

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**DERİN ÖĞRENME YÖNTEMİ İLE OPTİK UYDU GÖRÜNTÜLERİNDEN
GEMİ TESPİTİ**

YÜKSEK LİSANS TEZİ

Osman DUMAN

İletişim Sistemleri Anabilim Dalı

Uydu Haberleşmesi ve Uzaktan Algılama Programı

HAZİRAN 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**DERİN ÖĞRENME YÖNTEMİ İLE OPTİK UYDU GÖRÜNTÜLERİNDEN
GEMİ TESPİTİ**

YÜKSEK LİSANS TEZİ

**Osman DUMAN
(705151013)**

İletişim Sistemleri Anabilim Dalı

Uydu Haberleşmesi ve Uzaktan Algılama Programı

Tez Danışmanı: Prof. Dr. Mesut KARTAL

HAZİRAN 2019

İTÜ, Bilişim Enstitüsü'nün 705151013 numaralı Yüksek Lisans Öğrencisi **Osman DUMAN**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**DERİN ÖĞRENME YÖNTEMİ İLE OPTİK UYDU GÖRÜNTÜLERİNDEN GEMİ TESPİTİ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Mesut KARTAL**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Sedef KENT PINAR**
İstanbul Teknik Üniversitesi

Dr. Öğr. Üyesi Serdar KARGIN
Beykent Üniversitesi

Teslim Tarihi : **02 Mayıs 2019**
Savunma Tarihi : **13 Haziran 2019**





Aileme,



ÖNSÖZ

Yüksek lisans eğitimim boyunca şahsıma rehberlik ederek değerli zamanımı ayıran, bilgi ve tecrübelerini esirgemeyen, fikirleri ile vizyonumu geliştiren danışman hocam Sayın Prof. Dr. Mesut KARTAL'a teşekkür ederim. Ayrıca, hayatım boyunca beni hep seven, ilgilerini esirgemeyen ve bugünlere gelmemi sağlayan anneme ve babama, değerli kardeşime ve zorlu eğitim periyodum boyunca her türlü desteği esirgemeyen sevgili eşime şükranlarımı sunarım.

Mayıs 2019

Osman DUMAN
(Elektrik Elektronik Mühendisi)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY	xix
1. GİRİŞ	1
1.1 Tezin Amacı	3
1.2 Literatür Araştırması	4
2. DERİN ÖĞRENME.....	7
2.1 Tanım	7
2.2 Kullanım Alanları.....	7
2.3 Yapay Sinir Ağları	7
2.4 Konvolüsyonel Sinir Ağları	8
3. GEMİ TESPİT METODU	9
3.1 Amaç	9
3.2 Kullanılan Programlar ve Kütüphaneler	9
3.3 Model Seçimi	10
3.4 Eğitim Görüntü Setleri	10
3.5 Modelin Eğitilme Prensibi	11
3.6 Kayıp (Loss) Grafiği	12
3.7 Modellerin Hiperparametreleri.....	12
3.8 Kayıp Grafiklerinin Karşılaştırılması.....	13
3.9 Giriş (Input) Resimleri	15
3.10 Modellerin Gemi Tespit Çıktıları.....	16
3.10.1 Faster rcnn inception v2_coco modeli öğrenim oranı 0.0002.....	16
3.10.2 Faster rcnn inception v2_coco modeli öğrenim oranı 0.00002.....	23
3.10.3 Faster rcnn resnet101_coco.....	29
3.10.4 Ssd_inception_v2_coco	36
3.11 Modellerin Gemi Tespit Çıktılarının Karşılaştırılması	42
3.12 Seçilen Modelin Gemi Tespit Performansının Artırılması İçin Yapılan İşlemler.....	42
4. SONUÇ VE ÖNERİLER.....	53
KAYNAKLAR	55
ÖZGEÇMİŞ.....	57

KISALTMALAR

ANN	: Artificial Neural Network
API	: Application Programming Interface
APT	: Adaptive Progressive Thresholding
CFAR	: Constant False Alarm Rate
CNN	: Convolutional Neural Network
CPU	: Central Processing Unit
DPI	: Dots Per Inch
GPU	: Graphics Processing Unit
IR	: Infrared
IUDF	: Image Uniformity Description Factor
KNN	: K-Nearest Neighbor
LBP	: Local Binary Patterns
LGBPHS	: Local Gabor Binary Pattern Histogram Sequence
MLP	: Multi-Layer Perceptron
NDWI	: Normalized Difference Water Index
R-CNN	: Region Based Convolutional Neural Network
RGB	: Red Green Blue
ROI	: Reigions Of Interests
SAR	: Synthetic Aperture Radar
SCR	: Signal to Clutter Ratio
SIFT	: Scale-Invariant Feature Transform
SSD	: Single Shot Detector
SVM	: Support Vector Machine



ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Giriş (input) resimleri..... 16





ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Video kamera resimleri (Rahmani ve Behrad, 2011).....	2
Şekil 1.2 : SAR görüntüsü (Li ve diğ, 2016).....	3
Şekil 1.3 : Optik uydu görüntüsü (Li ve diğ, 2017).....	3
Şekil 2.1 : Yapay sinir ağı.	8
Şekil 2.2 : Konvolüsyonel sinir ağı mimarisi (Doğan and Türkoğlu, 2018).	8
Şekil 3.1 : Eğitimde kullanılan görüntü setleri (Rhammell, 2019).....	11
Şekil 3.2 : LabelImg programı ile görüntülerin gemi ismiyle etiketlenmesi.	11
Şekil 3.3 : Tensorboard kayıp grafiği.	12
Şekil 3.4 : Modellerin kayıp grafiklerinin karşılaştırılması.....	14
Şekil 3.5 : Modellerin kayıp grafiklerinin karşılaştırılması.....	15
Şekil 3.6 : Modellerin kayıp grafiklerinin karşılaştırılması.....	15
Şekil 3.7 : Resim-1'in tespit sonucu.	16
Şekil 3.8 : Resim-2'nin tespit sonucu.	17
Şekil 3.9 : Resim-3'ün tespit sonucu.	17
Şekil 3.10 : Resim-4'ün tespit sonucu.	18
Şekil 3.11 : Resim-5'in tespit sonucu.	18
Şekil 3.12 : Resim-6'nın tespit sonucu.	19
Şekil 3.13 : Resim-7'nin tespit sonucu.	19
Şekil 3.14 : Resim-8'in tespit sonucu.	20
Şekil 3.15 : Resim-9'un tespit sonucu.	20
Şekil 3.16 : Resim-10'un tespit sonucu.	21
Şekil 3.17 : Resim-11'in tespit sonucu.	21
Şekil 3.18 : Resim-12'nin tespit sonucu.	22
Şekil 3.19 : Resim-13'ün tespit sonucu.	22
Şekil 3.20 : Resim-1'in tespit sonucu.	23
Şekil 3.21 : Resim-2'nin tespit sonucu.	23
Şekil 3.22 : Resim-3'ün tespit sonucu.	24
Şekil 3.23 : Resim-4'ün tespit sonucu.	24
Şekil 3.24 : Resim-5'in tespit sonucu.	25
Şekil 3.25 : Resim-6'nın tespit sonucu.	25
Şekil 3.26 : Resim-7'nin tespit sonucu.	26
Şekil 3.27 : Resim-8'in tespit sonucu.	26
Şekil 3.28 : Resim-9'un tespit sonucu.	27
Şekil 3.29 : Resim-10'un tespit sonucu.	27
Şekil 3.30 : Resim-11'in tespit sonucu.	28
Şekil 3.31 : Resim-12'nin tespit sonucu.	28
Şekil 3.32 : Resim-13'ün tespit sonucu.	29
Şekil 3.33 : Resim-1'in tespit sonucu.	29
Şekil 3.34 : Resim-2'nin tespit sonucu.	30
Şekil 3.35 : Resim-3'ün tespit sonucu.	30
Şekil 3.36 : Resim-4'ün tespit sonucu.	31

Şekil 3.37 : Resim-5'in tespit sonucu.	31
Şekil 3.38 : Resim-6'nin tespit sonucu.	32
Şekil 3.39 : Resim-7'nin tespit sonucu.	32
Şekil 3.40 : Resim-8'in tespit sonucu.	33
Şekil 3.41 : Resim-9'un tespit sonucu.	33
Şekil 3.42 : Resim-10'un tespit sonucu.	34
Şekil 3.43 : Resim-11'in tespit sonucu.	34
Şekil 3.44 : Resim-12'nin tespit sonucu.	35
Şekil 3.45 : Resim-13'ün tespit sonucu.	35
Şekil 3.46 : Resim-1'in tespit sonucu.	36
Şekil 3.47 : Resim-2'nin tespit sonucu.	36
Şekil 3.48 : Resim-3'ün tespit sonucu.	37
Şekil 3.49 : Resim-4'ün tespit sonucu.	37
Şekil 3.50 : Resim-5'in tespit sonucu.	38
Şekil 3.51 : Resim-6'nin tespit sonucu.	38
Şekil 3.52 : Resim-7'nin tespit sonucu.	39
Şekil 3.53 : Resim-8'in tespit sonucu.	39
Şekil 3.54 : Resim-9'un tespit sonucu.	40
Şekil 3.55 : Resim-10'un tespit sonucu.	40
Şekil 3.56 : Resim-11'in tespit sonucu.	41
Şekil 3.57 : Resim-12'nin tespit sonucu.	41
Şekil 3.58 : Resim-13'ün tespit sonucu.	42
Şekil 3.59 : Eğitimde kullanılan ilave görüntü setleri (Gfeng, 2019).....	43
Şekil 3.60 : Kayıp grafiklerinin karşılaştırılması (Modelin ilk hali mavi, son hali yeşil renklidir).	44
Şekil 3.61 : Kayıp grafiklerinin karşılaştırılması (Modelin ilk hali mavi, son hali yeşil renklidir).	44
Şekil 3.62 : Resim-1'in tespit sonucu.	45
Şekil 3.63 : Resim-2'nin tespit sonucu.	45
Şekil 3.64 : Resim-3'ün tespit sonucu.	46
Şekil 3.65 : Resim-4'ün tespit sonucu.	46
Şekil 3.66 : Resim-5'in tespit sonucu.	47
Şekil 3.67 : Resim-6'nin tespit sonucu.	47
Şekil 3.68 : Resim-7'nin tespit sonucu.	48
Şekil 3.69 : Resim-8'in tespit sonucu.	48
Şekil 3.70 : Resim-9'un tespit sonucu.	49
Şekil 3.71 : Resim-10'un tespit sonucu.	49
Şekil 3.72 : Resim-11'in tespit sonucu.	50
Şekil 3.73 : Resim-12'nin tespit sonucu.	50
Şekil 3.74 : Resim-13'ün tespit sonucu.	51

DERİN ÖĞRENME YÖNTEMİ İLE OPTİK UYDU GÖRÜNTÜLERİNDEN GEMİ TESPİTİ

ÖZET

Deniz yollarının ticari/askeri alanda etkin şekilde kullanımı ve kontrolü; devletler için gün geçtikçe daha da artan ve önem arz eden bir husus haline gelmektedir. Bu önemli alanların gözetimi ve denetimi; gözlemci unsurun (gemi, insanlı/insansız hava aracı, vb.) olay yerine direkt olarak gitmesi ve alanı incelemesi suretiyle icra edilebilir, ancak bu şekilde bir yaklaşım ile bölgeye intikal ve kısıtlı görüş açısı/kapsama alanı ile gözlem işlemleri sonucunda zaman sarfiyatına sebebiyet verebileceği gibi savaş, terör veya kriz ortamlarında sıcak bölgeye yakınlığından dolayı gözlemci unsur büyük bir risk altında bulunacaktır. Tam bu noktada uzaktan algılama yöntemleri faydalı çözümler sunmakta, özellikle uydu görüntüleri ile yapılan uzaktan algılama ile operator güvenli bir şekilde, kendini ifşaa etmeden, kısa zamanda büyük alanları inceleyebilmekte ve aradığı unsuru tespit edebilmektedir.

Bu çalışma kapsamında, Tensorflow Nesne Tespiti API'sinin nesne tanıma üzerine halihazırda eğitilmiş modellerinin içerisinde gemi resmi bulunduran optik uydu görüntüleri ile eğitilmesi yolu ile optik uydu görüntülerinden gemi tespit eden algoritmalara ilave bir seçenek eklemektir. Gemi tespit metodu tasarlanırken, lisans sorunu yaşanmaksızın her ortamda kullanılabilmesi için sistemin açık kaynak koduna sahip olması, hızlı çalışması ve kullanımının kolay olması hedeflenmiştir.

Birinci bölümde, deniz yollarının önemi ve izlenme usulleri hakkında temel bilgilerden, bahse konu alan ile ilgili olarak uzaktan algılama yöntemlerinin kullanımından, yararından ve birbirleri ile karşılaştırılmasından, literatür araştırmasından, tezin konusundan, amacından ve kapsamından bahsedilmiştir.

İkinci bölümde, derin öğrenme ve kullanım alanları, yapay sinir ağları ve konvolüsyonel sinir ağları hakkında özet bilgiler verilmiştir.

Üçüncü bölümde, tez çalışması kapsamında kullanılan kütüphaneler, modelin eğitilmesi ve deneysel sonuçlardan bahsedilmiştir. Tezin amacı doğrultusunda gemi tespit yazılımının kaynak kodu Python 3.7 yazılım dili ile hazırlanmıştır. Python kütüphaneleri için Anaconda programı kurulmuştur. Derin öğrenme kütüphanesi olarak Google tarafından oluşturulmuş Tensorflow kütüphanesi, bilgisayarlı görü kütüphanesi olarak OpenCV kütüphanesi kullanılmıştır. Eğitilmiş model olarak; Tensorflow Object Detection Model Zoo'dan "faster rcnn inception v2_coco, faster rcnn resnet101_coco, ssd_inception_v2_coco" modelleri kullanılmıştır. Modellerin eğitiminde kullanılan gemi görüntü seti, çözünürlüğü 80 * 80 * 3, RGB bandı, 96 dpi olan 2085 adet görüntüden oluşmaktadır. Modellerin eğitimi Tensorboard arayüzü üzerinden kayıp (loss) grafikleri vasıtasıyla takip edilmiştir. Modeller arasındaki gemi tespit performansı sergileyen modelin gemi tespit performansının artırılması için değişken çözünürlüğe sahip (987 * 804 * 3, 1155*820*3, vb.), ilk gemi görüntü setinin çözünürlüğünden yaklaşık 10 kat büyük, RGB bandı, 96 dpi olan 1056 adet görüntüden oluşan görüntü seti ile model tekrar eğitilmiştir. Bahse konu modelin

gemi tespit performansında iyileşme olduğu gözlenmiştir.
Dördüncü bölümde sonuçlar ve öneriler kısmı bulunmaktadır.



SHIP DETECTION BY OPTICAL SATELLITE IMAGES WITH DEEP LEARNING METHOD

SUMMARY

The effective use and control of maritime routes in the commercial / military area is an increasing and important need for states. As a result of the rapid development in the shipbuilding industry, the number of ships and their size increase day by day. This poses a major threat to safe passage, especially in narrow traffic channels. In order to ensure the safety of the cruises within this scope; some ports allocate special anchorages and evacuation channels, but these canals and anchorages can be used illegally by ordinary ships. In addition, as another type of illegal use; sometimes fishing vessels can occupy the main channels irregularly as fishing areas. Due to the recent decline in the amount of fish reserves in the world, ship detection has become a more effective and efficient method of monitoring fishermen to identify illegal fishing activities. Remote monitoring of the regions of interest in terms of detection and tracking of foreign/enemy elements threatening the security of the area in the seas makes a major contribution to the early warning of military intelligence, crime and anti-terrorism units. It is also useful for detecting the lost ships, boats, aeroplanes, debris, containers, etc.

For these issues; the observer can check these areas directly by going there but it will cost time and in case of terror or war the observer will be in a huge risk. At this point "remote sensing" offers a great solution. Surface of the sea can be observed by radar, video cameras, optical satellite imagery or synthetic aperture radar (SAR) imagery. In case of war, active transmission devices can be dangerous for observer because enemy electronic warfare devices can detect your coordinates according to your transmission. In the war environment, this is never desired. So that devices like video cameras, satellite imageries are much more useful. Compared to satellite imaging systems, the viewing distance and field of view of the video cameras are more limited, more susceptible to damage or hitting by enemy.

Compared to optical satellite imagery, SAR imagery is much more useful at night because it doesn't need a lightning source but its images are usually with high-level speckles and, insensitive to wood materials. Compared to SAR imagery, optical satellite imagery has higher resolution and more detailed information but it needs lightning source so it can't be used at night. Deep learning is a kind of machine learning method that enables computers to learn from acquired experience and information and to extract useful patterns from raw data. It can learn information from pictures, texts or sounds. Based on the knowledge it learned, it can develop more advanced works, or even can work ahead of human abilities. It has a wide range of applications from driverless vehicles to satellite and defense systems, medical research to industrial automation systems, electronic devices and advertising. Nowadays, thanks to deep learning, autonomous vehicles are developed which can stop when they see the red light, park itself when it finds empty parking lot, follow the vehicles in front and rear, decide how to behave against objects that

suddenly come to the road. We know that objects in the world can be seen very clearly from satellites. What would happen if we could transfer this ability to the automated system? Deep Learning seems to have done this a lot. In medical research (especially in cancer screening), it is targeted to automatically scan items that can escape the eyes of technicians. It is aimed to detect abnormal situations with deep learning. With the help of deep learning in the field of Industrial Automation, it is planned to reduce the damage caused by occupational safety and work accidents. It is aimed at working by hearing and interpreting speech rather than manual intervention in the management of electronic devices.

Many deep learning methods use architectures called Artificial Neural Networks (ANN). It is the system developed for the purpose of technically realizing the ability to discover, create and derive new information by means of learning which is the most basic feature of human brain. This system consists of artificial cells that run in parallel and consist of hierarchically connected network nodes. ANN consists of input, hidden and output layers. Although the number of neurons in each layer can be more than one, the layer that has the most impact on the learning ability of the network is the hidden layer.

Convolutional neural network (CNN) in machine learning is a deep feed artificial neural network which has been successfully applied to the analysis of images. It is a multi-layer network architecture developed from the classic neural network process. CNN usually consists of input layer, convolution layer, pooling layer, full connection layer and output layer. It can execute feature extraction and mapping through fast training, and has high prediction accuracy, so it is often applied to classification and estimation.

The purpose of this work is to add an additional option to the ship detection algorithms from optical satellite imagery by training the Tensorflow Object Detection Application Programming Interface (API). When the ship detection method is designed, it is aimed that the system should have open source code, work quickly and be easy to use.

In the first chapter, the basic information about the importance and monitoring methods of maritime routes, the usage of remote sensing methods, the benefits and comparison with each other, the literature research, the subject, the purpose and the scope of the thesis are mentioned.

In the second chapter, deep learning, ANN and CNN are mentioned.

In the third chapter, the libraries used in the thesis work, the training of the models and the experimental results are mentioned. When the ship detection method is designed, it is aimed that the system should have open source code, work quickly, can be trained, can be operated via a average laptop and be easy to use. For this purpose, source code is written via Python 3.7.. Anaconda is installed for Python libraries. Tensorflow (made by Google) is installed which is an open source machine learning library for research and production. OpenCV is installed for computer vision library. Training a single “object detection model” can take weeks to months, even with GPU. For CPU it will be done much more long time.

Therefore, usage of an already trained model and training it for your object (for this work it is ship) is much more useful and time saver. For this purpose, Tensorflow Object Detection API’s models (trained for object detection by Google) are used for already trained object detection model. They are trained with optical satellite images

to detect ships. “faster rcnn inception v2_coco”, “faster rcnn resnet101_coco” and “ssd_inception_v2_coco” have been used as trained models from Tensorflow Object Detection Model Zoo. These models are trained by Google via Coco Image Dataset which is a large-scale object detection, segmentation, and captioning dataset. Next; a number of additional libraries, such as "matplotlib, pandas" are installed. Then ship image sets are downloaded and images which has ship labelled as “ship (in Turkish: Gemi)” with LabelImg. Non-ship images are erased. Dataset is consist of 2085 images whose resolution is 80*80*3, RGB band., 96 dpi. Training of the models is observed through loss graphs via Tensorboard interface. In order to increase the ship detection performance of the model which has the best ship detection performance between the models, it is retrained with additional image set. The additional image set consists of 1056 images with 96 dpi, in the RGB band with variable resolution (987 * 804 * 3, 1155 * 820 * 3, etc.) which are approximately 10 times larger than the resolution of the first ship image set. It is observed that the ship detection performance of this model is improved.

