

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OBJECT DETECTION FROM OPTICAL SATELLITE IMAGES
USING DEEP LEARNING TECHNIQUES

ESRA YILDIRIM
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF GEOMATICS ENGINEERING
GEODESY AND GEOGRAPHICAL INFORMATION TECHNOLOGIES PROGRAM

GEBZE
2022

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**OBJECT DETECTION FROM OPTICAL
SATELLITE IMAGES USING DEEP
LEARNING TECHNIQUES**

ESRA YILDIRIM
**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE**
DEPARTMENT OF GEOMATICS ENGINEERING
GEODESY AND GEOGRAPHICAL INFORMATION TECHNOLOGIES PROGRAM

THESIS SUPERVISOR
PROF. DR. TASKIN KAVZOGLU

GEBZE
2022

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TEKNİKLERİNİ
KULLANARAK OPTİK UYDU
GÖRÜNTÜLERİNDEN NESNE TESPİTİ

ESRA YILDIRIM
YÜKSEK LİSANS TEZİ
HARİTA MÜHENDİSLİĞİ ANABİLİM DALI
JEODEZİ VE COĞRAFİ BİLGİ TEKNOLOJİLERİ PROGRAMI

DANIŞMANI
PROF. DR. TAŞKIN KAVZOĞLU

GEBZE
2022



YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 09/06/2022 tarih ve 2022/28 sayılı kararıyla oluşturulan jüri tarafından 22/06/2022 tarihinde tez savunma sınavı yapılan Esra YILDIRIM'ın tez çalışması Harita Mühendisliği Anabilim Dalı, Jeodezi ve Coğrafi Bilgi Teknolojileri Programında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Prof. Dr. Taşkın KAVZOĞLU

ÜYE

: Prof. Dr. Fevzi KARSLI

ÜYE

: Doç. Dr. İsmail ÇÖLKESEN

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

SUMMARY

With the rapid improvements in remote sensing technologies and satellite systems, high-resolution images can be obtained, and many artificial objects can be distinguished from these images. In recent years, deep learning has become a state-of-the-art solution for computer vision problems, including image classification, object detection, and segmentation owing to its large amount of data usage and increasing computational power. In this thesis, automatic ship detection from remotely sensed images was performed using deep learning-based object detection algorithms. For this purpose, the ship dataset was created by combining images in High-Resolution Remote Sensing Detection (TGRS-HRRSD) and A Large-Scale Dataset for Object Detection in Aerial Images (DOTA) datasets. The created dataset was split into training and testing datasets used in the training and accuracy assessment stages. Four pre-trained deep learning algorithms were utilized for detecting ships, including Faster R-CNN, SSD, EfficientDet-D1, and CenterNet combined with ResNet-101 V1, ResNet-50 V1 FPN, EfficientNet-B1-BiFPN, and Hourglass-104 backbones, respectively. All models were trained and tested using the transfer learning approach. Hyperparameter optimization was carried out to obtain the best-performing model. The performance of applied models was analyzed using Common Objects in Context (COCO) evaluation metrics. An independent evaluation was implemented using satellite images from GÖKTÜRK-1 to investigate the transferability of the models. Experimental results revealed that CenterNet model achieved the highest accuracy on the testing dataset, according to COCO metrics, and had the highest generalization capability on GÖKTÜRK-1 images. Given these results, CenterNet was found to have crucial potential applicability in identifying ships in remotely sensed images.

Key Words: Deep Learning, Object Detection, Faster R-CNN, SSD, EfficientDet, CenterNet.

ÖZET

Uzaktan algılama teknolojileri ve uydu sistemlerinde meydana gelen hızlı gelişmelerle birlikte yüksek çözünürlüklü görüntüler elde edilebilmekte ve bu görüntülerden birçok yapay nesne ayırt edilebilmektedir. Son yıllarda, derin öğrenme büyük miktarda veri kullanımı ve artan hesaplama gücü nedeniyle görüntü sınıflandırma, nesne algılama ve segmentasyon dahil olmak üzere bilgisayarla görüş sorunları için popüler bir teknoloji haline gelmiştir. Bu tez çalışmasında, derin öğrenme tabanlı nesne algılama algoritmaları kullanılarak uzaktan algılanan görüntülerden otomatik gemi tespiti gerçekleştirilmiştir. Bu amaçla, High-Resolution Remote Sensing Detection (TGRS-HRRSD) ve A Large-Scale Dataset for Object Detection in Aerial Images (DOTA) veri setlerindeki görüntüler birleştirilerek bir gemi veri seti oluşturulmuştur. Oluşturulan veri seti, eğitim ve doğruluk değerlendirme aşamalarında kullanılan eğitim ve test veri setlerine bölünmüştür. Gemilerin tespiti için sırasıyla ResNet-101 V1, ResNet-50 V1 FPN, EfficientNet-B1-BiFPN ve Hourglass-104 omurgaları ile birleştirilen Faster R-CNN, SSD, EfficientDet-D1 ve CenterNet dahil olmak üzere önceden eğitilmiş dört derin öğrenme algoritması kullanılmıştır. Tüm modeller, transfer öğrenme yaklaşımı kullanılarak eğitilmiş ve test edilmiştir. En iyi performans gösteren modeli elde etmek için hiperparametre optimizasyonu gerçekleştirilmiştir. Uygulanan modellerin performansı, Common Objects in Context (COCO) değerlendirme metrikleri kullanılarak analiz edilmiştir. Modellerin aktarılabilirliğini araştırmak için GÖKTÜRK-1 uydu görüntüleri kullanılarak bağımsız bir değerlendirme yapılmıştır. Deneysel sonuçlar, COCO metriklerine göre test veri setinde CenterNet modelinin en yüksek doğruluğu elde ettiğini ve GÖKTÜRK-1 görüntüleri üzerinde en yüksek genelleştirme yeteneğine sahip olduğunu ortaya koymuştur. Bu sonuçlar göz önüne alındığında, CenterNet'in uzaktan algılanan görüntülerde gemileri tanımlamada önemli potansiyel uygulanabilirliğe sahip olduğu bulunmuştur.

Anahtar Kelimeler: Derin Öğrenme, Obje Tespiti, Faster R-CNN, SSD, EfficientDet, CenterNet.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude and thanks to my supervisor, Prof. Dr. Taskin KAVZOGLU for his support, suggestions, and encouragement. His guidance, sincerity, and motivation have deeply inspired me, and gave me the drive and will to complete this work throughout this process.

Besides my supervisor, I am very grateful to Assoc. Prof. Dr. Ismail COLKESEN for his valuable ideas and guidance during the thesis research and his contribution to the maturation of the thesis work. I want to thank all my co-workers and friends at Gebze Technical University for their encouragement and support.

Last but not least, my great appreciation and gratitude towards all my family, especially my mother and father for their support, inspiration, and prayers for my success.

TABLE of CONTENTS

	<u>Page</u>
SUMMARY	v
ÖZET	vi
ACKNOWLEDGEMENTS	vii
TABLE of CONTENTS	viii
LIST of ABBREVIATIONS and ACRONYMS	xi
LIST of FIGURES	xiv
LIST of TABLES	xviii
1. INTRODUCTION	1
1.1. Purpose of the Thesis	4
1.2. Literature Review	5
2. REMOTE SENSING	10
2.1. Energy Source and Electromagnetic Energy	11
2.2. Interaction of Electromagnetic Radiation with the Atmosphere	13
2.3. Interaction of Electromagnetic Radiation with the Earth Objects	15
2.4. The Concept of Resolution in Remote Sensing	18
2.4.1. Spatial Resolution	18
2.4.2. Spectral Resolution	19
2.4.3. Radiometric Resolution	19
2.4.4. Temporal Resolution	20
3. ARTIFICIAL NEURAL NETWORKS	21
3.1. History of Artificial Neural Networks	23
3.2. Basic Structure and Components of Artificial Neural Networks	25
3.3. Types of Artificial Neural Networks	28
3.3.1. Artificial Neural Networks According to Their Structure	28
3.3.1.1. Feed-forward Networks	28
3.3.1.2. Feed-back Networks	29
3.3.2. Artificial Neural Networks According to Learning Algorithms	30
3.3.2.1. Supervised Learning	30

3.3.2.2. Unsupervised Learning	31
3.3.2.3. Reinforcement Learning	31
3.4. Application Areas of Artificial Neural Networks	32
3.5. Use of Artificial Neural Networks in Remote Sensing	34
3.5.1. Land Use/Land Cover Classification	34
3.5.2. Object Recognition	35
3.5.3. Pixel Unmixing	37
4. DEEP LEARNING	38
4.1. A Brief History of Deep Learning	39
4.2. Deep Learning Architectures	40
4.2.1. Restricted Boltzmann Machines	41
4.2.2. Deep Belief Networks	42
4.2.3. Auto-Encoders	43
4.2.4. Recurrent Neural Networks	44
4.2.5. Long Short-Term Memory Networks	45
4.2.6. Convolutional Neural Networks	46
4.2.6.1. Convolutional Layer	47
4.2.6.2. Pooling Layer	48
4.2.6.3. Fully Connected Layer	49
4.3. Deep Learning-Based Object Detection Models	50
4.3.1. Faster Region-Based Convolutional Neural Network	51
4.3.2. Single Shot Multibox Detector	53
4.3.3. EfficientDet	55
4.3.4. CenterNet	57
4.4. Transfer Learning	59
5. APPLICATION	60
5.1. Dataset	60
5.1.1. DOTA	61
5.1.2. TGRS-HRRSD	62
5.1.3. Preparation of the Dataset	62
5.2. Training Methodology of Deep Learning Models	65
5.3. Hyperparameters Used in Model Training	66
5.4. Accuracy Assessment	68

5.5. Training and Testing Results of Deep Learning Models	70
5.5.1. Faster R-CNN	71
5.5.2. SSD	72
5.5.3. EfficientDet	73
5.5.4. CenterNet	74
5.5.5. Performance Comparison of Deep Learning Models	76
5.6. Visual Evaluation of Results	80
5.7. Generalization Ability Testing	86
6. CONCLUSIONS	97
REFERENCES	100
BIOGRAPHY	115
APPENDICES	116

LIST OF ABBREVIATIONS AND ACRONYMS

<u>Abbreviations and Acronyms</u>	<u>Explanations</u>
λ	: Wavelength
ν	: Frequency
c	: Velocity of light
E	: Electric field
m	: Meter
μm	: Micrometer
M	: Magnetic field
sec	: Second
Adaboost	: Adaptive Boosting
ADALINE	: ADaptive Linear Neuron
AE	: Auto-Encoder
ANN	: Artificial Neural Network
AP	: Average Precision
API	: Application Programming Interface
AR	: Average Recall
ART	: Adaptive Resonance Theory Network
AVHRR	: Advanced Very High-Resolution Radiometer
BiFPN	: Bi-directional Feature Pyramid Network
BOW	: Bag-Of-Words
BP	: Backpropagation
CFAR	: Constant False Alarm Rate
CNN	: Convolutional Neural Network
COCO	: Common Objects in Context
DBN	: Deep Belief Network
DL	: Deep Learning
DNN	: Deep Neural Network
DOTA	: A Large-Scale Dataset for Object Detection in Aerial Images
ECG	: Electrocardiogram
ELM	: Extreme Learning Machine

FCL	: Fully Connected Layer
FCN	: Fully Convolutional Network
FLOPs	: Floating-Point Operations per Second
FN	: False Negative
FP	: False Positive
FPN	: Feature Pyramid Network
GA	: Genetic Algorithm
GIS	: Geographical Information Systems
GPU	: Graphics Processing Unit
GRNN	: General Regression Neural Network
HECAN	: High-Efficiency Channel Attention Network
HOG	: Histogram of Oriented Gradient
IFOV	: Instantaneous Field of View
ILSVRC	: ImageNet Large-Scale Visual Recognition Challenge
IoU	: Intersection Over Union
KNN	: K-Nearest-Neighbor
KSOFM	: Kohonen Self-Organizing Feature Map
LIDAR	: Light Detection and Ranging
LR	: Learning Rate
LSTM	: Long Short-Term Memory
LULC	: Land Use and Land Cover
LVQ	: Learning Vector Quantization
mAP	: Mean Average Precision
MLP	: Multilayer Perceptron
MPANN	: Memetic Pareto Artificial Neural Network
MSNN	: Morphological Shared-Weight Neural Network
NAS-FPN	: Neural Architecture Search Feature Pyramid Network
NMS	: Non-maximum Suppression
PANet	: Path Aggregation Network
PNN	: Probabilistic Neural Network
RADAR	: Radio Detection and Ranging
RBM	: Restricted Boltzmann Machine
R-CNN	: Region-Based Convolutional Neural Network

R-DFPN	: Rotation Dense Feature Pyramid Network
ReLU	: Rectified Linear Unit
RESNET	: Residual Network
RNN	: Recurrent Neural Network
ROI	: Regions of Interest
RPN	: Region Proposal Network
RS	: Remote Sensing
SAR	: Synthetic Aperture Radar
SOM	: Self-Organizing Map
SPP-NET	: Spatial Pyramid Pooling Network
SSD	: Single Shot MultiBox Detector
SVDNET	: SVD Network
SVM	: Support Vector Machine
SWISH	: Self-Gated Activation Function
TGRS-HRRSD	: High-Resolution Remote Sensing Detection
TP	: True Positive
VGGNET	: Visual Geometry Group Network
XML	: Extensible Markup Language
XOR	: Exclusive OR
YOLO	: You Only Look Once

LIST OF FIGURES

<u>Figure No:</u>		<u>Page</u>
2.1:	Electromagnetic waves.	11
2.2:	Electromagnetic spectrum.	12
2.3:	Interaction between radiation and object.	15
2.4:	Typical spectral reflectance curves of vegetation, soil, and water.	16
2.5:	Landsat 8 images with different spatial resolutions.	18
2.6:	Images with different radiometric resolutions.	20
3.1:	Structure of a biological neuron.	25
3.2:	Structure of an artificial neuron.	26
3.3:	A simple three-layer neural network structure.	27
3.4:	Artificial neural network architectures in terms of their structure. (a) Feed-forward network, (b) Feed-back network.	29
4.1:	Relationships between artificial intelligence, machine learning, and deep learning.	38
4.2:	The architecture of the first known deep neural network.	39
4.3:	The basic structure of RBM with n hidden and m visible units.	41
4.4:	An architecture of DBN.	42
4.5:	A structure of an auto-encoder with its component.	43
4.6:	An unrolled representation of an RNN.	44
4.7:	Basic structure of an LSTM.	45
4.8:	Architecture of LeNet-5, a CNN.	46
4.9:	A simple depiction of 2D convolution operation.	48
4.10:	Example of average and maximum pooling operations with a stride of 2.	49
4.11:	The structure of the fully connected layer.	49
4.12:	Architecture of R-CNN model.	51
4.13:	Fast R-CNN model architecture.	52
4.14:	Architecture of Faster R-CNN.	52
4.15:	The network architecture of SSD.	53
4.16:	SSD model framework.	54

4.17:	The overall architecture of EfficientDet.	55
4.18:	Comparison of different networks for multi-scale feature fusion.	56
4.19:	Comparison of anchor-based detectors and center point detector. (a) Standard anchor-based detection, (b) Center point-based detection.	57
4.20:	Overview of the CenterNet network outputs for the object detection task. (a) Keypoint heatmap, (b) Local offset, (c) Object size.	58
5.1:	Samples of three annotated images per category in the DOTA dataset.	61
5.2:	Example images of each object category from the TGRS-HRRSD dataset.	62
5.3:	Representative images of ship dataset.	63
5.4:	Sample ship images collected from the TGRS-HRRSD and DOTA datasets, with ground-truth bounding boxes depicting some annotation mistakes. (a) Two small unannotated ships, (b) Bounding box that cannot correctly delimit the ship, i.e., larger than the ship, (c) An unannotated ship.	63
5.5:	The details of the image annotation process. (a) Original image, (b) Labeled image, (c) Annotation file containing the size of the image (blue rectangle), the category of the target (red rectangle), and the position of the ship target (green rectangle).	64
5.6:	Visual representation of IoU.	68
5.7:	Loss curves of ResNet-101 V1 based Faster R-CNN. (a) Box classifier classification loss, (b) Box classifier localization loss, (c) RPN localization loss, (d) RPN objectness loss, (e) Total training and testing loss.	71
5.8:	Loss curves of ResNet-50 V1 FPN based SSD. (a) Classification loss, (b) Localization loss, (c) Total training and testing loss.	73

5.9:	Loss curves of EfficientNet-B1-BiFPN based EfficientDet. (a) Classification loss, (b) Localization loss, (c) Total training and testing loss.	74
5.10:	Loss curves of Hourglass-104 based CenterNet. (a) Box offset loss, (b) Box scale loss, (c) Object center loss, (d) Total training and testing loss.	75
5.11:	Mean Average Precision scores (mAP) of all models. (a) mAP at 0.50:0.95 IoU, (b) mAP for large objects, (c) mAP for medium objects, (d) mAP for small objects, (e) mAP at 0.50 IoU, (f) mAP at 0.75 IoU.	77
5.12:	Average Recall (AR) scores of all models. (a) AR, given 1 detection per image, (b) AR, given 10 detection per image, (c) AR, given 100 detection per image, (d) AR for large objects, (e) AR for medium objects, (f) AR for small objects.	78
5.13:	Graphical comparison of all models with COCO metrics.	80
5.14:	Qualitative ship detection results of all models on test image 1. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	81
5.15:	Qualitative ship detection results of all models on test image 2. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	82
5.16:	Qualitative ship detection results of all models on test image 3. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	83
5.17:	Qualitative ship detection results of all models on test image 4. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	84
5.18:	Qualitative ship detection results of all models on test image 5. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	85
5.19:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	87

5.20:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	88
5.21:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	89
5.22:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	90
5.23:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	91
5.24:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	92
5.25:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	93
5.26:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	94
5.27:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	95
5.28:	Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.	96

LIST OF TABLES

<u>Table No:</u>		<u>Page</u>
5.1:	Number of images and ship instances in the dataset.	64
5.2:	Hardware and software specifications used in this study.	65
5.3:	Training hyper-parameters of the models.	66
5.4:	MS COCO dataset object detection evaluation metrics.	70
5.5:	Performance comparison of all models with MS COCO metrics on the testing dataset.	79
5.6:	Technical specifications of GÖKTÜRK-1 satellite.	87

1. INTRODUCTION

The detection and identification of objects is one of the primary issues in the aerial and satellite image processing field and is of utmost significance in many research domains such as urban monitoring, disaster management, and precision agriculture. In parallel with the developments in remote sensing technology and satellite systems for Earth observation, the number and quality of remotely sensed images are increasing daily. These images obtained from airborne and spaceborne sensors provide up-to-date and more detailed information allowing for more precise classification and positioning of the target object. With these high-resolution images, it has become possible to detect even small artificial objects. Nowadays, studies on the recognition of numerous targets such as buildings [Mei et al., 2021], ships [Yildirim and Kavzoglu, 2021], aircraft [Wu et al., 2021], and trees [Yildirim et al., 2022], in the field of automatic object recognition using remotely sensed images have become increasingly widespread. To be more specific, the detection and recognition of ship objects have a pivotal role in various applications, including monitoring maritime traffic, maritime rescue, monitoring sea pollution, and identifying illegal fishing [He et al., 2017a], [Abileah, 2009]. Additionally, it conduces to the surveillance and building of significant harbors and coastal zones. In particular, given its military and civilian reconnaissance, the detection and identification of ship targets have received considerable attention in the remote sensing field.

