdir.

2.2.1.4. Naive Bayes sınıflandırıcı

Naive Bayes sınıflandırıcısının çalışma prensibi, her durum için olasılıklar hesaplanır ve olasılık değeri en yüksek olana göre sınıflandırılır. Gerçek zamanlı tahmin, metin sınıflandırması ve öneri sistemleri başta olmak üzere birçok alanda kullanılabilir. Örnek olarak bir futbol müsabakasında hangi tarafın galip geleceği ile ilgili tahmin yapılırken maçın ne zaman kimin ev sahipliğinde oynanacağı, hava durumu ve seyirci sayısı gibi değişkenlere bağlı olarak tahmin edilebilir.

$P(F|C)$: C olayının oluştuğu anda, F olayının oluşma olasılığı

$P(C|F)$: F olayının oluştuğu anda, C olayının oluşma olasılığı

$P(F)$: F olayının oluşma olasılığı

$P(C)$: C olayının oluşma olasılığı

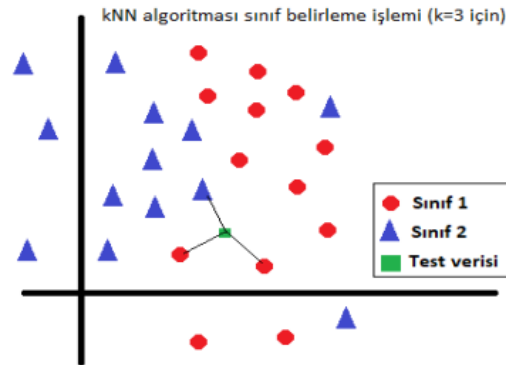
C harfi hedefi ve F harfi değişkenleri ifade etmektedir. Naive Bayes sınıflandırıcı kısaca tüm koşullu olasılıkların çarpımı olarak tanımlanabilmektedir. Denklem eşitlik 2.1’de gösterilmiştir.

$$p(C|F_1, \dots, F_n) = \frac{P(C)p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)} \quad (2.1)$$

2.2.1.5. En yakın komşu algoritmaları (KNN)

K-En en yakın komşu algoritması, Fix ve Hodges tarafından önerilen, makine öğrenmesi yöntemlerindendir (Fix ve Hodges, 1951). K-En en yakın komşu algoritmasında sınıflandırma veriler arasındaki bağlantılar göz önünde bulundurularak yapılmaktadır. Bu algoritma doğrusal ayrıştırma yolu ile koordinat düzleminde çalışmaktadır. Sınıflandırılacak olan obje, kendine en yakın k tane komşu veriyi baz alarak yoğunluk miktarına göre ilgili sınıfa atanır (Wu vd., 2002). Sonraki adımda ise komşu elemanların uzaklık mesafeleri belirlenmektedir. Komşu verilere ait uzaklıklar sıralanır ve test verisine en uygun olan sınıf seçilmektedir (Maxwell vd., 2018).

k=3 olduğu bir durum için örnek bir sınıf belirleme işlemi Şekil 2.8’de gösterilmiştir.

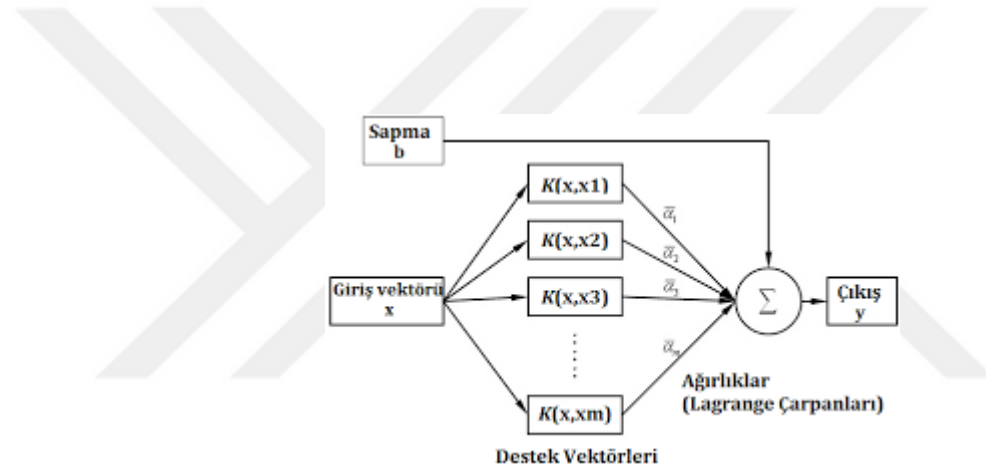


Şekil 2.8. En yakın komşu algoritması sınıf belirleme işlemi

2.2.1.6. Destek Vektör Makineleri (DVM)

Destek vektör makineleri (DVM), 1963 senesinde temeli Vladimir Vapnik tarafından atılan, sınıflandırma için kullanılan ve en küçük yapısal risk prensibiyle oluşturulan denetimli bir algoritma yöntemidir (Soman vd., 2011).

Bu algoritma, veri girdisine ilişkin birleşik dağılım fonksiyona ihtiyaç duymaması sebebiyle dağılımdan ayrı bağımsız öğrenme algoritmalarıdır (Soman vd., 2011). DVM, az sayıda eğitim verisi ve çok sayıda özellik olduğunda bile sınıflandırma ve regresyon problemlerinde iyi performans gösterir. Ayrıca kullanılan veri sayısının bir üst sınırı yoktur; ultra büyük veri setleri için de uygundur (Zigic ve Kecman, 2014). DVM yapısı Şekil 2.9'da gösterilmiştir.



Şekil 2.9. DVM yapısı (Ayhan & Erdoğan, 2014)

2.2.2. Gözetimsiz makine öğrenme stratejileri

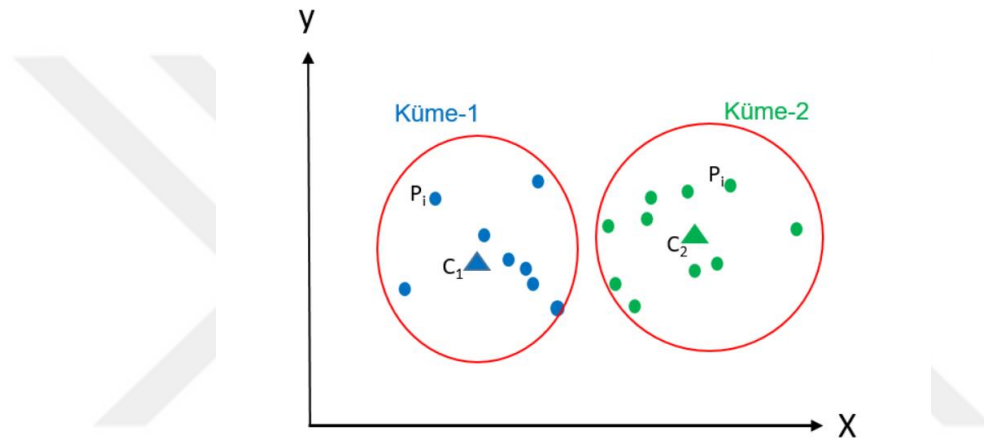
2.2.2.1. K-ortalamar (K-Means) kümeleme algoritması

K-ortalama kümeleme algoritması J.B. MacQueen tarafından 1967 yılında geliştirilmiştir (MacQueen, 1967). Algoritmada kümeleme işlemi, girişte verilen merkez sayısına, n adet veri nesnesinden oluşan bir veri kümesini bölerek yapılmaktadır. Veri setinde yer alan veriler sadece bir kümeye ait olmaktadır, küme kesişimine izin verilmemektedir (Sarıman, 2011). K-ortalamar algoritmasında işlem adımları aşağıda yer alan maddelerde sırası ile belirtilmiştir.

- Rassal olarak k adet küme merkez noktaları belirlenir.
- Belirlenen küme merkez noktaları ile verilerin arasındaki mesafeler hesaplanır.

- Veriler kendilerine en yakın mesafede olan kümelere atanır.
- Ardından yukarıda yer alan işlemler tekrarlanır.
- Verilerin küme değiştirmeyi bitirdiği durumda algoritma işlemi durdurur (Rao ve Srivinas, 2006).

İki kümeye ayrıştırılmış örnek bir K-ortalamlar şeması Şekil 2.10'da gösterilmiştir.



Şekil 2.10. K-ortalamlar (Means) algoritması

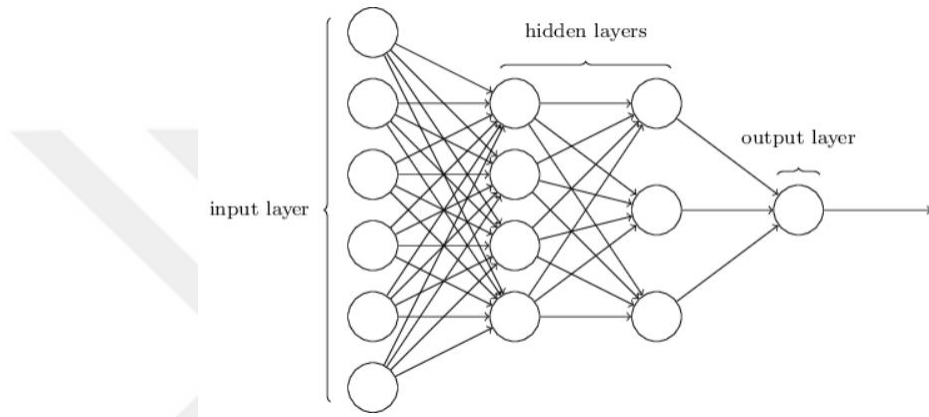
2.2.2.2. Hiyerarşik kümeleme

Hiyerarşik kümeleme yöntemi özelliklerin benzerliklerini baz alarak; tespitleri veya değişkenleri kümelemek amacıyla uygun mesafe veya benzerlik ölçülerini dikkate alır (Alpar, 2011).

Denetimsiz bir öğrenme modeli olan hiyerarşik kümeleme yönteminde kaç adet küme oluşturulacağı önceden bilinmez. Hiyerarşik kümelemede algoritma iki farklı yöntemle çalışmaktadır. Başlangıç aşamasında her bir verinin ayrı bir küme olarak başlayıp kendine en yakın mesafede olan kümeye atanarak devam ettiği işlemler dizisi birleştirici (yığılmacı) hiyerarşik kümeleme, başlangıç aşamasında tüm verileri tek bir küme olarak başlatıp en uzak mesafeleri hesaplayarak kümelere ayrıştırdığı yöntemle ayrıştırıcı (bölücü) hiyerarşik kümeleme adı verilmektedir (Yeşilbudak vd., 2011).

2.3. Yapay Sinir Ağları

Birçok derin öğrenme yönteminde, Yapay Sinir Ağları (YSA) adı verilen mimariler kullanılmaktadır. Bu ağlar, girdi (input), gizli (hidden) ve çıktı (output) katmanlarından oluşmaktadır. Ağın öğrenme yeteneği üzerinde en fazla etkiye sahip olan katman gizli katmandır (Karakaya ve Çemtay, 2017). Yapay sinir ağlarının genel şematik yapısı Şekil 2.11’de gösterilmiştir.



Şekil 2.11. Yapay sinir ağlarının mimarisi

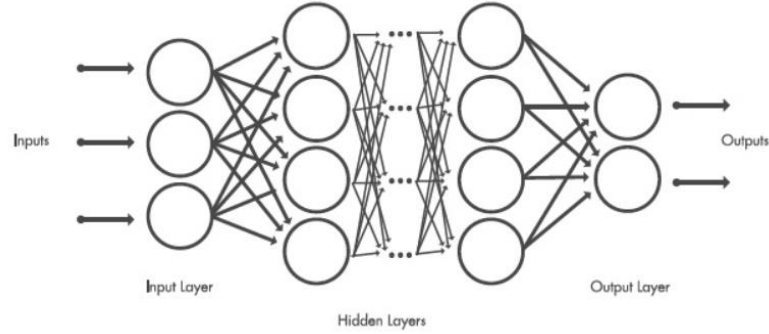
2.4. Derin Öğrenme

Derin öğrenme birçok farklı alanda çok sayıda katmana sahip yapay sinir ağlarından oluşan yapay zeka yöntemlerinden biridir. Literatürde makine öğrenmesinin bir alt dalı olarak kabul edilen derin öğrenme, veri setleriyle çıktılar üzerinde tahminde bulunacak olan yapay sinir ağlarının eğitilmesini mümkün kılmaktadır. (Doğan vd., 2019) Derin öğrenmenin temel mimari yapısı bir girdi katmanı ile bir çıktı katmanı arasında bulunan çok sayıda gizli katmandan oluşmaktadır (Ker vd., 2017).

Derin öğrenmenin sıklıkla kullanıldığı alanlar; yüz tanıma, ses tanıma, sürücüsüz araç kullanma, savunma sanayi, güvenlik sanayi ve sağlık sektörü başta olmak üzere birçok alanı kapsamaktadır. Yüz tanıma örneği olarak Facebook, fotoğraflarda etiketleme yapılabilmesi amacıyla DeepFace adlı derin öğrenme teknolojisini kullanmıştır. Ses tanıma yaygın olarak kullanıldığı sektör bankacılık sektörüdür. Savunma ve güvenlik sektöründe kamera kaydı sırasında sıra dışı bir faaliyet gerçekleştiğinde derin öğrenme sayesinde kameranın alarm vermesi bir örnek teşkil

etmektedir. Sağlık sektöründe ise günümüzde önemli bir örnek, derin öğrenme ile kanser tespiti yapılabilmesidir.