The fourth section contains the results and recommendations.





1. GİRİŞ

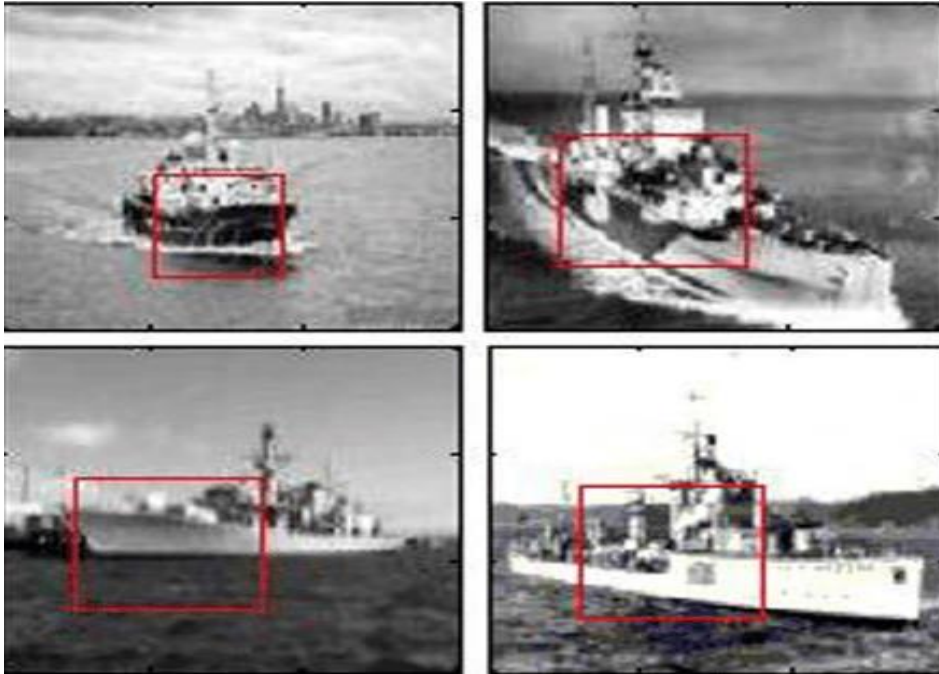
Deniz yollarının ticari/askeri alanda etkin şekilde kullanımı ve kontrolü; devletler için gün geçtikçe daha da artan ve önem arz eden bir husus haline gelmektedir. Gemi üretim endüstrisindeki hızlı gelişim sonucunda gemilerin sayısı ve tonajları gün geçtikçe artmaktadır. Bu durum özellikle yoğun trafiğe sahip dar geçiş kanallarındaki güvenli intikaller için büyük tehdit oluşturmaktadır. Bu kapsamdaki seyirlerin güvenliğini sağlamak amacıyla bazı limanlar özel intikal kanalları ve demirleme alanları tahsis etmekte ancak bahse konu kanallar ve demirleme alanları sıradan gemiler tarafından yasadışı olarak izinsizce kullanılabilir. Diğer bir yasadışı kullanım türü olarak balıkçı tekneleri; gemilerin intikal ettikleri ana kanalları balık tutmak amacıyla düzensiz biçimde işgal edip gemi trafiğini tehlikeye sokabilmekte ve kaza tehditi oluşturabilmektedirler (Du ve diğ, 2017). Dünyadaki balık rezerv miktarlarının son yıllardaki düşüşü sebebiyle gemi tespiti yasadışı balıkçılık faaliyetlerini tespit etmek için balıkçıları izleme hususunda çok daha etkin ve verimli bir yöntem haline gelmiş bulunmaktadır (Zhu ve diğ, 2010). Karasularındaki veya uluslararası sulardaki bölge güvenliğini tehdit eden yabancı unsurların önceden tespit edilmesi, takip edilmesi ve gerekli müdahalenin yapılması hususunda ilgilenilen bölgelerin uzaktan izlenmesi; askeri istihbarat, suçla ve terörle mücadele birimlerinin erken uyarılması açısından büyük katkı sağlamaktadır (Yaman ve Asari, 2007). Ayrıca kayıp gemileri, gemileri, uçakları, çöpleri, konteynerleri vb. tespit etmek için bahse konu yöntem çok kullanışlıdır (Soni ve diğ, 2015).

Bu önemli alanların gözetimi ve denetimi; gözlemci unsurun (gemi, insanlı/insansız hava aracı, vb.) olay yerine direkt olarak gitmesi ve alanı incelemesi suretiyle icra edilebilir, ancak bu şekilde bir yaklaşım ile bölgeye intikal ve kısıtlı görüş açısı/kapsama alanı ile gözlem işlemleri sonucunda zaman sarfiyatına sebebiyet verebileceği gibi savaş, terör veya kriz ortamlarında sıcak bölgeye yakınlığından dolayı gözlemci unsur büyük bir risk altında bulunacaktır. Tam bu noktada uzaktan algılama yöntemleri faydalı çözümler sunmaktadır. Deniz yüzeyi; kaideye monteli radarlar, video kameralar, optik uydular veya sentetik açıklıklı radarlı (SAR) uydular

ile gözlenebilir.

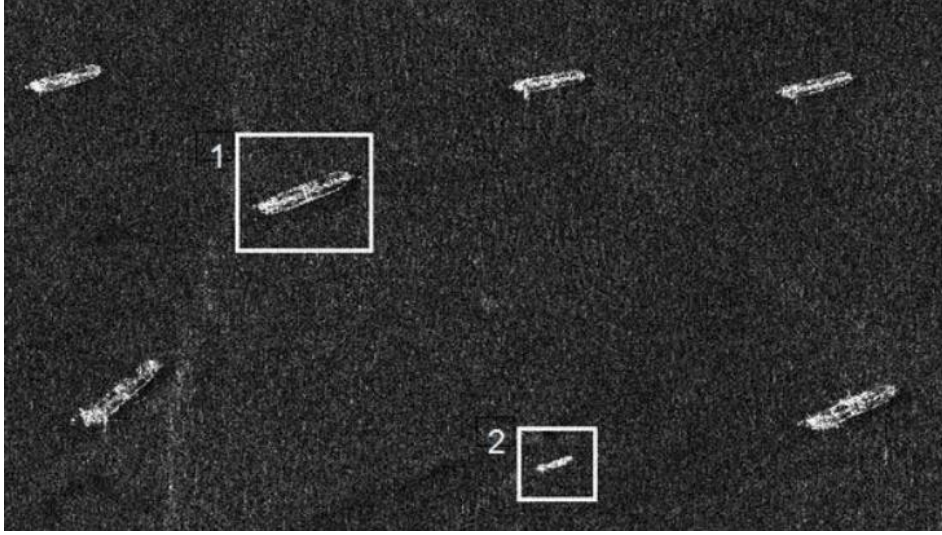
Savaş şartlarında, gözlemcinin aktif iletim (transmisyon) yapan cihazları gözlemcinin kendisi için büyük tehdit oluşturabilmektedir, çünkü düşman unsurların elektronik destek sistemleri gözlemcinin ortama yaydığı elektromanyetik dalgalardan gözlemcinin yönünü ve yerini tespit edebilmektedir. Gizliliğin büyük öneme sahip olduğu savaş ortamında böyle bir durum asla istenilmez. Bu şartlar altında video kameralar ve uydu sistemleri daha kullanışlı hale gelmektedir.

Uydu görüntüleme sistemleri ile karşılaştırıldığında, video kameraların görüş mesafesi ve görüş alanı daha sınırlıdır (Şekil 1). Hasar almaya veya düşman unsur tarafından vurularak etkisiz hale getirilmeye daha yatkındır.

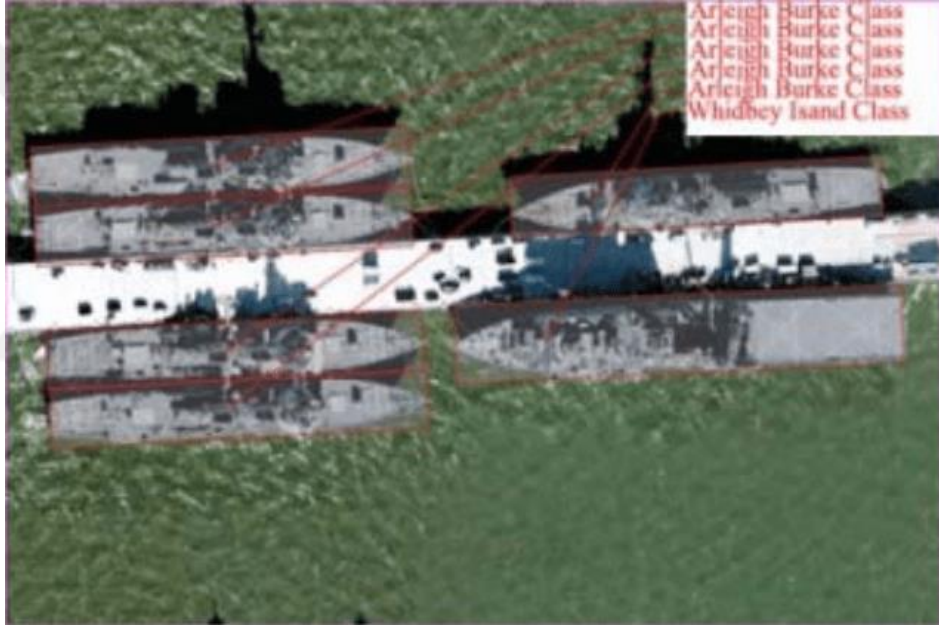


Şekil 1.1 : Video kamera resimleri (Rahmani ve Behrad, 2011).

Optik uydu görüntülerine kıyasla, SAR görüntüleri (Şekil 1.2) gece için çok daha kullanışlıdır çünkü aydınlatma kaynağına ihtiyaç duymaz ancak görüntüleri genellikle yüksek seviyeli beneklidir ve ahşap malzemelere karşı duyarsızdır. SAR görüntüleri ile karşılaştırıldığında, optik uydu görüntüleri daha yüksek çözünürlüğe ve daha ayrıntılı bilgilere sahiptir (Yang ve diğ, 2015). Aydınlatma kaynağına ihtiyaç duyarlar, bu sebeple gece kullanılamazlar (Şekil 1.3).



Şekil 1.2 : SAR görüntüsü (Li ve diğ, 2016).



Şekil 1.3 : Optik uydu görüntüsü (Li ve diğ, 2017).

1.1 Tezin Amacı

Bu çalışmanın amacı, Tensorflow Nesne Tespiti API'sini eğiterek optik uydu görüntülerinden gemi algılama algoritmalarına ilave bir seçenek eklemektir. Gemi tespit metodu tasarlanırken, lisans sorunu yaşanmaksızın her ortamda kullanılabilmesi için sistemin açık kaynak koduna sahip olması, hızlı çalışması ve kullanımının kolay olması hedeflenmiştir.

1.2 Literatür Araştırması

Du ve diğ. (2017) tarafından; Hu değişmeyen anları (invariant moments) ve ART katsayıları (ART coefficients) gemilerin video kamera görüntülerinin bölgesel şekil özelliğinin özdeğerler (eigenvalues) olarak çıkarılması için kullanılmıştır. K-En Yakın Komşu (K-Nearest Neighbor (KNN)) yöntemi ile seçilen özellikler, gemi özellik kütüphanesi ile eşleştirilmiştir. Kütüphanedeki en benzer özdeğere göre gemi sınıflaması yapılmıştır.

Zhu ve diğ. (2010) tarafından uzay-tabanlı optik görüntüden gemilerin tespiti üzerinde çalışma yapılmıştır. Bulutların, okyanus dalgalarının ve küçük adaların olumsuz etkilerini ortadan kaldırmak için basit şekil analizi (shape analysis), görüntü bölümlenme (image segmentation) ve yarı denetimli bir hiyerarşik sınıflandırma yöntemi kullanılmıştır.

Yaman ve Asari (2007) tarafından hem IR bandında hem de görünür bandta çalışan, oldukça karışık (cluttered) arka planlı video kamera görüntülerinden, küçük boyutlu tekneleri tespit edebilen bir sistem geliştirilmiştir. IR bandı için Graph-cut segmentasyon algoritması ve görünür bant için Uyarlamalı İlerici Eşikleme (Adaptive Progressive Thresholding (APT)) segmentasyon algoritması kullanılmıştır.

Soni ve diğ. (2015), optik uydu görüntüsünden gemilerin, batıkların ve uçakların tespiti üzerinde çalışmışlardır. Şekil ve boyut değerlendirmesi ile üç boyutlu renkli piksel (24-bit) tabanlı bir yaklaşım kullanılarak yüksek bir tespit oranı hedeflenmiştir. MATLAB programı kullanılmıştır.

Rahmani ve Behrad (2011) tarafından Yerel Gabor İkili Desen Histogram Dizisi (Local Gabor Binary Pattern Histogram Sequence (LGBPHS)) yöntemi ile video kamera görüntüsünden gemilerin algılanması hakkında çalışma yapılmıştır. Sınıflandırma için Çok Katmanlı Perceptron (Multi-Layer Perceptron (MLP)) ve Destek Vektör Makinası (Support Vector Machine (SVM)) kullanılmıştır.

Yang ve diğ. (2015) tarafından optik uydu görüntüsünün görsel arama mekanizmasına dayanan bir gemi tespit çalışması yapılmıştır. Aday bölgeler küresel karşıtlık modeliyle çıkarılmıştır. Bulut, ada, dalga vb. yanlış alarmlar, şekil ve komşuluk benzerliği analizi ile yok edilmiştir.

Li ve diğ. (2016) tarafından icra edilen çalışmada iki taraflı filtre aracılığıyla yüksek

çözünürlüklü TerraSAR görüntülerinin belirginlik haritasını oluşturulup homojenize edilmiştir. Daha sonra düşük seviyeli sinyal / dağınıklık oranı (Signal to Clutter Ratio (SCR)) şartında, kaba/çok dalgalı denizde, gemi tespit etmek için sabit yanlış alarm oranı (Constant False Alarm Rate (CFAR)) algılama algoritması kullanılmıştır.

Li ve diğ. (2017) tarafından kıyıdaki gemilerin otomatik olarak algılanması ve tanımlanmasına ilişkin büyük uydu görüntüleri üzerinde bir çalışma yapılmıştır. Port şablonu ile test görüntüsü arasındaki coğrafi koordinat hatası, ölçek değişmeyen özellik dönüşümü (Scale-Invariant Feature Transform (SIFT)) kaydı ile elimine edilmiştir. Daha sonra, ilgi alanları (Regions Of Interests (ROI)) bölgeleri çıkarılmış ve yatay olarak hizalanmıştır. Eğitilmiş multimodeller kaydırma penceresi yöntemi ile ROI'lere uygulanmış ve gemi adayları ortaya çıkartılmıştır. Son olarak, adaylar füzyon stratejisi kullanılarak birleştirilmiştir.

Beşbinar ve Alatan (2015) tarafından optik uydu görüntüsü ile kıyı bölgesinde gemi tespit metodu geliştirmiştir. Mevcut genel yükseklik verilerinin sıfır seviyesini kullanarak normalize edilmiş fark su endeksi (Normalized Difference Water Index (NDWI)) eşiklenerek bir ilk maske oluşturulmuştur. Segmentasyon sonucunun sınırı, grafik kesim algoritması (graph-cut algorithm) ile daha da geliştirilmiştir. Sonuçta ortaya çıkan sınır çizgisi, insan yapımı limanları temsil eden çizgi parçalarını bulmak için kullanılmıştır. Birleştirildikten ve uygun şekilde ortadan kaldırıldıktan sonra, bu çizgi bölümleri liman alanını çıkarmak için kullanılmıştır. Ardından ikili maskenin geri kalan kısımları şekillerine göre gemi olarak test edilmiştir.

Arguedas (2015) tarafından optik uydu görüntüsünde doku tabanlı gemi sınıflandırma sistemi üzerinde çalışma yapılmıştır. Kafa, ana güverte ve kış olmak üzere gemi üç bölüme ayrılmıştır. Bu üç bölümün bireysel özellikleri, Yerel İkili Desenler (Local Binary Patterns (LBP)) yöntemi kullanılarak alınmış ve kütüphanede saklanmıştır. Kütüphanedeki ve test görüntüsündeki doku özellikleri en yakın komşu yöntemi ile karşılaştırılmıştır. Ardından, geminin üç bölümünün özellikleri birleştirilmiş ve yeniden değerlendirilmiştir. Sınıflandırma; test görüntüsü ile kütüphane arasındaki en yakın gemi karakteristiğine göre yapılmıştır.

Song ve diğ. (2014) tarafından Yang ve diğ. (2015)'nin çalışmasına benzer bir çalışma yapılmış olmasına rağmen, çalışmanın belirli sabit boyutlu görüntüler yerine tüm boyutlara ve özellikle yüksek boyutlu uydu görüntülerine uygulanabileceği

belirtilmiştir. Algoritmanın başlangıcında, büyük ölçekli uydu görüntüsü küçük parçacıklara bölünmüş, her parçacık ayrı olarak işlenmiş ve SVM bu parçacıklarda gemi olup olmadığına karar vermiştir.

Li ve diğ. (2017) tarafından icra edilen çalışmada SAR görüntülerindeki gemileri tespit etmek için Görüntü Düzgünlüğü Tanımlama Faktörü (Image Uniformity Description Factor (IUDF)) bir doku özelliği olarak kullanılmıştır.



2. DERİN ÖĞRENME

2.1 Tanım

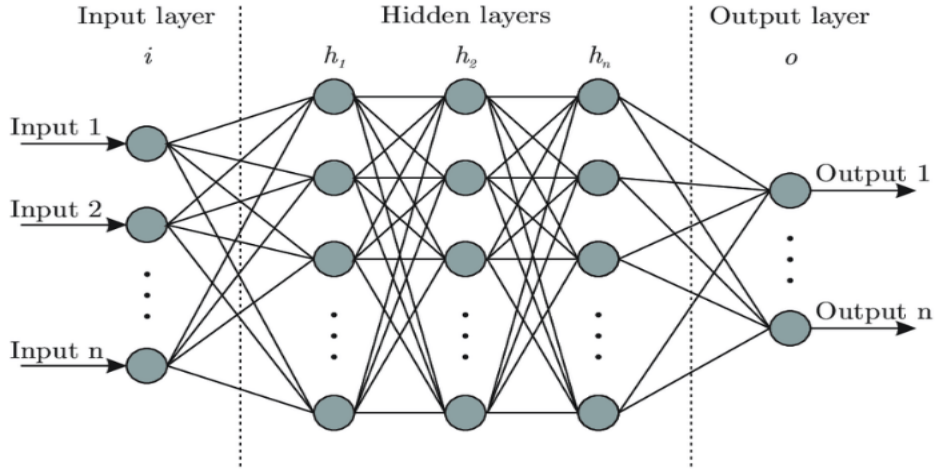
Derin öğrenme; bilgisayarların edinilmiş tecrübe ve bilgilerden istenilen şeyleri öğrenmelerini ve ham verilerden faydalı kalıpları çıkarmasını sağlayan bir tür makine öğrenmesi yöntemidir. Resimlerden, metinlerden veya seslerden bilgi öğrenebilir. Öğrendiği bilgiye istinaden daha ileri seviye işler geliştirebilir, hatta insanoğlunun yeteneklerinin çok ilerisinde işler çıkartabilir.

2.2 Kullanım Alanları

Günümüzde derin öğrenme; sürücüsüz araçlardan uydu ve savunma sistemlerine, tıbbi araştırmalardan endüstriyel otomasyon sistemlerine, elektronik cihazlardan reklam sektörüne kadar geniş bir yelpazede kullanım alanına sahiptir. Hadad, Y. (2018) tarafından derin öğrenmenin kullanıldığı 31 adet farklı alanla ilgili, içinde görsel barındıran güzel bir bilgilendirme çalışması yapılmıştır.