Object detection addresses a twofold problem that locates an object in an image (localization) and labels that object in a specific category (classification). However, this is a challenging task because of the varying sizes, orientations, types, and complex backgrounds that objects have in the images. In addition, this task includes various difficulties such as viewpoint variation, lighting, shadow, occlusion, and imaging sensor which cause major variations in the visual appearance of objects [Cheng and Han, 2016]. What is worse, the recognition of small-scale objects in the images is quite challenging. Driven by these requirements, many methods have been developed for detecting diverse types of objects in remotely sensed images, and overcoming the challenges mentioned above. Traditional object recognition methods commonly adopt a three-stage identification pipeline: informative region generation, feature extraction, and classification. Firstly, informative region generation is the search for regions of

interest that contain objects, and the sliding windows method is utilized for this search. Then, the feature extraction phase is carried out to withdraw the features of the generated regions and obtain the feature vectors. For this, manually designed features such as Haar-like [Viola and Jones, 2001], Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005], and Bag-of-words (BoW) [Fei-Fei and Perona, 2005] are used. Finally, in the classification stage, the labels of the object classes are matched to the regions generated and encoded with the features. At this phase, classifiers, with varying complexity such as support vector machine (SVM) [Mountrakis et al., 2011], AdaBoost [Freund and Schapire, 1997], and k-nearest-neighbor (kNN) [Peterson, 2009] are utilized. However, the traditional object recognition mechanism suffers from several deficiencies. The feature extraction phase is of vital importance as detection accuracy extremely depends on the quality of the hand-crafted features and well-extracted features boost detection accuracy. The sliding window strategy is generally blind and can produce a large amount of redundant candidate window, resulting in high computational complexity. Thus, achieving real-time performance is difficult for traditional methods. In addition, manually extracted features are not robust for objects with complex and variable backgrounds in the image. Consequently, these conventional methods lack high-level semantic information and have poor generalization ability.

In recent years, with the emergence of the deep learning paradigm, the computer vision community has witnessed considerable advances in the performance of image processing applications, including image segmentation [Badrinarayanan et al., 2017], object detection [Redmon et al., 2016], and image classification [Yang et al., 2018c]. The deep learning concept originates from artificial neural networks (ANNs), the core principle of which underlies the desire to create processing systems that mimic the activity of the human brain and the nervous system [Kavzoglu, 2001]. Deep learning algorithms can automatically and adaptively extract features from the given data with a self-learning approach by building a deep neural network, thus overcoming the drawbacks of traditional object recognition methods that perform feature extraction based on manual selection and expert experience. In particular, with the tremendous success of the famous AlexNet [Krizhevsky et al., 2012], a convolutional neural network (CNN) architecture, at the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012 [Deng et al., 2009], the implementation of CNNs in the field of image processing has started booming. After the AlexNet, many novel CNN

architectures including ZFNet [Simonyan and Zisserman, 2014], visual geometry group networks (VGGNet) [Zeiler and Fergus, 2014], GoogleNet [He et al., 2016], and residual networks (ResNet) [Szegedy et al., 2015] have been presented to boost its performance. Given the evolution of CNN architectures, the trend depicts that the networks are getting deeper thereby they can get better feature representations [Gu et al., 2018]. In this vein, the object detection and recognition architectures based on the CNNs began to improve rapidly and are increasingly applied in object detection tasks from remotely sensed images due to their strong feature representation capabilities [Ni and Hua, 2021], [Han et al., 2020]. Furthermore, the usage of a massive amount of data and the improvement of the performance of graphics processing units (GPUs) have also increased their popularity.

At present, there are broadly two kinds of deep learning-based object detectors, namely two-stage detectors and one-stage detectors. As the name implies, two-stage detectors consisting of two phases utilize a Region Proposal Network (RPN) to generate regions of interest (ROI) from the input images, and then each proposal is classified and the location of the target object in the input image is determined by bounding-box regression to get final detection result. These detectors mainly include Region-Based Convolutional Neural Network (R-CNN) [Girshick et al., 2014], Fast R-CNN [Girshick, 2015], Faster R-CNN [Ren et al., 2015], Spatial Pyramid Pooling Network (SPP-Net) [He et al., 2015], and Mask R-CNN [He et al., 2017b]. On the other hand, one-stage detectors omit an ROI generation process and regard object identification as a regression task by adopting a unified system that class probabilities and bounding box positions are regressed directly from the input images. The representative one-stage architectures are the You Only Look Once (YOLO) family [Redmon et al., 2016], [Redmon and Farhadi, 2017], [Redmon and Farhadi, 2018], [Bochkovskiy et al., 2020], [Web 1, 2022], Single Shot MultiBox Detector (SSD) [Liu et al., 2016], EfficientDet [Tan et al., 2020], and CenterNet [Zhou et al., 2019]. The principal difference between two-stage and one-stage detectors lies in whether ROI is produced in the detection process. In comparison with the two-stage object detection models, the detection speed of the one-stage algorithms is breakneck and apt for utilization in real-time detections, but their accuracy is lower. It should be highlighted that both one-stage and two-stage object detectors have their own merits based on the practices.

1.1. Purpose of the Thesis

This thesis evaluated and compared the identification performances of deep learning object detection meta-architectures from remotely sensed images in the context of ship objects. For the training and testing of models, a ship dataset was created by selecting images containing ship targets from the open-source A Large-Scale Dataset for Object Detection in Aerial Images (DOTA) and High-Resolution Remote Sensing Detection (TGRS-HRRSD) datasets which were created for object detection from high-resolution satellite imagery and are frequently used in the literature. Although these datasets were already labeled, it was observed that ground-truth boxes in some images were mislabeled, and some small ships were unlabeled. The ships in the collected images were relabeled manually with the bounding box to overcome these annotation errors and achieve more precise detection results. Four pre-trained CNN-based object detectors were employed to obtain an accurate and robust model that would precisely classify and locate ship samples. For this purpose, both one-stage and two-stage models were investigated, and their effectiveness was compared. These models are pre-trained on TensorFlow deep learning library with the Common Objects in Context (COCO) dataset. In this context, Faster R-CNN, SSD, EfficientDet-D1, CenterNet models, combined with ResNet-101 V1, ResNet-50 V1 FPN, EfficientNet-B1-BiFPN, and Hourglass-104 feature extractors, respectively, were evaluated. All deep learning models were trained using a transfer learning approach and fine-tuned for better performance through the TensorFlow Object Detection Application Programming Interface (API), making it straightforward to create, train, and use object detection algorithms. The transfer learning approach allows using weights/parameters from pre-trained models to train deep neural networks and overcome the excessive time and resource requirements to train the network from scratch. Then, the performance of the trained models was analyzed using COCO metrics. These metrics include a total of 12 average precision (AP) and average recall (AR) defined on different object scales, intersection over union (IoU) scales, and detections per image. Moreover, to investigate the generalization capability of the models on different images from the training and testing dataset, GÖKTÜRK-1 satellite imagery was introduced to all models and the detection results were examined.

1.2. Literature Review

Researchers have extensively studied the topic of ship detection and identification from remotely sensed images in the last decade due to its broad range of applications. In particular, the availability of an increasing number of remotely sensed images with high spectral and spatial resolution and the practicality of recognizing even small ship instances from these images have increased the interest in this topic. Before the advent of the deep learning paradigm that led to emerging breakthroughs in target detection, traditional techniques based on handcrafted features were utilized for this task. However, the deep learning-based object detectors have outperformed them with a large margin in terms of speed as well as accuracy and have become mainstream methods for detecting ships from remotely sensed images. Most of these methods utilize CNNs to extract the input image features and determine target position and ship class. One of the major causes behind this success is that deep learning architectures automatically perform feature extraction by themselves without the need for a manually handcrafted feature extraction process.

Zhu et al. [2010] introduced a hierarchically complete and operational method relying on texture and shape features for detecting ship objects from space-borne optical images. By utilizing image segmentation and simple shape analysis, several factors influencing ship detection performance such as sea clutter, islands, and clouds were removed, and ship candidates were obtained. Afterward, most false alarms were eliminated with a semi-supervised hierarchical classification process with diverse texture and shape features. Although the proposed algorithm was efficient in distinguishing ship and non-ship objects and achieved satisfying identification performance, it was stated that it needed performance improvements due to missing detections and false candidate issues [Zhu et al., 2010].

Corbane et al. [2010] designed an operational ship detection system based on signal processing algorithms, statistical methods, and mathematical morphology and utilize high-resolution optical imagery. The detection of potential ship instances was conducted with automatic threshold estimation and associated filtering. The probabilities of belonging to the ship class were predicted with a logistic model. This model was based on the variables derived from Radon and Wavelet transforms. It was

mentioned that the proposed algorithm was robust and fast in ship detection and showed promising results in identifying small ships [Corbane et al., 2010].

Liu et al. [2014] applied a novel method that utilizes context and shape information to identify inshore ships from high-resolution satellite images. Water and land were segmented in the images and consecutive shape analysis was performed to detect the ship targets in the segmented images. The position of the ship's head was determined by analyzing the shape, and the width and length of the ship were calculated through region growing. Context information was used to accurately locate ship objects and eliminate false alarms. According to the results, it was reported that the proposed framework was robust and effective under diverse conditions [Liu et al., 2014].

Shi et al. [2014] introduced an approach based on local shape features and anomaly detection to identify ship targets from high-resolution optical panchromatic images. In this approach, the panchromatic image was transformed into a hyperspectral data form, and object candidate regions were extracted utilizing a hyperspectral anomaly detector Reed–Xiaoli. An extended HOG feature was extracted and hypotheses for ship targets were produced using the AdaBoost algorithm. Although the presented approach achieved satisfactory results in ship identification, it produced inferior performance when dealing with ships close to land [Shi et al., 2014].

Yang et al. [2014] designed a sea surface analysis method aiming to automatically identify ship targets from high-resolution optical imagery with diverse sea surfaces. In this strategy, the homogeneousness of the sea surface was analyzed, and ship candidates were selected using a linear selection function combining region and pixel features. Afterward, false alarms were eliminated by using length-width ratio and compactness. It was mentioned that the proposed approach was the first to utilize the sea surface analysis for ship recognition from optical imagery and could have a broad application in real-time ship identification systems due to its computational efficiency and detection accuracy. It was also reported that the algorithm produced false alarms in cluttered seas, that is, under interference by ship wakes and clouds, etc. [Yang et al., 2014].

Qi et al. [2015] constructed an unsupervised framework based on saliency and ship histogram of oriented gradient, namely S-HOG descriptor for detecting ships in optical satellite images. First, the candidate regions were extracted by the visual salience, and suspected targets were removed using a homogeneous filter. Then, the

S-HOG descriptor was utilized to identify ships. It was stated that the applied approach was robust and performed well for complex scenes and varying ship sizes [Qi et al., 2015].

Tang et al. [2015] introduced a compressed-domain ship target identification approach that uses deep neural network (DNN) and extreme learning machine (ELM) to identify ship objects in optical spaceborne imagery. The compressed domain was utilized to extract ship target candidates quickly. DNN was adopted to obtain high-level ship feature representation providing more accurate classification. In addition, ELM, which has better generalization ability and learns extremely faster than other conventional learning methods, was utilized for effective feature fusion and decision making. It was reported that the features extracted in the proposed method were quite robust and the approach provided higher detection accuracy and required less detection time [Tang et al., 2015].

Zou and Shi [2016] presented a ship target detection algorithm from optical satellite images called SVD Networks (SVDNet). This algorithm leveraged CNNs and the singular value decomposition (SVD) algorithm. First, in the ship candidate extraction phase, convolutional and nonlinear mapping layers were utilized. Then, the linear SVM algorithm and the feature pooling process were adopted in the ship verification phase. According to experiments on optical images from the Venezuelan Remote Sensing Satellites and GaoFen-1, it was mentioned that the proposed framework provided the desirable time performance and high detection robustness. Besides, it was stated that future work was related to integrating information of both global and local environments and reducing the memory cost [Zou and Shi, 2016].

Kang et al. [2017] introduced a modified Faster R-CNN model using the constant false alarm rate (CFAR) algorithm to identify ship objects in synthetic aperture radar (SAR) imagery. In this presented framework, the Faster R-CNN object detector was utilized for generating proposals at various scales and evaluating the target with a classification score. Small-scaled ship objects, which obtained low classification scores and were regarded as false alarms, were detected by the CFAR algorithm according to the amplitude of the pixels. It was stated that the experiments conducted on SAR image demonstrated that the proposed approach dramatically improved the detection rate of small-sized ship targets [Kang et al., 2017].

Yang et al. [2018] designed Rotation Dense Feature Pyramid Networks (R-DFPN) framework to overcome ship detection challenges caused by the complexity of

scenes, dense objects, and redundancy of detection regions in remotely sensed images. In this framework, DFPN was put forward to enhance feature propagation and encourage feature reuse. Besides, rotational anchors (proposals) were utilized to generate high-quality proposals. Additionally, a multiscale ROI Align method was utilized to maintain the feature information completely. It was noticed that although experiments on Google Earth images demonstrated that the introduced approach had state-of-the-art performance, the false alarm issue still existed [Yang et al., 2018b].

Zhang et al. [2019] introduced an improved Faster R-CNN algorithm for detecting ship targets in high-resolution optical remotely sensed imagery. First, the non-water area was segmented from the water area in selected Gaofen-2 images, and ship candidate regions that may contain ship objects (ROI) were extracted. Afterward, Faster R-CNN with VGG-16 backbone was adopted to detect ships, and its structure was improved. According to the results of the comparison with different models and the baseline Faster R-CNN algorithm combined with the VGG-16 backbone, the proposed method provided higher performance in identifying small ships and gathering ships [Zhang et al., 2019a].

Nie et al. [2020] constructed a ship target identification and segmentation framework from remotely sensed imagery based on the improved Mask R-CNN instance segmentation algorithm. In this framework, a bottom-up structure was added to the FPN architecture of the Mask R-CNN model, and attention mechanisms were utilized. With these improvements applied, it was reported that the model detected and segmented ship objects with higher accuracy than the Mask R-CNN baseline model and other models, and the small ship detection and segmentation accuracy also needed improvement [Nie et al., 2020].

Chen et al. [2021] utilized an attention mechanism to improve the YOLOv3 object detection model for ship detection task and obtain the best trade-off between its recognition speed and accuracy. For this aim, a novel and lightweight dilated attention module highlighting the difference between ship objects and background was introduced, and this module was embedded into the YOLOv3 model. As a result of the study, it was stated that the improved algorithm demonstrated promising performance in the identification of ship targets at different scenes and scales and achieved a real-time speed [Chen et al., 2021].

Zhang et al. [2021] utilized the Mask R-CNN and Faster R-CNN algorithms to automatically recognize ships from optical remotely sensed images. The Faster R-

CNN and Mask R-CNN models were trained on Google Earth images with the VGG-16 and ResNet-50 backbones, respectively, and the results were compared. Experimental results demonstrated that the ship identification accuracies of Faster R-CNN and Mask R-CNN algorithms were 92.76% and 95.21%, respectively. It was stated that the Mask R-CNN achieved pixel-level segmentation and outperformed the Faster R-CNN model in ship object identification [Zhang et al., 2021].

Xie et al. [2021] introduced an improved EfficientDet model, called High-Efficiency Channel Attention Network (HECAN), to overcome missed detection and false alarm problems in detecting side-by-side and small-sized ship targets in optical remotely sensed imagery. In this framework, high-efficiency channel attention was used to increase the small-scale ship detection capability, and the Soft-NMS algorithm [Bodla et al., 2017] was utilized to enhance the recall rate of side-by-side ships. It was stated that the proposed framework better detected side-by-side and small-scale ship targets, with a 1.63% higher average precision and 4.75% higher recall rate than the baseline EfficientDet model [Xie et al., 2021].

Sun et al. [2022] made enhancements to the original YOLO V5s object detector to overcome the poor identification of dense small ship targets in detecting ships from remotely sensed images. The enhanced Kmeans and DIOU_loss were utilized for clustering the dataset and improving the candidate frame selection approach. In addition, the channel attention mechanism was presented. According to the experimental results, it was stated that although the detection speed decreased slightly compared to the baseline YOLOv5 model, the mean average precision (mAP) increased by 22.48% and smaller ship target instances could be detected [Sun et al., 2022].