Temel olarak giriş katmanı, çok sayıda gizli katman ve çıkış katmanından oluşan derin öğrenme mimarisi Şekil 2.12’de gösterilmiştir.



Şekil 2.12. Derin öğrenme mimarisi

Derin öğrenme algoritmalarında yapay sinir ağları denetimli ya da denetimsiz olarak eğitilebilmektedir. Algoritma yapısı verilerin özelliklerinin temsil değerlerinin öğrenilmesi temeline dayanmaktadır. Alt seviyedeki özelliklerin belirli hiyerarşik işlemlerden geçirilmesi ile üst seviyede özelliklerin ortaya çıkarılması sağlanmaktadır.

Yapay sinir ağlarını temel alan ve literatürde makine öğrenmesinin bir alt dalı olarak kabul edilen derin öğrenmeyi geçmişteki modellerden ayıran unsur girdi ve çıktı katmanları arasında yer alan çok sayıda gizli katmanlardır. Derin öğrenmenin sağladığı katkılar şunlardır;

- Özelliklerin otomatik olarak çıkarılması ve istenen sonuç için en uygun şekilde ayarlanması derin öğrenmenin bir avantajıdır. Özelliklerin önceden çıkarılması gerekmez. Bu yönüyle, zaman alıcı makine öğrenimi tekniklerinden ayrılır.

- Verilerdeki doğal varyasyonlara karşı dayanıklılık otomatik olarak öğrenilir.

- Aynı sinir ağı tabanlı yaklaşım birçok farklı uygulamaya ve veri türüne uygulanabilir.

- Grafik işlemci üniteleri (GPU) kullanılarak çok büyük paralel hesaplamalar gerçekleştirilebilir ve büyük hacimli veriler için ölçeklenebilir. Ayrıca, veri miktarı çok büyük olduğunda daha iyi performans sonuçları sağlar.

- Derin öğrenme mimarisi, gelecekteki yeni sorunlara uyarlanmak için esnektir.

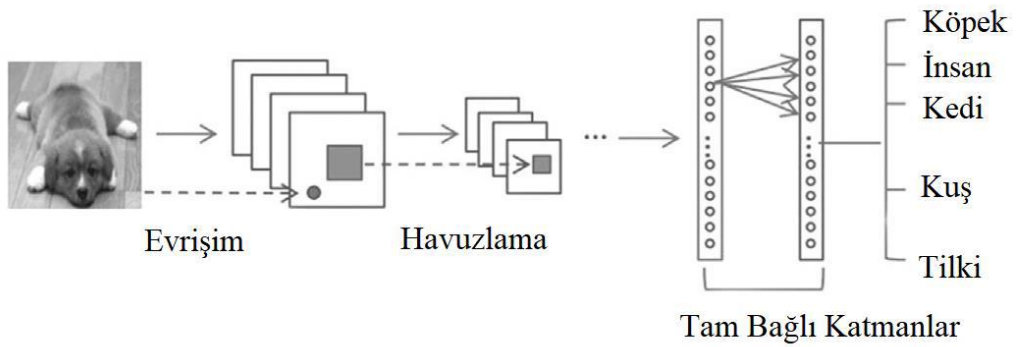


3. EVRİŞİMLİ SİNİR AĞLARI

Derin öğrenme kavramı temelde çok katmanlı yapay sinir ağlarını ifade etmektedir. Derin öğrenme kapsamında bilinen en popüler yapı evrişimsel sinir ağlarıdır. Evrişimsel sinir ağlarının (ESA) en önemli özelliği yapay sinir ağındaki parametre sayısını azaltmasıdır. Bir evrişimsel sinir ağı uygulamada görüntünün tamamını kullanmadan amaca ulaşabilmektedir. Farklı ebatlarda filtrelerin kullanılmasıyla her bir katmanda evrişim işlemi uygulayarak parametre sayısını yüksek oranda azaltabilmektedir (Murat, 2021).

Evrişimli Sinir Ağları görüntü sınıflandırmasında sıklıkla kullanılan başarılı bir derin sinir ağıdır. Çok katmanlı bir yapay sinir ağı modeli olan ESA'nın geliştirilme maksadı dijital görüntü ve videolardan çıkarım sağlamaktır. İnsan beyninin yapısından esinlenerek tasarlanmış, görüntüye ait piksellerin değerlerine göre özellik çıkarma ve sınıflandırmayı sağlayan ileri beslemeli yapay sinir ağı modelidir (Andrew, 2018).

ESA yapısal olarak üç katmandan oluşmaktadır. Bunlar her biri farklı görevler için tasarlanmış evrişim, havuzlama ve tam bağlı katmanlar olarak tanımlanmaktadır. Dijital görüntüler veya videolar belirtilen bu katmanlar ile çeşitli işlemlerden geçmektedir. ESA'ların kullanıldığı alanlar; objelerin tespiti, görüntülerin sınıflandırılması gibi problemlerdir. ESA'nın ağ yapısı Şekil 3.1.'de gösterilmiştir.

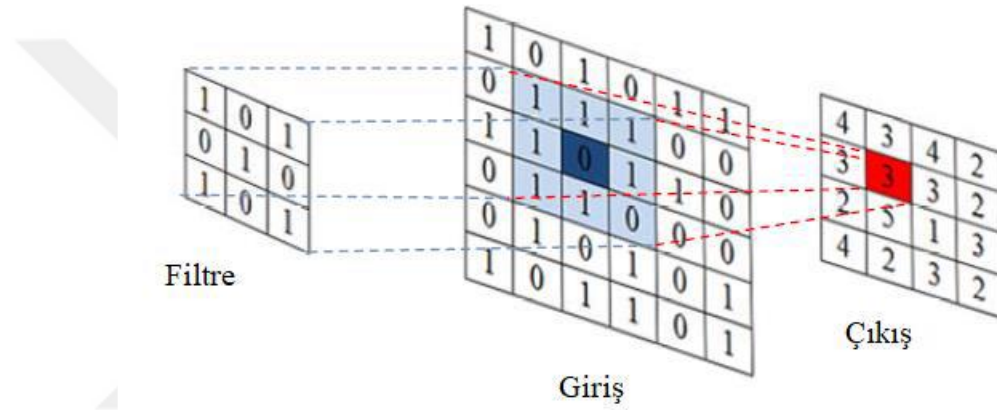


Şekil 3.1. Evrişimli sinir ağı yapısı (Zhang vd., 2019)

3.1. Evrişimli Sinir Ağları'nın Katmanları

3.1.1. Evrişim Katmanı

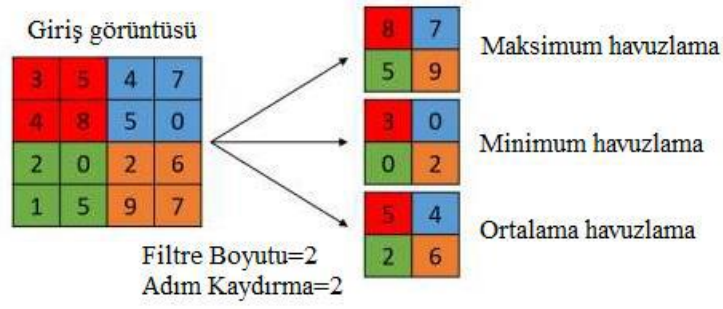
Evrişim, ESA algoritmalarında veri üzerinde ilk işlem yapan katmandır. Belirli bir filtre, görüntü üzerinde dolaşarak görüntü üzerinden özellik çıkarımı sağlamaktadır. Çıkarılan özellikler sayesinde girdi veri matrisinden daha küçük boyuta sahip yeni veri matrisi oluşmaktadır (Salur vd., 2019). Evrişim katmanının mimarisi, Şekil 3.2'de gösterilmiştir.



Şekil 3.2. Evrişim işlemi (Liu vd, 2016)

3.1.2. Havuzlama Katmanı

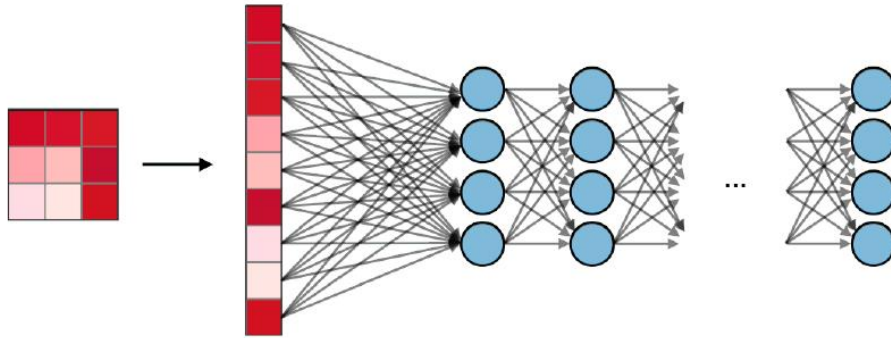
Havuzlama katmanı sıklıkla evrişim katmanından sonra kullanılmaktadır. Havuzlama katmanında, evrişim katmanında olduğu gibi filtreler görüntü üzerinde gezdirilir. Kendinden sonra gelen evrişim katmanı için giriş boyutunu azaltmaktadır. Filtreler görüntü üzerinde belli adımlarla gezdirilerek genellikle piksellerin maksimum değerlerini (maksimum havuzlama) alarak işlem yapar. Havuzlama katmanında amaç hesaplamayı hızlandırmak için verinin boyutunu küçültmek ve özelliklerini kaybetmemesini sağlamaktır. Örnek bir havuzlama işlemi Şekil 3.3'te gösterilmiştir.



Şekil 3.3. Havuzlama işlemi (Türkoğlu vd., 2020)

3.1.3. Tam Bağlı Katman

Uygulamalarda yer alan veri setlerinin boyutu çok yüksek olabilmektedir. Tam bağlı katman, kendinden önce gerçekleşen bir veya birden çok evrişim ve havuzlama katmanlarından sonra matris halindeki görüntüleri düz vektör haline getirmektedir (Doğan ve Türkoğlu, 2018). Bu katmanın amacı evrişim ve havuzlama katmanlarıyla çıkarılan özniteliklerin sınıflandırmasını yapmaktır. Tam bağlı katman yapısı Şekil 3.4.'te gösterilmiştir.



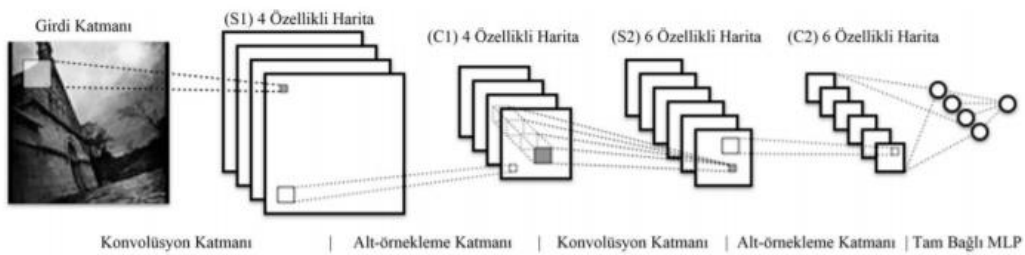
Şekil 3.4. Tam bağlı katman yapısı

3.2. Evrişimli Sinir Ağı Modelleri

Uzun süre önemini kaybetmiş olan derin öğrenme 2012 yılında düzenlenen ImageNet yarışmasıyla yeniden popülerlik kazanmaya başlamıştır. LeNet-5, AlexNet, VGG, ResNet ve GoogleNet ön plana çıkan evrişimli sinir ağı modelleridir.

3.2.1. Lenet-5

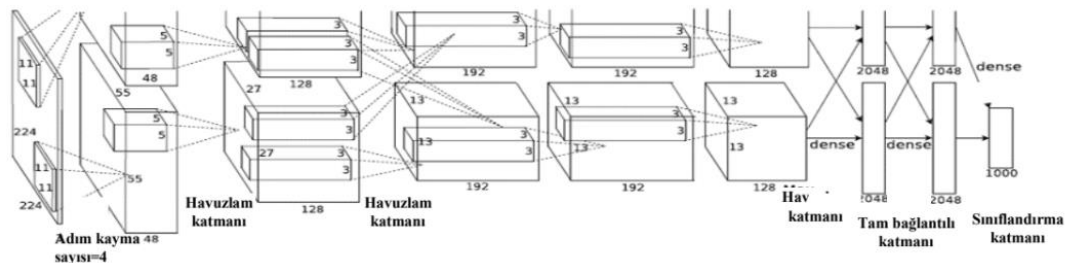
1998 yılında Yann Le Cun ve arkadaşları tarafından yayımlanan ve ilk defa başarılı sonuç veren evrişimli sinir ağı modelidir (Lecun vd., 1998). Bu modelin sonradan geliştirilecek modellerden farkı veri boyutunun küçültülmesi aşamasında max-pooling işlemi yerine ortalama (avarage) pooling işlemi yapıyor olmasıdır. LeNet-5 mimarisi Şekil 3.5’te gösterilmiştir.