2.3 Yapay Sinir Ağları

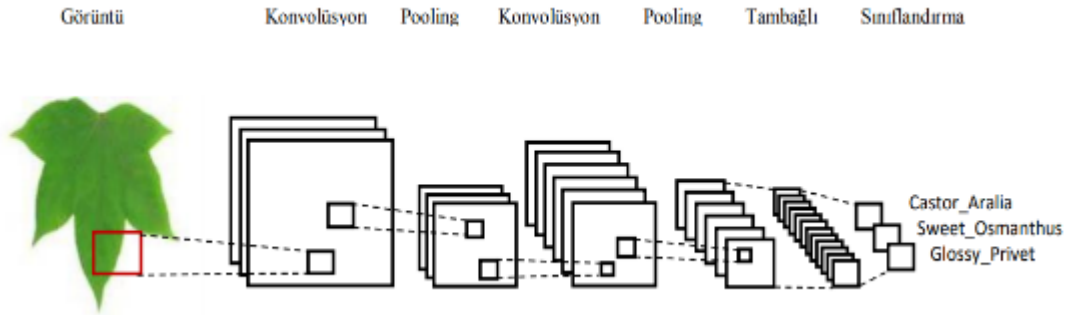
Birçok derin öğrenme yönteminde, Yapay Sinir Ağları adı verilen mimariler kullanılmaktadır (Şekil 2.1). Bu ağlar, girdi (input), gizli (hidden) ve çıktı (output) katmanlarından oluşmaktadır. Her katmandaki nöronların sayısı birden fazla olabilsede, ağın öğrenme yeteneği üzerinde en fazla etkiye sahip olan katman gizli katmandır (Karakaya ve Çemtay, 2017).



Şekil 2.1 : Yapay sinir ağı.

2.4 Konvolüsyonel Sinir Ağları

Makine öğreniminde Konvolüsyonel Sinir Ağı (Convolutional Neural Network (CNN)), görüntülerin analizinde başarıyla uygulanan derin bir yapay sinir ağıdır. Klasik sinir ağı sürecinden geliştirilmiş çok katmanlı bir ağ mimarisidir. CNN genellikle giriş (input) katmanı, evrişim (convolution) katmanı, havuz (pooling) katmanı, tam bağlantı (full connection) katmanı ve çıkış (output) katmanından oluşur (Şekil 2.2). Hızlı eğitim yoluyla, özellik çıkarma ve haritalama yapabilir ve yüksek tahmin doğruluğuna sahip olduğundan genellikle tanıma, tespit, sınıflandırma ve tahmin işlemlerinde kullanılmaktadır (Shen and Wang, 2018).



Şekil 2.2 : Konvolüsyonel sinir ağı mimarisi (Doğan and Türkoğlu, 2018).

3. GEMİ TESPİT METODU

3.1 Amaç

Gemi tespit metodu tasarlanırken, lisans sorunu yaşanmaksızın her ortamda kullanılabilmesi için sistemin açık kaynak koduna sahip olması, hızlı çalışması, eğitilebilmeye müsait olması ve kullanımının kolay olması amaçlanmıştır.

3.2 Kullanılan Programlar ve Kütüphaneler

Tezin amacı doğrultusunda gemi tespit yazılımının kaynak kodu Python 3.7 yazılım dili ile hazırlanmıştır. Python kütüphaneleri için Anaconda programı kurulmuştur. Derin öğrenme kütüphanesi olarak Google tarafından oluşturulmuş Tensorflow kütüphanesi, bilgisayarlı görü kütüphanesi (computer vision library) olarak OpenCV kütüphanesi kullanılmıştır.

Nesnelerin tespiti için aranılan nesnelerin özelliklerinin saklandığı bir özellik kütüphanesine ihtiyaç duyulur. Bu kütüphane yeni baştan oluşturulabileceği gibi daha önceden eğitilmiş bir kütüphane kullanılabilir. Nesne algılama modeli bu kütüphaneyi içinde barındırıp bu kütüphane ile beraber eğitilebilir. Tek bir nesne algılama modelinin eğitimi, Grafik İşlemci Ünitesi (Graphics Processing Unit (GPU))'nde bile haftalar veya aylar sürebilir. Bahse konu eğitim, Merkezi İşlem Birimi (Central Processing Unit (CPU)) için çok daha uzun zamanda yapılabilir. Önceden eğitilmiş bir modelin kullanımı ve tespit edilmek istenilen nesne için eğitilmesi ise çok daha faydalıdır ve zamandan tasarruf edilmesini sağlar. Bu tez çalışması kapsamında, Google tarafından eğitilmiş olan Tensorflow Nesne Algılama Uygulama Programlama Arabirimi (Tensorflow Object Detection Application Programming Interface (API)), önceden eğitilmiş nesne tespit modeli olarak kullanılmış olup içlerinde gemi barındıran optik uydu görüntüleri ile eğitilmiştir (Tensorflow Object Detection API, 2019).

3.3 Model Seçimi

Tensorflow Object Detection API (2019)'ya ait Tensorflow Object Detection Model Zoo (2019) listesinde Tensorflow'un paylaştığı önceden eğitilmiş modellerin listesi bulunmaktadır.

Bahse konu modeller Google tarafından, büyük ölçekli nesne algılama, bölümlendirme ve resim yazısı (captioning) veri kümesi olan Coco Image Data Set kullanılarak eğitilmiştir. Modellerin listesinde “Speed (ms)” ve “COCO mAP^[^1]” olmak üzere iki ölçüt bulunmaktadır. “Speed (ms)” değeri arttıkça modelin işlem süresi daha uzun olmaktadır. “COCO mAP^[^1]” değeri arttıkça modelin tespit oranı artmakta ancak kullandığı hafıza miktarı artmaktadır. Bu kapsamda, çalışılan platformun hafıza miktarının müsaade ettiği oranda, düşük “Speed (ms)” değerli ve yüksek “COCO mAP^[^1]” değerli modellerin seçilmesinin performans açısından uygun olacağı değerlendirilmektedir.

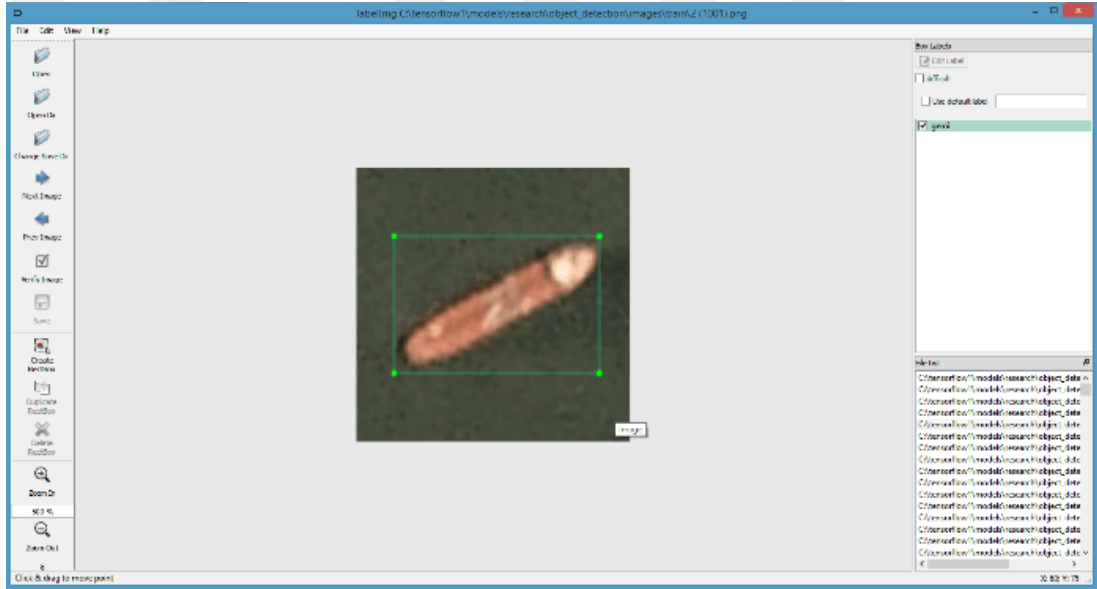
Eğitilmiş model olarak; “faster rcnn inception v2_coco, faster rcnn resnet101_coco, ssd_inception_v2_coco” modelleri seçilmiş, farklı değerlere sahip modellerin performanslarının karşılaştırılması hedeflenmiştir. Bu tez çalışmasının yapıldığı platformun ekran kartı hafızasının daha yüksek isabet oranına sahip modeller için yeterli olmamasından ötürü bahse konu modeller ile çalışma yapılmamıştır.

3.4 Eğitim Görüntü Setleri

Modelin belirlenmesine müteakiben modelin eğitiminde kullanılacak gemi görüntü setleri internet ortamından indirilmiştir (Rhammell, 2019). Görüntü seti, çözünürlüğü 80 * 80 * 3, RGB bandı, 96 dpi olan 2085 adet görüntüden oluşur (Şekil 3.1). İçinde gemi görüntüsü bulunduran resimler LabelImg programı vasıtasıyla “gemi” ismi ile etiketlenmiştir (LabelImg, 2019). Data sette bulunan bir resim üzerindeki etiketleme işlemi Şekil 3.2’de gösterilmiştir. İçinde gemi görüntüsü olmayan resimler silinmiştir. Görüntülerin % 80’i eğitim (training) görüntüsü, % 20’si test görüntüsü olarak gruplandırılmıştır.



Şekil 3.1 : Eğitimde kullanılan görüntü setleri (Rhammell, 2019).



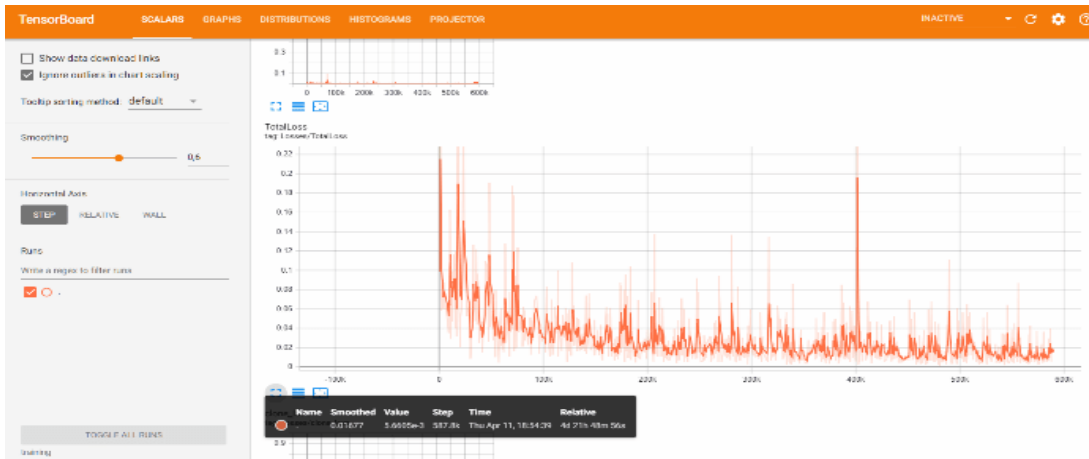
Şekil 3.2 : LabelImg programı ile görüntülerin gemi ismiyle etiketlenmesi.

3.5 Modelin Eğitilme Prensibi

Eğitim basamağında model; gemi etiketli eğitim resimleri ile kendisini eğitmekte ve test görüntülerindeki gemileri tespit etmeye çalışmaktadır. Tespit sonuçları ile test resimlerindeki işaretlenmiş gemiler karşılaştırılmakta, daha iyi tespit sonucu için model otomatik olarak nöron katsayılarını güncellemek suretiyle eğitimi tekrarlamakta ve en iyi tespit performansına ulaşmaya çalışmaktadır.

3.6 Kayıp (Loss) Grafiği

Eğitim süreci, kayıp grafiği ile kontrol edilir. Model kütüphanesi eğitilirken, kayıp grafiğinin kademeli olarak düşmesi beklenilmektedir. Eğer düşme işlemi durursa, kütüphane ya öğrenmeyi kesti demektir ya da öğrenebileceği her şeyi öğrenmiş demektir. Bu şartlar altında eğitim kullanıcı tarafından durdurulabilir. Örnek eğitim kapsamında model kütüphanesi 587000 kez eğitilmiş olup kayıp grafiğinin düşme işleminin azaldığı, nerdeyse yatay bir bandta seyrettiği gözlemlenmiş olup kütüphanenin öğrenebileceği şeyleri öğrendiği değerlendirilmiş ve eğitim durdurulmuştur (Şekil 3.3).



Şekil 3.3 : Tensorboard kayıp grafiği.

3.7 Modellerin Hiperparametreleri

“faster rcnn inception v2_coco” modelinin öğrenim oranı (learning rate) 0.0002, parça (batch) sayısı 1, eğitim adımı (training step) sayısı 587316’dır.

Aynı modelin öğrenim oranı 10 kat azaltılmış ve tespit performansının nasıl değiştiği gözlenmiştir.

“faster rcnn resnet101_coco” modelinin öğrenim oranı 0.0003, parça sayısı 1, eğitim adımı sayısı 490213’tür. Bahse konu model sayesinde daha yüksek bir öğrenim oranı ile tespit performansının nasıl değiştiği gözlenmiştir.

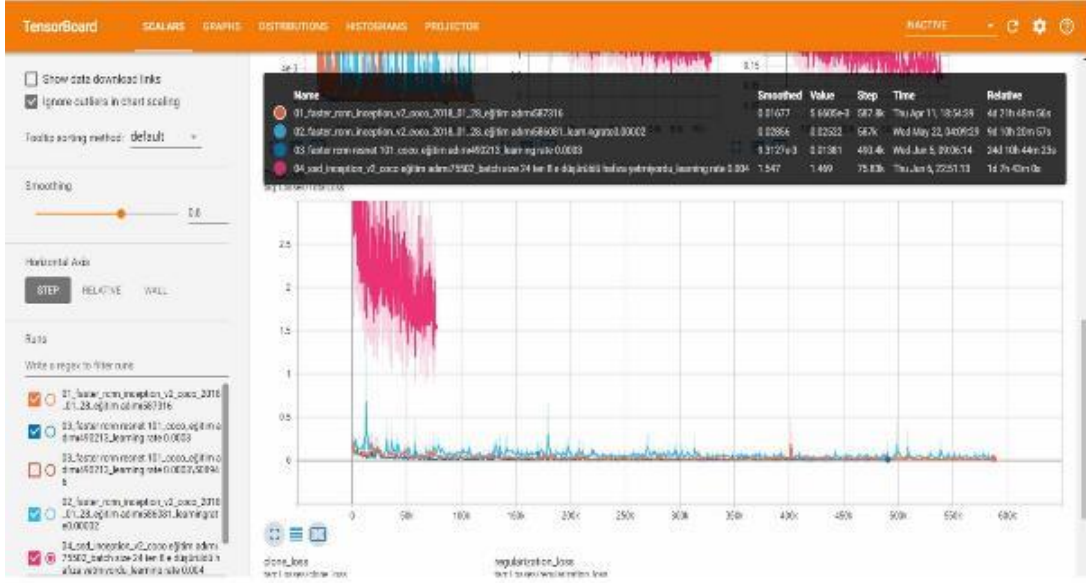
“ssd_inception_v2_coco” modelinin öğrenim oranı 0.004, parça sayısı 8, eğitim adımı sayısı 75502’dir. Bahse konu model; “single shot detector (ssd)” olup işlem hızı daha yüksek ancak tespit performansı Faster R-CNN’e nispeten daha düşüktür. Öğrenim oranı ne kadar düşük ise eğitim o kadar hassas olur ancak aynı oranla uzun süreli hale

gelir ve şayet öğrenim oranı kayıp grafiklerinin azalma istikametindeki “local minimumları” aşamayacak kadar küçük olursa model kendini geliştirebileceği en iyi seviyeye gelemeyebilir. Öğrenim oranı yüksek olursa eğitim o kadar hızlı gerçekleşir ancak tespit hassasiyeti düşebilir. İlâveten, model kendini geliştirebileceği en iyi seviyeye gelmeye çalışırken nöron katsayılarını değiştirdiği için yüksek orandaki değişim sonucu en iyi seviyeden uzaklaşılma ihtimali bulunmaktadır. Bu sebeple optimal bir öğrenim oranı seçimi modelin başarısı için önem arz etmektedir.

Parça sayısı; modelin aynı anda kaç adet resim ile kendini eğitebildiği rakamdır. Ne kadar yüksek olursa modelin tespit oranı artar, ancak ekran kartının belleğinin kullanım oranı aynı oranla artar. Ne kadar düşük olursa, modelin resimlerdeki gürültüyü ezberleme ihtimali artar. Bu sebeple donanımın kapasitesinin müsaade ettiği sürece; eğitim görüntüleri sayısı ile 1 sayısı arasında, 2 sayısının katları olacak şekilde (örneğin 16, 32, vb.) bir sayı belirlemek gerekmektedir. Bu çalışmanın yapıldığı platformun ekran kartının hafızasının yeterli olmaması sebebiyle üç adet modelin parça sayısı 1 ile sınırlı tutulmuştur.

3.8 Kayıp Grafiklerinin Karşılaştırılması

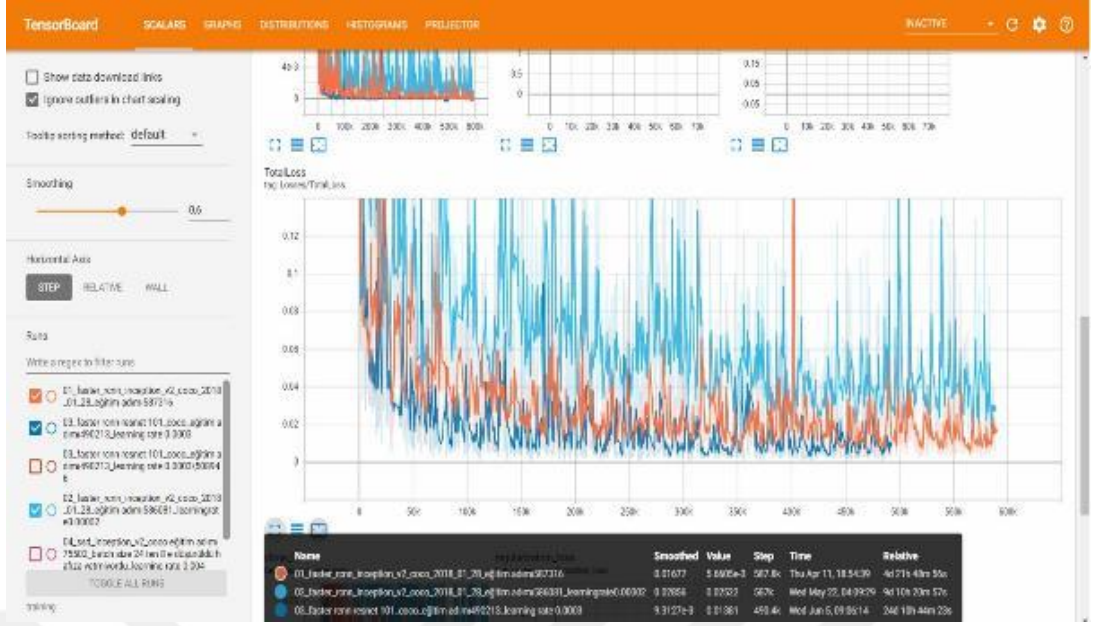
Modellerin eğitimi devam ederken kayıp grafikleri birbirleri ile karşılaştırılmıştır (Şekil 3.4). Grafikte pembe renk ile gösterilen “ssd_inception_v2_coco” modelinin eğitiminin diğer modellere kıyasla daha yüksek değer aralığında seyrettiği gözlenmiştir. Bu sebeple diğer modellere karşılaştırıldığında daha kötü performans sergileyeceği değerlendirilmiş olup bahse konu modelin eğitimi 75502’inci adımda sonlandırılmıştır.



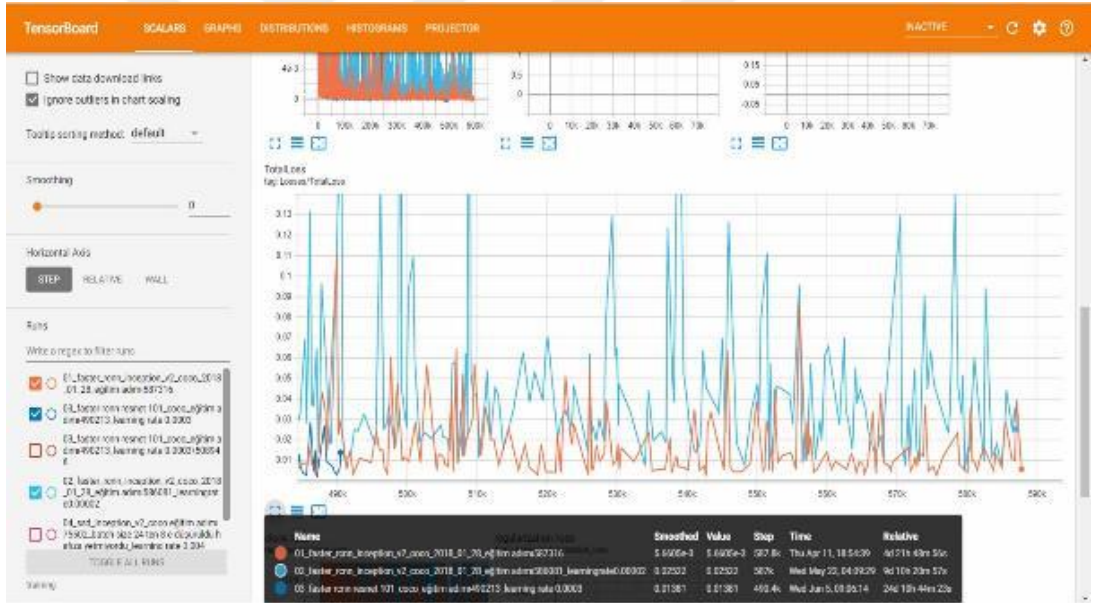
Şekil 3.4 : Modellerin kayıp grafiklerinin karşılaştırılması.