2. REMOTE SENSING

Remote sensing (RS) is the science and art of acquiring information about an area, object, or phenomenon through measuring instruments placed at a certain distance from the earth that is not in physical or intimate contact with the area, object, or phenomenon under investigation [Lillesand et al., 2015]. The RS is based on measuring the amount of energy reflected or emitted from earth objects by remote sensing sensors. In the historical development process, first in the 1800s aerial photographs obtained with cameras placed on balloons, kites, and pigeons' chest gained popularity with the discovery of the aircraft and even became a vital reconnaissance tool in the First and Second World Wars. The remote sensing technology, which was first started in the 1960s to collect intelligence with spy satellites, has been utilized for civilian purposes since the 1970s using the Landsat series satellites developed by the United States, and has developed and extended to the present day [Kavzoğlu and Çölkesen, 2011]. In particular, after the 1980s, after sending high-resolution satellites to space, remote sensing data began to be utilized as a base dataset by many disciplines. More specifically, remotely sensed data has various application fields in many disciplines such as geology, forestry, cartography, hydrology, environment, agriculture, and urbanism [Çölkesen, 2009]. These studies include detecting marine mucilage [Kavzoglu et al., 2021], monitoring forest fires [Tonbul et al., 2022], determining land use and land cover [Colkesen and Kavzoglu, 2019], and identifying tree species [Tonbul et al., 2020]. Besides, many Geographical Information Systems (GIS) applications integrate remotely sensed images for various analyses, e.g., [Weng, 2002], [Brivio et al., 2002], [Pradhan, 2009].

Remote sensing systems are categorized into passive and active sensing systems in terms of the type of energy they use. Passive remote sensing systems measure the energy emitted or reflected from the target object through the rays of the sun, that is, they do not produce any energy. On the other hand, active remote sensing systems produce their own energy, transmit it, and measure the energy reflected back from the target object. In this context, aerial photography can be given as an example of passive remote sensing systems, while active remote sensing systems include Radio Detection and Ranging (RADAR) and Light Detection and Ranging (LIDAR).

2.1. Energy Source and Electromagnetic Energy

In remote sensing, observations and measurements of electromagnetic energy emitted from the object of interest made with sensors carried on satellites or aircraft are utilized. The first requirement of remote sensing is to have an energy source that illuminates the target if the target does not emit the detected energy. Considering this requirement, the sun is the most significant energy source in remote sensing and the light energy from the sun reaches the earth in the form of electromagnetic waves. The propagation of electromagnetic energy as waves from a source is defined as electromagnetic radiation. Visible light, ultraviolet rays, radio waves, radiant heat, and X-rays are forms of electromagnetic radiation. All this electromagnetic radiation has basic properties and propagates according to the basic wave theory. According to this theory, electromagnetic energy travels in harmonic sinusoidal waves at the velocity of light ($c = 3 \times 10^8 \text{ m/sec}$). The electromagnetic wave structure is shown in Figure 2.1 [Web 2, 2022]. The electromagnetic wave consists of electric (E) and magnetic (M) fields perpendicular to each other and to the direction of propagation. The wave characteristic of electromagnetic energy is defined by the wavelength (λ), which is the distance from one electromagnetic wave peak to the next, and the wave frequency (ν), which is the number of peaks that pass through a fixed point in space per unit time. The speed of the electromagnetic wave is the velocity of light, and it is equal to the multiplication of its frequency and wavelength [Lillesand et al., 2015].

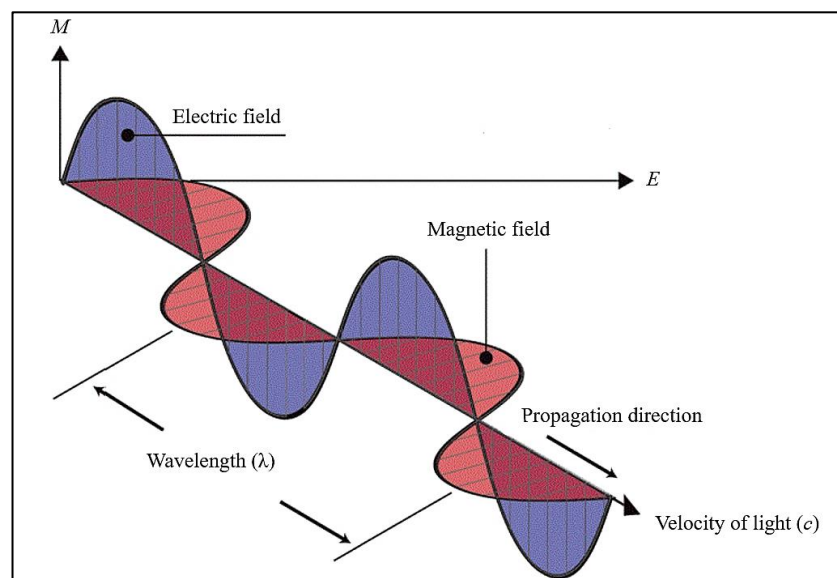


Figure 2.1: Electromagnetic waves.

The electromagnetic energy spectrum is a continuous energy field from wavelengths of 10^{-10} μm (cosmic rays) to wavelengths of 10^{10} μm (radio waves). In remote sensing, electromagnetic waves are categorized according to their wavelength locations in the electromagnetic spectrum. The unit utilized to measure wavelength along the electromagnetic spectrum is micrometer (μm), and one micrometer is equivalent to 1×10^{-6} m. The visible portion of the electromagnetic spectrum shown in Figure 2.2 is extremely small, with the spectral sensitivity of the human eye only extending between wavelengths of 0.4 to 0.7 micrometers [Web 3, 2022]. In this region, the blue color has a range of 0.4 to 0.5 μm , the green color 0.5 to 0.6 μm , and the red color 0.6 to 0.7 μm about. Ultraviolet energy is adjacent to the blue wavelength in the visible portion of the spectrum. There are three distinct categories of infrared waves adjacent to the red wavelength located at the end of the visible portion. These are, respectively, near-infrared (0.7 to 1.3 μm), mid-infrared (1.3 to 3 μm), and thermal infrared (beyond 3 to 14 μm). Longer wavelengths (1 mm to 1 m) are defined as the microwave portion of the spectrum. Remote sensing sensors mostly use the microwave, infrared, and visible wavelength ranges to collect information about the earth. The electromagnetic radiation in distinct portions of the electromagnetic spectrum interacts differently with different objects. The thermal properties of surfaces can be best analyzed in the thermal infrared portion of the spectrum, and their dielectric properties and roughness in the microwave portion. Consequently, different sensing systems are designed to determine the properties of objects with different chemical and physical structures, and to discriminate them from each other, using the energy emitted and reflected by earth objects in different portions of the spectrum.

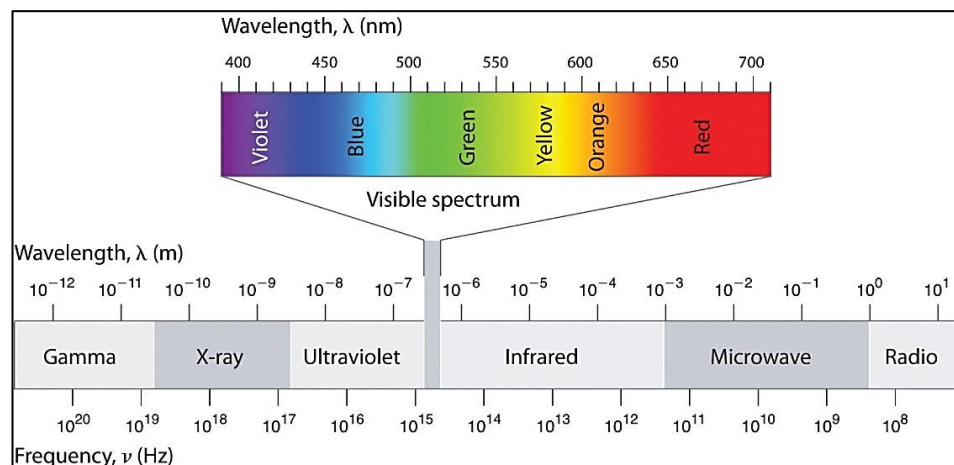


Figure 2.2: Electromagnetic spectrum.

2.2. Interaction of Electromagnetic Radiation with the Atmosphere

The radiation (energy) used for remote sensing must pass through some distance of the earth's atmosphere before reaching the earth's surface. As the radiation passes through the atmosphere, it interacts with the particles and gases in the atmosphere and undergoes some changes and weakens. As a result of this interaction, the net impact of the atmosphere on the radiation varies depending on parameters such as the length of the path taken in the atmosphere, the magnitude of the sensed energy signal, the current atmospheric conditions, and the wavelengths involved. Radiation interacting with the atmosphere can be scattered, absorbed, or reflected into space. Particles suspended in the atmosphere such as gas molecules, fog and clouds, dust and water droplets cause electromagnetic radiation to be redirected from its original path. This phenomenon is called scattering, and the amount of scattering depends on the wavelength of the radiation, the density of the particles, and the distance that radiation takes through the atmosphere. In this context, scattering occurs in three ways, depending on the relationship between the size of these particles causing scattering in the atmosphere and the size of the wavelength of the scattered radiation. These are Rayleigh, Mie, and nonselective scatterings [Lillesand et al., 2015].

Rayleigh scattering, which occurs in the upper layers of the atmosphere, happens when the particles causing the scattering (small specks of dust, nitrogen, and oxygen molecules, etc.) are smaller than the wavelength of the radiation. The amount of Rayleigh scattering is inversely proportional to the fourth power of the wavelength. For this reason, short wavelengths tend to scatter much more than long wavelengths. The sky appears blue as most of the energy in the blue wavelength, the shortest wavelength in the visible region of the spectrum, is scattered by particles and gas molecules in the atmosphere. Moreover, at sunrise and sunset, the scattering at longer wavelengths in the visible region is much greater, as the sun's rays travel farther through the atmosphere, compared to the middle of the day, and this causes the sky to appear red. However, Rayleigh scattering is one of the main causes of haze that reduces the crispness or contrast of the image. Also, Rayleigh scattering makes it arduous to obtain and evaluate images in the blue part of the electromagnetic spectrum in remote sensing. To eliminate this impact, special filters are utilized in aerial photographs.

Mie scattering occurs when the size of the atmospheric particles (pollen, smoke, dust, water vapor, etc.) is almost equal in size to the wavelength of the radiation. This scattering, which happens in the lower parts of the atmosphere where larger particles are more abundant, tends to influence longer wavelengths than Rayleigh scattering. Mie scattering dominates under overcast atmospheric conditions or when the particles in the atmosphere are slightly larger.

Non-selective scattering is caused by the particles (water droplets and large dust particles, ice crystals, etc.) much larger than the wavelength of the radiation. This scattering that takes place in the lower atmosphere is not wavelength-dependent, that is, it scatters all wavelengths equally. Fog and clouds appear white because of approximately equal quantities of scattering of blue, green, and red light in the visible region. This scattering creates haze in remotely sensed images, resulting in loss of crispness. It lowers the contrast in the image and reduces the spatial detail information.

Another phenomenon that occurs during the interaction of electromagnetic radiation with the atmosphere is atmospheric absorption. In contrast to scattering, absorption is a process in which molecules absorb radiation of various wavelengths in the atmosphere. In this context, three principal atmospheric components that absorb radiation are ozone (O_3), carbon dioxide (CO_2), and water vapour. Others are methane (CH_4), nitrous oxide (NO), and chlorofluorocarbons (CFCs). These gases cause the absorption of radiation of specific wavelengths. Ozone in the upper atmosphere causes the absorption of ultraviolet rays, which are harmful to living things. Carbon dioxide in the lower atmosphere, known as a greenhouse gas, tends to absorb infrared light. Water vapour in the lower atmosphere absorbs thermal infrared lights and near microwave rays. Since the presence of water vapour changes according to location and time, its influence is different for each location and time.

The wavelength ranges where radiation passes through the atmosphere with low attenuation and is not heavily affected by atmospheric absorption are called atmospheric windows. In this context, the permeability of the atmosphere changes with the point of view and, accordingly, the length of the path traveled by the energy in the atmosphere. The visible portion of the electromagnetic spectrum, to which the human eye is sensitive, corresponds to an atmospheric window. Only specific wavelengths can be sensed due to atmospheric absorption in remote sensing, remote sensors are designed to operate in one or more of these atmospheric windows.

2.3. Interaction of Electromagnetic Radiation with the Earth Objects

Electromagnetic radiation reaching the earth's surface without being absorbed or scattering from the atmosphere can interact with objects on the earth's surface in three distinct ways. When energy (I) passing through the atmospheric window is incident upon an object on the earth, it can be reflected by the surface of that object (R), absorbed by the electrons or molecules that make up the object (A), and/or pass through the object, that is, it can be transmitted (T). These energy interactions are illustrated in Figure 2.3. The total incident energy interacts with the earth's surface through one or more of these three impacts, and the proportions of each vary depending on the wavelength of the energy, the material, and conditions of the surface. This relationship between energy and object forms the basis of remote sensing [Çölkesen, 2015].

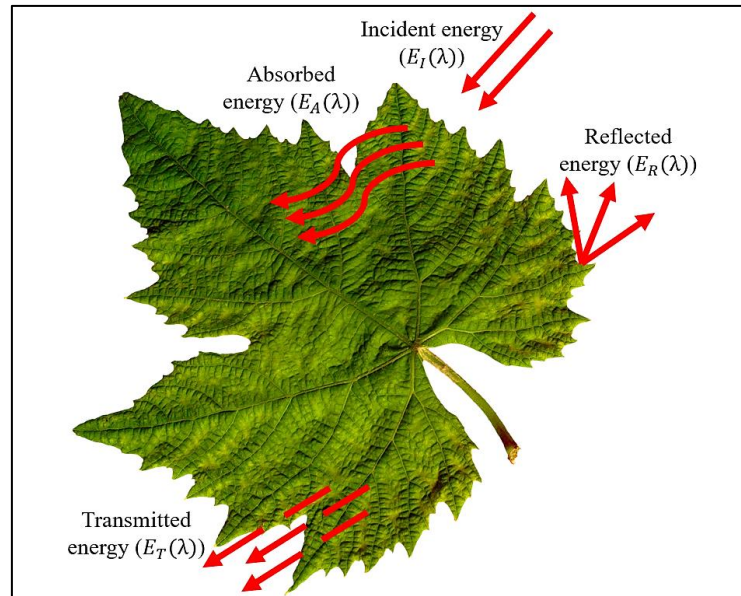


Figure 2.3: Interaction between radiation and object.

Considering that the energy is never lost, but merely changes its form, the total energy incident on the object is equal to the sum of the energies reflected by the object surface, absorbed, and/or transmitted by the object. The energy balance equation expresses the interrelationship among these three types of energy interactions in 2.1.

$$E_I(\lambda) = E_R(\lambda) + E_A(\lambda) + E_T(\lambda) \quad (2.1)$$

Remote sensing systems record changes in the amount of energy reflected and emitted from objects on the earth's surface at selected wavelength ranges [Çölkesen, 2015]. The fact that many remote sensing systems operate in the wavelength regions in which reflected energy is predominant makes it crucial to know the reflectance properties of objects on earth. The spectral reflectance of the object is measured by the ratio of energy reflected by the object to the energy incident to the object, and it is expressed in percentage. Each object on the earth's surface displays different reflectance at different wavelengths, and the spectral reflectance curve shows this behavior. The graphical representation of the spectral reflectance of an object as a function of wavelength is called the spectral reflectance curve and gives insight into the spectral properties of the objects. Moreover, spectral reflectance curves of objects play a crucial role in determining the spectral region to be studied in remote sensing for a particular application and in selecting the sensor to be used [Sunar et al., 2011].

Identifying objects on the earth's surface in remotely sensed data is possible by knowing the spectral characteristics of objects. In this respect, objects on the remotely sensed images are distinguished using spectral libraries developed for this purpose or based on measurements made with spectroradiometer in the field. In such circumstances, it is of utmost significance to know the spectral properties of various vegetation, soil, water, and other ground surface properties of interest. Generalized spectral reflectance curves of vegetation, soil, and water are depicted in Figure 2.3, [Web 4, 2022]. In this graph, the x-axis and y-axis represent the various wavelengths (μm) of the electromagnetic spectrum and the amount of reflectance (%), respectively.

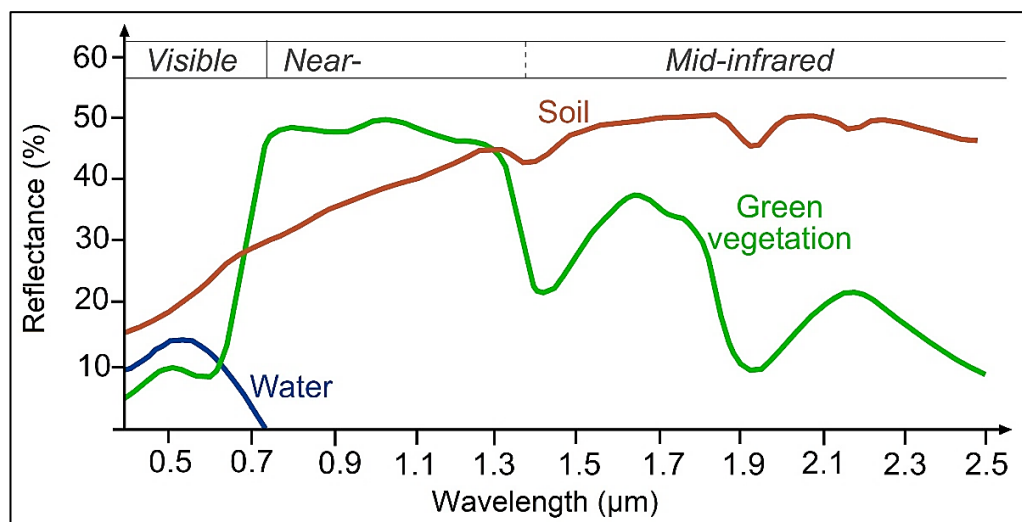


Figure 2.4: Typical spectral reflectance curves of vegetation, soil, and water.

The spectral reflectance of the vegetation varies depending on the wavelength. In the visible region (0.4–0.7 μm), pigments such as chlorophyll, xanthophyll, carotene, and anthocyanin in green plants absorb most of the incident energy. Higher absorption at blue and red wavelengths causes leaves to appear green. As the plant ages, chlorophyll, the most important pigment in the plant, loses its effectiveness. In such circumstances, the impact of other pigments increases, which leads to the leaves appearing yellow and red. More specifically, the red appearance of some trees in autumn is explained by the decrease in chlorophyll and the predominance of anthocyanin pigments in the tree structure. Most of the incident energy is reflected in the near-infrared region because the absorbed energy is reduced. At specific wavelengths (water absorption bands) in the mid-infrared region, energy reflectance is decreased because the water in the leaf absorbs energy. The visible, near-infrared, and mid-infrared regions give insight into the pigments, cell structure, and water content of plants, respectively [Sunar et al., 2011].