Şekil 3.5. *LeNet ağıının mimarisi (Şeker & Diri & Balık, 2017)*

3.2.2. AlexNet

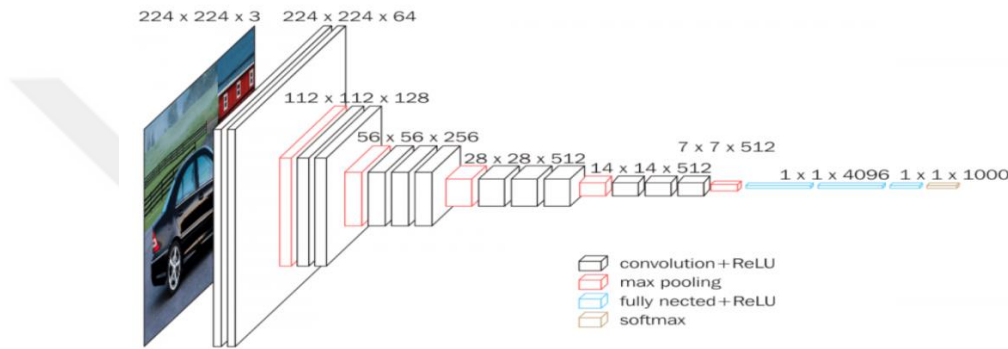
LeNet mimarisi baz alınarak 2012 yılında Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton tarafından geliştirilmiş ve aynı yıl içerisinde düzenlenen ILSVRC ImageNet yarışmasını kazanan model olmuştur. Bu model ile dijital ortamda obje tanıma hata oranı %26,2'den %15,4'e düşürülmüştür (Krizhevsky vd., 2012). AlexNet mimari yapısı Şekil 3.6'da gösterilmiştir.



Şekil 3.6. AlexNet mimarisi (Krizhevsky, Sutskever et al. 2012)

3.2.3. VGGNet

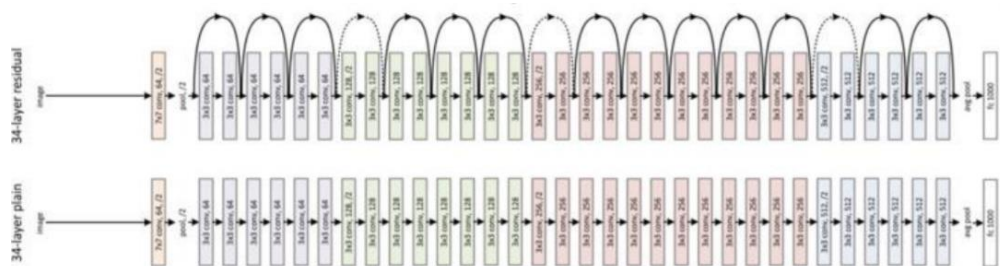
Ağ modeli, daha önce oluşturulmuş olan modellere göre basittir ancak önemli bir farkı evrişim katmanları ikili veya üçlü kullanılmasıdır. 2014 yılında düzenlenen ILSVRC yarışmasında ise derin ESA modeli olan VGG16 modeli yerleştirmede birinci, sınıflandırma alanında ise ikinci olmuştur. ImageNet verileriyle eğitilerek test edilmiş olan bu model ilk 5 sınıflandırmada %92.7 tutarlılığa ulaşmıştır (Simonyan vd., 2014). VGG modelinin mimarisi Şekil 3.7’de gösterilmiştir.



Şekil 3.7. VGG mimarisi

3.2.4. Microsoft ResNet

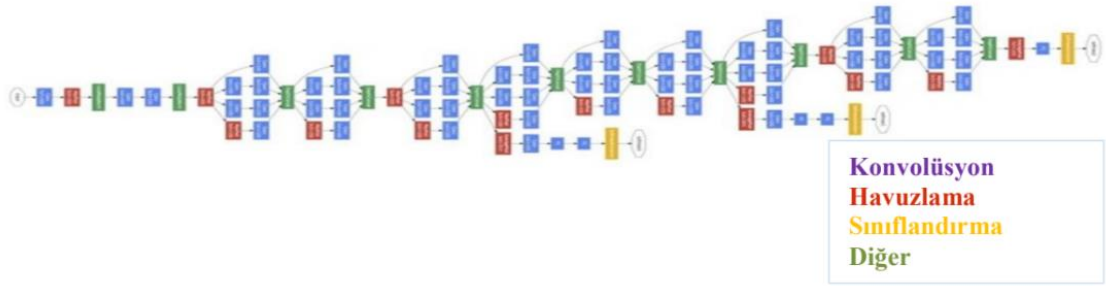
ResNet daha önce tasarlanmış algoritmalarla göre daha derin tasarlanmıştır. Toplamda 152 katmandan oluşmaktadır. 2015 yılında düzenlenen ImageNet yarışmasını %3.6 gibi bir hata oranı ile kazanmıştır. Model, hata oranını azaltmak için çok sayıda katman içeren artık bloklar kullanmaktadır (He vd., 2015). ResNet modelinin yapısı Şekil 3.8.’de gösterilmiştir.



Şekil 3.8. Microsoft ResNet ilk 34 katmanının ağ mimarisi (He vd. 2016)

3.2.5. GoogleNet

GoogleNet diğer modellere göre daha karmaşık bir yapıya sahiptir. Inception modüllerinden meydana gelmektedir. Her modül birbirinden farklı boyutlara sahip evrişim ve maksimum havuzlama işlemlerinden oluşmaktadır. Bu ağ modelinin diğerlerinden farkı giriş resmine uygulanan filtre ve işlemlerin sıralı bir şekilde değil aynı anda olmasıdır. GoogleNet 2014 yılında düzenlenen ImageNet yarışmasını %5,7 hata oranı ile kazanmıştır. GoogleNet modelinin mimarisi Şekil 3.9.'da gösterilmiştir.

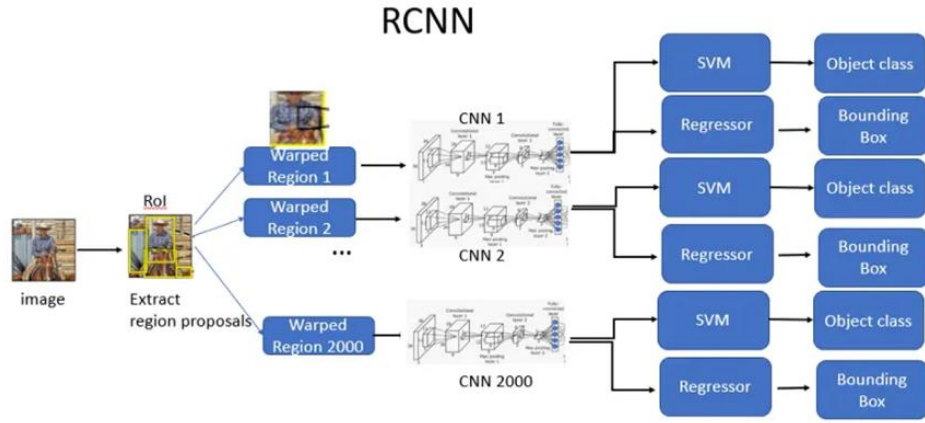


Şekil 3.9. GoogLeNet ağ mimarisi (Szegedy vd., 2015)

3.3. Derin Öğrenme ile Obje Tespiti

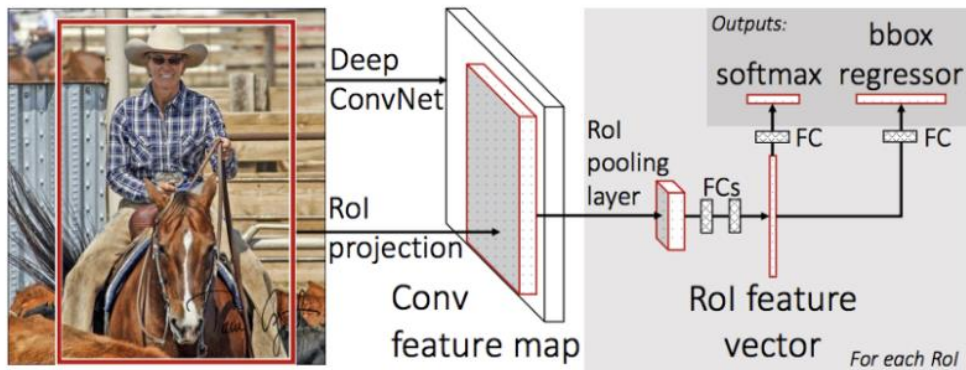
ESA algoritmaları yüz tanıma, görüntü sınıflandırma, video işleme ve nesne tespiti gibi alanlarda kullanılabilmektedir. Ancak ESA'nın en büyük eksikliği nesne tespitinde direk olarak kullanılamamalarıdır; ESA yalın haliyle bir görüntüyü ancak sınıflandırma yeteneğindedir. Bu sebepten dolayı ESA'ları nesne tespitinde kullanmak üzere R-CNN modeli geliştirilmiştir.

R-CNN (Region-based Convolutional Neural Network) algoritmasında girdi verisi üzerinden seçici arama (selective search) ile görüntü üzerinde nesne olma ihtimali olan 2000 adet bölge önerisi oluşturmaktadır. Oluşturulan bölge önerilerinin boyutlarını CNN'den (ConvNet) geçebilmesi için önerilen bölgelerin boyutlarının eşitlenmesi gerekmektedir. CNN'den çıkan öznetelik verileri kullanılarak lineer regresyon ile nesnenin sınırları, destek vektör makinaları (DVM) ile nesnenin ait olduğu sınıf belirlenmektedir. R-CNN mimarisi Şekil 3.10'da gösterilmiştir.



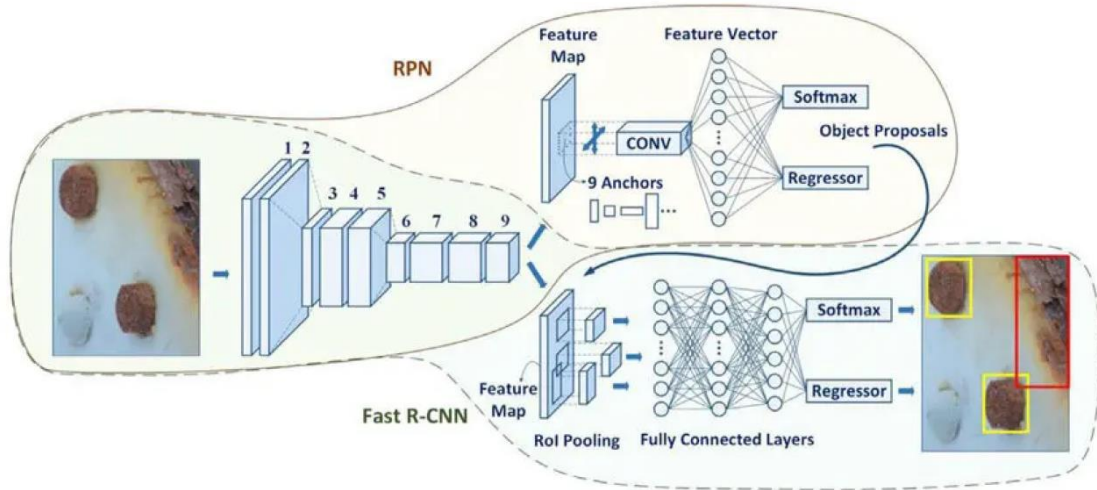
Şekil 3.10. R-CNN mimarisi

R-CNN algoritmasındaki yavaşlığın giderilmesi için Fast R-CNN algoritması geliştirilmiştir. Fast R-CNN algoritmasında görülen en büyük gelişme DVM ve regresörü kombinlemesidir. Bu yöntemde R-CNN’de olduğu gibi bölge önerisi yerine giriş resmi direkt olarak ESA’dan geçirilerek Conv5 özellik haritası oluşturulmaktadır. Özellik haritası üzerinden seçici arama (selective search) ile bölge önerisi çıkarılmaktadır. Böylece giriş resminin direkt ESA’dan geçirilmesiyle zaman kazanılmaktadır. Sonraki aşamada bölge önerilerinin boyutları eşitlenmektedir. Bölgeler tam bağlı katmandan geçirilir ve nesneler Softmax ile sınıflandırılır. Fast R-CNN mimarisi Şekil 3.11’de gösterilmiştir.



Şekil 3.11. Fast R-CNN mimarisi (Girshick, 2015)

Faster R-CNN’de ise modelin hızının azalmasına sebep olan seçici arama (selective search) ile bölge önerisi almak yerine bölge önerisi sağlayan ve evrişimli bir sinir ağı olan Region Proposal Network (RPN) kullanılmaktadır. Faster R-CNN modelinin mimarisi iki aşamada incelenmektedir. Birinci aşamada RPN, evrişim katmanında oluşturulan öznitelik haritası üzerinde nesne skoruna göre nesne olabilecek sınırları belirleyerek nesne teklifi ortaya çıkarmaktadır. İkinci aşamada ise RPN ile ortaya çıkan sonuçlar Fast R-CNN algoritmasına dahil edilmektedir. Fast R-CNN algoritmasının çıktı olarak ortaya çıkan iki katmandan birisi olan Softmax sınıflayıcı katmanı ile nesnenin ait olduğu sınıf tahmin edilmekte, ikinci katman olan regresyon katmanında ise nesne kutucuğunun doğruluğunun doğruluğu belirlenmektedir (Cömert vd., 2019). Faster R-CNN mimarisi Şekil 3.12’de gösterilmiştir.