Diğer modellerin kayıp grafikleri incelendiğinde; eğitim süreci boyunca “faster rcnn resnet101_coco” modelinin (koyu mavi) sıfır kayıp seviyesine en yakın model olduğu gözlenmiş, ardından “faster rcnn inception v2_coco” modelinin öğrenim oranı 0.0002 olan halinin geldiği (turuncu) ve en son olarak “faster rcnn inception v2_coco” modelinin öğrenim oranının on kat az olduğu halinin (açık mavi) geldiği gözlenmiştir (Şekil 3.5).

Modellerin en sonuncu eğitim basamaklarına geldiği an itibariyle sıfır kayıp seviyesine en yakın modelin “faster rcnn inception v2_coco” modelinin öğrenim oranı 0.0002 olan hali (turuncu) olduğu, ardından “faster rcnn resnet101_coco” modelinin (koyu mavi) geldiği ve en son olarak “faster rcnn inception v2_coco” modelinin öğrenim oranının on kat az olduğu halinin (açık mavi) geldiği, bahse konu değerlerin birbirlerine çok yakın olduğu, en düşük olanın değerinin 0.0056 ve en yüksek değerinin 0.0252 olduğu gözlenmiştir (Şekil 3.6).



Şekil 3.5 : Modellerin kayıp grafiklerinin karşılaştırılması.



Şekil 3.6 : Modellerin kayıp grafiklerinin karşılaştırılması.

3.9 Giriş (Input) Resimleri

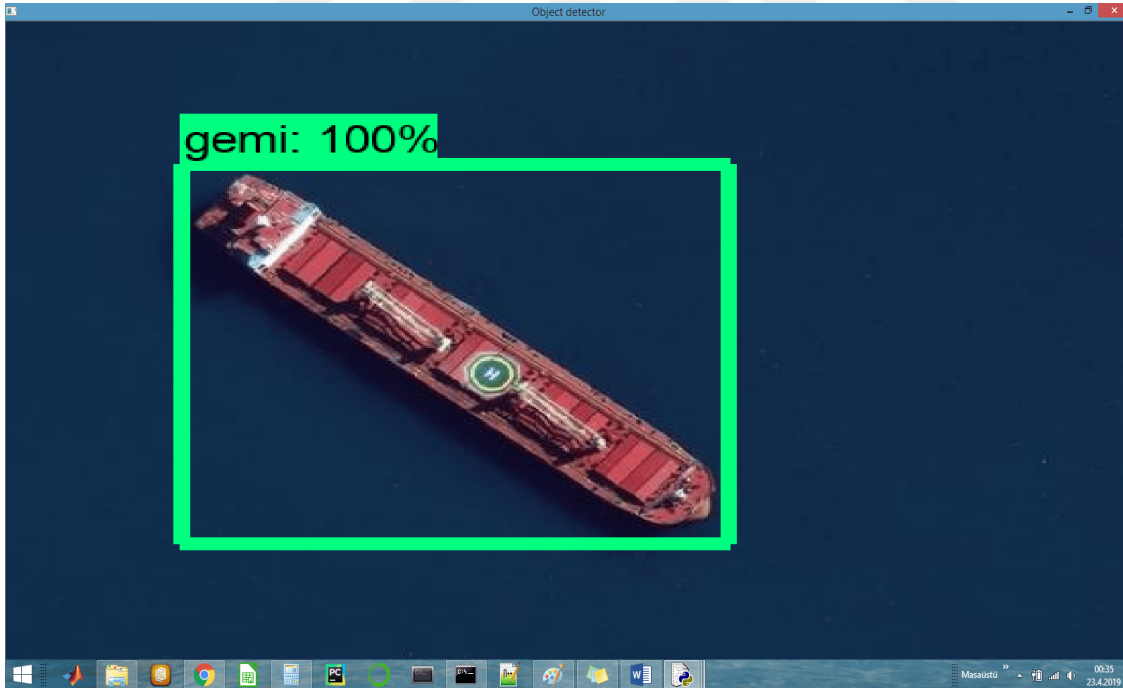
Eğitim tamamlandıktan sonra içinde gemi görüntüsü barındıran 13 adet optik uydu resmi modele input olarak verilmiştir. Bahse konu resimler ile ilgili bilgiler Çizelge 3.1’de gösterilmiştir.

Çizelge 3.1 : Giriş (input) resimleri.

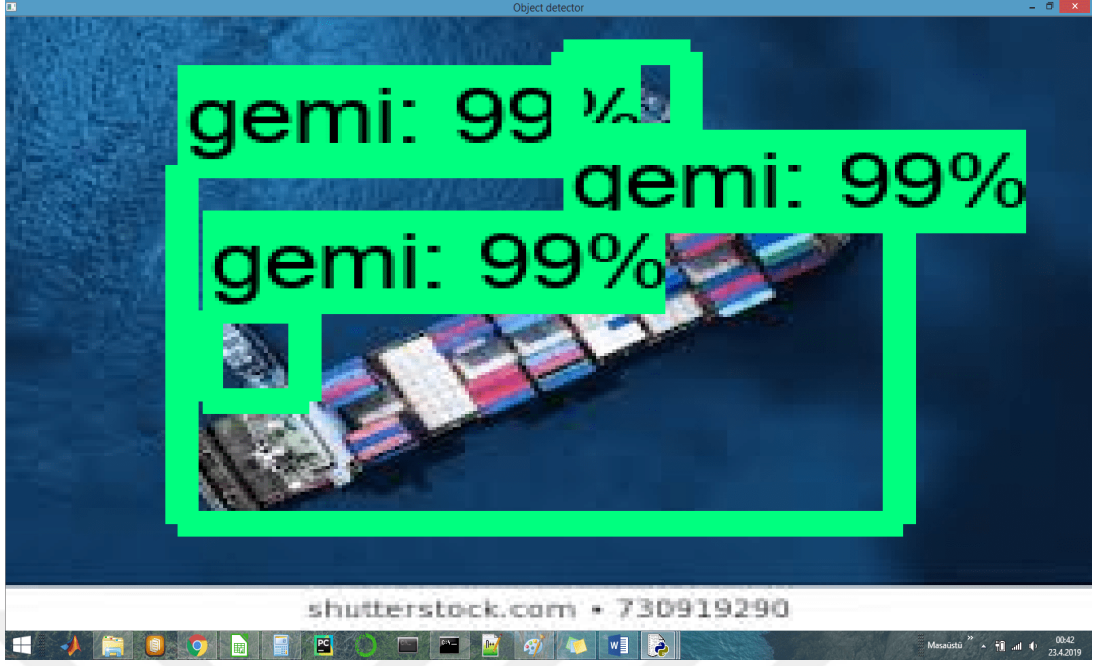
Çözünürlük	Band	DPI	Gemi Sayısı
550*393*3	RGB	96	1
265*190*3	RGB	96	3
577*348*3	RGB	96	4
246*205*3	RGB	96	3
261*193*3	RGB	96	1
313*313*3	RGB	96	1
1291*799*3	RGB	96	2
1197*1043*3	RGB	72	1
954*954*3	RGB	96	6
827*395*3	RGB	96	5
756*402*3	RGB	96	23
500*200*3	RGB	96	11
1950*1950*3	RGB	96	16

3.10 Modellerin Gemi Tespit Çıktıları

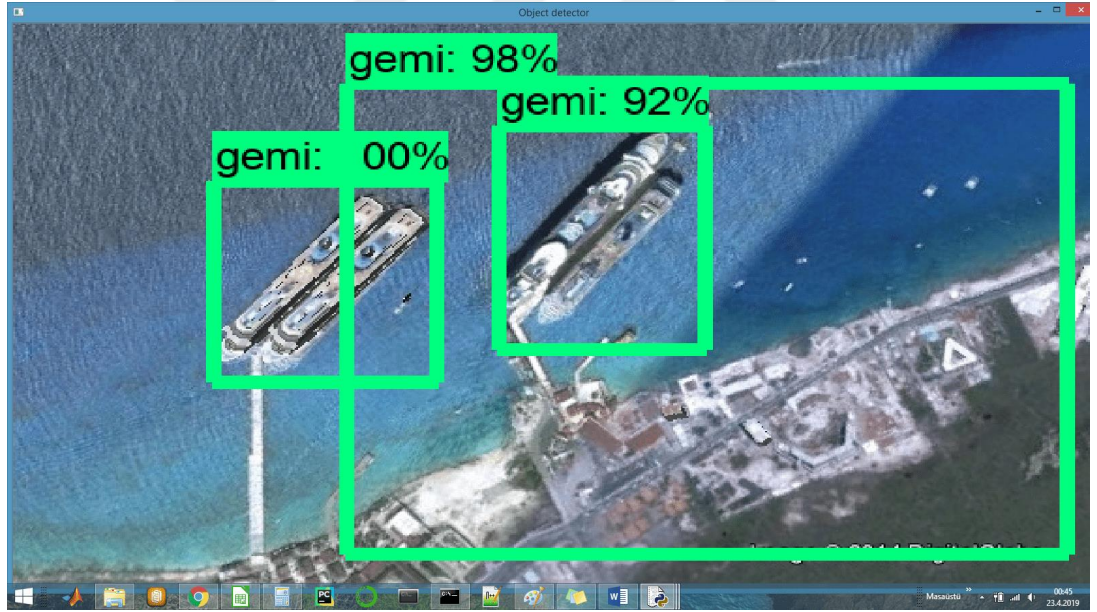
3.10.1 Faster rcnn inception v2_coco modeli öğrenim oranı 0.0002



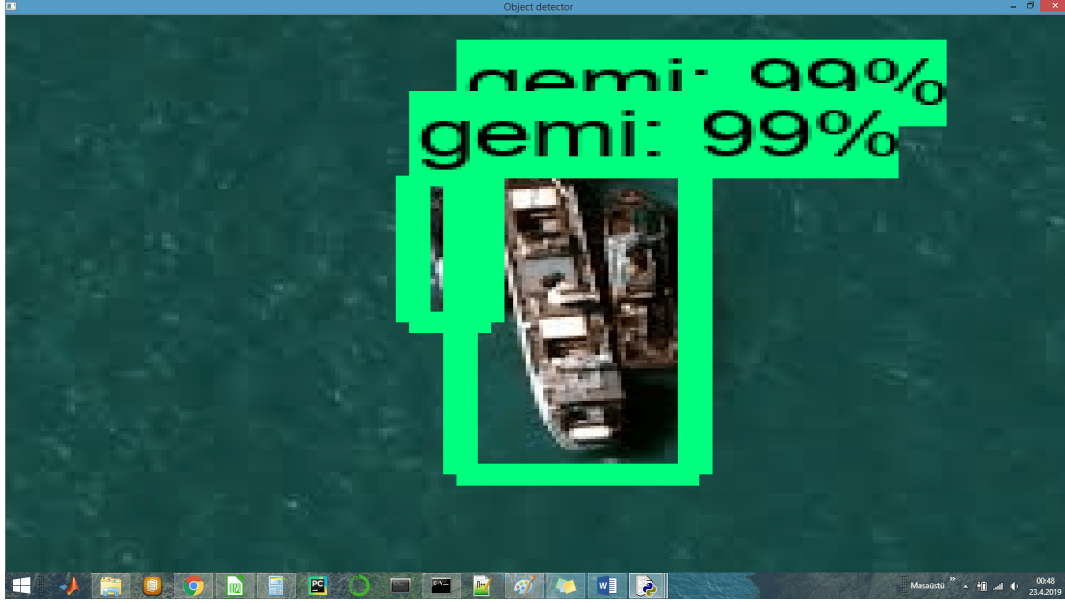
Şekil 3.7 : Resim-1'in tespit sonucu.



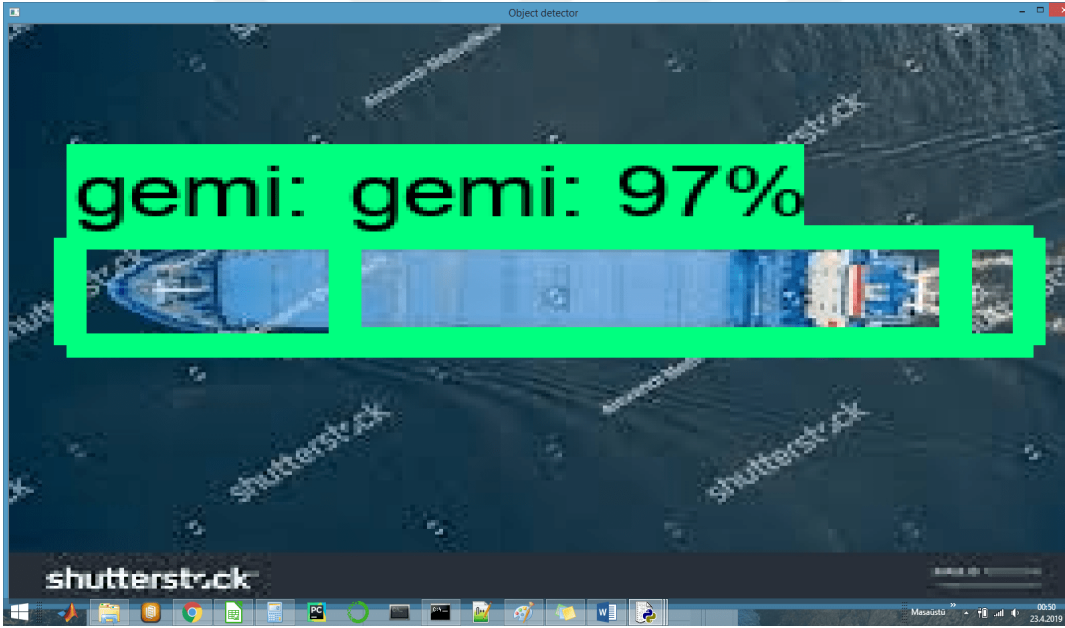
Şekil 3.8 : Resim-2'nin tespit sonucu.



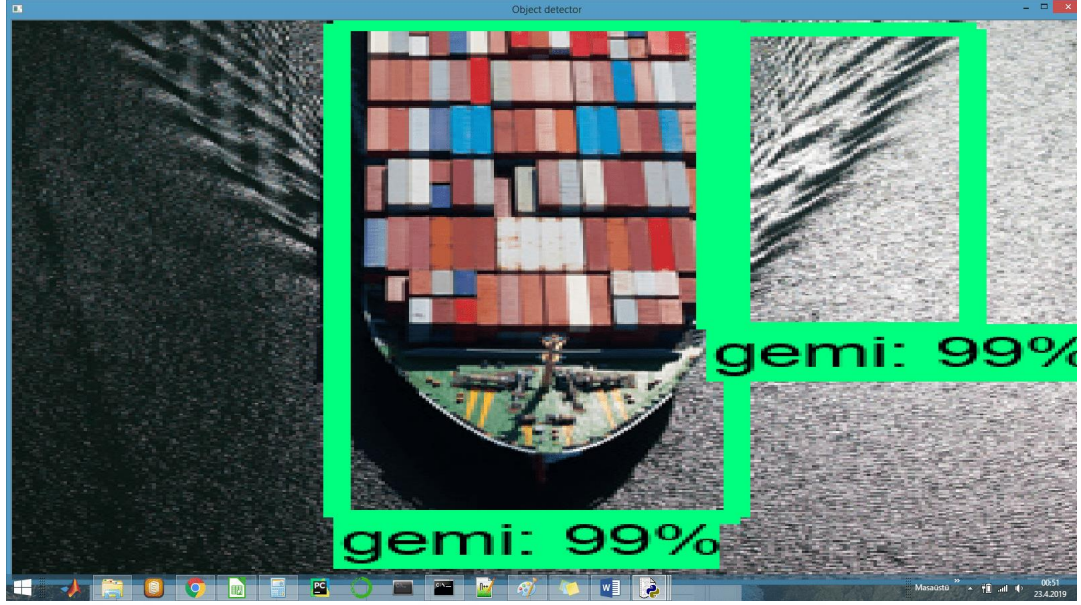
Şekil 3.9 : Resim-3'ün tespit sonucu.



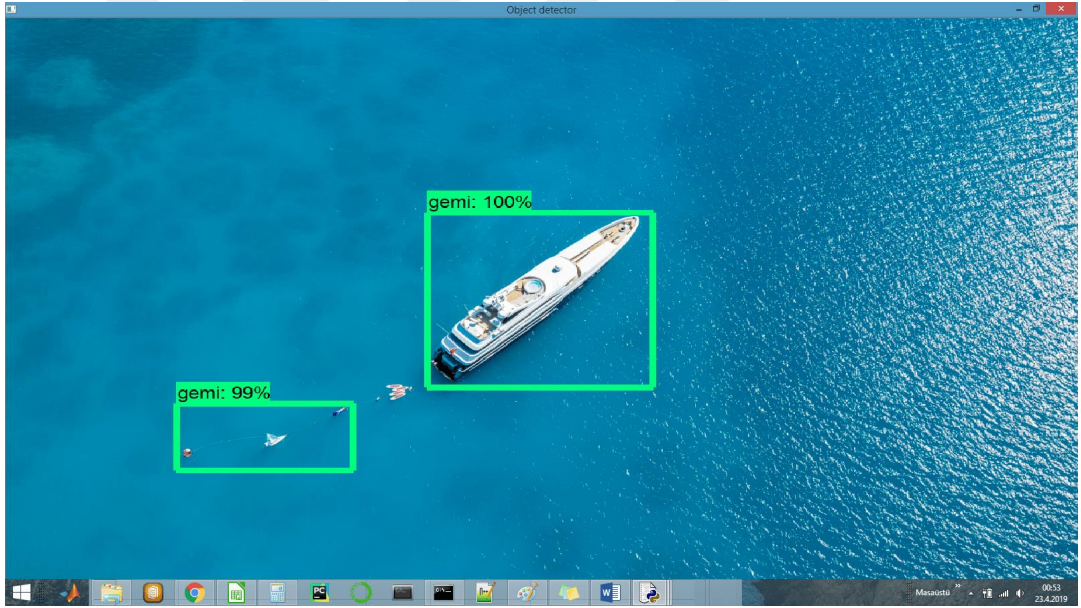
Şekil 3.10 : Resim-4'ün tespit sonucu.



Şekil 3.11 : Resim-5'in tespit sonucu.



Şekil 3.12 : Resim-6'nın tespit sonucu.



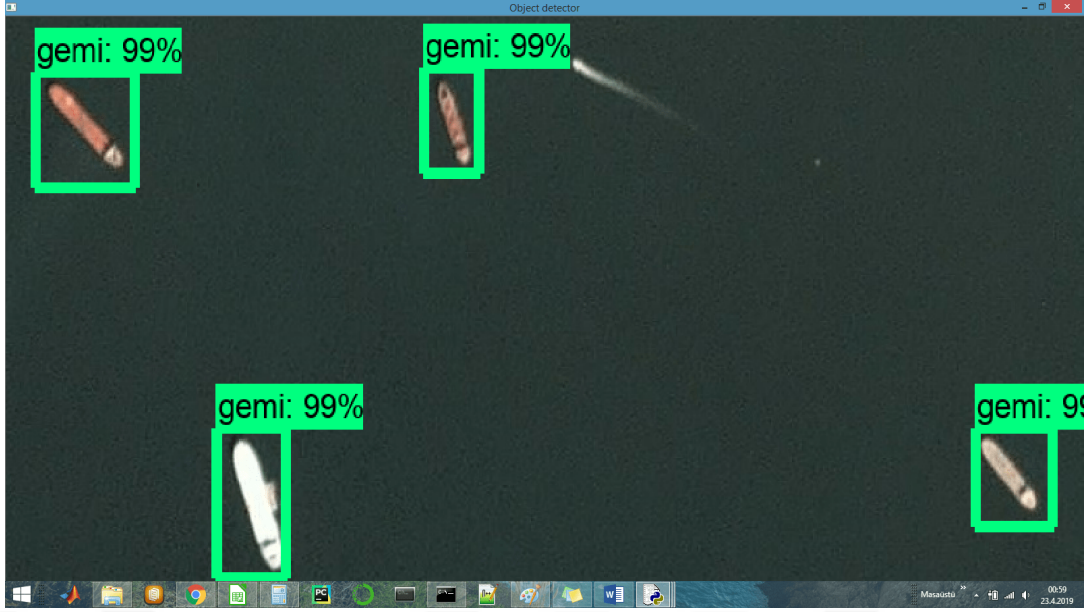
Şekil 3.13 : Resim-7'nin tespit sonucu.