Many factors influence the spectral reflectance of the soil. Some of these are soil texture (proportion of sand, silt, and clay), moisture content, organic matter content, presence of iron oxide, and surface roughness. In general, the increase of moisture in the soil reduces its reflectance. This impact is especially dominant in water absorption bands. On the other hand, soil moisture content has a strong relationship with soil texture. Coarse, sandy soils have low moisture content and high reflectance because they drain well. In contrast, poorly drained, fine-textured soils demonstrate lower reflectance. In addition to the foregoing, surface roughness and organic matter content also decrease reflectance. Especially at visible wavelengths, the presence of iron oxide in the soil remarkably diminishes the reflectance of energy [Lillesand et al., 2015].

When the spectral reflectance of water is examined, it is obviously seen that it has high reflectance in the visible region (0.4–0.7 μm) and robust absorption at wavelengths greater than 0.7 (near-infrared and mid-infrared regions). Given these characteristics, water can be easily distinguishable from other objects in the near-infrared and mid-infrared regions. However, these phenomena are acceptable for deep and clean water sources. Different spectral reflectance is observed in water sources that are not very permeable and contain abundant chlorophyll sediment, etc. In addition, the spectral reflectance of water is influenced by its depth, materials, and surface roughness.

2.4. The Concept of Resolution in Remote Sensing

Although many remote sensing sensors sense data based on the basic principles of remote sensing, the format and quality of the outcoming image can vary greatly [Sunar et al., 2011]. The resolution of satellites explains these differences. In this context, there are four kinds of resolution in remote sensing: spatial, spectral, radiometric, and temporal. These resolutions are different for distinct sensors as they depend on the satellite orbit configuration and sensor design.

2.4.1. Spatial Resolution

The smallest distance between two neighboring objects that the sensor can record refers to the spatial resolution. In other words, it corresponds to the area on the earth's surface occupied by a pixel, which is the smallest element of the image. The sensor's spatial resolution depends on the platform altitude and the Instantaneous Field of View (IFOV) of the sensor, which determines the area on the earth's surface that is "seen" from a given altitude at one particular moment in time. The area occupied by a pixel on the earth's surface is determined by multiplying the IFOV by the altitude of the platform from the ground. Landsat-8 satellite images of Iceland Reykjavik with different spatial resolutions are depicted in Figure 2.5 [Web 5, 2022]. It is clear from the figure that as the spatial resolution increases, the distinguishability of objects increases, and smaller details can also be discriminated.



Figure 2.5: Landsat 8 images with different spatial resolutions.

2.4.2. Spectral Resolution

Objects and details on the earth's surface are distinguished by their spectral reflectance characteristics at distinct wavelengths. In this respect, spectral resolution is a significant feature that is considered in the selection of satellite data convenient for the project. Spectral resolution refers to the specific wavelength range that a sensor records in the electromagnetic spectrum and the number of bands. If the wavelength range that the detector records is narrow, it contains more spectral detail. Conversely, the widening of the wavelength range decreases the spectral resolution. Similarly, decreasing the number of bands has the impact of reducing the spectral resolution. According to the number of bands included by the sensor, there are multispectral and hyperspectral sensor systems. While multispectral sensors can detect in a few wavelength ranges with different spectral resolutions, hyperspectral sensors can detect in hundreds of very narrow spectral bands in the visible, near-infrared, and mid-infrared regions of the electromagnetic spectrum. Hyperspectral images have diverse application domains such as precision agriculture, classification of soil types and coastal management, and mapping of mineral sources.

2.4.3. Radiometric Resolution

The radiometric resolution of sensor is described as its capability to discriminate the slightest differences in energy. In other words, it is the amount of information that each pixel in the image contains. The sensitivity to detect smaller differences in reflected or emitted energy refers to the high radiometric resolution. These energy differences correspond to the number of shades of gray that can discriminate in the image. Radiometric resolution is measured in bits, each recording an exponent of power 2. For instance, if a sensor uses 8 bits to record data, it has 256 (2^8) digital values ranging from 0 to 255. However, if it uses 4 bits, it takes 16 (2^4) values ranging from 0 to 15, and has less radiometric resolution. The higher the radiometric resolution, the more value there is to store information, making it more sensitive to the slightest differences in energy. When the images with different radiometric resolutions of the same region are compared in Figure 2.6, it is observed that the level of detail discrimination increases as the resolution increases from 2 to 8 bits [Web 5, 2022].

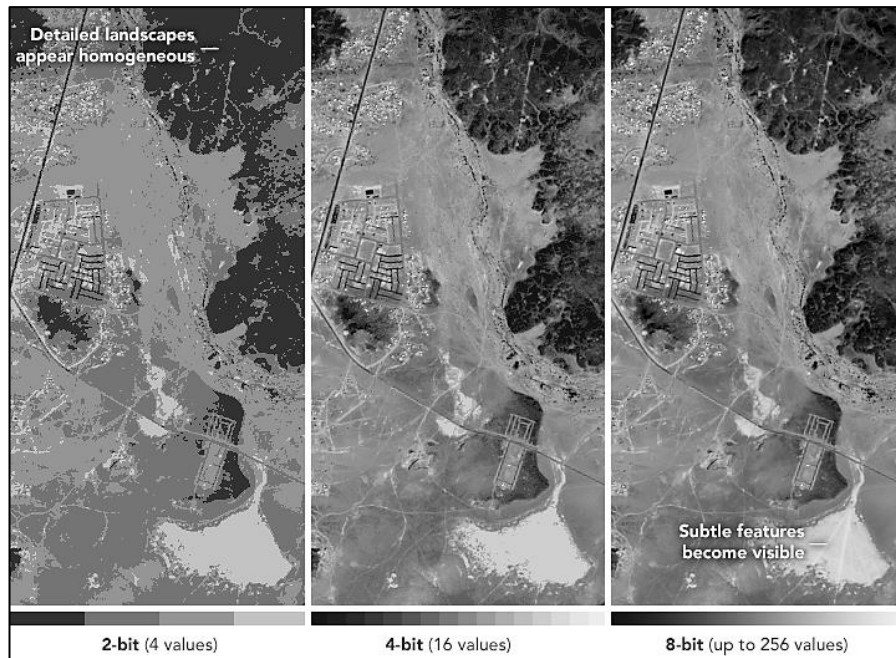


Figure 2.6: Images with different radiometric resolutions.

2.4.4. Temporal Resolution

The temporal resolution refers to how often a satellite sensor collects data for the same geographical area. That is, it is the time elapsed between consecutive observations of the same area taken by the sensor. The shorter this period, the higher the temporal resolution of the sensor. Temporal resolution can vary depending on the sensor's characteristics, the orbital characteristics of the satellite, and the swath width. For instance, the LANDSAT 8 satellite captures the image of a given region on the earth's surface every 16 days, while the NOAA/AVHRR sensor captures it every 12 hours. To put it another way, the temporal resolution of the NOAA satellite is quite higher than LANDSAT 8. Since temporal resolution specifies the images that a sensor captures sequentially, it is of paramount importance in selecting sensors to be utilized in remote sensing applications, particularly change detection using multi-temporal image sets. For example, multi-temporal images are utilized to monitor plant change during growth, monitor mucilage bloom in the sea, and temporal analysis of forest fires.

3. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are computational models that emerged in parallel with cognitive and computer science research developments. These networks have recently attracted increasing attention from researchers due to their remarkable success in various engineering and scientific fields such as image processing, signal processing, and pattern recognition. Besides, the fact that new generation high-performance computers can meet parallel and enormous computational power requirements in the implementation of ANNs has also increased the popularity of ANNs. They are heuristic algorithms that are extensively applied today to handle problems in diverse fields, including medical, security, finance, transportation, speech, agriculture, geography, and management [Kavzoglu, 2001].

Artificial neural networks adopt an alternate computational strategy than that of conventional methods. Neural computing is an information processing paradigm performed by networks of neurons working to solve specific problems. The core principle of ANNs emanates from the desire to create computer systems that can automatically perform the capabilities such as deriving, generating, and discovering new knowledge through learning which is one of the characteristics of the human brain, without any aid [Öztemel, 2012], [Kavzoglu, 2001]. They perform this learning process from experience through samples like humans. Generally, an ANN system imitates the behavior of the network of neurons in the human brain [Dave and Dutta, 2014]. These networks are composed of a vast amount of processing elements (neurons) that are highly interconnected by weighted links. These elements work together in distributed way to serve the aims of learning from the input information of the network, coordinating the internal processing, and optimizing the final output of the network [Jiang et al., 2010]. The unique structural representation of ANNs is based on these processing elements being connected in such a way that each input is locally processed among neighboring elements. The robustness of ANNs underlies these structural representations of them.

ANNs have many advantages such as adapting to their environment, making decisions under uncertainties, non-linearity, parallelism, working with missing data, learning, generalization ability, real-time adaptability, multidirectional execution, using an unlimited number of variables and parameters, and tolerance for faults

[Öztemel, 2012], [Kononenko and Kukar, 2007]. However, in addition to these superiorities, they have a black box nature because they are not straightforwardly interpretable by humans. That is, it cannot be explained how they behave in learning about some specific issues and drawing conclusions from trained networks. Nevertheless, these powerful advantages have enabled ANNs to be successfully implemented in numerous remote sensing tasks lately, including image classification, object detection, change detection, image compression, forecasting, spectral unmixing, and data fusion. In addition to the foregoing, ANNs are also applied to inversion problems such as the inversion of radar scattering from vegetation canopies, inversion of microwave brightness temperatures, and inversion of snow parameters from passive microwave remote sensing measurements [Kavzoglu, 2009].

Up to date, many artificial neural network models and learning algorithms have been introduced by adopting different interconnection strategies for the processing elements to enhance the performance of the network. In this context, artificial neural network models can be classified according to two criteria. These are the structure of the network and the learning algorithm utilized. Regarding their structure, ANNs appear as feed-forward and feed-back neural networks according to the direction of the learning algorithm associated with the network model. In a feed-forward network, information is transmitted from the input layer to the output layer by unidirectional connections. Conversely, in a feed-back network information is transmitted from the output layer to the input layer. On the other hand, there are three types ANNs depending on whether the model utilizes a supervised, unsupervised, or reinforcement learning strategy. In supervised models, output data is also provided for the input data given to the network, while in unsupervised models only input data is provided. Also, in the reinforcement learning algorithm, the output data of the input data is not provided, and a critic is used to evaluate the suitability of the obtained output corresponding to the given input. Examples of supervised neural network models include the Delta rule, Perceptron, and Multilayer Perceptron (MLP), while unsupervised models include the Adaptive Resonance Theory Networks (ART), Kohonen Self Organizing Map (SOM), and Grossberg networks. In addition, the Genetic Algorithm (GA) and Boltzmann Machine are given as examples of reinforcement learning algorithms.

3.1. History of Artificial Neural Networks

For many years, there has been an increasing interest in examining the structure of the human brain and imitating its ability to think. With this increasing research attention, new computational models or artificial neural networks based on biological background have been introduced to sort out many complex problems. The first milestone in developing artificial nervous systems was presented in 1943. Although it is known that some scientists worked on artificial neural networks in the years before 1940, it cannot be said that these researches have engineering value [Zurada, 1992], [Öztemel, 2006].

In 1943, Warren McCulloch and Walter Pitts proposed the first formal model of a basic computing neuron. This model could function as an arithmetic logic computing element because it contains all the essential elements to perform logic operations. Nonetheless, this neuron model paved the way for future developments in ANNs. Donald Hebb, a psychologist, presented a learning approach to update neuron connections in 1949. In this approach, it has been postulated that information can be stored in connections between neurons [Hebb, 1949]. This technique, now referred to as the Hebbian learning rule, made vital contributions to the theory of artificial neural networks. Marvin Minsky and Dean Edmonds built the stochastic neural analog reinforcement calculator in 1951, known as the first artificial neural network in history. Afterward, Frank Rosenblatt produced a neuron-like element in 1958, which he named the Perceptron, inspired by the mathematical side of Hebb's ideas. The Perceptron model adopts the principle that a neuron takes more than one input and produces an output. The output of the network is obtained by comparing the weighted sum of the input values with a particular threshold value [Rosenblatt, 1958].

In the early 1960s, a new learning rule called the Widrow-Hoff learning rule was introduced with the ADaptive Linear Neuron (ADALINE) network by Widrow and his graduate student Marcian Hoff. Although the ADALINE neuron model is structurally similar to the perceptron, the perceptron uses the threshold function as the activation function, while ADALINE uses the linear function. Both models can only solve linearly separable problems. The Widrow-Hoff rule is more powerful than the learning rule of the perceptron and minimizes the mean square error during training. In addition, Widrow and Hoff developed the MADALINE (for Many ADALINES)

neural network model, which includes more than one adaptive element, and this model has formed the basis for beneficial future studies. In the late 1960s, Marvin Minsky and Seymour Papert demonstrated that perceptrons cannot solve very straightforward nonlinear problems such as the XOR (Exclusive OR) problem in their book called “Perceptron” [Minsky and Papert, 1969]. After this claim, progress in artificial neural network studies dramatically stopped. However, by the 1980s, these limitations of perceptrons were overcome with the development of the multilayer perceptron model.

Although the studies on ANNs stopped in 1969 and even the necessary financial support was withdrawn, some scientists continued to work in this domain. In 1972, neuropsychologist James A. Anderson and electrical engineer Tuevo Kohonen published similar studies about associative memory independently of one another. These studies became the basis of unsupervised learning rules to be developed later. Kunihiko Fukushima [1980] developed the neocognitron model, which is capable of robust visual pattern recognition through learning and is related to biological plausibility. Tuevo Kohonen [1982] proposed the Self-Organizing Map (SOM) using unsupervised learning, and it has become the most popular neural network architecture. In the same year, Hopfield demonstrated that neural networks can be generalized, and solutions can be produced for problems that are arduous to sort out with prevalent computer software [Hopfield, 1982], [Hopfield, 1984]. He confirmed his claim by solving the traveling salesperson problem. After these studies, interest in artificial neural networks began to be re-established, and consequently, Hinton and Sejnowski [1983] developed the Boltzmann machine. In 1986, David Rumelhart and James McClelland invented the Backpropagation Algorithm, which uses a new learning algorithm in feed-forward models and showed that the previously claimed limitations in this regard could be overcome. Because the XOR problem, which the single-layer perceptron could not solve, has been solved with multi-layer perceptrons. With the development of this algorithm, a breakthrough was made in the field of ANNs, and they are still one of the most utilized training algorithms [Zurada, 1992]. These developments have led to increased interest in ANNs. Then, Broomhead and Lowe introduced the Radial basis functions model in 1988 as an alternative to multilayer perceptrons. With this model, much higher success has been achieved in filtering processes. Specht has developed more improved models of these networks, namely the General Regression Neural Network (GRNN) [Specht, 1991] and Probabilistic Neural Network (PNN) [Specht, 1988].

3.2. Basic Structure and Components of Artificial Neural Networks

Artificial neural networks are systems that mimic the biological nervous structure and working principle of the human brain. In this context, it is necessary to examine the biological nervous system to understand artificial neural networks. The biological nervous system is a three-layered system in which the brain (central neural network) continuously interprets signals (information) from sensory organs and other nerves and produces convenient decisions. This system consists of the brain and the receptor and effector nerves under the control of the brain. In the central neural network, appropriate signals are produced by continuously making feed-forward and feed-back between the receptor nerves that convert the signals from the environment into electrical signals and the effector nerves that convert the signals produced by the brain into appropriate responses. The fundamental processing element of the biological nervous system is special nerve cells called neurons, which perform their functions by connecting with each other. As seen in Figure 3.1, a basic biological nerve cell consists of the cell body or soma, dendrite, axon, and synapse [Web 6, 2022]. The cell body has a nucleus that includes information about hereditary characteristics. A neuron receives signals from other neurons through its dendrites, and transmits the signals produced by the cell body along the axon. These signals eventually branch into strands and sub-strands. The synapses at the ends of these strands are the elementary functional unit that provides signal transmission between two neurons. Signal exchange between two neurons is provided by neurotransmitters with synaptic connections [Jain et al., 1996].

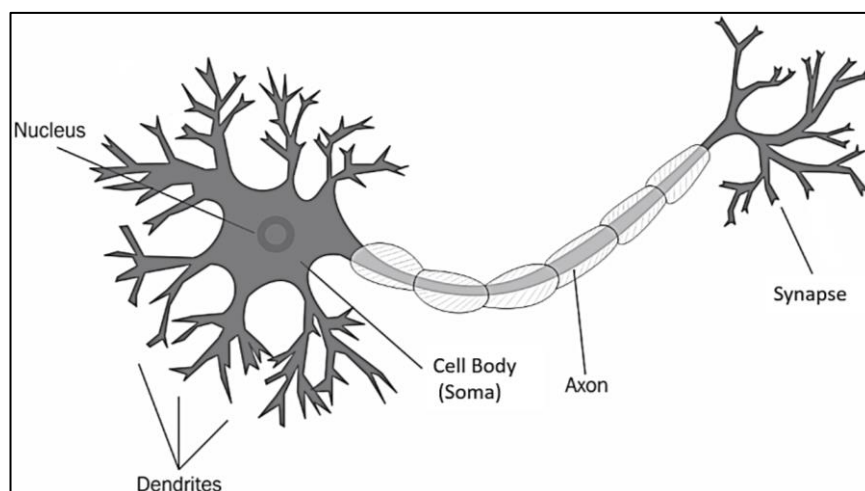


Figure 3.1: Structure of a biological neuron.

As in biological neural networks, the fundamental element of artificial neural networks is artificial neurons, also known as processing elements. ANNs are formed by creating connections between these artificial neurons and grouping them into layers. The artificial neuron consists of components similar to the basic components of biological neurons. In this context, as can be seen in Figure 3.2, an artificial neuron has five components: input, weight, summation function, activation function, and output neuron [Web 7, 2022].