Şekil 3.12. Faster R-CNN mimarisi (Ren vd., 2017)

4. STADYUM TESPİT METODU

Tez çalışması kapsamında veri setinin stadyum görüntülerinden oluşturulmasında ki en önemli etken savaş, doğal afet vb. gibi olumsuz durumlarda helikopter iniş/kalkışına ve acil toplanma alanı olarak kullanılmasına elverişli olmasıdır.

4.1. Kullanılan Programlar ve Kütüphaneler

Stadyum tespiti için “Python 3.6” programlama dili kullanılmıştır. Python programlama dili için geliştirme ortamı olarak “Anaconda Prompt” tercih edilmiştir. Derin öğrenme kütüphanesi olarak Google Brain tarafından geliştirilmiş Tensorflow kütüphanesi, görüntülerin işlenmesi ve gösterilmesi maksadıyla bilgisayarlı görü kütüphanesi olarak OpenCV ve nesne tespit modeli olarak Tensorflow Object Detection Application Programming Interface (API) (Tensorflow Nesne Algılama Uygulama Programlama Arabirimi) kullanılmıştır. Tensorflow Object Detection API, Google tarafından nesne tanıma maksadıyla oluşturulan, önceden eğitilmiş modelleri kullanıcıya sunan araçtır. Bu modeller COCO (Microsoft Common Objects in Context) veri seti ile eğitilmiştir (Tensorflow Object Detection API, 2019). COCO veri seti 91 sınıftan oluşmakta olup bunlardan 85 tanesi etiketlenmiş 5 binden fazla veriye sahiptir. Toplamda 328 bin resimde yaklaşık olarak 2 milyon etiketli veri bulunmaktadır.

4.2. Model Seçimi

Tablo 4.1’de Tensorflow Object Detection Zoo (2019) listesinde COCO veri seti ile eğitilmiş modeller yer almaktadır. Speed (ms) sütununda bulunan değerler modellerin nesne tespitinde ms cinsinden hızlarını, COCO mAP[^1] sütununda bulunan değerler ise modellerin COCO veri setindeki nesneleri tanıma başarısını ifade etmektedir. Speed (ms) değeri modelin işlem süresi ile doğru orantılıdır. COCO mAP[^1] değerinin yüksek olması nesne tespitinde başarı oranının yüksek olduğunu ifade etmektedir.

Tablo 4.1. COCO veri seti ile eğitilmiş modeller

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco	26	18	Boxes
ssd_mobilenet_v1_quantized_coco	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco	29	16	Boxes
ssd_mobilenet_v1_ppn_coco	26	20	Boxes
ssd_mobilenet_v1_fpn_coco	56	32	Boxes
ssd_resnet_50_fpn_coco	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes
faster_rcnn_resnet101_lowproposals_coco	82		Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	620	37	Boxes
faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241		Boxes
faster_rcnn_nas	1833	43	Boxes
faster_rcnn_nas_lowproposals_coco	540		Boxes
mask_rcnn_inception_resnet_v2_atrous_coco	771	36	Masks
mask_rcnn_inception_v2_coco	79	25	Masks
mask_rcnn_resnet101_atrous_coco	470	33	Masks
mask_rcnn_resnet50_atrous_coco	343	29	Masks

Modeller merkezi işlem birimi (CPU) veya grafik işlemci birimi (GPU) olmak üzere iki farklı şekilde çalıştırılır. Modelin seçiminde en önemli unsur tez çalışması kapsamında kullanılan bilgisayarın ekran kartının hesaplama kapasitesidir. Modellerin GPU ile çalıştırılması için literatürde minimum hesaplama kapasitesinin yaklaşık 3,5 değerinde olması beklenmektedir. Tez çalışması kapsamında kullanılan bilgisayarın donanımsal özellikleri Tablo 4.2’de gösterilmiştir.

Tablo 4.2. Modelin eğitilmesi ve test edilmesinde kullanılan bilgisayarın teknik özellikleri

Donanım	Özellik
İşlemci	Intel (R) Core (TM) i7-4700HQ CPU @ 2.40 GHz
Yüklü Bellek (RAM)	8,00 GB
Sistem Türü	64 Bit
Ekran Kartı	NVIDIA GEFORCE 750M
Hesaplama Kapasitesi	3.0

Donanımsal özelliklerin yeterli düzeyde olmaması sebebiyle tez çalışması kapsamında model olarak “Faster R-CNN Inception v2 COCO” ve “Faster R-CNN Resnet101 COCO” modelleri merkezi işlem birimi (CPU) ile çalıştırılmıştır.

4.3. Eğitim Veri Setleri

Google ve Yandex şirketleri açık kaynak üzerinde işlenmiş harita katmanı sunmak için çeşitli çözünürlükte ticari uydu görüntülerine ihtiyaç duymaktadır. Bu bağlamda şirketler yeryüzünün yoğunluğuna göre uygun çözünürlükte uydu görüntüsü temin etmektedir. Boş arazilerin olduğu bölge için yüksek çözünürlüklü uydu görüntüsüne ihtiyaç duymayıp çözünürlüğü daha düşük uydu görüntülerini kullanmayı tercih ederken, yapılaşmanın ve gelişmenin yoğun olduğu bölgeler için yüksek çözünürlüklü uydu görüntülerine ihtiyaç duymaktadırlar.

Tez kapsamında stadyumların bulunduğu konumlar farklılık göstermektedir. Bu durumda bir stadyumun Google tarafından hazırlanan işlenmiş harita katmanının çözünürlüğü ile Yandex tarafından hazırlanan işlenmiş harita katmanının çözünürlüğü birbirinden farklı olabilmektedir. Ancak elde edilen veri setleri ham uydu görüntüsü olmayıp SAS Planet uygulaması üzerinden direkt olarak işlenmiş harita katmanı olarak temin edildiği için net çözünürlük değerleri hakkında bilgi edinilememiştir. Sonuç olarak tez kapsamında kullanılan modeller parametreleri sabit tutulmak kaydıyla Google ve Yandex şirketlerinin SAS Planet uygulaması üzerinde paylaşıma sunduğu işlenmiş harita katmanlarıyla eğitilmiştir. Temin edilen veri setleri ise birebir aynı ölçekte ve aynı köşe koordinatlarından kırılarak temin edilmiştir. Tez kapsamında veri setlerinin temin edildiği SAS Planet uygulaması Delphi yazılımı ile oluşturulmuş, Google Earth, Google Maps, Bing Maps, DigitalGlobe, Kosmosnimki, Yandex Maps, Yahoo gibi servisler tarafından sağlanan yüksek çözünürlüklü uydu görüntülerini ve geleneksel haritaları görüntülemek ve indirmek için tasarlanmış ücretsiz bir programdır.

SAS Planet uygulamasının birçok işlevi mevcuttur. Bunlar;

- GPS alıcısı ile çalışma,
- Rota oluşturma,
- Mesafe ölçümü,

- KML uzantılı dosyaların görüntülenmesi,
- Panaromio servis desteği,
- CBS uygulamalarında kullanılmak üzere harita üzerinde seçili alanın dışarıya aktarılması,
- Harita üzerinde seçili konum hakkında bilgi sağlaması,
- Haritaları iPhone ve Yandex haritalar tarafından desteklenen bir formatta dışarıya aktarma yeteneği,
- Wikimapia dosyalarının yüklenmesi ve görüntülenmesini sağlaması,
- Google ve Yandex aracılığıyla yer arama.

Eğitimde kullanılacak görüntü veri seti SAS Planet uygulaması üzerinde mevcut, Google ve Yandex şirketlerinin paylaşımına sunduğu işlenmiş harita katmanlarından temin edilmiştir. Tespit objesi olarak belirlenen her bir stadyum görüntüsü için köşe koordinatı Google ve Yandex veri setleri için sabit tutulmuştur.

SAS Planet uygulamasından temin edilen görüntülerin tespit modelinin hata vermesini önlemek maksadıyla 200 kilobayt'tan küçük olmasına önem verilmiştir. Eğitim için kullanılan, Google şirketinden temin edilen işlenmiş harita katmanlarından oluşan veri setinin bir kısmı Şekil 4.1.'de, Yandex şirketinden temin edilen işlenmiş harita katmanlarından oluşan veri setinin bir kısmı Şekil 4.2'de gösterilmiştir.

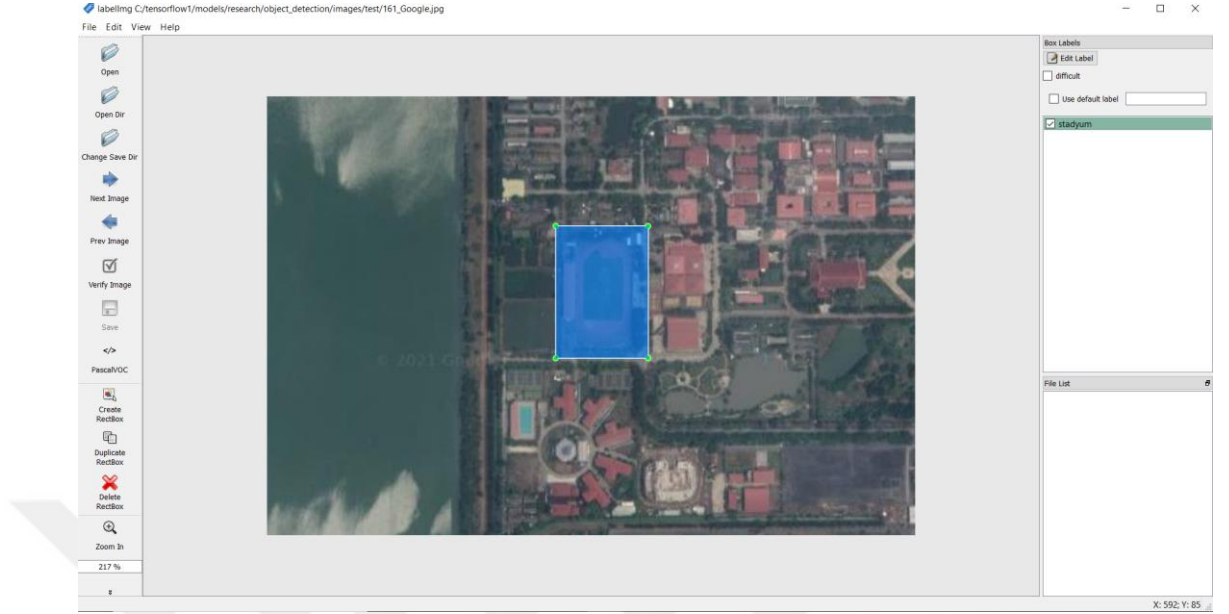


Şekil 4.1. Tez kapsamında kullanılan Google veri setinden örnekler



Şekil 4.2. Tez kapsamında kullanılan Yandex veri setinden örnekler

Veri setinde bulunan 310 adet görüntü dört ana yön istikametinde döndürülerek görüntü çoğaltma işlemi gerçekleştirilmiştir. 1240 adet görüntü içerisinde 1120 adedi stadyum kapsayan görüntü, 120 adedi ise stadyum kapsamayan görüntülerden oluşmaktadır. 1240 görüntü %70'i eğitim %30'u test verisi olacak şekilde 868 adet eğitim, 372 adet test verisi olarak ayrıştırılmıştır. Veri setine ait görüntülerin etiketlenmesi işlemi LabelImg_v1.8.0 programı ile gerçekleştirilmiş, görüntüler 'Stadyum' ismi ile etiketlenmiştir. Etiketleme işlemi Şekil 4.3'te gösterilmiştir.



Şekil 4.3. *LabelImg_v1.8.0 programı ile veriyi etiketleme işlemi*

4.4. Modelin Eğitilme Prensibi

Tez çalışmasında kullanılan modeller önce eğitim basamaklarında ‘Stadyum’ ismiyle etiketlenmiş veri setleriyle denetimli olarak eğitilmekte sonrasında test için oluşturulan veri setlerinde yer alan stadyumları tespit etmeye çalışmaktadırlar. Test veri setlerindeki tespitlere göre modellerin gizli katmanlarını oluşturan nöronlar ağırlık değerlerini güncelleyerek kayıp (loss) değerlerini minimize etmek için algoritmayı tekrar etmektedirler. Tez kapsamında kullanılan Faster R-CNN algoritması nesne tespiti yapmak üzere geliştirilmiş gradyan vektör temelli öğrenmeye dayalı bir algoritmadır.

Tez kapsamında Google şirketinden temin edilen uydu görüntülerinden oluşan veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” modeli 50010 adımda, “Faster R-CNN Resnet101 Coco” modeli 16810 adımda sonlandırılmış, Yandex şirketinden temin edilen uydu görüntülerinden oluşan veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” modeli 50030 adımda, “Faster R-CNN Resnet101 COCO” modeli ise 16670 adımda sonlandırılmıştır.

4.5. Hiper Parametre Seçimi

“Faster R-CNN Inception v2 COCO” modelinin hem Google hem de Yandex görüntülerinden oluşan veri setinde öğrenim oranı (learning rate) 0.0002, parça (batch) sayısı 1’dir. Google veri seti için eğitim adımı (training step) sayısı 50010, Yandex veri seti için eğitim adımı 50030’dur.

“Faster R-CNN Resnet101 COCO” modelinin hem Google hem de Yandex uydu görüntülerinden oluşan veri setinde öğrenim oranı (learning rate) 0.0003, parça (batch) sayısı 1’dir. Google veri seti için eğitim adımı (training step) sayısı 16810, Yandex veri seti için eğitim adımı 16670’dir.