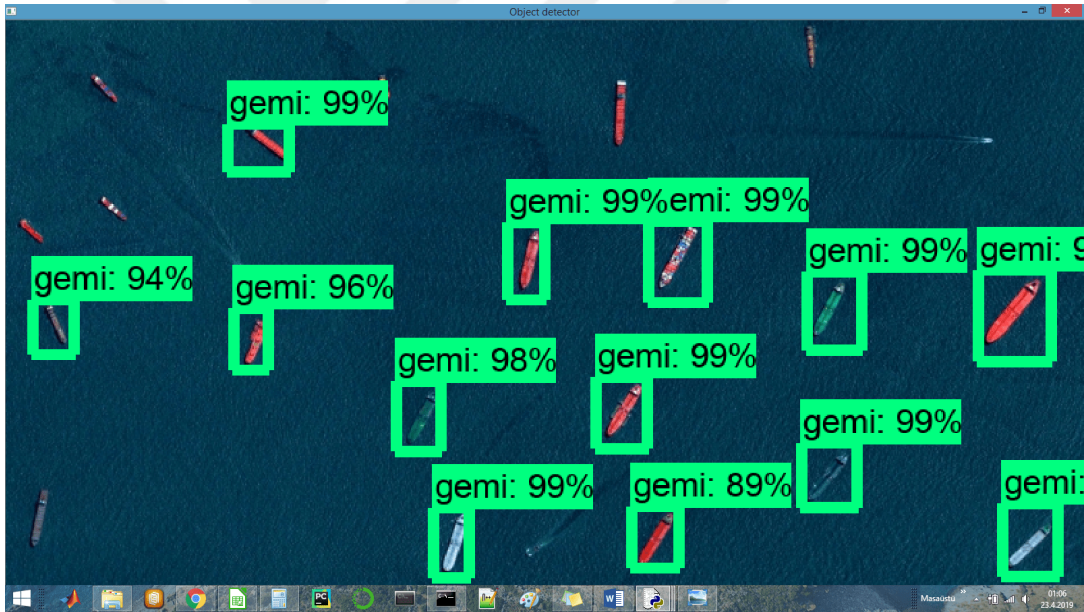
Şekil 3.14 : Resim-8'in tespit sonucu.



Şekil 3.15 : Resim-9'un tespit sonucu.



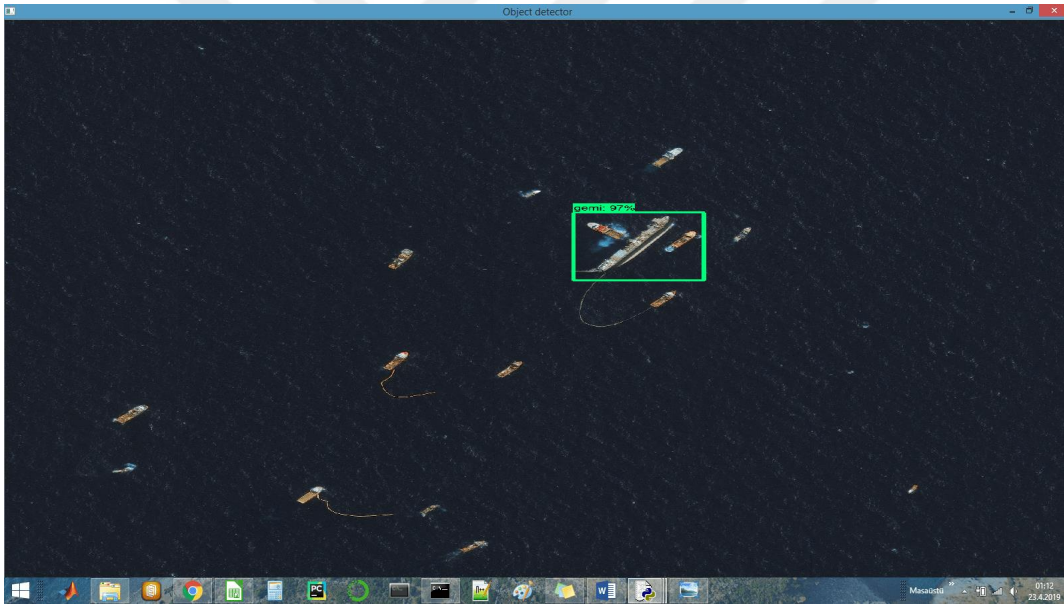
Şekil 3.16 : Resim-10'un tespit sonucu.



Şekil 3.17 : Resim-11'in tespit sonucu.

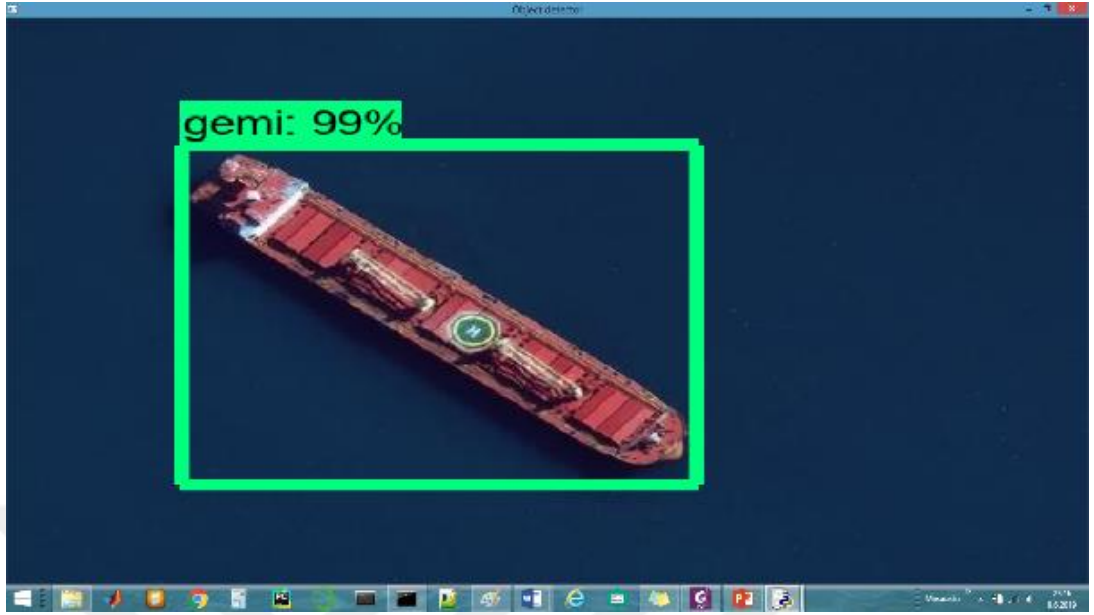


Şekil 3.18 : Resim-12'in tespit sonucu.

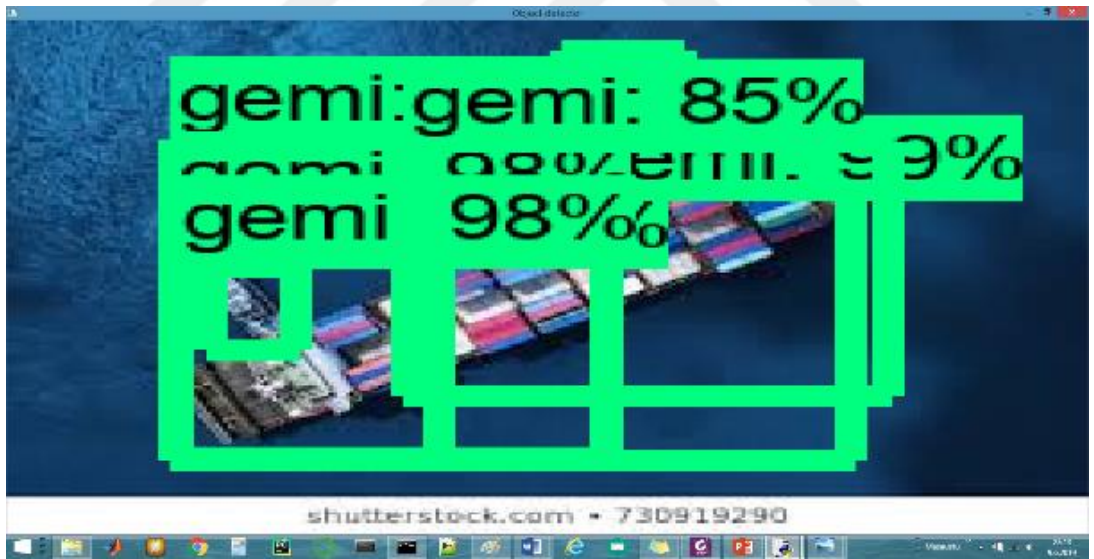


Şekil 3.19 : Resim-13'ün tespit sonucu.

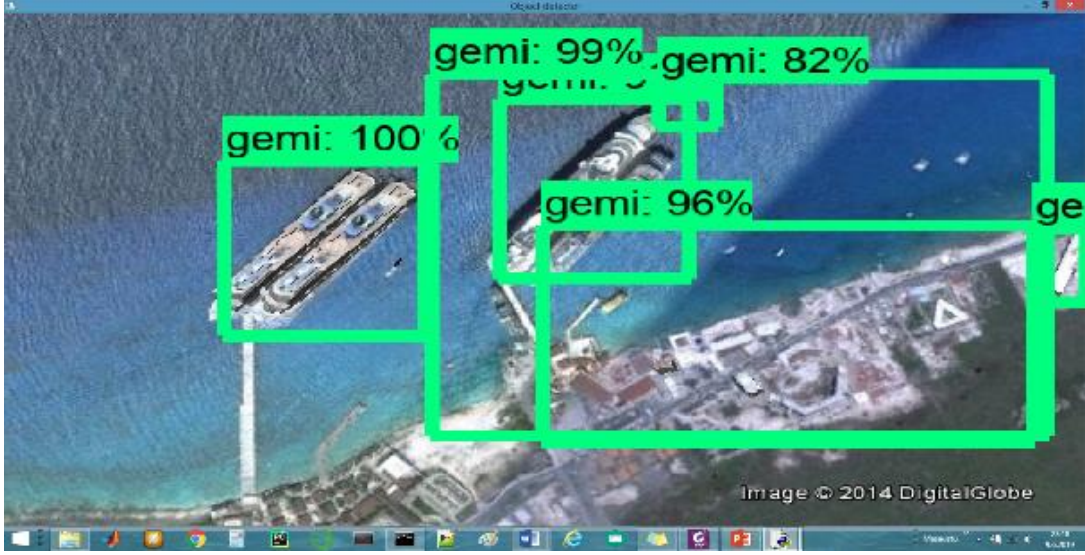
3.10.2 Faster rcnn inception v2_coco modeli öğrenim oranı 0.00002



Şekil 3.20 : Resim-1'in tespit sonucu.



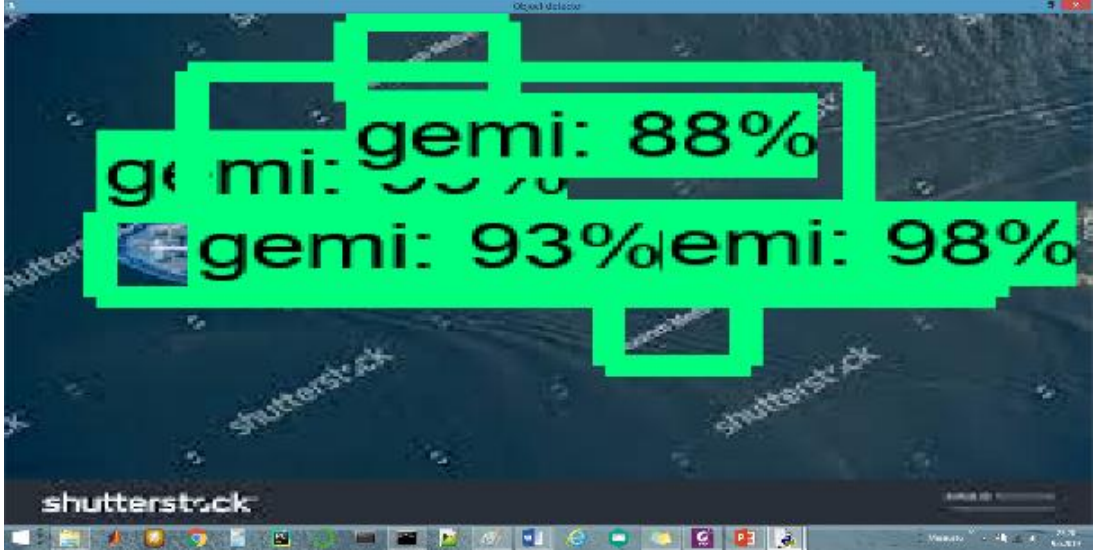
Şekil 3.21 : Resim-2'nin tespit sonucu.



Şekil 3.22 : Resim-3'ün tespit sonucu.



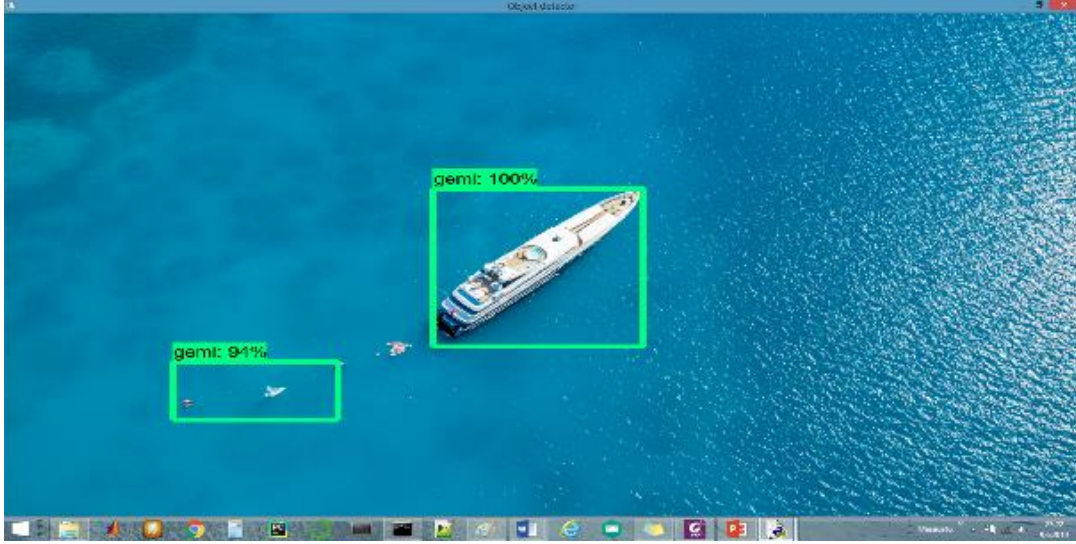
Şekil 3.23 : Resim-4'ün tespit sonucu.



Şekil 3.24 : Resim-5'in tespit sonucu.



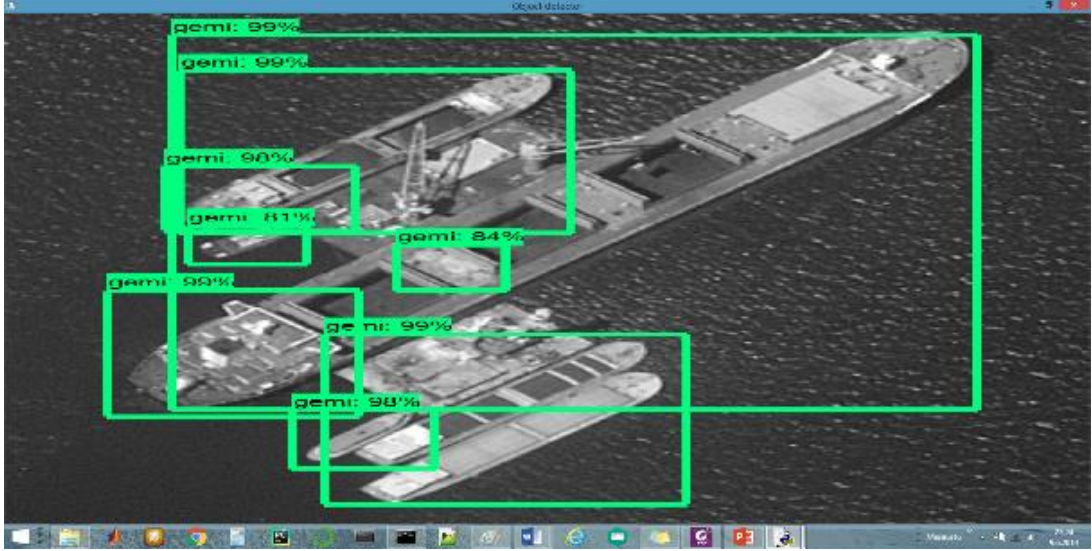
Şekil 3.25 : Resim-6'nın tespit sonucu.



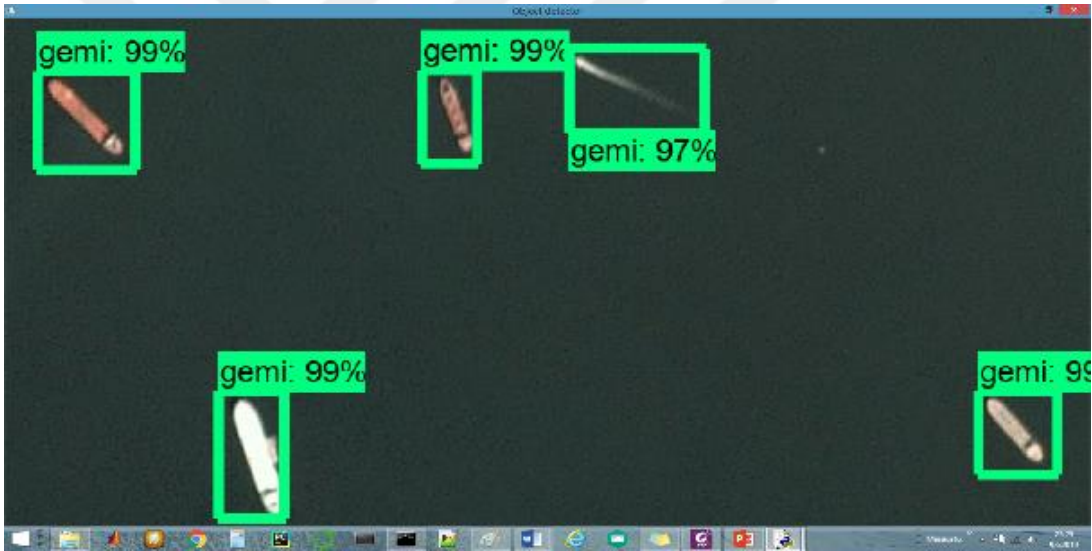
Şekil 3.26 : Resim-7'nin tespit sonucu.



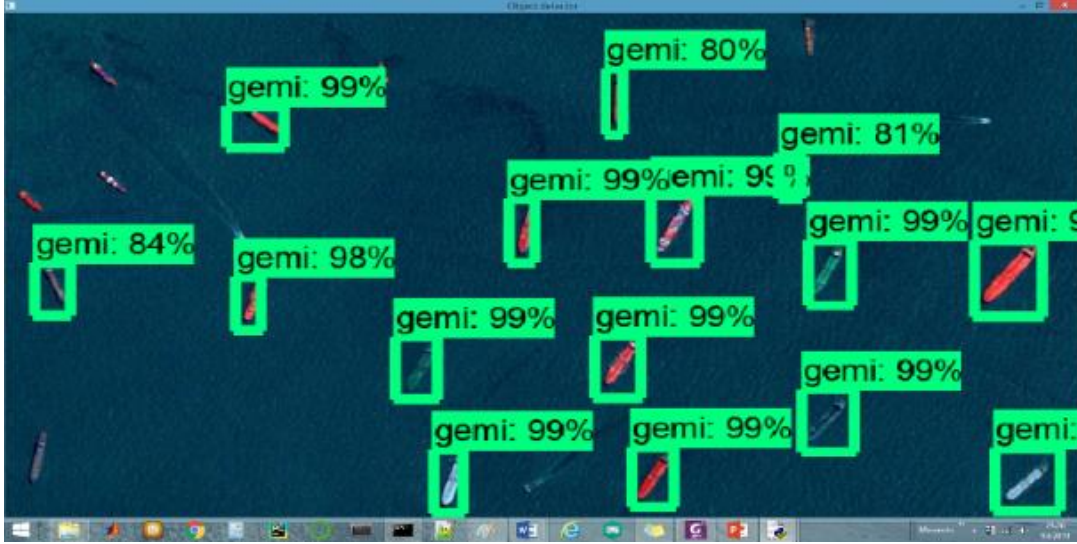
Şekil 3.27 : Resim-8'in tespit sonucu.