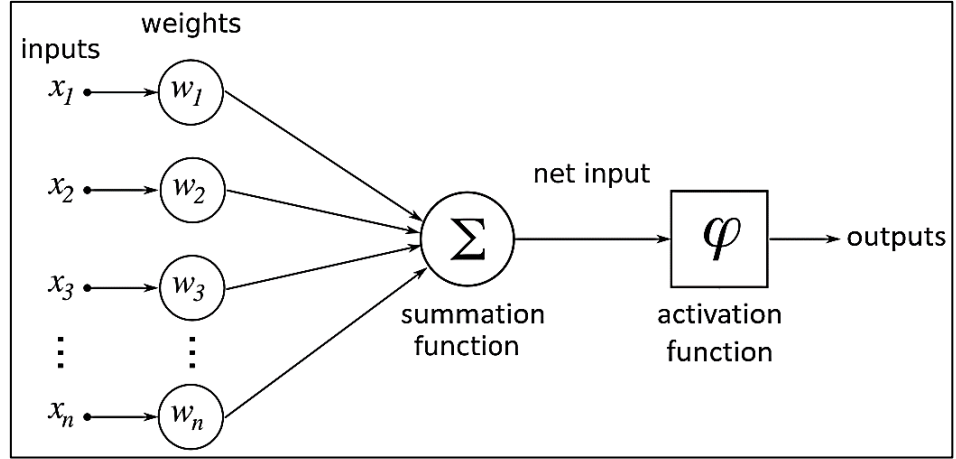


Figure 3.2: Structure of an artificial neuron.

Here, $x_1, x_2, \dots, x_n$ are signals (information) coming to the artificial neuron from the external world, other cells, or itself. The weights are represented as $w_1, w_2, \dots, w_n$ indicate the significance of the information coming to the artificial neuron and its impact on the cell. The sum function calculates the net input value coming to this artificial neuron. This value is commonly calculated as a weighted sum based on the sum of each input multiplied by its own weight value. This process is expressed as in equation 3.1 where X_i , W_i , and n represent the inputs, weights, and the total number of inputs coming to the neuron, respectively. However, different summation functions can also be used according to the implemented network structure [Öztemel, 2012].

$$NET = \sum_{i=1}^n X_i W_i \quad (3.1)$$

The activation function determines the output that the neuron will produce in response to this input by passing the net input to the neuron via diverse functions. As

with the summation function, different formulations can be used as activation function. In such circumstances, the optimal function for the studied problem is determined by the practitioner through trial and error. For instance, nowadays, the Sigmoid activation function is generally utilized for the multilayer perceptron model.

Artificial neural networks are formed by connecting the aforementioned artificial neurons in various ways and are usually organized in layers. However, the coming together of neurons is not random, and they generally come together in three layers and in parallel within each layer. These layers that construct the neural network are the input layer, hidden layer, and output layer. In the input layer, neurons receive information from the external world and transmit it to the hidden layers. Information coming to the hidden layer is processed and sent to the output layer. These hidden layers can be one or more than one. The neurons in the output layer process the information coming from the hidden layer and produce the output value for the input set. The structure of a typical three-layer artificial neural network is given in Figure 3.3 [Wang et al., 2010].

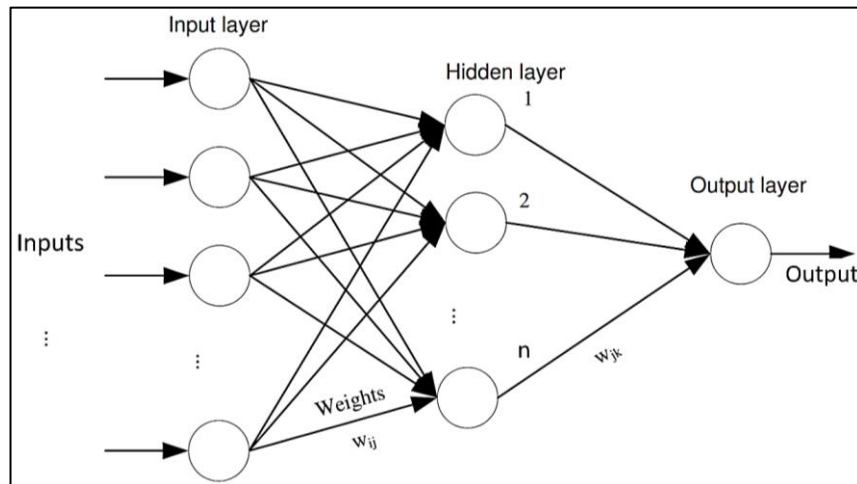


Figure 3.3: A simple three layer neural network structure.

The correct output of the artificial neural network is ensured by the proper determination of the weight values of the neuron's connections. This phenomenon is called training of the neural network, and the aim of the training of the network is to find out the optimal weight values that will produce the correct output for the input set of the network. When the neural network reaches the proper weight values, it has the capability to generalize about the events represented by the input set, that is, the network learns [Sudheer et al., 2002], [Öztemel, 2012].

3.3. Types of Artificial Neural Networks

Artificial neural networks are structures made up of interconnected artificial neurons. Although ANNs have similar working principles, how the connections between artificial neurons are arranged and the nature of the connections determine the structure of the network. In other words, while creating an artificial neural network architecture, the arrangement of artificial neurons and the determination of the weights of the connections between these neurons are of utmost significance for the characteristic of the network. The learning algorithm determines how the connections are adjusted so that the desired overall behavior of the network can be achieved. Within this framework, artificial neural networks are classified according to their structure and the learning algorithms they utilize [Çölkesen, 2009].

3.3.1. Artificial Neural Networks According to Their Structure

Artificial neural network architectures can fall into two categories in terms of their structure or the connections between artificial neurons: feed-forward networks and feed-back or recurrent networks.

3.3.1.1. Feed-forward Networks

In a feed-forward artificial neural network, the neurons are usually grouped into layers. Information is transmitted from the input layer to the output layer by one-way connections. That is, the network has one-way connections having no loop backward. With this nature, feed-forward networks perform a static function [Alaloul and Qureshi, 2020]. The feed-forward network architecture is illustrated in Figure 3.4 (a). While neurons make connections from one layer to the next layer, they do not have connections within the same layer. In this network, outputs from one layer are given as inputs to the next layer. There is no information processing in the input layer, and the information is transmitted to the hidden layer. The neurons in the hidden layer receive the information from the neurons in the input layer with the effect of the connection weights and sent them to the output layer in processed form. Similarly, at the output layer, information is processed, and the output of the network is determined.

Examples of feed-forward networks include the multi-layer perception (MLP) [Rumelhart and McClelland, 1986] and the learning vector quantization (LVQ) network [Kohonen, 1989].

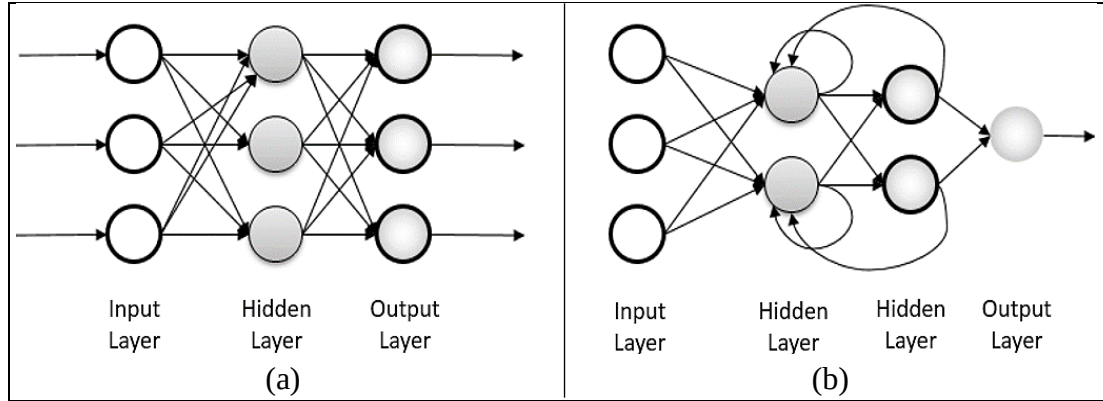


Figure 3.4: Artificial neural network architectures in terms of their structure. (a) Feed-forward network, (b) Feed-back network [Pekel and Soner Kara, 2017].

3.3.1.2. Feed-back Networks

Contrary to feed-forward networks, in feed-back networks, information is transmitted both forward and backward. That is, there is a two-way flow of information. With this characteristic, these networks have a dynamic memory structure, unlike feed-forward networks. In these networks, the outputs of neurons in the output layer or hidden layers can be given as input to the input layer or previous hidden layers, to any neuron in its own layer, or to the neuron itself (Figure 3.4 (b)). That is, the outputs at a given moment in these networks reflect current input values as well as previous input and output values [Pham et al., 2007]. Due to this nature, feed-back networks have a non-linear structure. Feed-back networks with different behavior and structure can be obtained according to the type of connections that give the feed-back property to these networks. In backward calculation logic, the output value produced by the network is compared with the expected outputs of the network and the difference is determined. This difference is considered an error. By distributing the error to the weight values of the network, it is aimed to reduce the error in the next iteration. These networks are used in forecasting applications such as forecasting diverse types of time series. Hopfield [Hopfield, 1982], Elman [Elman, 1990], and Jordan [Jordan, 1986] networks are examples of feed-back networks.

3.3.2. Artificial Neural Networks According to Learning Algorithms

One of the most distinctive characteristics of artificial neural networks is that they have the ability to learn. In the training process of artificial neural networks, determining the proper weight values by changing the weight values of the connections between the artificial neurons is expressed as the learning of the network. This learning process is performed according to the determined learning strategy. From this perspective, artificial neural networks are also classified according to the learning algorithms utilized in the training of the networks. In general, there are three basic learning paradigms: supervised learning, unsupervised learning, and reinforcement learning. These algorithms can be utilized by any type of artificial neural network architecture.

3.3.2.1. Supervised Learning

In supervised learning algorithm, a training data is presented to the network as input data. This training data includes the input examples and the target outputs corresponding to these input examples. The network updates the weights of the connections between neurons to produce the desired outputs corresponding to these inputs. In other words, in the supervised learning algorithm, the connection weights between artificial neurons are updated according to the difference between the actual output values in the training data and the output values obtained in the output layer of the network [Çölkesen, 2009]. For this, the difference between all outputs obtained from the network and the actual output values is calculated, and the margin of error for each neuron is adjusted. Thus, these neurons also update their weights. In this respect, supervised learning requires a teacher or supervisor to ensure desired or target output values [Pham et al., 2007]. An essential aspect of supervised learning is the construction of training data. It should consist of data examples that satisfactorily represent the problem space under investigation. Otherwise, a reliable neural network model with desired results cannot be obtained with a non-representational training data. Examples of supervised learning algorithms include the Delta rule [Widrow and Hoff, 1960], generalized Delta rule, and LVQ algorithm.

3.3.2.2. Unsupervised Learning

An unsupervised learning algorithm, also known as self-organization learning, does not adopt a targeted output approach. In this learning algorithm, only input examples are presented to the network during training. That is, unlike supervised learning, target outputs corresponding to these inputs are not given to the network. In this context, unsupervised learning does not require any teacher or supervisor. The network is expected to learn the relationships between parameters in these input examples by itself. In other words, the neural network recognizes patterns and differences in input data provided without any external support [Alaloul and Qureshi, 2020]. The network automatically adjusts the neuron connection weights so that it can cluster the input patterns into categories with similar characteristics. After the learning process of the neural network is finished, labeling by the user is required to indicate what the outputs of the network mean [Öztemel, 2006]. ART [Carpenter and Grossberg, 1988] and SOM [Kohonen, 1982] are artificial neural networks that utilize unsupervised learning algorithm.

3.3.2.3. Reinforcement Learning

In reinforcement learning algorithm, as in unsupervised learning, the target output values of the input data are not given to the network, only the input values. Instead, the network is expected to produce its own outputs in response to these input values. A critic is utilized to evaluate the goodness of these outputs. The training process continues according to the information given by this critic, indicating whether the output is good or bad. The most active and well-performing neuron connections for the input are strengthened by consecutive iterations [Pham et al., 2007]. The reinforcement learning algorithm is especially convenient for problems such as long-term versus short-term reward trade-off, and has been successfully employed in diversity of problems, including games such as chess, telecommunications, and robot control [Krenker et al., 2011]. Genetic algorithm [Holland, 1975], [Goldberg, 1989] and Boltzmann machine [Hinton and Sejnowski, 1983] are examples of the reinforcement learning algorithm.

3.4. Application Areas of Artificial Neural Networks

Nowadays, the theory of artificial neural networks is used in many research areas including medicine, economics, finance, biology, and industry. Moreover, ANNs have achieved remarkable success in various applications such as pattern recognition, forecasting, medical diagnosis, and signal processing, due to their being more robust and outperforming other computational methods in solving various problems. In this context, the application of ANNs in some fields is summarized below.

In recent years, ANNs have been widely utilized in pattern recognition, and are gradually replacing the traditional pattern recognition methods. ANNs mimic the behavior of the human brain, allowing computer programs to recognize patterns and solve common problems. In pattern recognition with ANNs, the pattern term mentioned may be a personality's face, a handwritten word, a fingerprint image, or a speech signal. Nazeer et al. [2007] presented a face recognition system using ANNs. They evaluated the performance of their system by applying two photometric normalization techniques and found that their system gave promising results for face recognition and face verification. In another study, Electrocardiogram (ECG) pattern recognition was implemented using ANNs. In the study, three types of ANN models, SOM, Backpropagation (BP), and LVQ networks, were trained separately. Afterward, these models were tested for ECG pattern recognition, and it was noted that the SOM network performed the best [He et al., 2006]. An ANN approach was used for optical character recognition in a study conducted by Mani and Srinivasan [1997]. For this, a simple feed-forward ANN model was trained with distinct set of noisy data, and the backpropagation technique was utilized for learning in the network. Experimental results revealed that the recognition rate was 70% for noisy data to up to 99%. Han et al. [1992] proposed a learning pattern recognition system using ANN, which could diagnose faults and aging conditions for an induction motor. In this system, multilayer perceptron model with error backpropagation learning algorithm was utilized for pattern recognition. In another study, a pattern recognition approach using ANN for an Interactive Voice Response was introduced [Shah et al., 2009].

The recent popularity of artificial neural network-based forecasting methods has led to an enormous increase in research activity in academia. The core principle underlying this popularity is that ANNs are data-driven and self-adaptive unlike

traditional model-based methods and can capture subtle functional relationships among the data [Zhang et al., 1998]. In this way, ANNs have been successfully applied in many forecasting applications, including time series forecasting, crop yield forecasting, load forecasting, stock market prediction, and inflation forecasting. Oancea and Ciucu [2014] utilized ANNs for financial time series analysis. They employed different feed-forward and recurrent neural networks and training algorithms to predict the EUR/RON and USD/RON exchange rates using data series with daily exchange rates from 2005 to 2013. Consequently, they reported that the recurrent network is more successful than the classical feed-forward network. In the field of crop yield forecasting, Dahikar and Rode [2014] used feed-forward backpropagation ANN for the prediction of crop yields in rural districts. Moghaddam et al. [2016] employed feed-forward ANN models trained by backpropagation algorithm to forecast the daily NASDAQ stock exchange rate. In another study, the short-term forecast about energy demand of the Benin Electric Community was implemented using Multilayer Perceptron (MLP) and Radial Basis Function (RBF) ANN approaches [Salami et al., 2018].

Artificial neural networks are finding more and more use in medical diagnosis applications recently. ANNs have the capability to handle various types of medical data and integrate them into categorized outputs. In this way, ANNs have been applied in the diagnosis of colon cancer, colorectal cancer, pancreatic disease, early diabetes, and multiple sclerosis lesions. Yılmaz and Bozkurt [2012] employed feed-forward networks, cascade-forward networks, time delay networks, distributed delay networks, probabilistic neural network, and learning vector quantification for the diagnosis of iron deficiency anemia in women. For this, they handled the blood test results of women over the age of 18. As a result of the study, it has been shown that medical data can be classified with ANN approaches. Abbass [2002] presented an evolutionary artificial neural network approach for breast cancer diagnosis. In the study, it was revealed that the approach called memetic pareto artificial neural network (MPANN) had better generalization and lower computational cost compared to standard backpropagation. In another study, a multilayer neural network structure trained with the Levenberg-Marquardt algorithm, and a probabilistic neural network structure were utilized for diabetes disease diagnosis. Qualitative analysis on the Pima-diabetes disease dataset has shown that ANN structures can be used successfully for this task [Temurtas et al., 2009].

3.5. Use of Artificial Neural Networks in Remote Sensing

Artificial neural networks hold a significant place in the remote sensing field and have been widely applied to the analysis of remotely sensed data with high-quality results. The rapid adoption of ANN approaches in remote sensing research stems from some of the unique properties of ANNs [Mas and Flores, 2008]. They have the capability to learn complex patterns and to consider complex nonlinear relationships between explanatory and dependent variables in this learning process [Lek and Guégan, 1999]. ANNs are tolerant to noise present in the data, and thus can produce robust results even with missing or imprecise data [Hewitson and Crane, 1994], [Kavzoglu, 2001]. Furthermore, since they have no underlying assumptions about the dataset, they permit incorporating and analyzing diverse types of data and input structures [Kavzoglu, 2001]. To be more specific, ANNs can incorporate data from different sensors, ancillary data, and categorical data [Foody, 1995a]. In addition to the foregoing, ANNs can incorporate *a priori* knowledge and constraints about the process into the analysis [Foody, 1995b]. Considering these characteristics, ANNs play a crucial role in the analysis of remotely sensed data, and are the subject of many research in the literature [Benediktsson and Sveinsson, 2003], [Tatem et al., 2001], [Frizzelle and Moody, 2001]. The tasks implemented in these researches are generally land use and land cover (LULC) classification, object detection, change detection, image compression, georeferencing, unmixing, forecasting, and data fusion. In this context, three of these tasks are examined in detail in the following subsections.