Öğrenim oranı düşük olduğu takdirde modelin eğitimi daha hassas olmakla beraber eğitim süresi de bir o kadar uzamaktadır. Ancak öğrenim oranının yüksek olması modelin tespit hassasiyetini düşürmektedir. Her adımdan sonra model eğitimi kapsamında gizli katmanlarda bulunan nöron katsayılarını güncellemektedir. Tespit hassasiyetinin düşmesi sonucu en başarılı seviyeden sıklıkla sapmaya neden olmaktadır. Bu sebeple modelin başarısı için en optimal öğrenim oranı belirlenmelidir.

Parça sayısı olarak belirlenen değer, modelin aynı anda kaç görüntü ile eğitileceği anlamına gelmektedir. Parça sayısının yüksek olması modelin tespit oranına doğru oranda etki yapmaktadır ancak ekran kartı belleğinin kullanım oranını da aynı doğrultuda artırmaktadır. Tez çalışmasında kullanılan bilgisayarın ekran kartı belleğinin yüksek olmaması sebebiyle tüm model hiper parametrelerinde parça sayısı 1 değerinde sabit tutulmuştur.

4.6. Model Performanslarının Karşılaştırılması

Modellerin performans karşılaştırmalarında dört çeşit metrik kullanılmıştır. Bunlar; kesinlik (precision), duyarlılık (recall), F1-Skoru (F1-Score) ve kalite (quality) metrikleridir.

Kesinlik metriği (Denklem 4.1); tahmin sonucu pozitif olan değerlerin aslında ne kadarının pozitif olduğunu belirtmek için kullanılmaktadır.

Duyarlılık metriği (Denklem 4.2); aslında pozitif olması gereken seçeneklerin kaç adetinin pozitif tahmin edildiğini belirtmek için kullanılmaktadır.

F1-Skoru metriği (Denklem 4.3); kesinlik ve hassaslık değerlerinin harmonik ortalamasını göstermektedir.

Kalite metriği (Denklem 4.4); eğitilmiş algoritmanın başka bir girdide gösterebileceği tahmini başarı oranını belirtmektedir. Metrik denklemleri aşağıda yer almaktadır;

$$\text{Kesinlik} = \frac{DP}{DP + YP} \quad (4.1)$$

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \quad (4.2)$$

$$\text{F1-Skoru} = \frac{2 * \text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (4.3)$$

$$\text{Kalite} = \frac{DP}{DP + YP + YN} \quad (4.4)$$

DP: Doğru Pozitif (Modelin pozitif olarak tahminde bulunduğu ve aslında pozitif olan objeler)

YP: Yanlış Pozitif (Modelin pozitif olarak tahminde bulunduğu ve aslında pozitif olmayan objeler)

YN: Yanlış Negatif (Modelin negatif olarak tahminde bulunduğu ve aslında pozitif olan objeler) (Albayrak, 2021)

Eğitim ve test verisi olarak toplam 1240 adet görüntünün 372 tanesi test verisine dahil edilmiştir. 372 adet veri görüntü çoğaltma işlemi yapılmadığı durumda 93 adet görüntüden oluşmaktadır. 93 adet görüntünün 84 tanesi toplamda 121 adet stadyum görüntüsünü kapsamakta, 9 tanesi ise stadyum görüntüsünü kapsamamaktadır.

Metriklerin hesaplanması test veri seti içerisinde görüntü çoğaltma işlemi uygulanmamış 93 adet görüntü üzerinde yapılmıştır. Metrik sonuçları Tablo 4.3'te gösterilmiştir.

Tablo 4.3. Google ve Yandex görüntüleriyle eğitilmiş model performanslarının karşılaştırılması

Model	Veri seti	DP	YP	YN	Kesinlik	Duyarlılık	F1_Skoru	Kalite
Faster R-CNN	Google	97	1	23	0,9897	0,8083	0,889849	0,801653
Inception v2	Yandex	102	1	18	0,9902	0,8500	0,914759	0,842975
COCO								
Faster R-CNN	Google	101	0	20	1	0,8347	0,909904	0,834711
Resnet101 COCO	Yandex	98	1	22	0,9898	0,8166	0,894897	0,809917

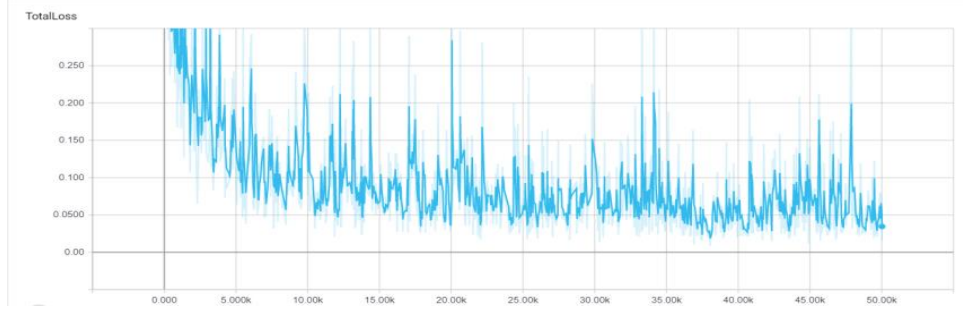
Tablo 4.3. incelendiğinde kesinlik metriği hariç, duyarlılık, F1-skoru ve kalite metriklerinde en başarılı model Yandex görüntüleri ile eğitilen “Faster R-CNN Inception v2 COCO” modelidir.

4.7. Kayıp (Loss) Grafikleri ve Karşılaştırılması

Google uydu görüntüleriyle eğitilen “Faster R-CNN Inception v2 COCO” modelinin eğitimdeki kayıp grafiği Şekil 4.4.’te, Yandex uydu görüntüleriyle eğitilen “Faster R-CNN Inception v2 COCO” modelinin eğitimdeki kayıp grafiği ise Şekil 4.5.’te gösterilmiştir. Grafik dikey ekseninde modelin eğitim veri seti için kayıp değerini, yatay ekseninde adım sayısını ifade etmektedir.



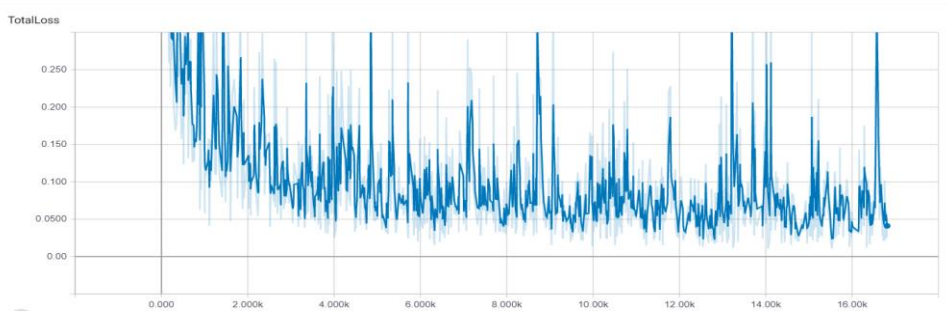
Şekil 4.4. Google uydu görüntüleri ile eğitilen Faster R-CNN Inception v2 COCO modelinin eğitim veri seti için kayıp grafiği



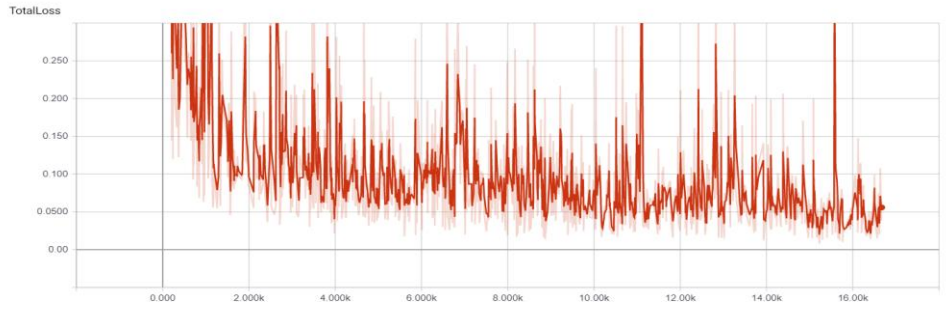
Şekil 4.5. Yandex uydu görüntüleri ile eğitilen *Faster R-CNN Inception v2 COCO* modelinin eğitim veri seti için kayıp grafiği

Şekil 4.4. ve Şekil 4.5. incelendiğinde “Faster R-CNN Inception v2 COCO” modelinin Google şirketinden temin edilen uydu görüntüleri ile eğitilmesinde kayıp dalgalanmalarının Yandex şirketinden temin edilen uydu görüntüleri ile eğitilmesinden daha sık olduğu ve algoritmaya ait gizli katmanların ağırlık değerlerindeki değişimlerin daha keskin olduğu gözlenmektedir. Sonuç olarak Yandex uydu görüntüleri ile eğitilen “Faster R-CNN Inception v2 COCO” modelinin kayıp değeri daha düşük çıkmıştır.

Google uydu görüntüleriyle eğitilen “Faster R-CNN Resnet101 COCO” kayıp grafiği Şekil 4.6.’da Yandex uydu görüntüleriyle eğitilen “Faster R-CNN Resnet101 COCO” modelinin kayıp grafiği ise Şekil 4.7.’de gösterilmiştir.



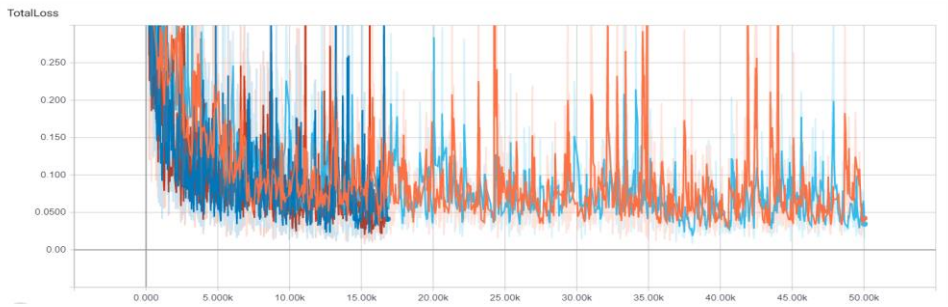
Şekil 4.6. Google uydu görüntüleri ile eğitilen *Faster R-CNN ResNet101 COCO* modelinin kayıp grafiği



Şekil 4.7. Yandex uydu görüntüleri ile eğitilen *Faster R-CNN ResNet101 COCO* modelinin kayıp grafiği

Şekil 4.6. ve Şekil 4.7. incelendiğinde “Faster R-CNN Resnet101 COCO” modelinin Yandex şirketinden temin edilen uydu görüntüleri ile eğitilmesinde kayıp dalgalanmalarının Google şirketinden temin edilen uydu görüntüleri ile eğitilmesinden daha sık olduğu ve algoritmaya ait gizli katmanların ağırlık değerlerindeki değişimlerin daha keskin olduğu gözlenmektedir. Sonuç olarak Google veri seti ile eğitilen “Faster R-CNN Resnet101 COCO” modelinin kayıp değeri daha düşük çıkmıştır.

Tüm model ve veri seti parametrelerine ait kayıp grafikleri Şekil 4.8’de gösterilmiştir.



- Google veri seti ile eğitilen Faster R-CNN Inception v2 COCO modeli
- Google veri seti ile eğitilen Faster R-CNN Resnet101 COCO modeli
- Yandex veri seti ile eğitilen Faster R-CNN Inception v2 COCO modeli
- Yandex veri seti ile eğitilen Faster R-CNN Resnet101 COCO modeli

Şekil 4.8. Google ve Yandex veri setleri ile eğitilen *Faster R-CNN inception v2* ve *Faster R-CNN Resnet101 COCO* modellerinin kayıp grafiği

Çalışmaya ait model türü, veri seti temin bilgisi, eğitim adım sayısı ve kayıp değerleri Tablo 4.4’te gösterilmiştir.

Tablo 4.4. *Eğitilen modellerin kayıp değerleri*

Model	Veri Seti	Eğitim Adım Sayısı	Kayıp Değeri
Faster R-CNN Inception v2 COCO	Google	50010	0,048
Faster R-CNN Inception v2 COCO	Yandex	50030	0,015
Faster R-CNN Resnet101 COCO	Google	16810	0,025
Faster R-CNN Resnet101 COCO	Yandex	16670	0,057

“Faster R-CNN Inception v2 COCO” modelinin Google uydu görüntüleri ile oluşturulan veri seti ile eğitilmesi 50010 adımda gerçekleşmiş olup kayıp değeri 0,048’dir. Yandex uydu görüntüleri ile oluşturulan veri seti ile eğitilmesi 50030 adımda gerçekleşmiş olup kayıp değeri 0,015’tir.