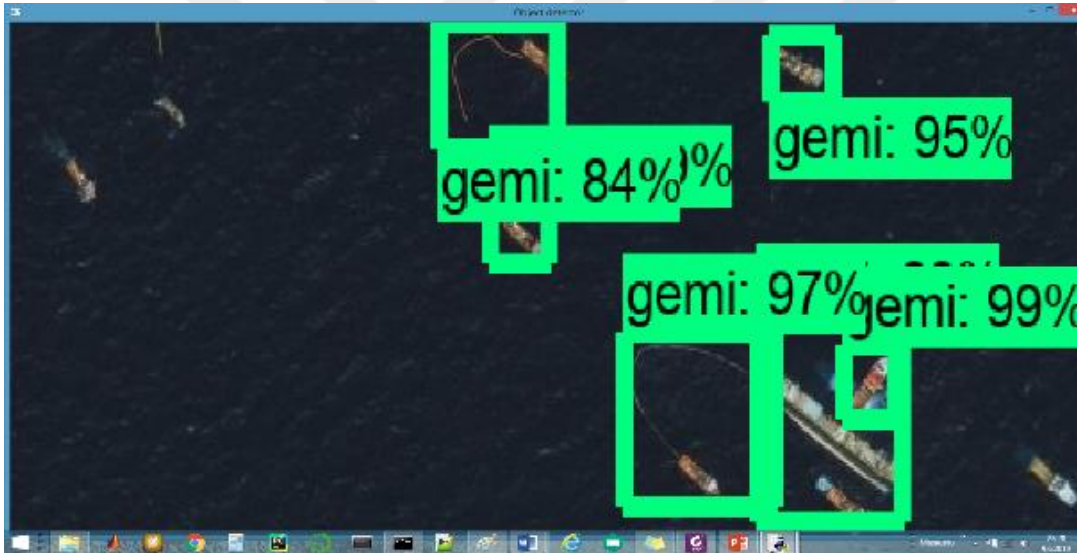
Şekil 3.28 : Resim-9'un tespit sonucu.



Şekil 3.29 : Resim-10'un tespit sonucu.



Şekil 3.30 : Resim-11'in tespit sonucu.

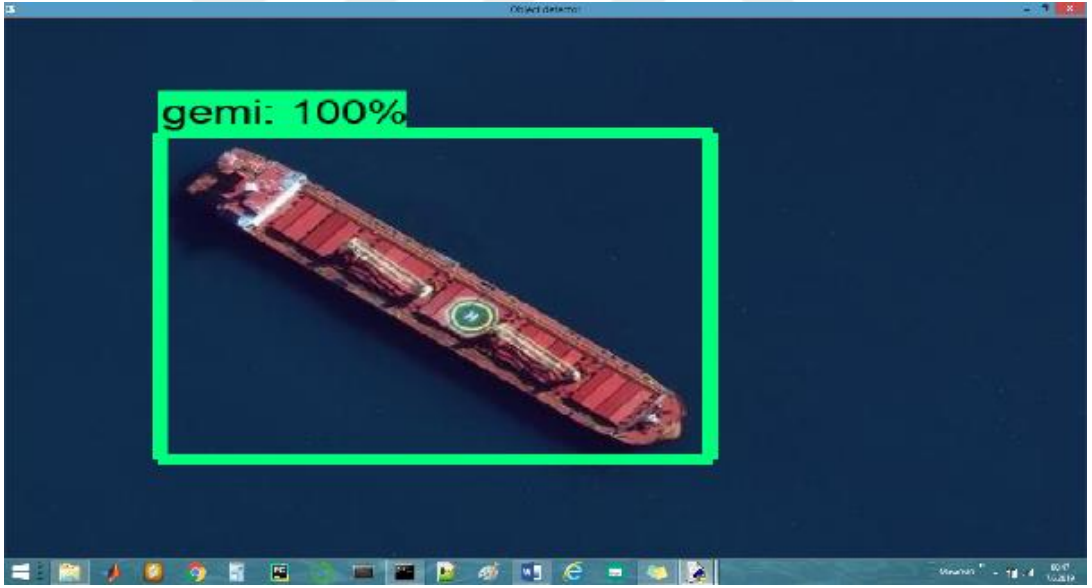


Şekil 3.31 : Resim-12'nin tespit sonucu.



Şekil 3.32 : Resim-13'ün tespit sonucu.

3.10.3 Faster rcnn resnet101_coco



Şekil 3.33 : Resim-1'in tespit sonucu.



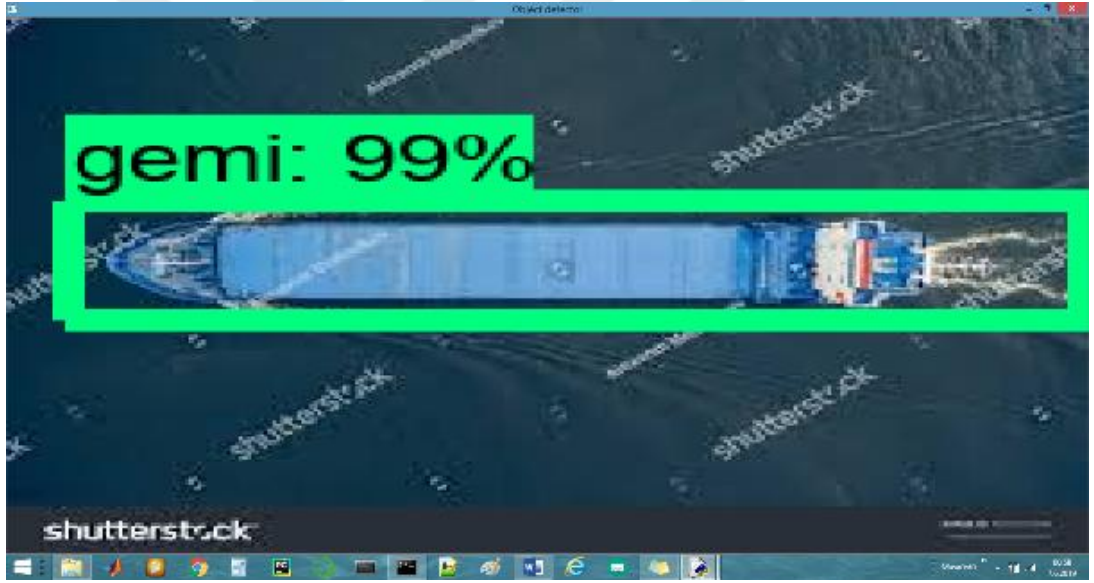
Şekil 3.34 : Resim-2'nin tespit sonucu.



Şekil 3.35 : Resim-3'ün tespit sonucu.



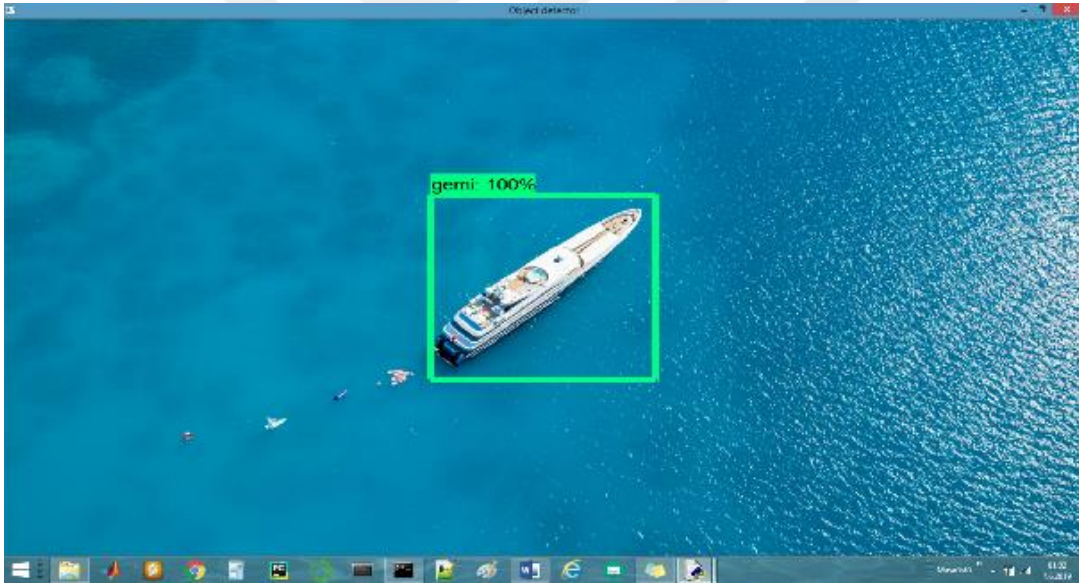
Şekil 3.36 : Resim-4'ün tespit sonucu.



Şekil 3.37 : Resim-5'in tespit sonucu.



Şekil 3.38 : Resim-6'nın tespit sonucu.



Şekil 3.39 : Resim-7'nin tespit sonucu.



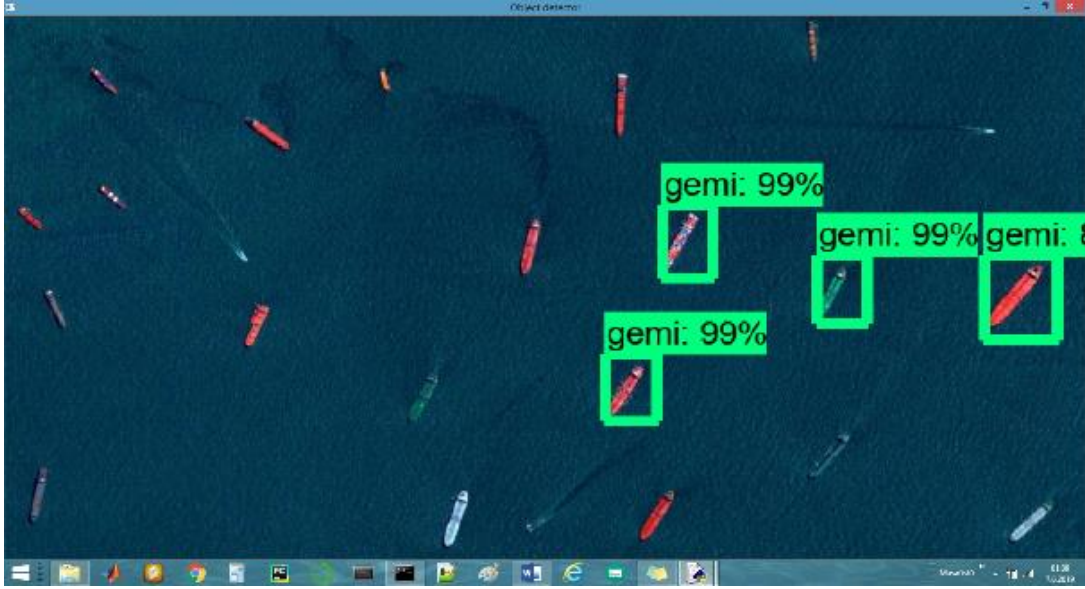
Şekil 3.40 : Resim-8'in tespit sonucu.



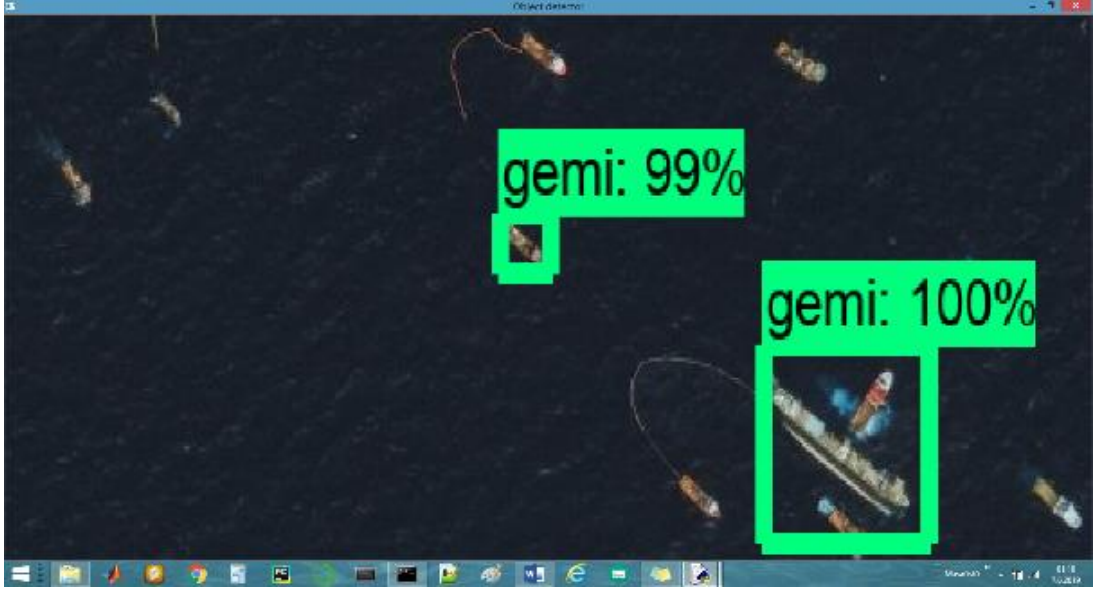
Şekil 3.41 : Resim-9'un tespit sonucu.



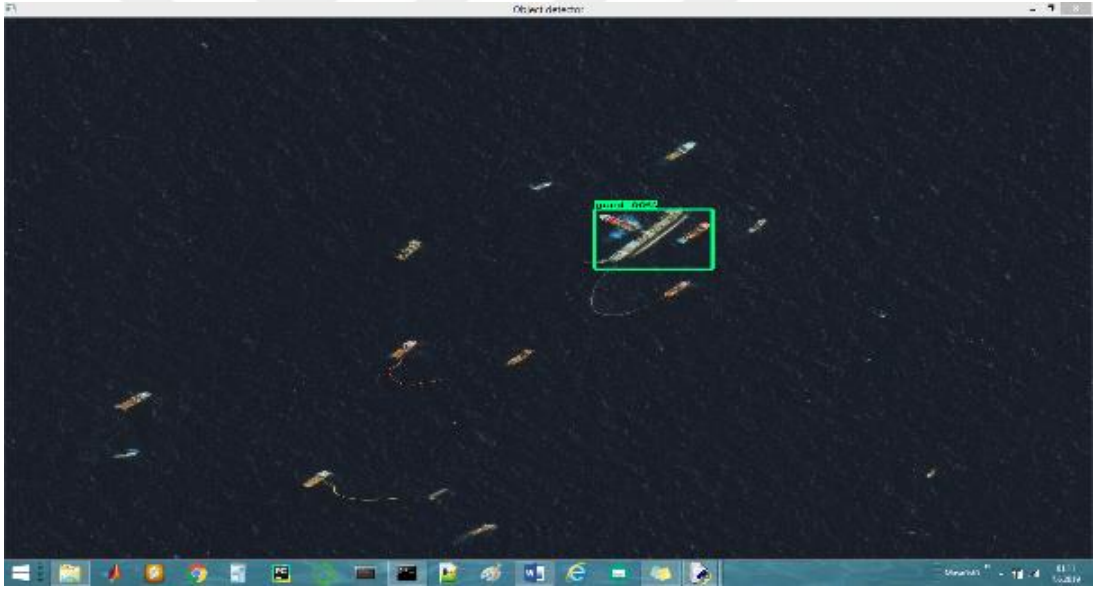
Şekil 3.42 : Resim-10'un tespit sonucu.



Şekil 3.43 : Resim-11'in tespit sonucu.

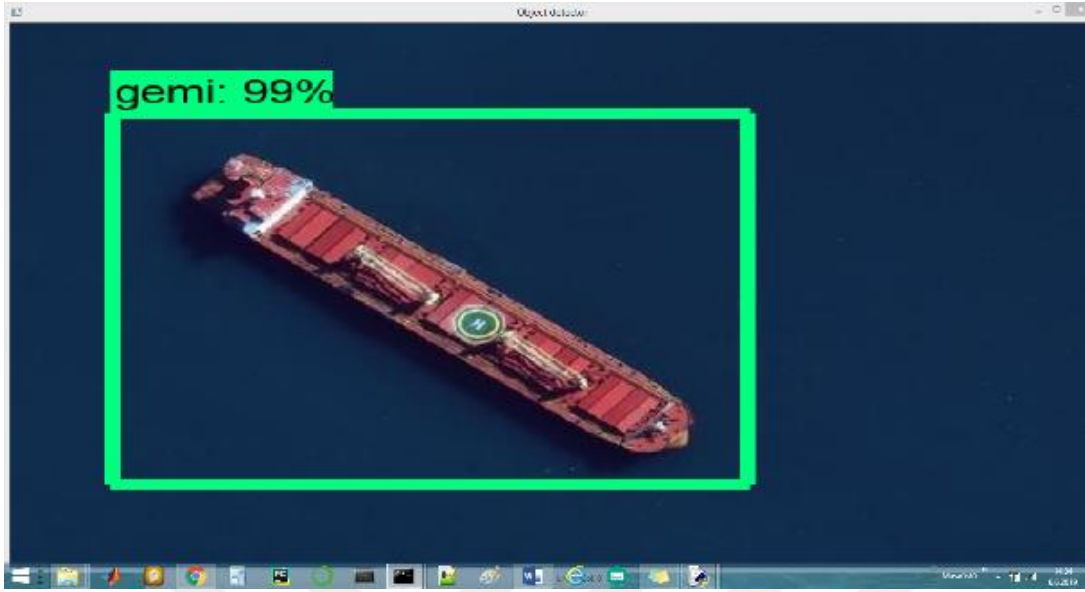


Şekil 3.44 : Resim-12'nin tespit sonucu.



Şekil 3.45 : Resim-13'ün tespit sonucu.

3.10.4 Ssd_inception_v2_coco



Şekil 3.46 : Resim-1'in tespit sonucu.



Şekil 3.47 : Resim-2'nin tespit sonucu.



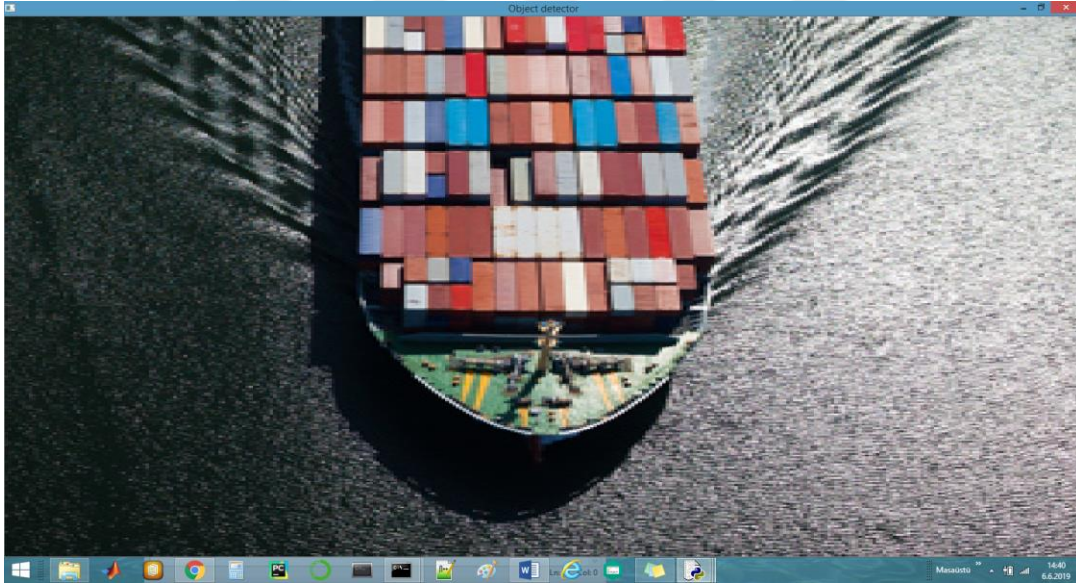
Şekil 3.48 : Resim-3'ün tespit sonucu.