3.5.1. Land Use/Land Cover Classification

In the remote sensing area, one of the most popular applications of the ANNs is the classification of LULC information. Classification of land cover is a standard task in the remote sensing field, where each satellite image pixel is assigned a class label indicating the physical material of the object surface, while land use defines the socio-economic function of a piece of land [Yang et al., 2018a]. Although several classification methods have been developed to fulfill these tasks, the most popular are classification approaches based on ANNs. The popularity of ANN approaches is due to their unique advantages over conventional statistical classification methods

[Kavzoglu and Vieira, 1998]. The most evident advantage of ANNs may be that they have a non-parametric structure and require little or no *a priori* knowledge of the statistical distribution of the data [Kavzoglu and Mather, 2001]. Due to these superiorities, the use of ANNs in remotely sensed image classification applications, especially LULC classification, has spread and led to an increasing number of studies.

Yuan et al. [2009] developed an automated ANN classification system consisting of unsupervised SOM and supervised MLP neural network modules. BP training algorithm was utilized for the MLP, and the standard SOM, refined SOM algorithms were utilized for the SOM module. The LULC classification capability of the system was tested on the Landsat TM image, and it was reported that the system successfully accomplished network training pattern creation, network training and network generalization tasks. In another study, Kohonen Self-organizing Feature Map (KSOFM) network was presented for LULC classification. The algorithm was tested on the Landsat TM scene and compared with the maximum likelihood method and the BP network. It was observed that while the KSOFM network outperformed the maximum likelihood method in terms of classification accuracy, there was no significant difference with the BP network [Ji, 2000]. Zhou and Yang [2008] applied MLP networks trained by the BP algorithm with different combinations of parameters, including activation function, learning rate, iteration number, momentum to classify Landsat ETM+ image. The impacts of different parameter settings on classification performance were evaluated. Nagarkar and Rege [2019] designed an optimized ANN for the classification of multispectral images. The Taguchi method was utilized to choose the hyperparameters of the ANN model and optimize the model. AWiFS sensor images of Resourcessat-2 satellite were classified into diverse LULC classes with the proposed model. The results were compared with the minimum distance and maximum likelihood algorithms, and it was mentioned that the proposed model demonstrated higher classification performance in terms of accuracy and Kappa metrics.

3.5.2. Object Recognition

The application of artificial neural networks for object recognition from remotely sensed images is fast growing research field in the literature. Object recognition from remotely sensed images is a problem that requires the determination

of the classes of objects and their locations on the image [Yildirim and Kavzoglu, 2022]. In particular, high-resolution images obtained from the increasing number of sensors make it possible to distinguish even many small artificial objects. In this context, ANNs are a high precision tool capable of recognizing complex patterns that are difficult to analyze with other traditional approaches. In the literature, ANN approaches such as Hopfield neural networks, MLP, convolutional neural networks, and extreme learning machines have been employed to detect various objects including ships, trees, roads, and vehicles from remotely sensed images [Cheng and Han, 2016].

In a study carried out by Goswami et al. [2014], a framework based on ANN was presented to automatically extract water bodies from satellite data. For this, they trained MLP using the error backpropagation learning algorithm. As a result of the accuracy evaluation made according to the confusion matrix and Kappa coefficient, it was observed that the MLP model produced successful results in the automatic extraction of the water bodies. In addition, it was concluded that the accuracy of the model heavily depended on the quality of the training data. Li et al. [2001] employed ANN and threshold approaches to detect smoke from forest fires. Advanced Very High-Resolution Radiometer (AVHRR) images aboard NOAA satellites were utilized for the implementation. A multithreshold algorithm was developed due to the time-consuming application of the ANN approach over large areas. These two techniques were used separately or in combination, depending on the size of the image or the smoke conditions. In another study, the usability of ANNs for road detection from high-resolution satellite images was investigated on some of the RGB Ikonos and QuickBird images. For this purpose, backpropagation neural network with different hidden layer sizes and different iteration times was implemented to determine the best network structure and termination condition. In addition, it was tried to find out the optimum input vector by verifying the impacts of different input parameters on the network's capability [Mokhtarzade and Zoej, 2007]. A vector-guided framework for vehicle detection from 1-m panchromatic IKONOS high-resolution satellite images was introduced in a study conducted by Jin and Davis [2007]. A morphological shared-weight neural network (MSNN) was utilized to learn an implicit vehicle model in the study. Experimental results revealed that MSNN had very outstanding generalization ability and demonstrated high detection performance with very few false alarms. It was also observed that the model performed better than a baseline blob detection algorithm.

3.5.3. Pixel Unmixing

While coarser resolution satellite data is generally easier and less expensive to obtain and process, their spatial characteristics limit accurate land cover mapping at the regional or global scale. Land covers, which characteristically occur in relatively small patches or stripes in such an image, make supervised thematic classification arduous. In this case, mixed pixels are assigned to the most similar class, and the classified image no longer contains most of the original spectral information. In such circumstance, it is beneficial to unmix mixed pixel signatures to identify the land cover classes available at the subpixel level. Many methods have been proposed for this, such as linear regression-based techniques. However, since they are based on the assumption that mixing is linear, the ANN approach has become a more effective method for this task [Bastin, 1997], [Mas and Flores, 2008].

In the study conducted by Licciardi and Del Frate [2011], an ANN-based approach was employed to solve the unmixing problem in hyperspectral data. For this purpose, ANN architecture was utilized to perform both the dimensionality reduction and the final unmixing processes. The performance of the developed methodology was evaluated on the CHRIS-PROBA satellite images, the AVIRIS imagery, and the images obtained by the INTA-AHS instrument. As a result of the comparison with the appropriate benchmark techniques, it was pointed out that the proposed approach had evident advantages for both tasks. Fernandes et al. [2004] investigated pixel unmixing of a coarse SPOT-VGT image using a classical backpropagated MLP and four conventional algorithms. They implemented the algorithms using both proximate (<100 km) and distant (>400 km) separation between training and validation regions. It was found that the ANN model performed best in mapping fractional cover and continuous areas when the training and validation datasets were within ~ 100 km. However, when these datasets were well separated (>400 km), ANN performance deteriorated more. Foody et al. [1997] presented an ANN approach for unmixing the composition of image pixels. As a result of the case study conducted to provide accurate estimations of sub-pixel land cover composition, it was reported that the approach could accurately estimate the class composition of image pixels. Besides, the results showed the potential of ANNs to obtain accurate information from remotely sensed data containing dense mixed pixels.

4. DEEP LEARNING

Deep learning (DL) is part of machine learning methods that is a subfield of artificial intelligence and is emerging as a novel and rapidly growing field of study in big data analysis (Figure 4.1). Artificial intelligence consists of techniques aimed at developing computer systems capable of solving problems that can typically be solved with human intelligence and discernment. Machine learning includes algorithms that allow computers to learn by themselves without being explicitly programmed. Deep learning algorithms differ from existing machine learning methods in that they require a very large quantity of data, and that they also need hardware with very high-performance computing capable of processing this data with their complex architecture. Especially with the developments in computer and graphics processing units (GPUs), deep learning algorithms have attracted increasing attention from researchers and academicians in recent years. In this respect, deep learning approaches have achieved outstanding performance in many applications, including object recognition [Ribeiro et al., 2018], image processing [Ciregan et al., 2012], natural language processing [Socher, 2014], character recognition [Acharya et al., 2015], and speech recognition [Noda et al., 2015]. Besides, big technology companies with high data and computing power such as Google, Microsoft, and Facebook also utilize deep learning methods and integrate them into their products.

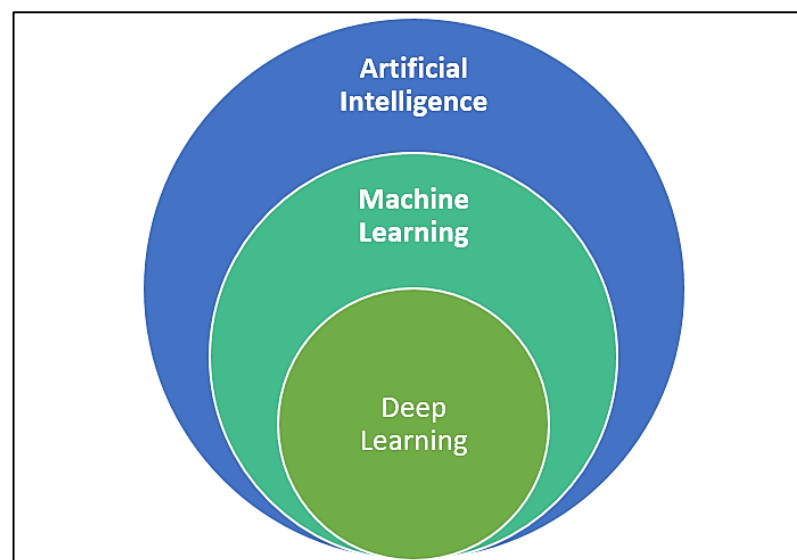


Figure 4.1: Relationships between artificial intelligence, machine learning, and deep learning.

4.1. A Brief History of Deep Learning

Historically, the deep learning paradigm relies on the concept of artificial neural networks and computing models that emulate the human perception system. In this regard, the history of deep learning dates back to the first formal model of a basic computing neuron proposed by Warren McCulloch and Walter Pitts in 1943. Afterward, Frank Rosenblatt produced the perceptron model, which adopts the principle that a neuron takes more than one input and produces an output [Rosenblatt, 1958]. In 1965, mathematician Alexey Ivakhnenko and Valentin Grigor'evich Lapa introduced the first general, workable learning algorithm for supervised deep feed-forward multilayer neural networks (perceptron). The architecture of this network is illustrated in Figure 4.2 [Web 8, 2022]. In this deep neural network, the best features in each layer are selected using statistical techniques, and then these are transferred to the next layer. Additionally, Ivakhnenko described an eight-layer deep neural network trained by the "group method of data handling" [Ivakhnenko, 1971].

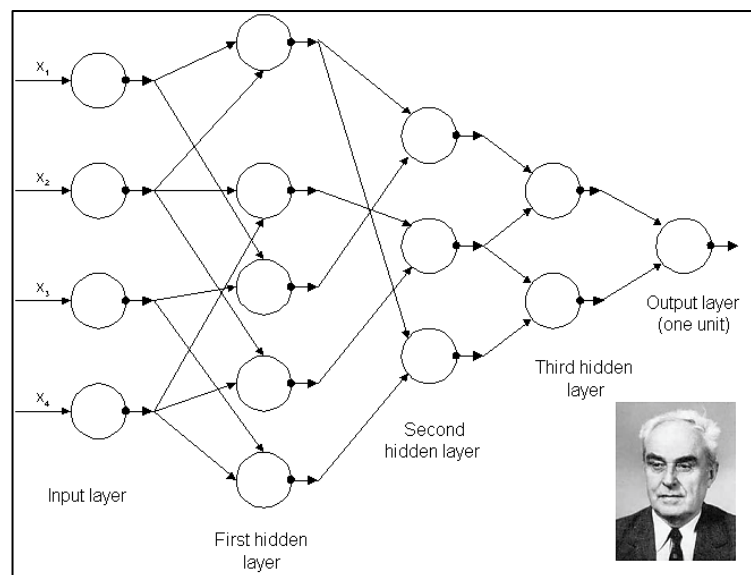


Figure 4.2: The architecture of the first known deep neural network.

In the 1980s, Kunihiko Fukushima proposed Neocognitron, the first convolutional neural network architecture used for handwriting character recognition and other pattern recognition tasks. LeCun et al. combined convolutional neural networks with backpropagation theories to recognize handwritten digits (MNIST) [LeCun et al., 1989]. In 1995, Corinna Cortes and Vladimir Vapnik developed support

vector machines to classify two groups with similar data. Long Short-Term Memory (LSTM) for recurrent neural networks, proposed by Hochreiter and Schmidhuber, aims at overcoming long-term dependency problems [Hochreiter and Schmidhuber, 1997]. In the 2000s, the concept of deep learning began to gain popularity with the study of Hinton and Salakhutdinov showing how a many-layered artificial neural network could be pre-trained one layer at a time. Fei Fei Li launched ImageNet in 2009, a massive and free database of more than 14 million labeled images that can be used to train neural networks with supervised learning [Deng et al., 2009]. In 2012, AlexNet, the convolutional neural network model designed by Alex Krizhevsky, won ImageNet's image classification competition by a large margin [Krizhevsky et al., 2012]. The core reason underlying this success was the efficient use of graphics processing units, many training examples, and rectified linear units (ReLU) [Ma et al., 2019]. This success has caused deep learning to attract much more attention in distinct sub-fields of computer vision. In 2014, Facebook utilized DeepFace, a system developed with deep learning, to automatically detect and tag users in photos. AlphaGo, Google's DeepMind algorithm, beat one of the world's strongest go players at the Go game tournament in 2016 [Gibney, 2016]. Because this game was much more complex than chess, this success captured everyone's imagination and took the promise of deep learning to a new level. After these successes and thanks to increasing data and computational resources, the use of deep learning has increased for many applications in various fields, including remote sensing, medicine, and the defense industry.

4.2. Deep Learning Architectures

In the literature, various deep learning techniques have been developed that can be used depending on the application and the field of study. These include Recurrent Neural Networks (RNNs), Deep Belief Networks (DBNs), Restricted Boltzmann Machines (RBMs), Long Short-Term Memory Networks (LSTMs), Auto-Encoders (AEs), and Convolutional Neural Networks (CNNs). These architectures can be utilized to solve problems in distinct domains or use cases. For instance, CNN is generally used in computer vision and image recognition applications and RNN is used in time series problems. These architectures are reviewed in the following subsections.

4.2.1. Restricted Boltzmann Machines

Restricted Boltzmann Machine (RBM) was first proposed by Paul Smolensky [1986] under the name Harmonium and came to prominence after the work of Geoffrey Hinton and collaborators. RBMs are generative stochastic undirected artificial neural networks that can learn probability distributions across their inputs and consist of just two layers. Also, RBMs are a subset of Boltzmann machines. As shown in Figure 4.3, an RBM contains the visible layer (input layer), the hidden layer, and the connections between these two layers [Chu et al., 2018]. In these networks, the connections between neurons are symmetrical and bidirectional, as the weights are the same in both directions during training and usage of the neural network and information flows in both directions [Coşkun et al., 2017]. The visible layer that receives the input data has connections with the neurons of the hidden layer. However, there are no connections between neurons in the same layer. That is, visible neurons contain no connections between themselves. Likewise, hidden neurons also have no connections between themselves. Therefore, RBMs are considered restricted. On the other hand, there is no such restriction on Boltzmann machines. Noteworthy is that this restriction enables training algorithms that are more effective than those for Boltzmann machines. RBMs are utilized in various applications, including dimensionality reduction, regression, classification, feature learning, and collaborative filtering. Also, RBMs can form a deep belief network.

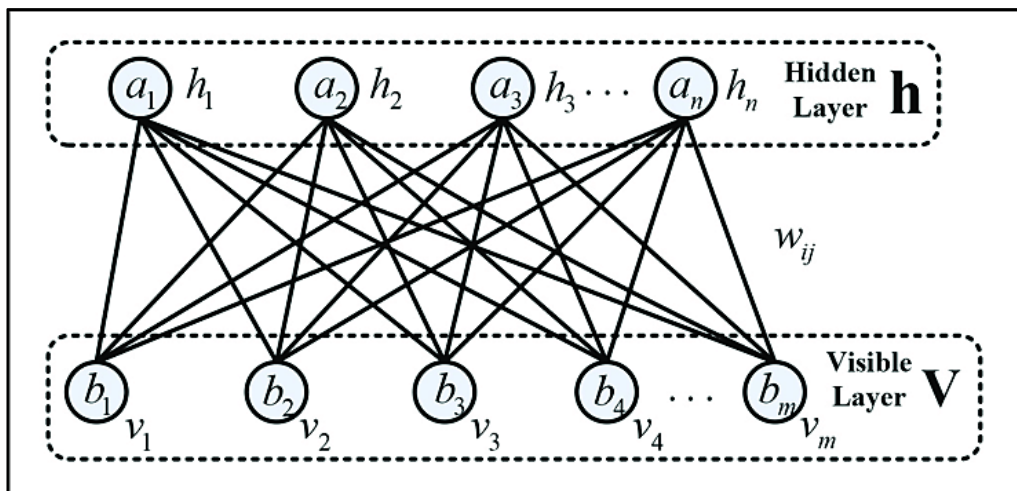


Figure 4.3: The basic structure of RBM with n hidden and m visible units.

4.2.2. Deep Belief Networks

Deep belief networks (DBNs), presented by Geoffrey Hinton [2009], are multilayer generative neural networks made up of stacked RBMs. In the DBN, each pair of connected layers is formed with an RBM. Each sub-network's hidden layer functions as the next visible layer. In this way, a model consisting of stacked RBMs has a stronger learning ability than a single RBM [Liu et al., 2017]. The architecture of DBN is depicted in Figure 4.4 [Web 9, 2022]. The hidden layer of RBM1 serves as the visible layer of RBM2. Outputs from RBM1 are fed to RBM2 as inputs for training RBM2. This operation is iterated until the network's output layer is trained, and then the network undergoes fine-tuning where the output layer is utilized to label the network. The training process of DBNs consists of these two phases, which are pre-training and fine-tuning. The core principle behind DBN is to train the network with an unsupervised learning method layer by layer using unlabeled data for each RBM layer before fine-tuning the whole neural network with supervised learning using labeled input [Ou et al., 2019], [Sarker, 2021]. Training of DBNs is carried out in these two phases. DBNs can play a crucial role in high-dimensional data applications with their classification and powerful feature extraction capabilities [Sarker, 2021]. DBN architectures are utilized in applications such as image recognition, information retrieval, natural language processing, failure prediction, and scene labeling.

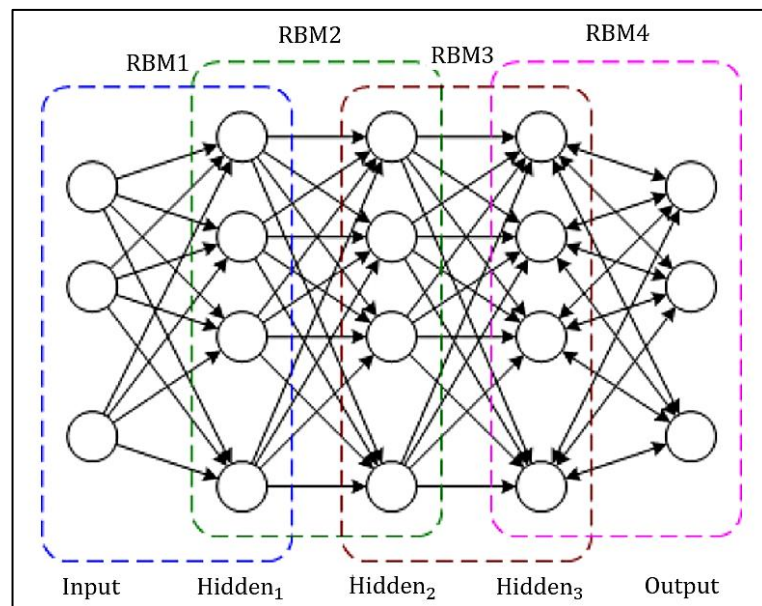


Figure 4.4: An architecture of DBN.