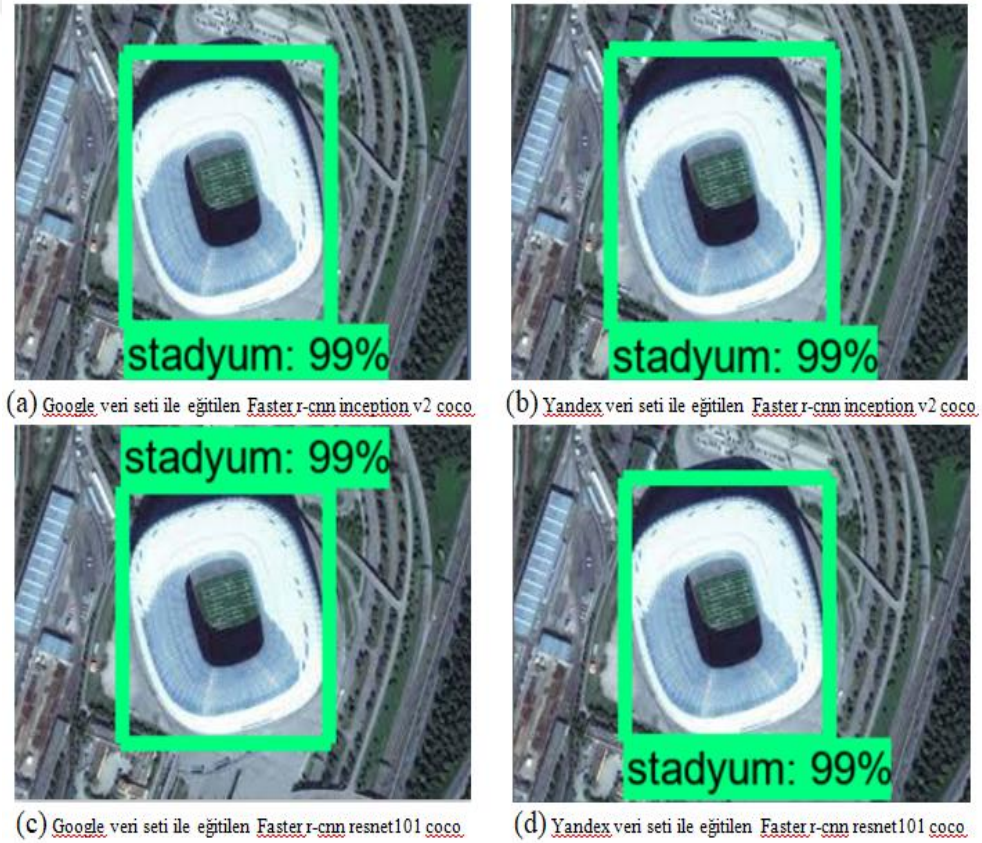
“Faster R-CNN Resnet101 COCO” modelinin Google uydu görüntüleri ile oluşturulan veri seti ile eğitilmesi 16810 adımda gerçekleşmiş olup kayıp değeri 0,025’tir. Yandex uydu görüntüleri ile oluşturulan veri seti ile eğitilmesi 16670 adımda gerçekleşmiş olup kayıp değeri 0,057’dir.

Kayıp grafiği üzerinde dört ayrı çalışmanın incelenmesinde en çok dikkat çeken husus Google uydu görüntülerinden oluşan veri seti ile eğitilen modellerde dalgalanmanın daha sık olmasıdır. Bunun dışında başarımlar oranı en yüksek olan model 0,015 kayıp değeri ile Yandex görüntüleri ile eğitilen “Faster R-CNN Inception v2 COCO” modelidir.

5. EĞİTİLEN MODELLERİN TEST EDİLMESİ

Eğitilmiş modellerin test edilmesi aşamasında Google nesne tespit uygulaması olan ‘object_detection_image.py’ arayüzü kullanılmıştır.

Modellerin eğitimi tamamlandıktan sonra açık kaynak üzerinden rastgele seçilen on adet hava fotoğrafı üzerinde modeller çalıştırılmıştır. Google veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” modelinin sonucu ‘(a)’ ile, Yandex veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” modelinin sonucu ‘(b)’ ile, Google veri seti ile eğitilen “Faster R-CNN Resnet101 COCO” modelinin sonucu ‘(c)’ ile, Yandex veri seti ile eğitilen “Faster R-CNN Resnet101 COCO” modelinin sonucu ‘(d)’ ile gösterilmiştir.



Şekil 5.1. Birinci test görüntüsü

Şekil 5.1’de birinci test sonucunda tüm algoritmalar görüntüdeki stadyumu %99’luk nesne tahmin oranında tespit edebilmiştir.



(a) Google veri seti ile eğitilen Faster r-cnn inception v2 coco



(b) Yandex veri seti ile eğitilen Faster r-cnn inception v2 coco



(c) Google veri seti ile eğitilen Faster r-cnn resnet101 coco



(d) Yandex veri seti ile eğitilen Faster r-cnn resnet101 coco

Şekil 5.2. İkinci test görüntüsü

Şekil 5.2’de ikinci test sonucunda bitişik konumda yer alan stadyumlardan sol tarafta olan stadyumu tüm algoritmalar %99’luk nesne tahmin oranıyla tespit edebilirken diğer stadyumu yalnızca Google veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” algoritması %66 nesne tahmin oranıyla tespit edebilmiştir.



(a) Google veri seti ile eğitilen Faster r-cnn inception v2 coco



(b) Yandex veri seti ile eğitilen Faster r-cnn inception v2 coco



(c) Google veri seti ile eğitilen Faster r-cnn resnet101 coco



(d) Yandex veri seti ile eğitilen Faster r-cnn resnet101 coco

Şekil 5.3. Üçüncü test görüntüsü

Şekil 5.3'te üçüncü test sonucunda tüm algoritmalar mevcut tek stadyumu %99'luk nesne tahmin oranıyla tespit edebilmiştir.



(a) Google veri seti ile eğitilen Faster r-cnn inception v2 coco



(b) Yandex veri seti ile eğitilen Faster r-cnn inception v2 coco



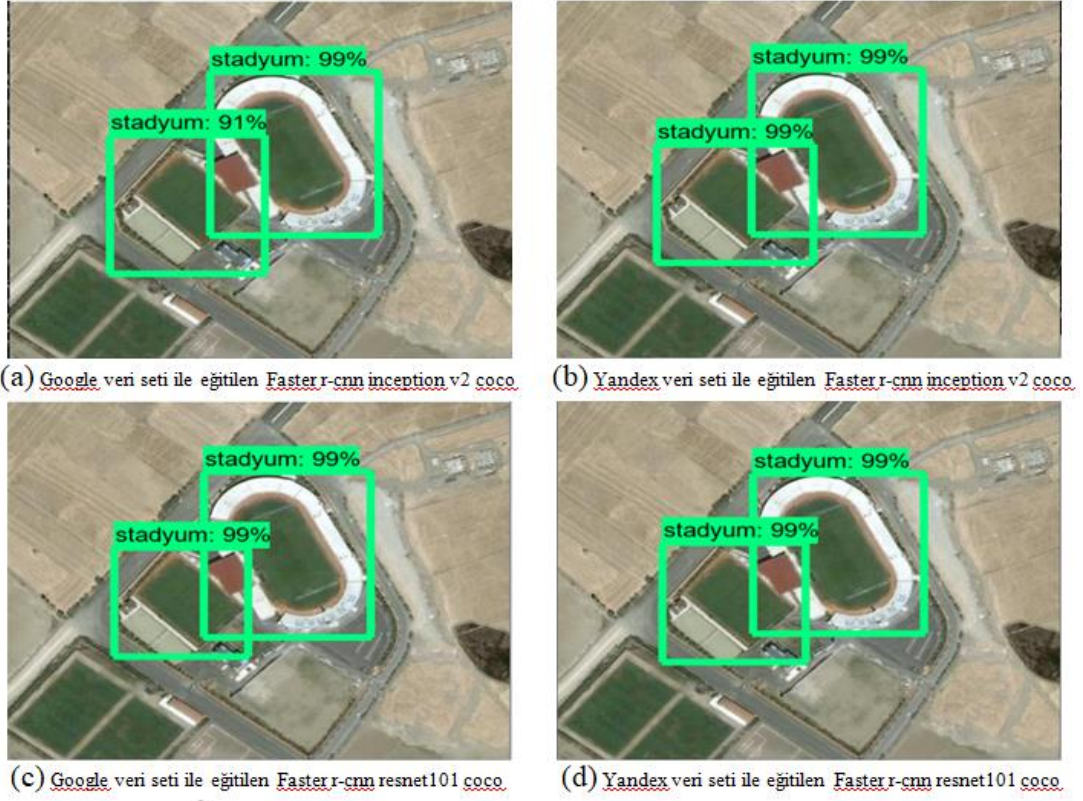
(c) Google veri seti ile eğitilen Faster r-cnn resnet101 coco



(d) Yandex veri seti ile eğitilen Faster r-cnn resnet101 coco

Şekil 5.4. Dördüncü test görüntüsü

Şekil 5.4.'te dördüncü test sonucunda tüm algoritmalar %99'luk nesne tahmin oranında stadyumları tespit edebilmiştir.



Şekil 5.5. Beşinci test görüntüsü

Şekil 5.5.'te beşinci test sonucunda bitişik konumda yer alan stadyumlardan üst kısımda kalanı tüm algoritmalar %99'luk nesne tahmin oranında tespit edebilirken, alt kısımda kalan stadyumu Google veri seti ile eğitilen "Faster R-CNN Inception v2 Coco" algoritması %91 tahmin oranında tespit edebilmiş, diğer algoritmalar yine %99 tahmin oranında tespit edebilmiştir.



(a) Google veri seti ile eğitilen Faster r-cnn inception v2 coco



(b) Yandex veri seti ile eğitilen Faster r-cnn inception v2 coco



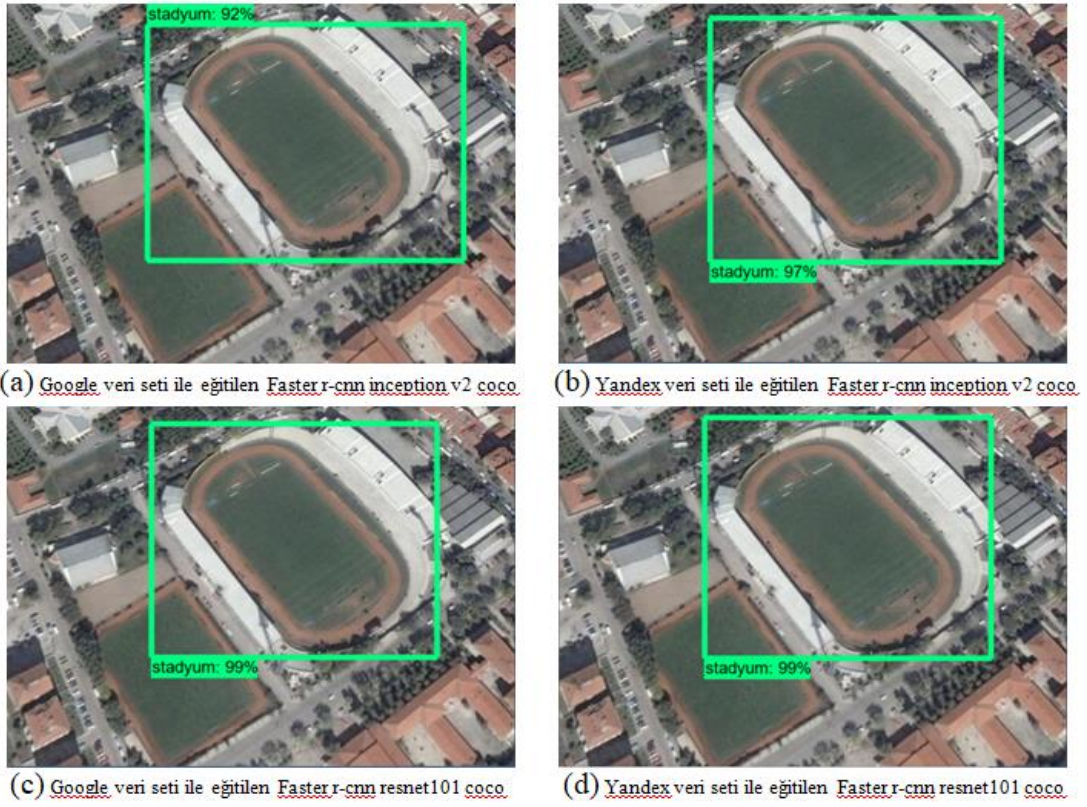
(c) Google veri seti ile eğitilen Faster r-cnn resnet101 coco