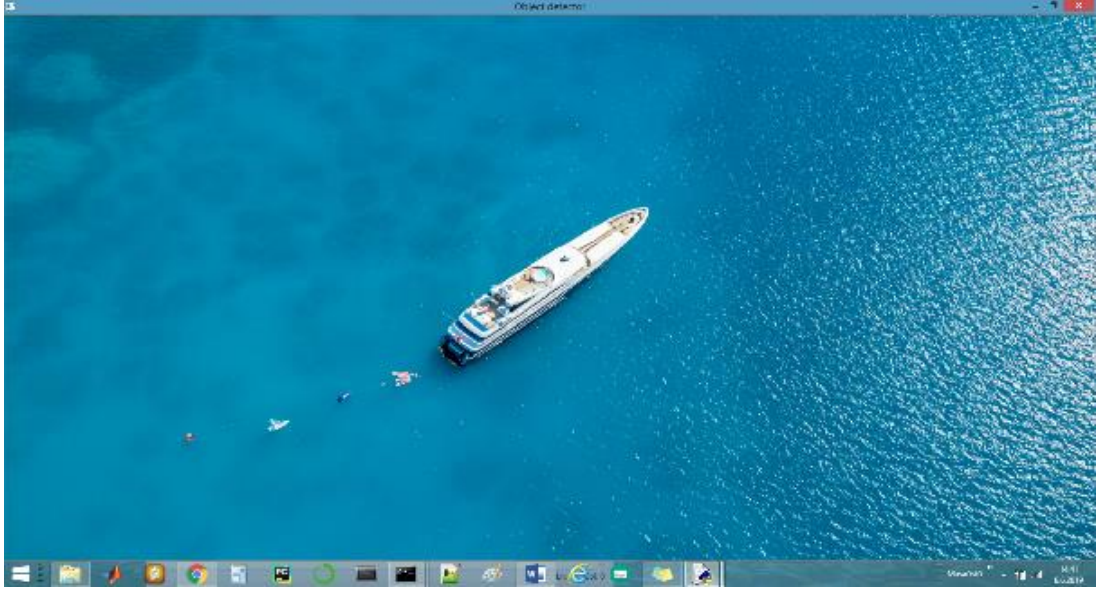
Şekil 3.49 : Resim-4'ün tespit sonucu.



Şekil 3.50 : Resim-5'in tespit sonucu.



Şekil 3.51 : Resim-6'nın tespit sonucu.



Şekil 3.52 : Resim-7'nin tespit sonucu.



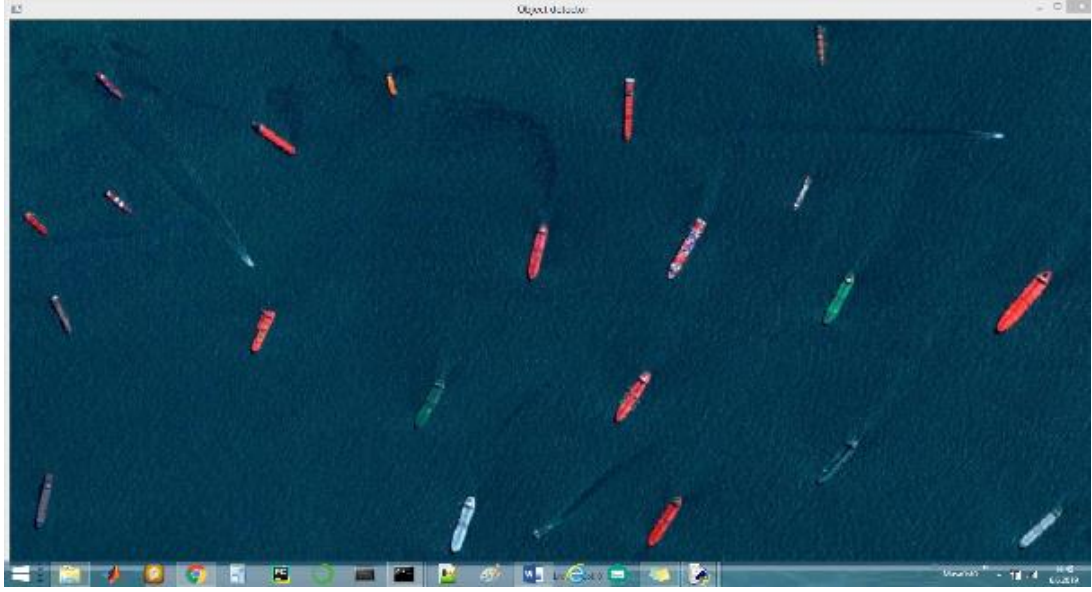
Şekil 3.53 : Resim-8'in tespit sonucu.



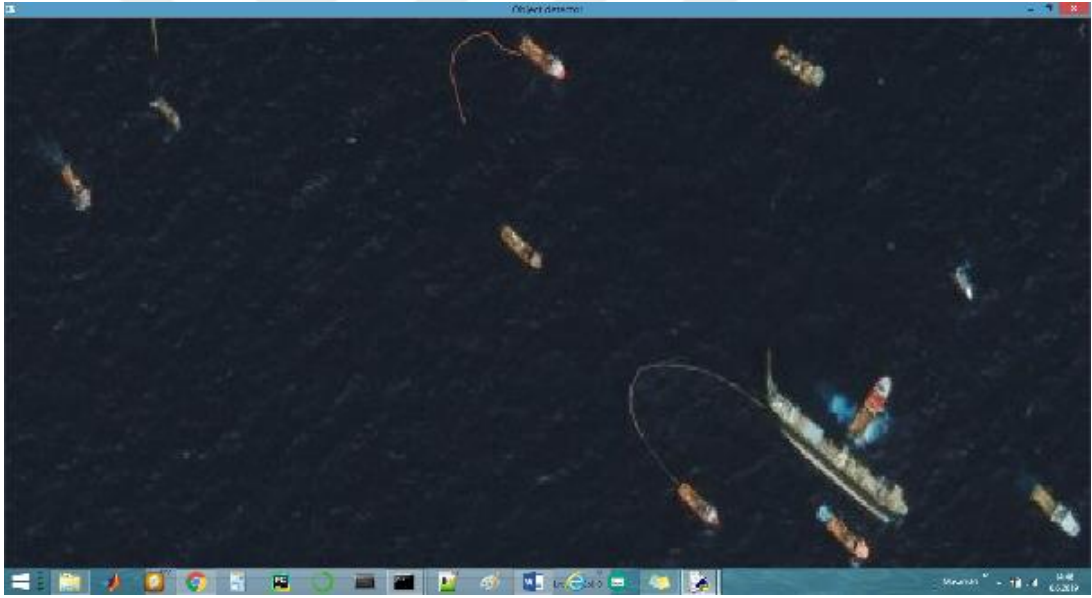
Şekil 3.54 : Resim-9'un tespit sonucu.



Şekil 3.55 : Resim-10'un tespit sonucu.



Şekil 3.56 : Resim-11'in tespit sonucu.



Şekil 3.57 : Resim-12'nin tespit sonucu.



Şekil 3.58 : Resim-13'ün tespit sonucu.

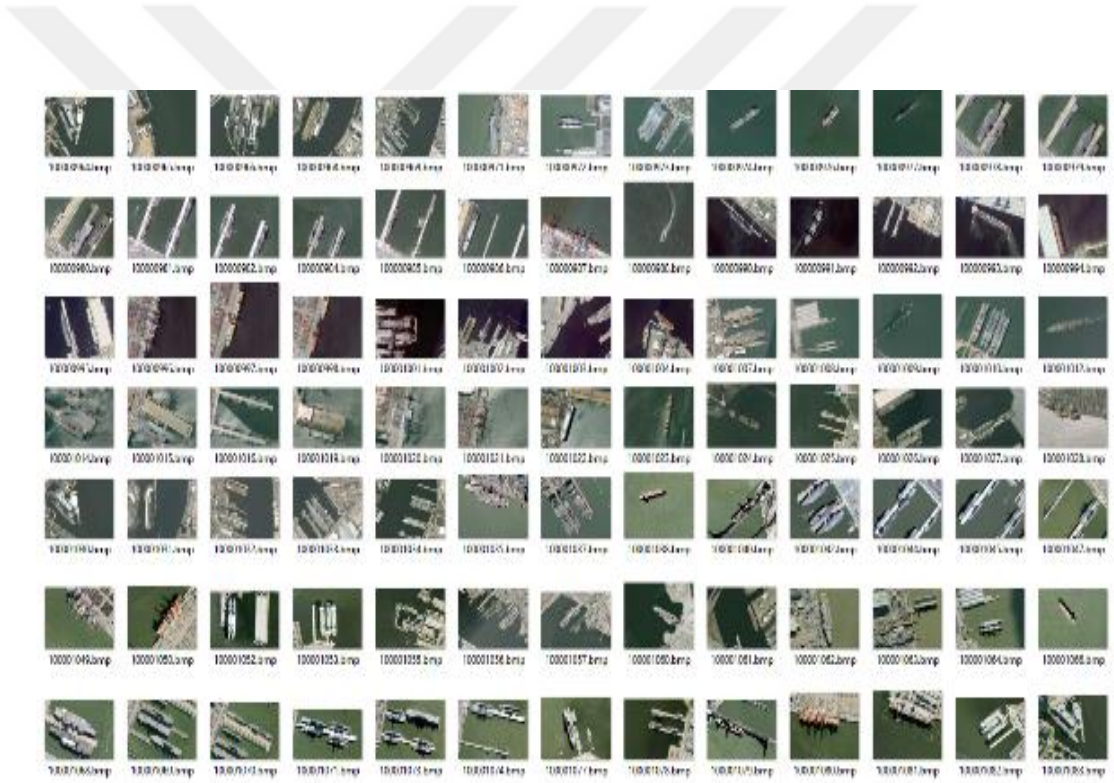
3.11 Modellerin Gemi Tespit Çıktılarının Karşılaştırılması

Yapılan testlerin sonucunda; ssd modelinin tespit performansının faster r-cnn modellerine kıyasla çok daha zayıf olduğu, bazı resimlerde gemi tespiti yapamadığı gözlenmiştir. Faster r-cnn modelleri arasında ise “faster rcnn inception v2_coco” modelinin öğrenim oranının on kat azaltılmış halinin (kayıp grafik rengi açık mavi olan) en iyi tespit performansına sahip olduğu gözlenmiştir. Bahse konu modelin; içinde 6'dan az sayıda gemi barındıran optik uydu resimleri ile az sayıda yanlış alarm verdiği ve düzgün çalıştığı değerlendirilmiştir. Resmin kapsadığı alan büyüdükçe veya gemi sayısı arttıkça, modelin performansının düştüğü gözlenmiştir. Kütüphanenin eğitimi esnasında kullanılan eğitim resimlerindeki tanker tipindeki gemilerin tespit oranının yüksek olduğu ancak daha ufak boyutlu gemilerin tespit edilmesinde modelin geliştirilmeye ihtiyaç duyulduğu değerlendirilmiştir.

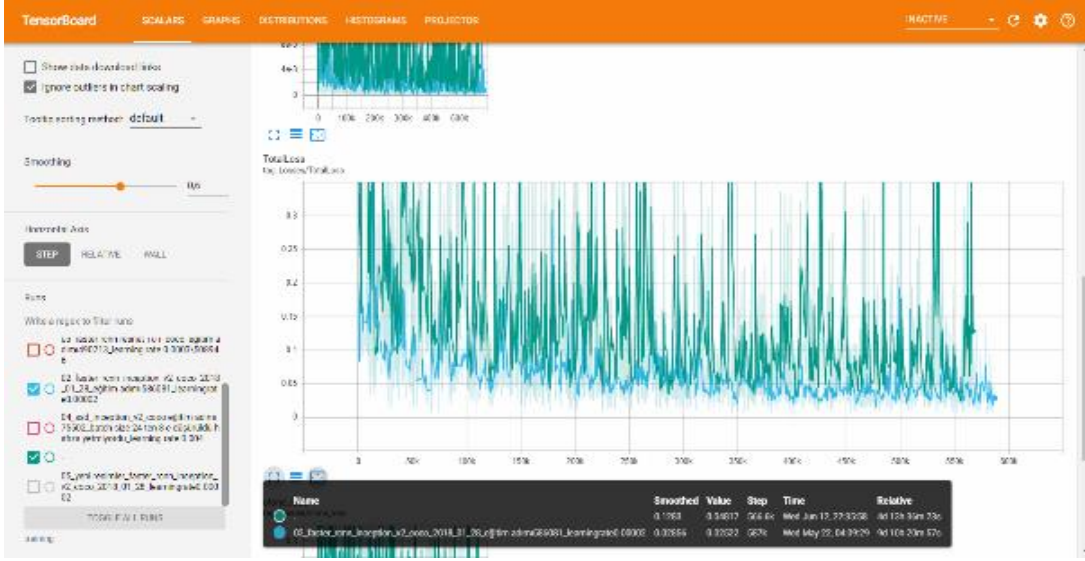
3.12 Seçilen Modelin Gemi Tespit Performansının Artırılması İçin Yapılan İşlemler

Eğitimler sonucunda en iyi gemi tespit performansı sergileyen modelin performansını artırmak için eğitim kütüphanesinin büyütülmesine karar verilmiştir. Bu kapsamda seçilen modelin eğitiminde kullanılacak ilave gemi görüntü setleri internet ortamından indirilmiştir (Gfeng, 2019).

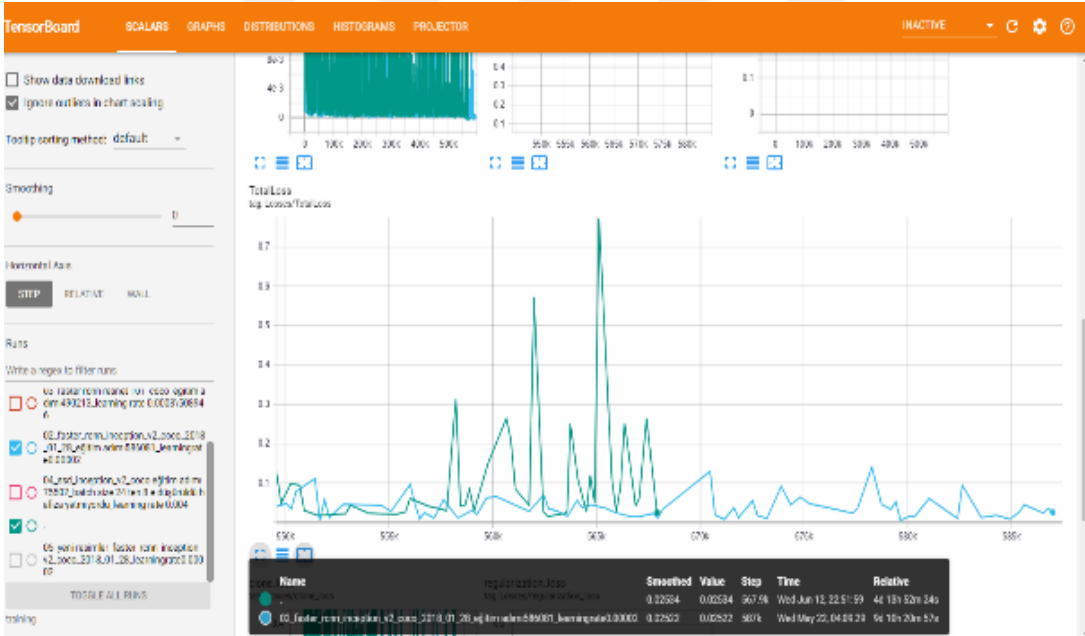
Görüntü seti, değişken çözünürlüğe sahip (987 * 804 * 3, 1155*820*3, vb.), ilk gemi görüntü setinin çözünürlüğünden yaklaşık 10 kat büyük, RGB bandı, 96 dpi olan 1056 adet görüntüden oluşur (Şekil 3.59). İçinde gemi görüntüsü bulunduran resimler LabelImg programı vasıtasıyla “gemi” ismi ile etiketlenmiştir (LabelImg, 2019). İçinde gemi görüntüsü olmayan resimler silinmiştir. Görüntülerin % 80'i eğitim (training) görüntüsü, % 20'si test görüntüsü olarak gruplandırılmıştır. Müteakiben 567900'üncü eğitim adımına kadar bahse konu model eğitime tabi tutulmuştur. Modelin eski hali ile son halinin kayıp grafiği karşılaştırması Şekil 3.60 ve Şekil 3.61'de gösterilmiştir. Modellerin en sonuncu eğitim basamaklarına geldiği an itibariyle sıfır kayıp seviyesine neredeyse aynı mesafede olduğu gözlenmiştir.



Şekil 3.59 : Eğitimde kullanılan ilave görüntü setleri (Gfeng, 2019).

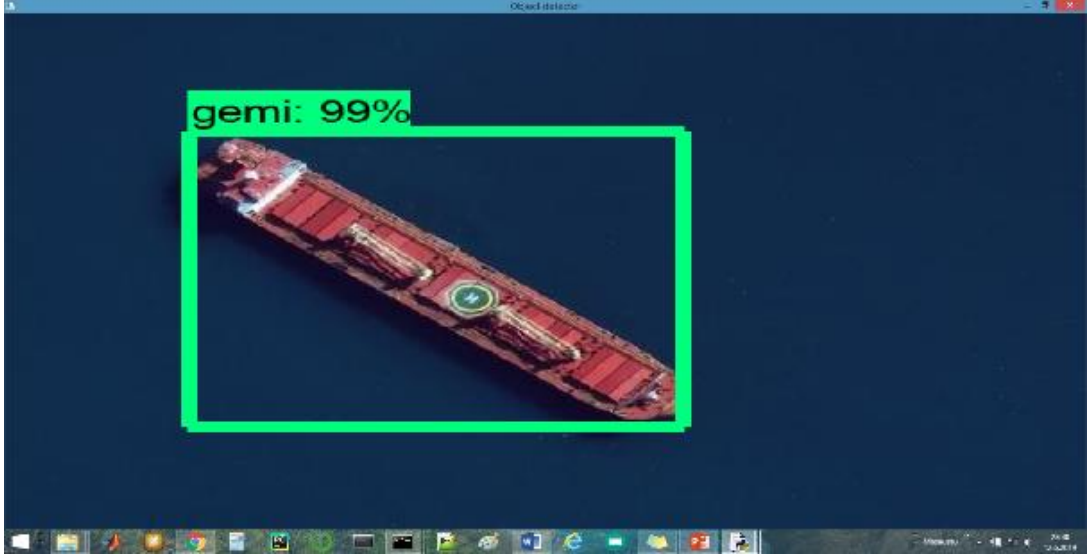


Şekil 3.60 : Kayıp grafiklerinin karşılaştırılması (Modelin ilk hali kırmızı, son hali yeşil renklidir).



Şekil 3.61 : Kayıp grafiklerinin karşılaştırılması (Modelin ilk hali kırmızı, son hali yeşil renklidir).

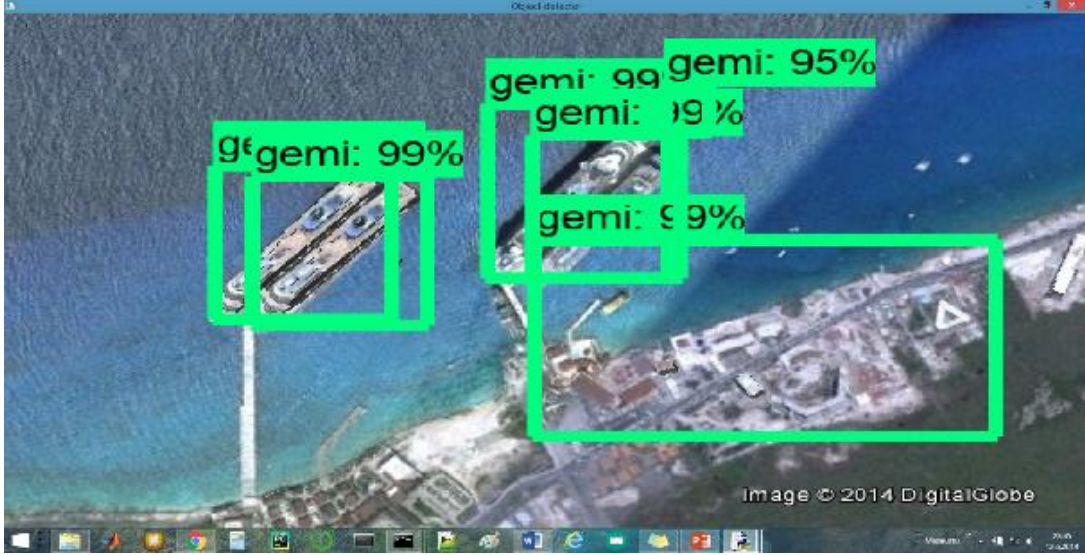
Eğitim tamamlandıktan sonra içinde gemi görüntüsü barındıran 13 adet optik uydu resmi modele tekrardan input olarak verilmiştir. Modelin son halinin gemi tespit çıktıları Şekil 3.62 ile Şekil 3.74 arasında gösterilmiştir.