4.2.3. Auto-Encoders

Auto-Encoders (AEs) [Goodfellow et al., 2016] are popular unsupervised learning algorithms in which the input layer is the same as the output layer. These networks first perform an operation of compressing the input data into a latent space representation, and after that try to reconstruct the output from this representation. This compression process aims to diminish the size of this input and increase the efficiency of the learning process, since a large input data set will cause low computation speed [Coşkun et al., 2017]. The structure of an auto-encoder is shown in Figure 4.5 [Sublime and Kalinicheva, 2019]. An AE contains three main components: encoder, code, and decoder. Among these, the encoder's task is to compress the given input data to generate code that the decoder then utilizes to reconstruct the input. Afterward, the decoder is responsible for reconstructing the input using this code. In this context, this network works like a feature extraction engine that tries to determine which of the data's features are most significant [Coşkun et al., 2017]. Given these characteristics, AEs are particularly useful for dimensionality reduction implementations [Hinton and Salakhutdinov, 2006]. Besides, it is widely applied in many unsupervised learning tasks, including denoising, feature extraction, generative modeling, efficient coding, anomaly, or outlier detection [Goodfellow et al., 2016]. Also, there are different variants of AEs, including, Sparse AE, Variational AE, Denoising AE, and Contractive AE. Among these, denoising, sparse, and contractive AEs are beneficial for later classification applications, while variational AEs are beneficial for generative models.

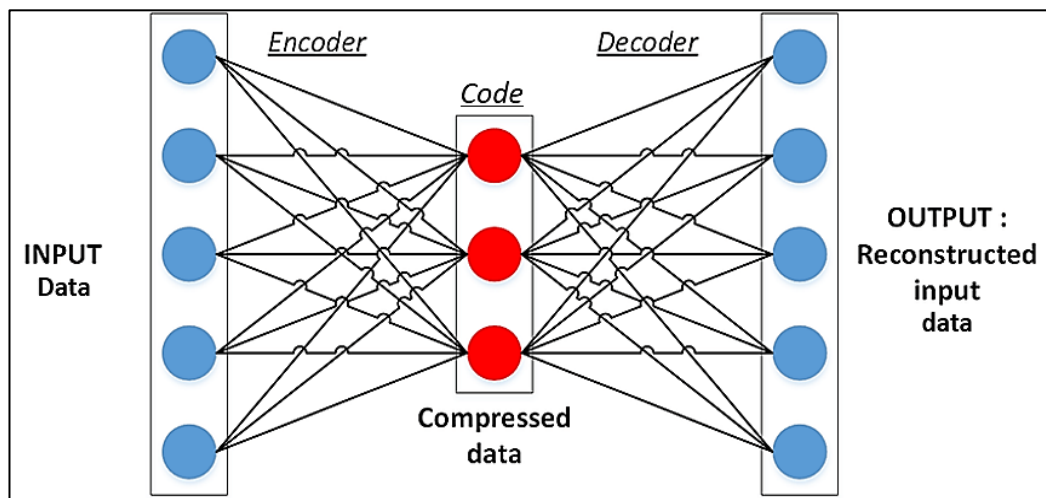


Figure 4.5: A structure of an auto-encoder with its components.

4.2.4. Recurrent Neural Networks

A simple Recurrent Neural Network (RNN) example was first introduced by Elman [1990]. RNN is a neural network architecture in which connections between neurons form a loop using recurrent computations to extract sequential data. They have a memory that can affect current input and output by capturing information about previous inputs. It differs from a multi-layered feed-forward network with these connections between its neurons. In deep neural networks, inputs and outputs are assumed to be independent of each other, while RNNs are structures whose output depends on the previous elements in the sequence [Sarker, 2021]. That is, for each element in a sequence, they perform the same process based on previous outputs, and for this reason, they are called recurrent. The working principle of a conventional architecture of RNN is shown in Figure 4.6 [Web 10, 2022]. The right part of the figure is the unrolled RNN structure where the previous information is transformed to the next time step, while the left part is the typical RNN architecture. Here, x , h , and o represent the input value, output value, and hidden layer of the network, respectively. As can be seen, a hidden layer (h) at time t depends on the input x_t at time t and the previous hidden layer at time $t-1$, that is, the output at time $t-1$. This neural network has variants, including Long Short-Term Memory network (LSTM), Multidimensional LSTM network, Bidirectional LSTM network, and Hierarchical LSTM network. RNNs are popular models with many application areas, including machine translation [Datta et al., 2020], document summarization [Nallapati et al., 2017], language modeling [Mikolov et al., 2010], and speech recognition [Graves et al., 2013].

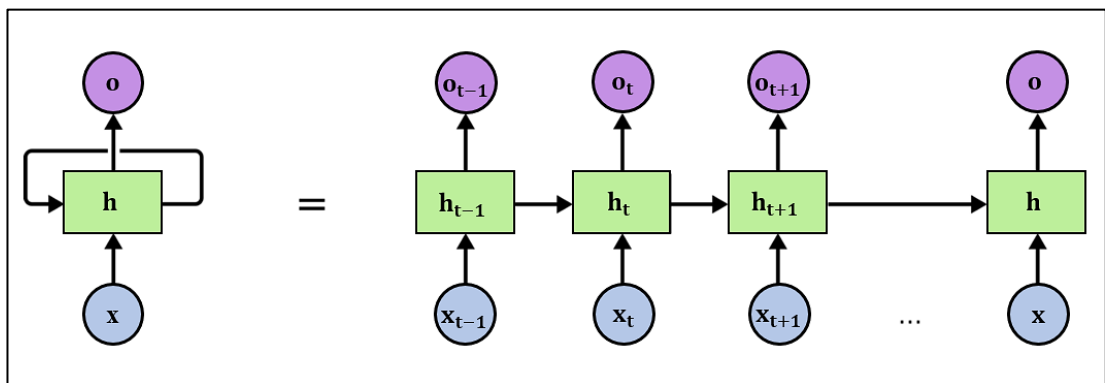


Figure 4.6: An unrolled representation of an RNN.

4.2.5. Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) is an RNN method first proposed by Hochreiter and Schmidhuber [1997]. Since the RNN models have problems remembering the long-term dependency, that is, they have short-term memory, LSTMs have been proposed to deal with these shortcomings. In addition to the feed-back loops that RNNs utilize to keep information in memory over time, LSTMs have special units to address long-term dependency and overcome the vanishing gradient problem. A typical LSTM neural network consists of multiple memory blocks (units). Generally, an LSTM has forget gate, input gate, and output gate. These work together to control the flow of information in an LSTM cell [Sarker, 2021]. A basic LSTM network structure is demonstrated in Figure 4.7 [Web 11, 2022]. The input and output gate serve the purposes of deciding what information should be passed to the cell state and deciding and controlling the outputs, respectively. The forget gate is used to decide the information to be stored from the prior state cell and to forget the information that is no longer required. The capability to decide all these based on the network's current input is one of the LSTM's important strengths [Mosavi et al., 2019]. LSTMs are widely used in sequential or time-series problems because they can learn long-term dependencies with their transactive memory mechanism [Ahmadi et al., 2019]. In addition, LSTMs are also applied in tasks, including machine translation, question-answering chatbots, natural language processing, speech recognition, and classification. Besides, essential technology companies such as Amazon, Google, and Apple also use LSTM in their voice recognition platforms.

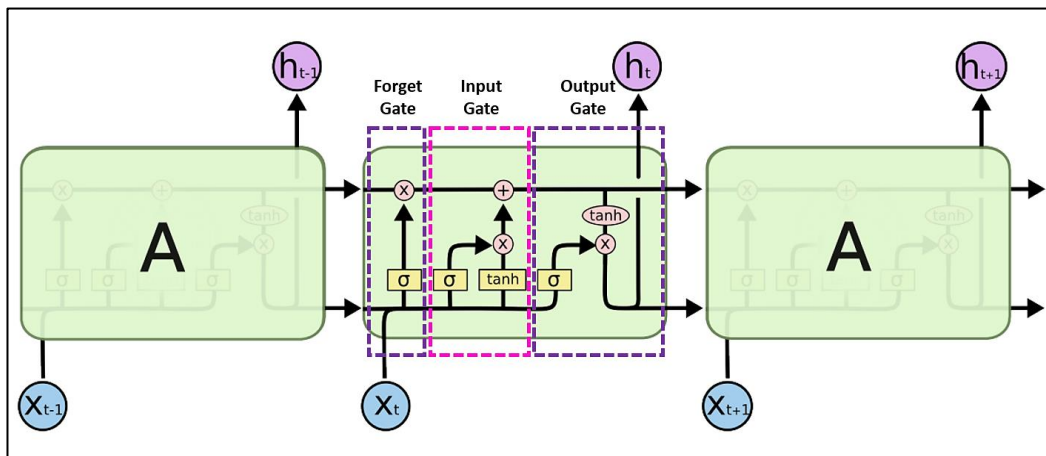


Figure 4.7: Basic structure of an LSTM.

4.2.6. Convolutional Neural Networks

Convolutional neural networks (CNN) [LeCun et al., 1998] are one of the most widely utilized deep learning algorithms with remarkable performance for image processing applications, including object detection, image classification, and image segmentation. CNNs contain many hidden layers, and deeper layers can detect more abstract and specific features. Also, it automatically learns features from the given input data without the need for manually extracted features (handcrafted features) with conventional learning techniques. Since CNNs were originally designed to process data that come from multiple arrays, they are very useful models for processing multi-band remote sensing image data [LeCun et al., 2015], [Ma et al., 2019]. Besides, CNNs are successfully implemented in many areas such as natural language processing, biomedicine, and signal processing. CNNs consist of three hierarchical structures: convolutional layers, pooling layers, and fully connected layers. Among these, convolutional layers perform feature extraction by scanning over the input with matrices called filters or kernels. The feature maps from this layer are made smaller with various functions in the pooling layer. These features are then converted into a one-dimensional vector and subjected to classification in the fully connected layer. The first famous CNN architecture, named LeNet-5, developed by LeCun for the task of recognizing handwritten digits is illustrated in Figure 4.8 [LeCun et al., 1998]. This neural network contains three convolutions ($C1$, $C3$, $C5$), two pooling or subsampling ($S2$, $S4$), a fully connected ($F6$), and an output layer. The input of the network is handwritten number image, and the network's output is a probability of over ten possible numerical values. With this architecture, handwritten numbers in the MNIST dataset were predicted with high accuracy.

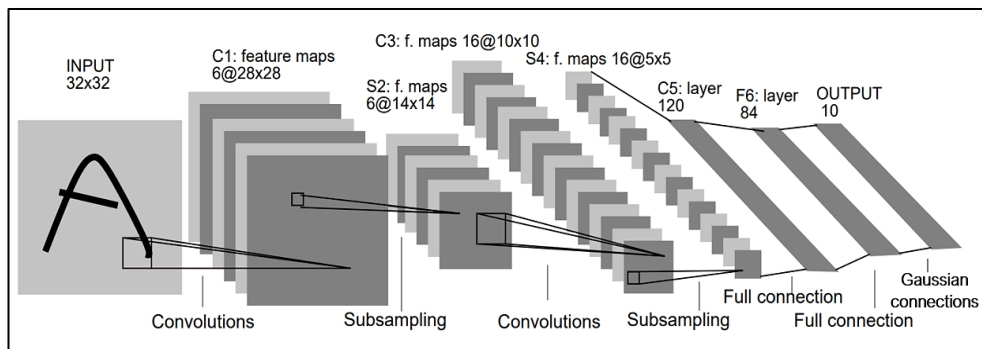


Figure 4.8: Architecture of LeNet-5, a CNN.

With the development of GPUs and the need for shorter training times, CNNs have attracted the attention of an increasing number of researchers. Also, newly developed deep learning techniques also contributed to the development of CNN models. For example, Glorot et al. [2011] introduced the use of Rectified Linear Units (ReLU) as an activation function in the CNN models. Srivastava et al. [2014] presented the dropout technique for implementing CNNs. Subsequently, the application of CNNs in image processing began to boom, with AlexNet [Krizhevsky et al., 2012], the CNN architecture designed using dropout, ReLU layers, and data augmentation techniques, winning the ImageNet competition in 2012. Later, many popular deep CNN architectures including ZFNet, VGGNet, GoogleNet, and ResNet were developed. On the other hand, various robust CNN architectures have been developed for object detection and recognition task, and these models have also been successfully employed to remotely sensed images. These models are mainly regional CNN structures (R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN), YOLO series (YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5), SSD, EfficientDet, and CenterNet models.

4.2.6.1. Convolutional Layer

Convolution is an integral operation defined between two functions that states the amount of overlap of one function as it is shifted over the other function [Coşkun et al., 2017]. In CNN architectures, the convolution layer generates many feature maps from the input data as a result of a convolution operation using filters. This layer contains the parameters of the number of filters (d), filter size ($k \times k$), stride (s), and padding size (p). Stride (s) refers to the number of pixels the filter will move to. Padding size (p) refers to the process of padding the edges of the image in size p to control the output size. When the input image size is $W_i \times H_i \times C_i$ (W_i indicates the width, H_i the height, and C_i the number of channels), the output feature map size ($W_o \times H_o \times C_o$) is calculated by equations 4.1 and 4.2. Also, C_o is equivalent to the number of filters (d).

$$W_o = \frac{W_i - k + 2p}{s} + 1 \quad (4.1)$$

$$H_o = \frac{H_i - k + 2p}{s} + 1 \quad (4.2)$$

For 2D data such as images, 2D convolution operation is considered. A simple 2D convolution operation is illustrated in Figure 4.9 [Web 12, 2022]. The convolution filter moves successively over every pixel on the image according to the stride size, starting from the upper left corner of the given input image. Here, the dot product of the 5x5 sized input image matrix and the 3x3 sized filter matrix is performed and the sum of all values is assigned as the corresponding element of the output matrix (feature map matrix).

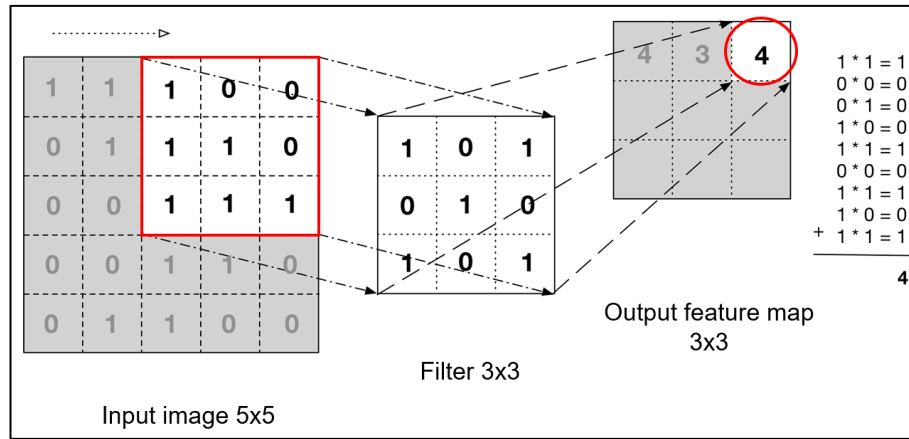


Figure 4.9: A simple depiction of 2D convolution operation.

After the convolution operation, an activation function (f) is used to reduce nonlinearity. For this purpose, the ReLU activation function is commonly utilized. This function is defined by the formula $f(x) = \max(0, x)$, which replaces all the negative pixel values with zero. When K is a $k \times k$ filter, X is a $k \times k$ patch of the input image, b is a bias, and $*$ is 2D convolution operation, the output activation from a feature map $Z(x)$ is obtained by calculating $Z(x) = f(K * X + b)$.

4.2.6.2. Pooling Layer

The pooling (or subsampling) layer is responsible for reducing the width and height of the feature maps. For this, it retains the most relevant attributes and eliminates the redundant ones. Although this process causes feature loss, it reduces the number of parameters of the network and speeds up its training. Hence it also minimizes the possibility of overfitting. There are no learnable parameters in this layer. Like convolutional layers, pooling layers have filter size ($k \times k$) and step (s) parameters.

Average and maximum pooling methods are mostly used for the pooling operation. Average pooling averages the values of the feature map underlying the pooling filter, while maximum pooling takes its greatest value. In Figure 4.10, maximum and average pooling operations with a 2x2 filter size are applied to the input with a stride of 2 [Web 13, 2022]. These operations divide the height and width of the input in half.

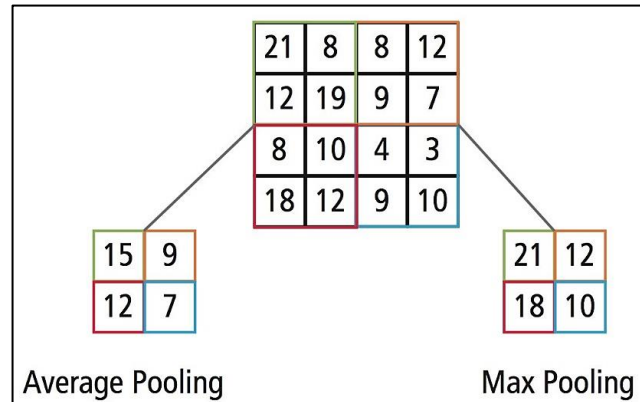


Figure 4.10: Example of average and maximum pooling operations with a stride of 2.

4.2.6.3. Fully Connected Layer

After the convolution and pooling layers, the fully connected layer (FCL) is responsible for high-level reasoning in the network. The structure of the FCL is shown in Figure 4.11 [Web 14, 2022]. As can be seen, all neurons in the FCL are connected to every neuron in the previous layer.