(d) Yandex veri seti ile eğitilen Faster r-cnn resnet101 coco

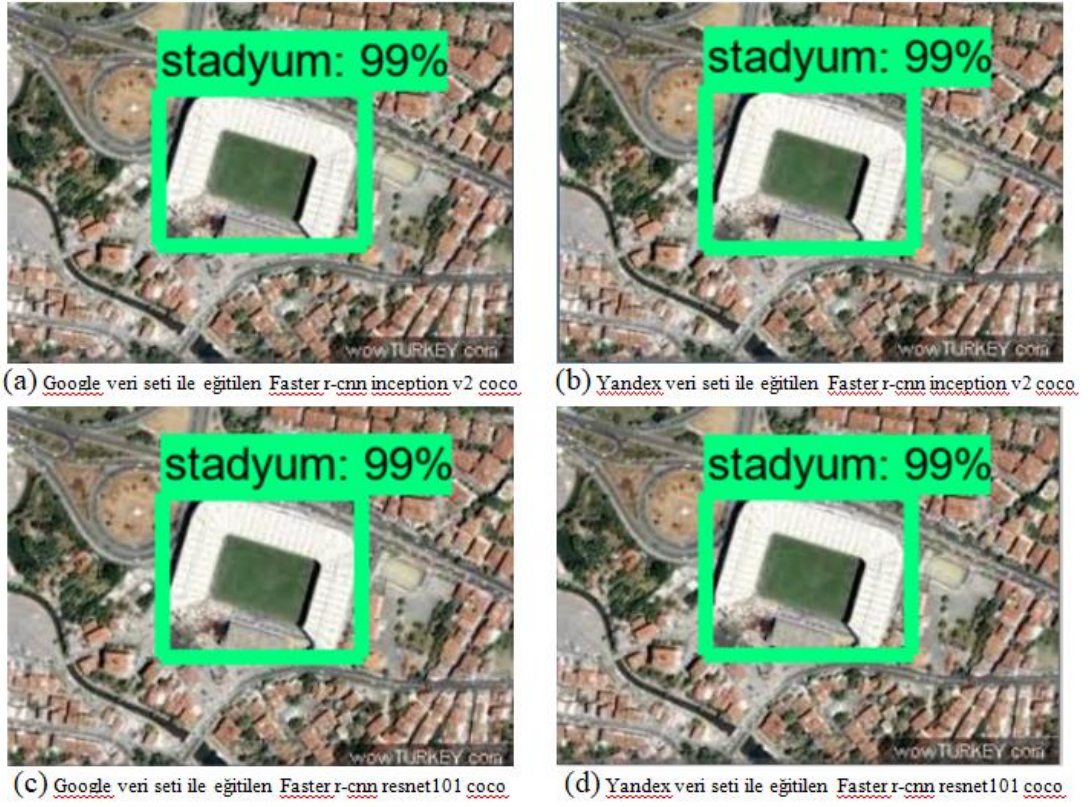
Şekil 5.6. Altıncı test görüntüsü

Şekil 5.6.'da altıncı test sonucunda Yandex veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” algoritması stadyumu tespit edememiş, Google veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” algoritması %77’lik nesne tahmin oranında, Google veri seti ile eğitilen “Faster R-CNN Resnet101 COCO” algoritması %94’lük nesne tahmin oranında, Yandex veri seti ile eğitilen “Faster R-CNN Resnet101 COCO” algoritması ise %93’lük nesne tahmin oranında tespit edebilmiştir.



Şekil 5.7. Yedinci test görüntüsü

Şekil 5.7.'de yedinci test sonucunda Google ve Yandex veri setleri ile ayrı ayrı eğitilen “Faster R-CNN Resnet101 COCO” algoritması stadyumları %99'luk nesne tahmin oranında tespit edebilmişken; Yandex veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” algoritması stadyumu %97'lik nesne tahmin oranında, Google veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” algoritması ise stadyumu %92'lik nesne tahmin oranında tespit edebilmiştir.



Şekil 5.8. Sekizinci test görüntüsü

Şekil 5.8.'de sekizinci test sonucunda tüm algoritmalar %99'luk nesne tahmin oranında stadyumları tespit edebilmiştir.



(a) Google veri seti ile eğitilen Faster r-cnn inception v2 coco



(b) Yandex veri seti ile eğitilen Faster r-cnn inception v2 coco



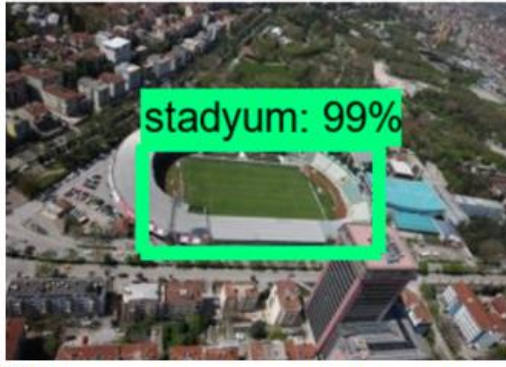
(c) Google veri seti ile eğitilen Faster r-cnn resnet101 coco



(d) Yandex veri seti ile eğitilen Faster r-cnn resnet101 coco

Şekil 5.9. Dokuzuncu test görüntüsü

Şekil 5.9.'da dokuzuncu testte kullanılan gayri faal durumdaki stadyumu eğitilmiş modellerden hiçbirisi tespit edememiştir.



(a) Google veri seti ile eğitilen Faster r-cnn inception v2 coco



(b) Yandex veri seti ile eğitilen Faster r-cnn inception v2 coco



(c) Google veri seti ile eğitilen Faster r-cnn resnet101 coco



(d) Yandex veri seti ile eğitilen Faster r-cnn resnet101 coco

Şekil 5.10. Onuncu test görüntüsü

Şekil 5.10.'da onuncu test sonucunda tüm algoritmalar %99'luk nesne tahmin oranında stadyumları tespit edebilmiştir.

6. SONUÇ

Tez çalışması kapsamında veri setleri SAS Planet uygulaması üzerinden aynı köşe koordinatlarına sahip Yandex ve Google uydu görüntülerinden temin edilmiştir. Yani bir stadyum Google uydu görüntüsü üzerinden hangi sınırlar dahilinde dışa aktarıldıysa, program üzerinden katman değiştirerek Yandex uydu görüntüsü üzerinden de aynı sınırlar dahilinde dışa aktarılmıştır.

Görüntülerin eğitilmesi işlemi iki ayrı model üzerinde dört aşamada yapılmıştır. Birinci aşamada “Faster R-CNN Inception v2 COCO” algoritması Google uydu görüntüleri ile eğitilmiş ikinci aşamada yine aynı algoritma Yandex uydu görüntüleri ile eğitilmiştir. Üçüncü aşamada “Faster R-CNN Resnet101 COCO” algoritması Google uydu görüntüleri ile eğitilmiş dördüncü aşamada ise yine aynı algoritma Yandex uydu görüntüleri ile eğitilmiştir.

Yapılan bu işlemler neticesinde ortaya çıkarılmak istenen iki ayrı sonuç vardır. Birincisi aynı veri seti ile eğitilen iki farklı algoritmanın birbirlerine karşı başarımlarının kıyaslanması. İkincisi ise aynı hiper parametrelere sahip bir algoritmanın iki ayrı veri seti ile eğitilmesi ile ortaya çıkacak sonuç bağlamında Google ve Yandex şirketlerinin açık kaynak üzerinde paylaşım sunduğu uydu görüntülerinin derin öğrenme algoritmaları üzerinde başarımlarının tespit edilmesidir.

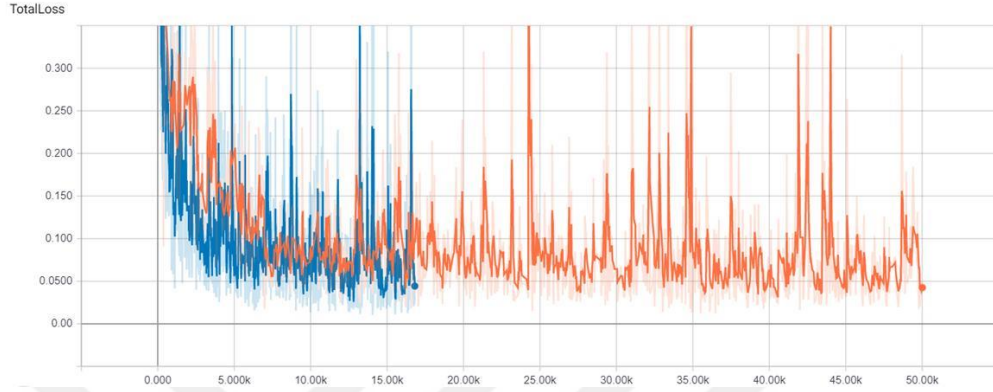
Analiz sonuçları F1-Skoru doğruluk metriği ve algoritmaların eğitim aşamasında gerçekleşen kayıp değerleri üzerinden değerlendirilmiştir. 0,914759 doğruluk değeri ile en başarılı model Yandex görüntüleri ile eğitilen “Faster R-CNN Inception v2 COCO” modeli olarak ortaya çıkmıştır. Kayıp değeri bazında yapılan analizde ise 0,015 ile en düşük kayıp değerine sahip Yandex görüntüleri ile eğitilen “Faster R-CNN Inception v2 COCO” modeli yine en başarılı model olarak ortaya çıkmıştır.

Kayıp grafiklerinin karşılaştırması iki farklı yöntem ile yapılmıştır. Birinci yöntem aynı veri seti ile eğitilen farklı modellerin karşılaştırılması, ikinci yöntem ise aynı modeli eğiten farklı veri setlerinin başarıma olan etkisinin karşılaştırılmasıdır.

Birinci yöntem: Aynı veri seti ile eğitilen farklı modellerin karşılaştırılması

Google uydu görüntülerinden oluşan veri seti ile “Faster R-CNN Inception v2 COCO” ile “Faster R-CNN Resnet101 COCO” modelleri karşılaştırılmıştır. “Faster R-CNN Inception v2 COCO” modeli 50010 adımda eğitilmiş olup kayıp değeri 0,048’dir.

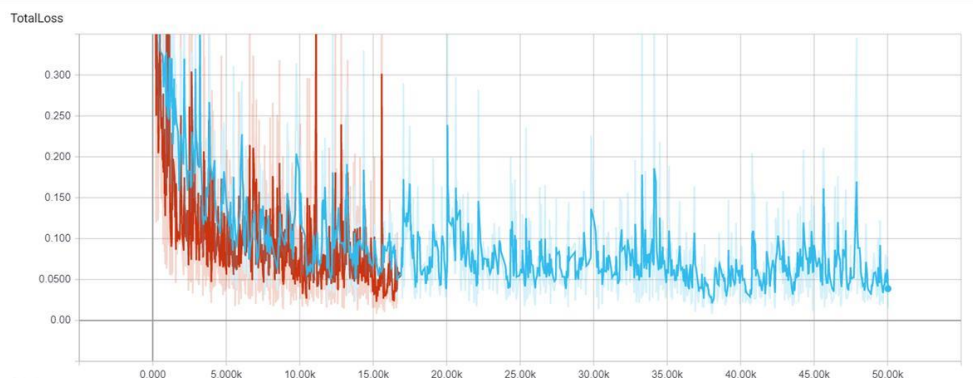
“Faster R-CNN Resnet101 COCO” modeli ise 16810 adımda eğitilmiş olup kayıp değeri 0,025’dir. Google veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” ve “Faster R-CNN Resnet101 COCO” modellerin grafiği Şekil 6.1.’de gösterilmiştir.



Şekil 6.1. Google veri seti ile eğitilen *Faster R-CNN Inception v2 COCO* ve *Faster R-CNN Resnet101 COCO* modellerin grafiği

Şekil 6.1. incelendiğinde “Faster R-CNN Resnet101 COCO” modeli daha az adımda eğitilmesine rağmen kayıp değerinin daha düşük olduğu görülmektedir. F-1 Skoru metriğine göre kıyaslandığında ise “Faster R-CNN Resnet101 COCO” modelinin doğruluk değeri 0,909904, “Faster R-CNN Inception v2 COCO” modelinin doğruluk değeri ise 0,889849 olarak ortaya çıkmıştır. Bu aşamada iki algoritma Google uydu görüntüleri ile eğitildiğinde “Faster R-CNN Resnet101 COCO” modelinin daha başarılı olduğu tespit edilmiştir.

Yandex uydu görüntülerinden oluşan veri seti ile “Faster R-CNN Inception v2 COCO” ile “Faster R-CNN Resnet101 COCO” modelleri karşılaştırılmıştır. “Faster R-CNN Inception v2 COCO” modeli 50030 adımda eğitilmiş olup kayıp değeri 0,015’dir. “Faster R-CNN Resnet101 COCO” modeli ise 16670 adımda eğitilmiş olup kayıp değeri 0,057’dir. Yandex veri seti ile eğitilen “Faster R-CNN Inception v2 COCO” ve “Faster R-CNN Resnet101 COCO” modellerin grafiği Şekil 6.2.’de gösterilmiştir.

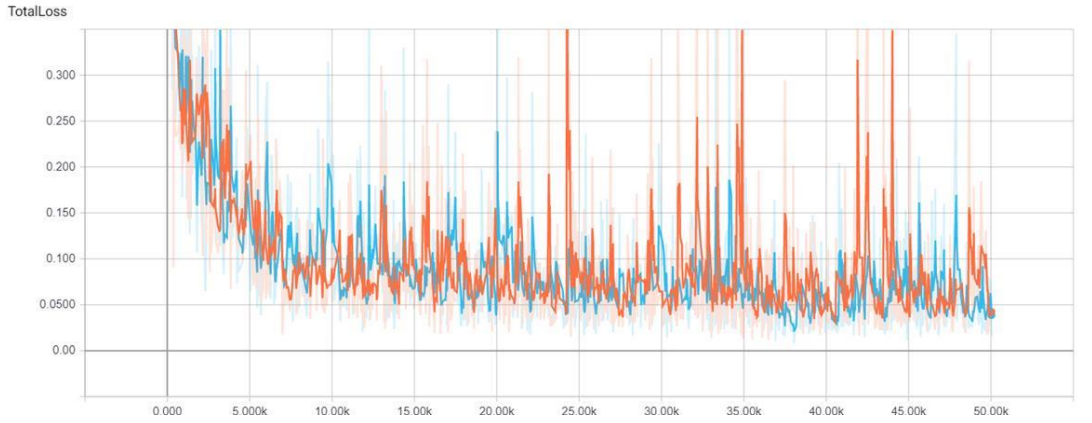


Şekil 6.2. Yandex veri seti ile eğitilen *Faster R-CNN Inception v2 COCO* ve *Faster R-CNN Resnet101 COCO* modellerin grafiği

Şekil 6.2. incelendiğinde *Faster R-CNN Inception v2 COCO* modelinin kayıp değerinin daha düşük olduğu görülmüştür. F-1 Skoru metriğine göre kıyaslandığında ise *Faster R-CNN Inception v2 COCO* modelinin doğruluk değeri 0,914759, *Faster R-CNN Resnet101 COCO* modelinin doğruluk değeri ise 0,894897 olarak ortaya çıkmıştır. Bu aşamada iki algoritma Yandex uydu görüntüleri ile eğitildiğinde *Faster R-CNN Inception v2 COCO* modelinin daha başarılı olduğu tespit edilmiştir.