Şekil 3.62 : Resim-1'in tespit sonucu.



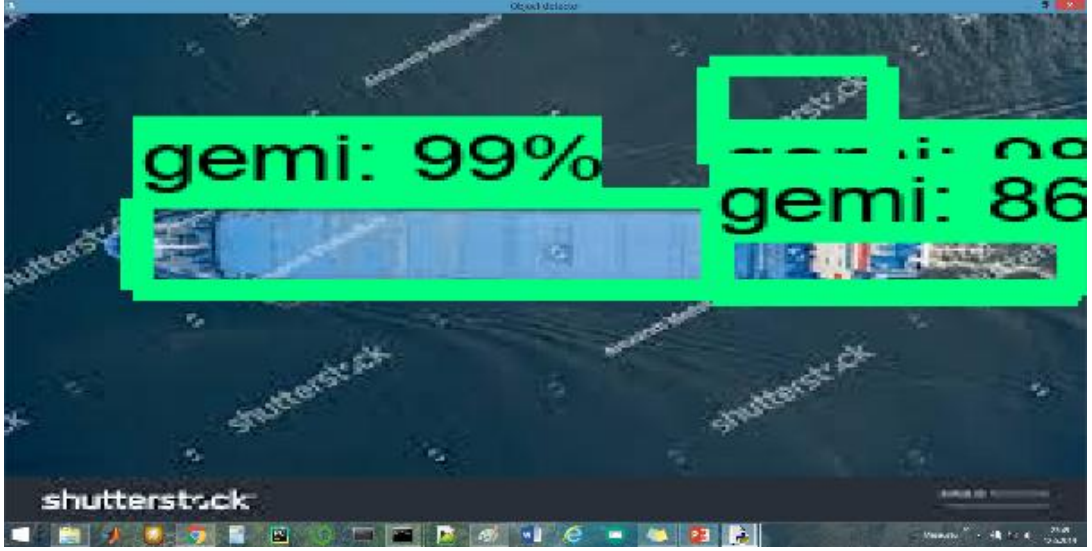
Şekil 3.63 : Resim-2'nin tespit sonucu.



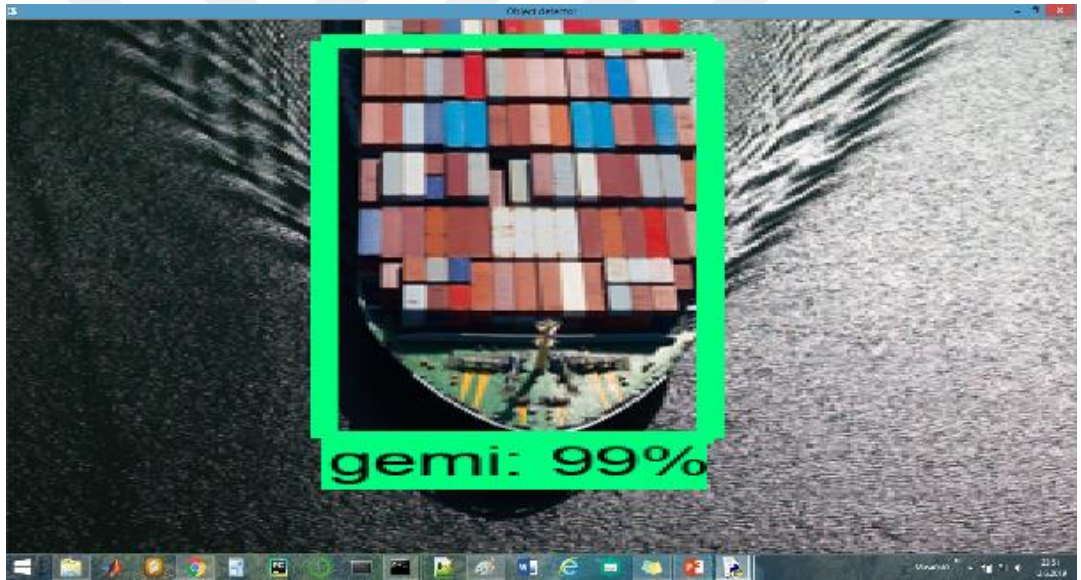
Şekil 3.64 : Resim-3'ün tespit sonucu.



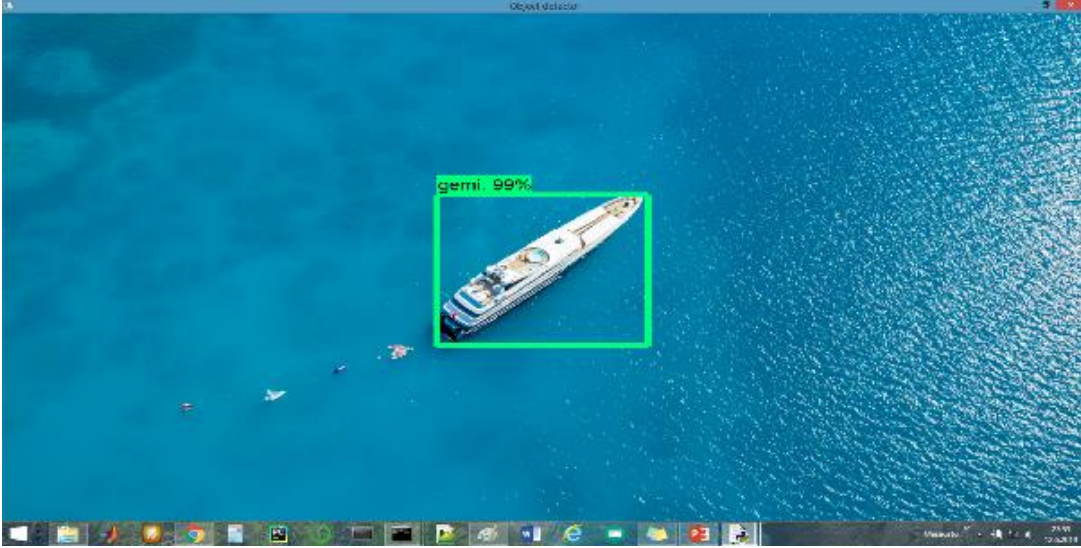
Şekil 3.65 : Resim-4'ün tespit sonucu.



Şekil 3.66 : Resim-5'in tespit sonucu.



Şekil 3.67 : Resim-6'nın tespit sonucu.



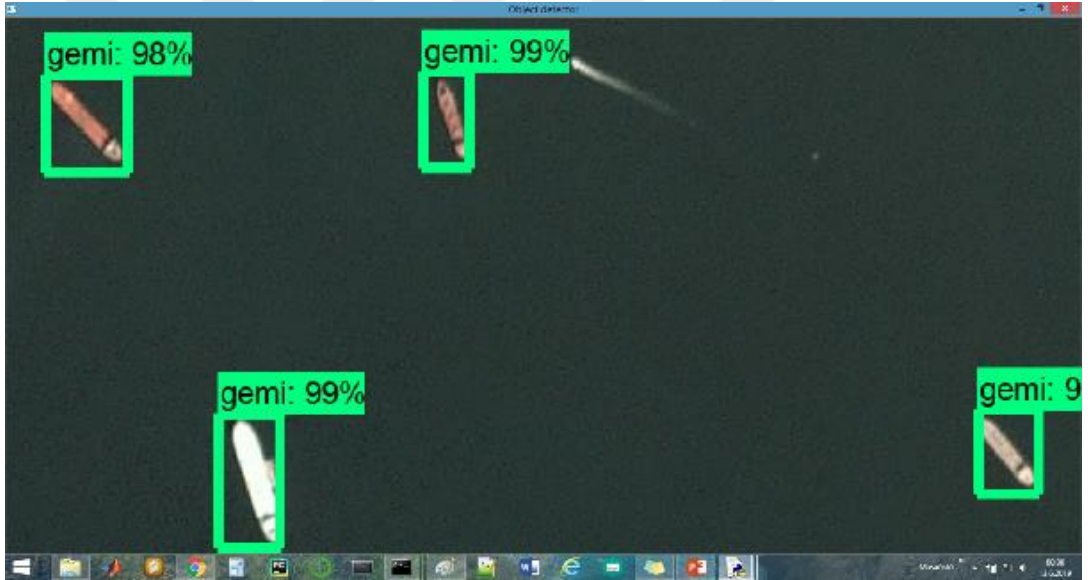
Şekil 3.68 : Resim-7'nin tespit sonucu.



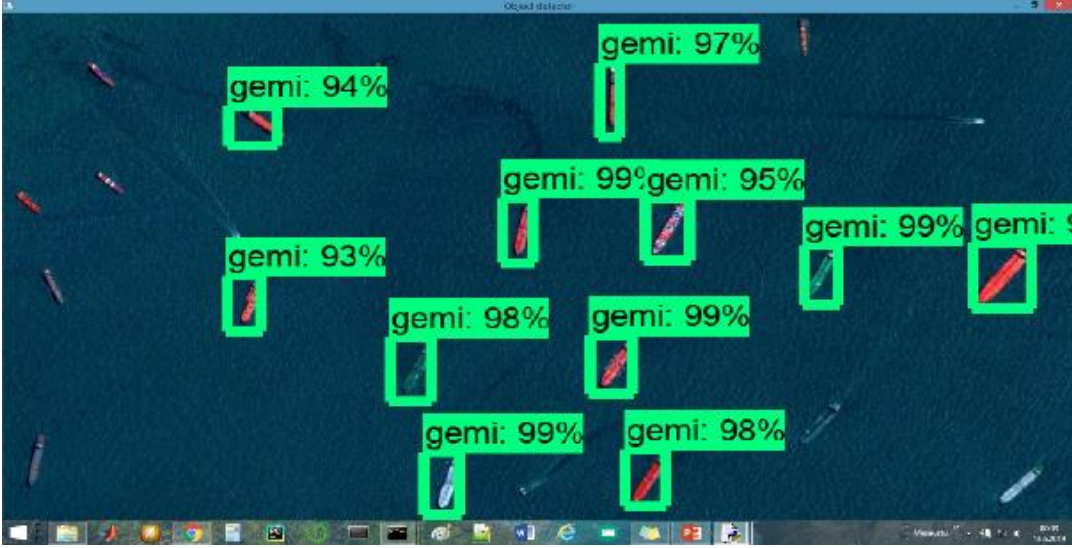
Şekil 3.69 : Resim-8'in tespit sonucu.



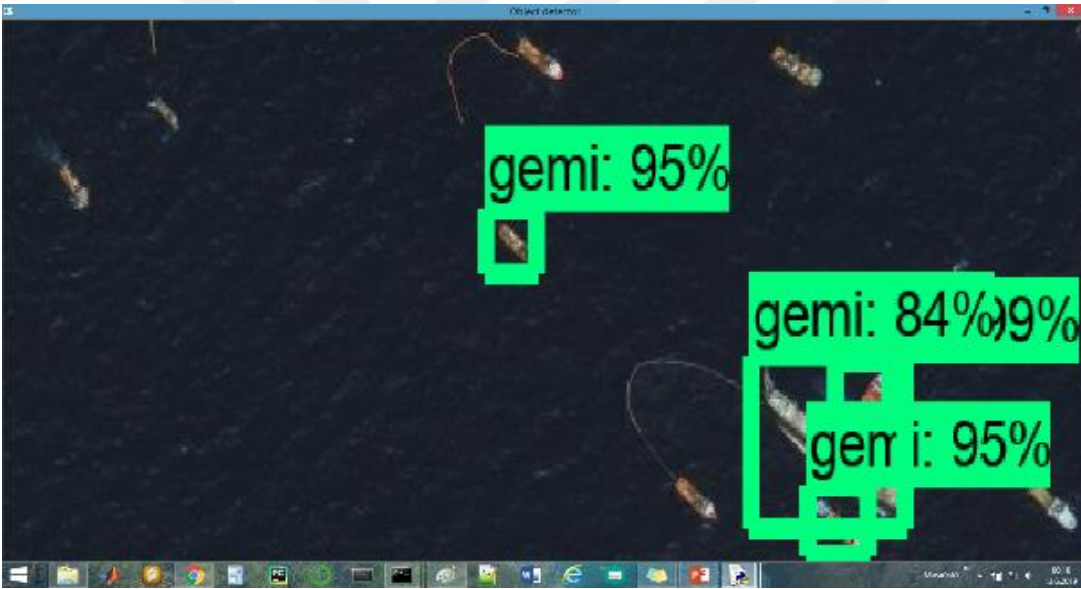
Şekil 3.70 : Resim-9'un tespit sonucu.



Şekil 3.71 : Resim-10'un tespit sonucu.



Şekil 3.72 : Resim-11'in tespit sonucu.



Şekil 3.73 : Resim-12'nin tespit sonucu.



Şekil 3.74 : Resim-13'ün tespit sonucu.

İlave görüntü setinin eklenmesi ile Resim-2'nin tespit sonucunda aynı gemi üzerindeki birden fazla tespit hatası giderilmiş, Resim-3'te yan yana duran gemiler ayrı ayrı tespit edilebilir hale gelmiş, Resim-5'teki yanlış tespit hataları azalmış, Resim-6'daki yanlış tespit hatası giderilmiş, Resim-9'daki yan yana duran 2 adet gemi ayrı ayrı tespit edilebilir hale gelmiş ve aynı gemi üzerindeki birden fazla tespit hatası giderilmiştir.



4. SONUÇ VE ÖNERİLER

Bu tez çalışması ile derin öğrenme algoritması ile optik uydu görüntülerinden gemi tespiti yapabilen bir sistem önerilmiştir. Gemi tespit metodu tasarlanırken, lisans sorunu yaşanmaksızın her ortamda kullanılabilmesi için sistemin açık kaynak koduna sahip olması, hızlı çalışması, eğitilebilmesi, derin öğrenme algoritmasına sahip olması ve kullanımının kolay olması hedeflenmiştir. Sistem çok kapsamlı bir donanıma ihtiyaç duymamaktadır, ortalama bir dizüstü bilgisayarda bile çalışabilmektedir. Programlama dili olarak Python 3.7, Python kütüphanesi olarak Anaconda, derin öğrenme kütüphanesi olarak Tensorflow, bilgisayar görü kütüphanesi olarak OpenCV kullanılmıştır. Uygulama Programlama Arabirimi (API) olarak Tensorflow Nesne Tespit API kullanılmış olup, halihazırda farklı nesne tiplerini tanıyabilecek şekilde eğitilmiş ancak optik uydu görüntüsünden gemi tespiti yapamayan nesne tespit modelleri, hiperparametrelerinin değiştirilmesi suretiyle, gemilerin bulunduğu optik uydu görüntüleri ile eğitilmiş, eğitim süreci Tensorboard kayıp grafikleri üzerinden izlenmiş ve modeller arası kıyas yapılmış, eğitilen modeller arasında optimum gemi tespit performansına sahip olan modelin performansının artırılması için eğitim kütüphanesine farklı çözünürlük ve gemi sayısına sahip uydu görüntüleri eklenerek model tekrar eğitilmiş, bahse konu modelin gemi tespit performansında artış olduğu gözlenmiştir.

Sonuç olarak, Tensorflow Nesne Tespit API modellerinin eğitilmesi suretiyle gemi tespit algoritmalarına ilave bir yöntem sunulmuştur. Sonuçların gelecek için umut verici olduğu değerlendirilmektedir. Modelin performansının artırılması için kütüphanedeki görüntülerin sayısının arttırılabileceği, kütüphaneye farklı çözünürlükte ve/veya daha farklı/büyükteki gemi tiplerini barındıran resimler eklenebileceği, resim çözünürlüklerinin standart hale getirilmesinin sisteme etkisinin incelenebileceği, eğitimin farklı modeller, farklı hiperparametreler ve farklı platform/donanımlar ile tekrarlanabileceği veya literatürde belirtilen diğer tespit yöntemleri ile halihazırdaki gemi tespit modelinin beraber kullanılabileceği değerlendirilmektedir.



KAYNAKLAR

- Arguedas, V., F.** (2015). Texture-based vessel classifier for electro-optical satellite imagery, *IEEE International Conference on Image Processing (ICIP)*, 3866-3870.
- Beşbinar, B. and Alatan, A.** (2015). Inshore ship detection in high resolution satellite images: approximation of harbours using sea-land segmentation, *Proc. of SPIE*, 9643(2D), 2-11.
- Doğan, F. and Türkoğlu, İ.** (2018). Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması, *Sakarya University Journal Of Computer And Information Sciences*
- Du, Q., Zhang, Y. Yang, X. and Liu, W.** (2017). Ship target classification based on Hu invariant moments and ART for maritime video surveillance, *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, 414-419.
- Gfeng** (2019). HRSC2016. In Kaggle.com, Alındığı tarih: 28.05.2019 adres: <https://www.kaggle.com/guofeng/hrsc2016>.
- Hadad, Y.** (2018). 30 amazing applications of deep learning, In *yonhadad.com*, Alındığı tarih: 01.05.2019, adres: <http://www.yonhadad.com/deep-learning-most-amazing-applications/>
- 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)*, 1-4.
- LabelImg**, (2019). In *GitHub*, Alındığı tarih: 30.04.2019, adres: <https://github.com/tzutalin/labelImg>.
- Li, T., Liu, Z., Ran, L. and Xie, R.** (2016). An efficient scheme for ship detection in high-resolution TerraSAR-X images, *CIE International Conference on Radar (RADAR)*, 1-4.
- Li, W., Zou, B. and Zhang, L.** (2017). Ship detection in a large scene SAR image using image uniformity description factor, *SAR in Big Data Era: Models, Methods and Applications (BIGSAR DATA)*, 1-5.
- Li, W., et al.**, (2017). Integrated Localization and Recognition for Inshore Ships in Large Scene Remote Sensing Images, *IEEE Geoscience and Remote Sensing Letters*, 14(6), 936-940.
- Rahmani, N. and Behrad, A.** (2011). Automatic marine targets detection using features based on Local Gabor Binary Pattern Histogram

Sequence, *2011 1st International eConference on Computer and Knowledge Engineering (ICCKE)*, 195-201.

Rhammell, (2019). Ships in Satellite Imagery, In *kaggle*, Alındığı tarih: 30.04.2019, adres:<https://www.kaggle.com/rhammell/ships-in-satellite-imagery>.

Shen, W. and Wang, W. (2018). Node Identification in Wireless Network Based on Convolutional Neural Network, *14th International Conference on Computational Intelligence and Security (CIS)*, 238-241.

Song, Z., Sui, H. and Wang, Y. (2014). Automatic ship detection for optical satellite images based on visual attention model and LBP, *IEEE Workshop on Electronics, Computer and Applications*, 722-725.

Soni, G., Singh, A. and Sharma, N. (2015). Inshore ship and hybrid object detection and recognition using context-aware color and shape model, *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 699-703.

Tensorflow Detection Model Zoo, (2019). In *GitHub*, Alındığı tarih: 30.04.2019, adres:https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md.

Tensorflow Object Detection API, (2019). In *GitHub*, Alındığı tarih: 30.04.2019, adres:https://github.com/tensorflow/models/tree/master/research/object_detection.

Yaman, C. and Asari, V. (2007). Long-Range Target Classification in a Cluttered Environment Using Multi-Sensor Image Sequences, *2007 3rd International Conference on Recent Advances in Space Technologies*, 304-308.

Yang, F., Xu, Q., Gao, F. and Hu, L. (2015). Ship detection from optical satellite images based on visual search mechanism, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 3679-3682.

Zhu, C., Zhou, H., Wang, R. and Guo, J. (2010). A Novel Hierarchical Method of Ship Detection from Spaceborne Optical Image Based on Shape and Texture Features, *IEEE Transactions on Geoscience and Remote Sensing*, 48(9), 3446-3456.

ÖZGEÇMİŞ



Ad Soyad : Osman DUMAN
Doğum Yeri ve Tarihi : Karşıyaka – 16.11.1987
E-posta : osmanduman_1987@outlook.com

ÖĞRENİM DURUMU:

- **Lisans** : 2009, Pamukkale Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği.

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2010-2019 yılları arasında Deniz Kuvvetleri Komutanlığında Mühendis Subay olarak çalıştı.

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER

- Kartal, M., and Duman, O., 2019. Ship Detection From Optical Satellite Images With Deep Learning. *International Congress – Recent Advances in Space Technologies*, June 11-14, 2019 İstanbul, Turkey. (Sunum Örneği)