İkinci yöntem: Aynı modeli eğiten farklı veri setlerinin başarıma olan etkisinin karşılaştırılması

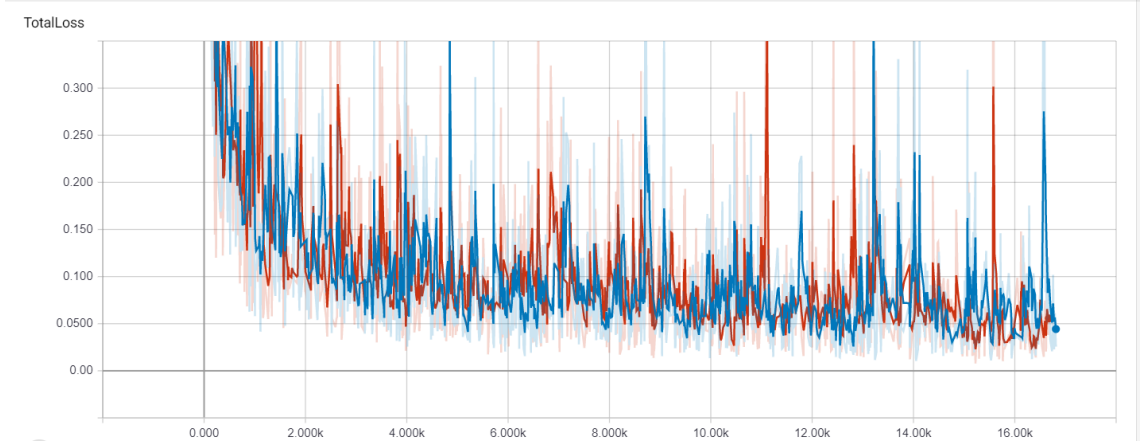
Tez çalışması kapsamında bir amaçta bir modeli farklı iki veri seti ile eğitmek suretiyle veri setinin modelin başarı oranına etkisini tespit etmektir. Bu doğrultuda ilk olarak “*Faster R-CNN Inception v2 COCO*” modeli Google ve Yandex şirketlerinin oluşturduğu uydu görüntüleri ile eğitilmiştir. Google şirketine ait uydu görüntüleri ile eğitilen “*Faster R-CNN Inception v2 COCO*” modeli 50010 adımda eğitilmiş olup kayıp değeri 0.048 olarak tespit edilmiş, Yandex şirketine ait uydu görüntüleri ile eğitilen “*Faster R-CNN Inception v2 COCO*” modeli 50030 adımda eğitilmiş kayıp değeri 0.01526 olarak tespit edilmiştir. Ayrıca kayıp grafiğinde de görüldüğü üzere Yandex veri setiyle eğitilen modelde kayıp değeri sapmalarının daha az olduğu ve eğitim sonu ortaya çıkan kayıp değeri ile Yandex veri setinin daha başarılı olduğu tespit edilmiştir. *Faster R-CNN Inception v2 COCO* modelinin Google ve Yandex uydu görüntüleriyle eğitilmesi aşamasında ortaya çıkan kayıp değerleri Şekil 6.3.’te gösterilmiştir.



Şekil 6.3. Google ve Yandex veri seti ile eğitilen Faster R-CNN Inception v2 COCO modelinin grafiği

Şekil 6.3. kayıp değeri bazında incelendiğinde Faster R-CNN Inception v2 COCO modelinin Yandex uydu görüntüleri ile eğitilmesinde başarı oranının daha yüksek olduğu görülmektedir. F-1 Skoru metriğine göre kıyaslandığında modelin Google uydu görüntüleri ile eğitilmesi durumunda doğruluk değeri 0,889849, Yandex uydu görüntüleri ile eğitilmesi durumunda ise doğruluk değerinin 0,914759 olduğu tespit edilmiştir. F-1 Skoru metriğine göre ise yine modelin Yandex uydu görüntüleri ile eğitilmesinde daha başarılı olduğu tespit edilmiştir.

Faster R-CNN Resnet101 COCO modelinin Google uydu görüntüleri ile eğitilmesi 16810 adımda tamamlanmış ve kayıp değeri 0,02566 olarak ortaya çıkmıştır. Modelin Yandex uydu görüntüleri ile eğitilmesi ise 16670 adımda tamamlanmış ve kayıp değeri 0,05702 olarak ortaya çıkmıştır. Google ve Yandex uydu görüntüleri ile eğitilmiş olan Faster R-CNN Inception v2 COCO modelinin kayıp değerine ait grafiği Şekil 6.4.'te gösterilmiştir.



Google ve Yandex görüntüleriyle eğitilmiş model performanslarının karşılaştırılması
Google ve Yandex görüntüleriyle eğitilmiş model performanslarının karşılaştırılması

Şekil 6.4. Google ve Yandex veri seti ile eğitilen Faster R-CNN Resnet101 COCO modelinin grafiği

Şekil 6.4. kayıp değeri bazında incelendiğinde Faster R-CNN Resnet101 COCO modelinin Google uydu görüntüleri ile eğitilmesinde başarı oranının daha yüksek olduğu görülmektedir. F-1 Skoru metriğine göre kıyaslandığında modelin Google uydu görüntüleri ile eğitilmesi durumunda doğruluk değeri 0,909904, Yandex uydu görüntüleri ile eğitilmesi durumunda ise doğruluk değerinin 0,894897 olduğu tespit edilmiştir. F-1 Skoru metriğine göre ise yine modelin Google uydu görüntüleri ile eğitilmesinde daha başarılı olduğu tespit edilmiştir.

Tüm sonuçlar göz önünde bulundurulduğunda Yandex görüntüleri ile eğitilen Faster R-CNN Inception v2 COCO modelinin en başarılı model olduğu tespit edilmiştir. Ayrıca Google uydu görüntüleri ile eğitilen modellerin kayıp grafiklerinde dalgalanmaların daha sık olduğu tespit edilmiştir.

KAYNAKÇA

- Song, H., ve Lee S. (2013). Hierarchical representation using NMF. International Conference on Neural Information Processing, pp. 466–473.
- Körez, A. (2020). Derin öğrenme kullanarak uzaktan algılama görüntülerindeki nesnelerin tespiti. Doktora Tezi. Ankara: Gazi Üniversitesi, Fen Bilimleri Enstitüsü.
- McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. AI Magazine, 27(4), pp.12.
- King, D. (1997). Kasparov versus Deeper Blue: The Ultimate man versus machine challenge. ICGA Journal, 20(3), pp. 187-187.
- McCulloch, W.S. and Pitts, W.H. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5, pp. 115-133.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), pp. 386–408.
- Ivakhnenko, A.G. and Lapa, V.G. (1966). Cybernetic predicting devices. CCM Information Corporation.
- Fukushima, K. (1979). Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron. IEICE Technical Report, 62(10), pp. 658-665.
- Weng, J., Cohen, P. and Herniou M. (1992). Camera calibration with distortion models and accuracy evaluation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 10, pp. 965-980.
- Cortes, C., and Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), pp. 273-297.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), pp. 735–1780.

- Taigman, Y., Yang, M., Ranzato M. and Wolf L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1701–1708.
- Hinton, G.E. and Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), pp. 504-507.
- Lee, C.S., Wang, M.H., Yen, S.J., Wei, T.H., Wu, I.C., Chou, P.C., Chou, C.H., Wang, M.W. and Yan, T. H. (2016). Human vs. computer go: Review and prospect [discussion forum]. IEEE Computational intelligence magazine, 11(3), pp. 67-72.
- Atalay, M. ve Çelik, E. (2017). Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamaları. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 9(22), 155-172.
- Kılıç, S. (2013). Doğrusal regresyon analizi. Journal of Mood Disorders, 3(2), pp. 90-92.
- Bircan, H. (2004). Lojistik Regresyon Analizi: Tıp Verileri Üzerine Bir Uygulama. Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 2, pp. 185-208.
- Kavzoğlu, T. ve Çölkesen, İ. (2010). Karar Ağaçları İle Uydu Görüntülerinin Sınıflandırılması: Kocaeli Örneği. Harita Teknolojileri Elektronik Dergisi, 2(1), pp. 36-45.
- Fix, E., and Hodges, J.L. (1951). Nonparametric discrimination: consistency properties. Randolph Field, Texas, Project, pp. 21-49.
- Wu, Y., Ianiakiev, K. and Govindaraju, V. (2002). Improved k-nearest neighbor classification. Pattern recognition, 35(10), pp. 2311-2318.
- Maxwell, A.E., Warner, T.A. ve Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. International Journal of Remote Sensing, 39(9), pp. 2784-2817.
- Soman, K.P., Loganathan, R. and Ajay, V. (2009). Machine learning with SVM and other kernel methods. PHI Learning Pvt. Ltd.
- Kecman, V. and Zigic, L. (2014). Algorithms for direct L2 support vector machines. IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings pp. 419-424.

- Karaman, S., Kocaarslan, I., Tiryaki, H. and Emrah, B.A.L. (2017). The Modelling Recorded Faults In Railways And Prediction. *IU-Journal of Electrical & Electronics Engineering*, 17(2), pp. 3417-3423.
- Ayhan, S., ve Erdoğan, Ş. (2014). Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 9(1), pp. 175-201.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1(14), pp. 281-297.
- Sarıman, G. (2011). Veri madenciliğinde kümeleme teknikleri üzerine bir çalışma: k-means ve kmedoids kümeleme algoritmalarının karşılaştırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 15(3), pp. 192-202.
- Rao, A.R. and Srinivas, V.V. (2006). Regionalization of watersheds by fuzzy cluster analysis. *Journal of Hydrology*, 318(1-4), pp. 57-79.
- Alpar, R. 2011. Uygulamalı Çok Değişkenli İstatistiksel Yöntemler. Detay Yayıncılık, 3. Baskı, Ankara.
- Yeşilbudak, M., Kahraman H.T. ve Karacan H. (2011). Veri Madenciliğinde Nesne Yönelimli Birleştirici Hiyerarşik Kümeleme Modeli. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 26(1), pp. 27-39.
- Karakaya, I. and Çemtay, Y. (2017). The effect of bant selection to success of artificial neural network in hyperspectral classification. *25th Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4.
- Doğan, F., & Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), pp. 409-445.
- Ker, J., Wang, L., Rao, J. and Lim, T. (2017). Deep learning applications in medical image analysis. *IEEE Access*, 6, pp. 9375-9389.

- Murat, S. (2021). İnsansız hava aracı görüntülerinden derin öğrenme yöntemleriyle nesne tanıma. Yüksek Lisans Tezi. İstanbul: Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü
- Andrew, N. (2018). Unsupervised Feature Learning and Deep Learning Tutorial. Convolutional Neural Network, <http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/> (Erişim Tarihi: 26.03.2022)
- Salur, M.U., Aydın, İ. ve Karaköse, M. (2019). Gömülü derin öğrenme ile tehdit içeren nesnelerin gerçek zamanda tespiti. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), pp. 497-509.
- Doğan, F. ve Türkoğlu, İ. (2018). Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması. Sakarya University Journal of Computer and Information Sciences, 1(1), pp. 10-21.
- LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp. 2278-2324.
- Krizhevsky, A., Sutskever, I. and Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems (NIPS), Nevada, ABD, pp. 645-651.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- He, K., Zhang, X. Ren, S. ve Sun, J. (2015). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition, 770-778
- Cömert, O., Hekim, M. ve Adem, K. (2019). Faster R-CNN Kullanarak Elmalarda Çürük Tespiti. Uluslararası Araştırma ve Geliştirme Dergisi, 11(1), pp. 335-341.
- Albayrak, E. (2021). Derin Öğrenme ile İHA görüntülerinden Nesne tespitinin yapılması. Yüksek Lisans Tezi. Bilecik Şeyh Edebali Üniversitesi, Fen Bilimleri Enstitüsü.

ÖZGEÇMİŞ

Adı SOYADI: Emre Batuhan SAMUR

Yabancı Dil: İngilizce

Eğitim Özgeçmişi:

- 2014, Kırıkkale Üniversitesi / Mühendislik Fakültesi / Endüstri Mühendisliği

Meslek Özgeçmişi:

- 2018, Türk Silahlı Kuvvetleri / Hava Kuvvetleri Komutanlığı / Askeri Personel

Yayınlar ve Bilimsel Çalışmalar:

Çağlayan, E.B., Erel, F., Samur, E.B., Deniz, M., Mobariz M.A. ve Kaplan, G. (2020). Uzaktan Algılama Teknikler ile Akşehir Gölü’ndeki Alansal Değişiminin İzlenmesi. Türkiye Uzaktan Algılama Dergisi, 2(2), pp. 70-76.