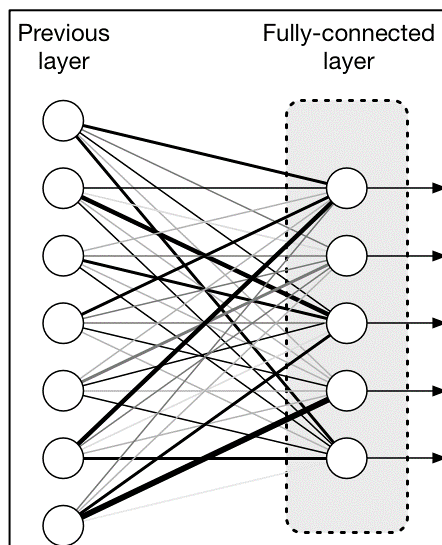


Figure 4.11: The structure of the fully connected layer.

In the convolutional layer, high-level features of the input data are obtained. Outputs from the last convolutional layer in the network are flattened, and then connected to the output layer by the FCL. This process is the most straightforward way to learn nonlinear combinations of features. In FCL, it is determined to which class the features, which will identify the object belong. That's why FCL looks for high-level features that have a high degree of relevance to a class. The weights (connections) that indicate these high-level features are shown thick in Figure 4.11 because the relationship between neurons is very strong. Taking these weights into account, the FCL determines which class the input data belongs to.

4.3. Deep Learning-Based Object Detection Models

Object detection is a technique that allows locating and classifying an object in an image. Various models have been developed for detecting objects from images in the field of deep learning. Generally, deep learning-based object detection algorithms are categorized into one-stage models and two-stage models. As the name implies, the object detection process in two-stage models happens in two phases. In the first stage, regions of interest are generated from the given input images using a Region Proposal Network (RPN). Afterward, the generated proposals are classified and the location of the objects in the image is determined using the bounding box regression. In contrast, one-stage models omit the process of producing a region of interest. It performs object identification in just one stage, adopting a unified system that regresses class probabilities and bounding box locations directly from the input images. To sum up, the main difference between two-stage and one-stage detectors is whether ROI is generated in the object detection process. Considering this difference, the detection speed of one-stage algorithms is extremely high and convenient for use in real-time detections. On the other hand, two-stage models generally demonstrate higher performance in object detection tasks. However, it should be noticed that the models in both categories have their own merits based on the practices.

In this study, Faster Region-Based Convolutional Neural Network (Faster R-CNN) from two-stage models and Single Shot MultiBox Detector (SSD), EfficienDet, and CenterNet from one-stage models are discussed in order to detect ship targets from remotely sensed images. These models will be examined in detail in the sub-headings.

4.3.1. Faster Region-Based Convolutional Neural Network

Faster Region-Based Convolutional Neural Network (Faster R-CNN) introduced by Ren et al. [2015], is the latest architecture in the R-CNN series and one of the most popular object detection models. The first version R-CNN [Girshick et al., 2014], utilizes a selective search algorithm to propose regions of potential interest with a hierarchical grouping method, eventually producing around 2,000 region proposals. Each proposal is independently passed through a CNN for feature extraction, and then the output feature maps are subjected to classification with SVM. However, having a vast quantity of the region proposals and passing them independently through a CNN makes R-CNN time-consuming. Hence, R-CNN is not ideal for real-time applications. These drawbacks in R-CNN have led to the emergence of the Fast R-CNN model. The workflow of the R-CNN model is presented in Figure 4.12 [Girshick et al., 2014].

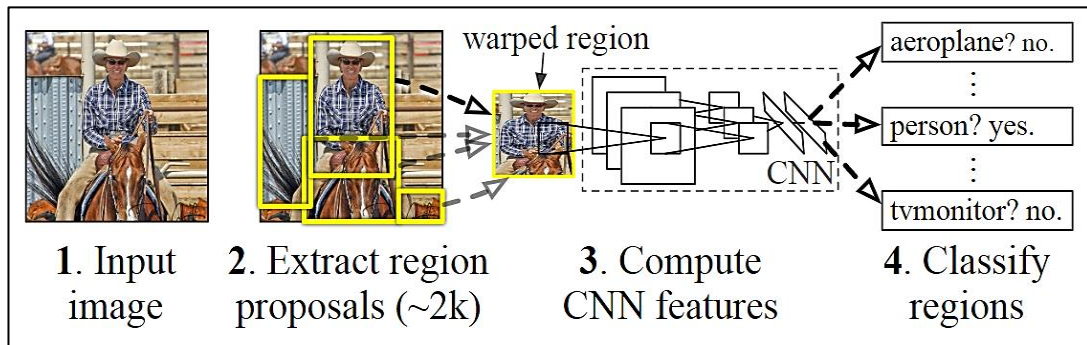


Figure 4.12: Architecture of R-CNN model.

In the Fast R-CNN model [Girshick, 2015], whose workflow is depicted in Figure 4.13, some of the bottlenecks of R-CNN have been overcome. In this model, the whole image is directly passed through a CNN, and regions are proposed on the output feature map of the CNN. In this way, CNN inference is not applied independently for 2,000 region proposals, thus the processing time is significantly reduced. Besides, this network utilizes the Region of Interest (RoI) pooling layer, to provide a fixed output size. Then, a SoftMax layer predicts the class of region proposals, and locating these region proposals is performed by bounding box regression. Both R-CNN and Fast R-CNN utilize selective search algorithm to produce region proposals. However, time-consuming, and slow nature of this external mechanism is regarded as the bottleneck of these models.

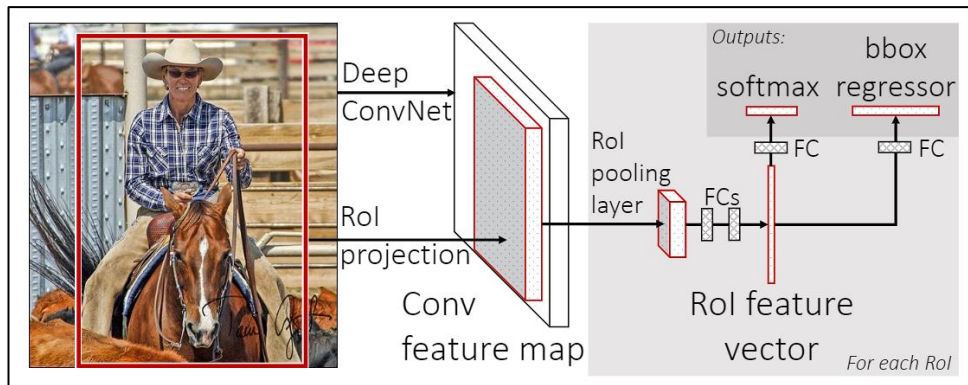


Figure 4.13: Fast R-CNN model architecture.

Faster R-CNN [Ren et al., 2015] uses the Region Proposal Network (RPN) to propose regions for areas where objects can be found from feature maps, and the Fast R-CNN detector to identify objects in these proposals. The architecture of faster R-CNN is depicted in Figure 4.14 [Ren et al., 2015]. This network eliminates the need for selective search algorithm. It replaces this algorithm with RPN. First, to produce feature maps the input image passes through the CNN. Next, feature maps extracted from CNN are passed to RPN, the deep fully convolutional network, to propose regions. The RPN takes these feature maps as input and proposes regions of interest with an objectness score for each of them. Afterward, a RoI pooling layer is employed to convert these region proposals to a fixed size. Finally, proposals are classified by SoftMax, and the object's bounding boxes are determined by bounding box regression.

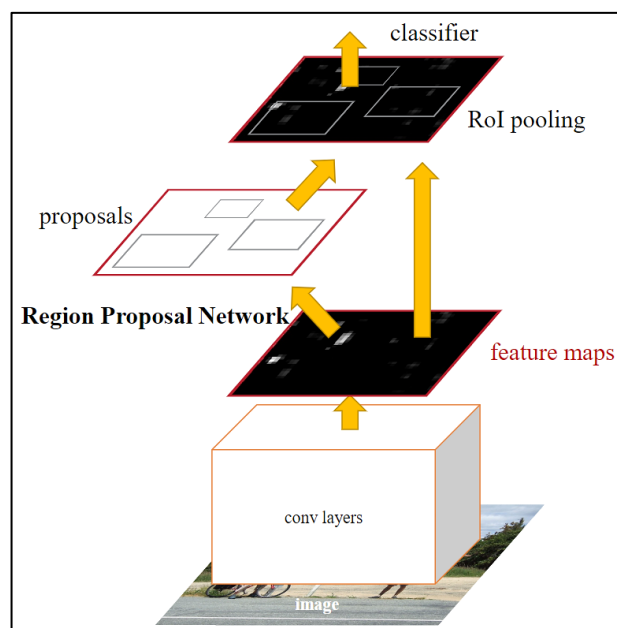


Figure 4.14: Architecture of Faster R-CNN.

4.3.2. Single Shot Multibox Detector

Single Shot Multibox Detector (SSD), introduced by Liu et al. [2016], refers to a one-stage detection approach using a single feed-forward convolutional neural network to predict object categories (classes) and anchor offsets (predicted bounding boxes). This characteristic of SSD increases its processing speed, while decreasing its detection accuracy. It generates a fixed-size set of bounding boxes and category scores. The category score refers to the presence of instances of object classes in these bounding boxes. Then, it generates the final detection results using a non-maximum suppression (NMS) algorithm. As seen in Figure 4.15, the early layers of the SSD model network are based on the VGG-16 base network. However, successful results can be produced using other base networks [Liu et al., 2016].

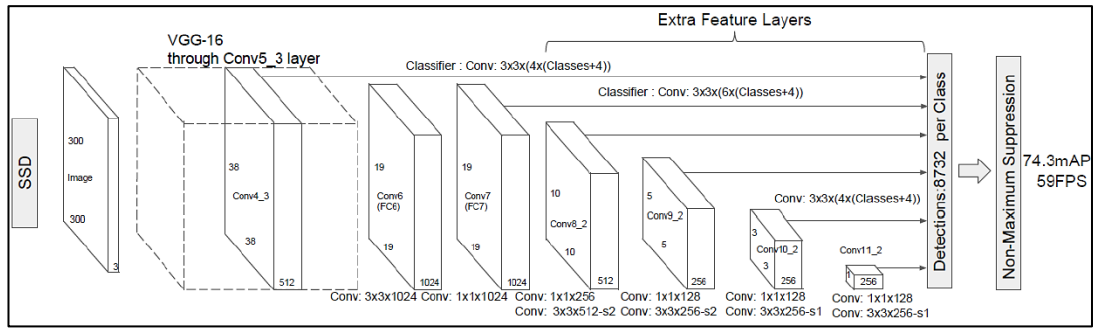


Figure 4.15: The network architecture of SSD.

SSD utilizes a set of default anchor boxes with distinct aspect ratios and scales. In this model, convolutional feature layers that produce feature maps at various scales are added to the end of the basic network to deal with the detection of objects at different scales and achieve high accuracy. These layers are tasked with predicting the offsets to default bounding boxes of distinct scales and aspect ratios and their associated confidences. In addition, these layers are in a structure that is progressively decreasing in size. In the SSD network shown in Figure 4.15, these extra feature layers are pointed out on the top of the network. These added feature layers utilize a set of convolutional filters to generate a fixed set of detection predictions. The loss of the network is expressed as the weighted sum between the confidence loss (e.g., SoftMax) and the localization loss (e.g., Smooth L1).

The SSD model framework is shown in Figure 4.16 [Liu et al., 2016]. According to the framework, training of the model needs an input image, as well as ground truth (GT) boxes for each object which represent their actual locations (Figure 4.16 (a)). In feature maps at different scales, the default boxes with different aspect ratios and distinct locations are evaluated. For example, Figure 4.16 (b) shows an example of 8×8 feature map, and (c) shows an example of 4×4 feature map. Then, the localization offsets ($\Delta(c_x, c_y, w, h)$) of all these default boxes, as well as the confidence scores ($c_1, c_2, \dots, c_p$) indicating the existence of the class instance for all object categories are predicted. In these offsets, c_x and c_y represent offsets from the center of the default box, and w, h represent offsets from the dimensions of the default box. In the training process, these default boxes are first matched with ground truth boxes. For instance, in Figure 4.16, the two boxes indicated in blue in the 8×8 feature map are matched with the cat object, and the two boxes indicated in the red in the 4×4 feature map are matched with the dog object. The reason why the dog cannot be matched with any box in the 8×8 feature map is that these boxes do not match the dog object box because they are at different scales. For the same reason, the cat cannot be matched to any box in the 4×4 feature map. Thus, during training, these three boxes are considered as positive, and the remaining boxes are considered as negative [Liu et al., 2016].

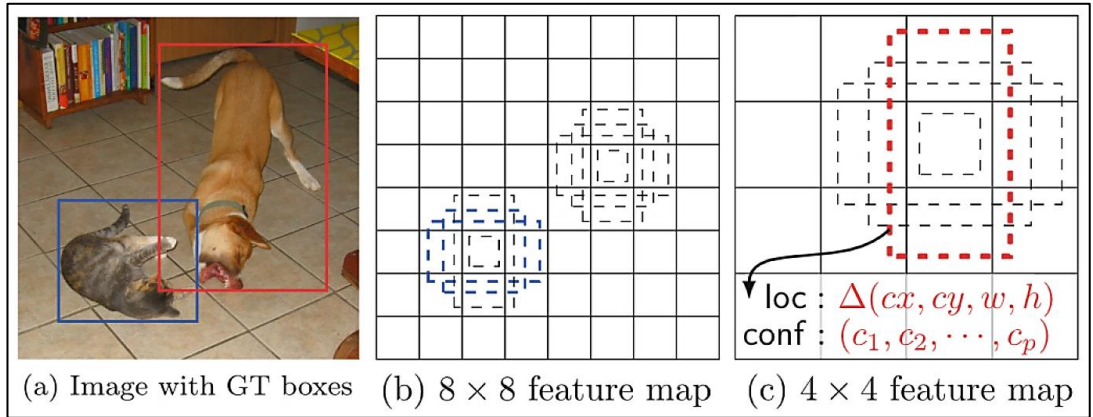


Figure 4.16: SSD model framework.

In addition to the set of default boxes and scales for object recognition, data augmentation, and hard negative mining strategies are also integrated into the training process of the SSD model. All these characteristics have significantly improved the performance of the SSD model.

4.3.3. EfficientDet

The core principle underlying the EfficientDet model [Tan et al., 2020] is to build a scalable detector with high accuracy and efficiency based on the one-stage detector approach. EfficientNets pre-trained by ImageNet are used as the backbone network in the model. For efficient multi-scale feature fusion, the Weighted Bi-directional Feature Pyramid Network (BiFPN) is employed, which presents learnable weights to learn the significance of distinct input features. The basis of multi-scale feature fusion is to aggregate features extracted at different resolutions. A compound scaling technique that is a heuristic-based approach is adopted for model scaling. This method is based on jointly scaling up the input resolution, depth, and width for the entire backbone, feature network, and class/box prediction network. EfficientDet architecture, built by the integration of EfficientNet backbone, BiFPN feature network, and compound scaling, demonstrates high performance with few parameters and the number of floating-point operations per second (FLOPs) [Tan et al., 2020].

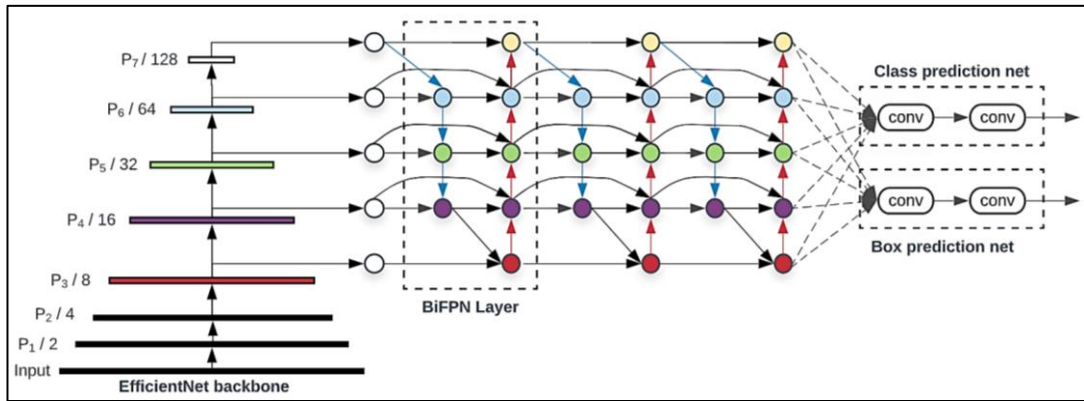


Figure 4.17: The overall architecture of EfficientDet.

The EfficientDet architecture built on the EfficientNet [Tan and Le, 2019] backbone network is illustrated in Figure 4.17 [Tan et al., 2020]. EfficientNet architecture, which is a convolutional neural network, consists of seven different feature levels ($P_1, P_2, P_3, P_4, P_5, P_6, P_7$). Features from level 3 to 7 ($P_3 - P_7$) are sent as input to the BiFPN network, where they are subjected to top-down and bottom-up bidirectional feature fusion. Then, these fused features are fed into the shared class/box network to predict object class and bounding boxes. The layers of the class/box network and the BiFPN layers are repeated multiple times.

Diverse network structures for multi-scale feature fusion are illustrated in Figure 4.18 [Tan et al., 2020]. The Feature Pyramid Network (FPN) provides a one-way top-down information flow for feature fusion, while the Path Aggregation Network (PANet) adds an extra bottom-up pathway to FPN. Neural Architecture Search Feature Pyramid Network (NAS-FPN) utilizes neural architecture search and repeatedly employs the same block. On the other hand, the design of Bi-FPN, which has better accuracy and efficiency trade-offs, was inspired by the PANet. Bi-FPN optimizes the FPN by removing connections that less contribute to performance and reduces the training of the network. Also, BiFPN uses weighted feature fusion. This means that the BiFPN layers add extra weight for each input feature, allowing the network to learn about their significance. Moreover, depthwise separable convolution is employed instead of regular convolution for feature fusion. After each convolution, batch normalization and Self-Gated Activation Function (SWISH) are utilized [Tan et al., 2020].

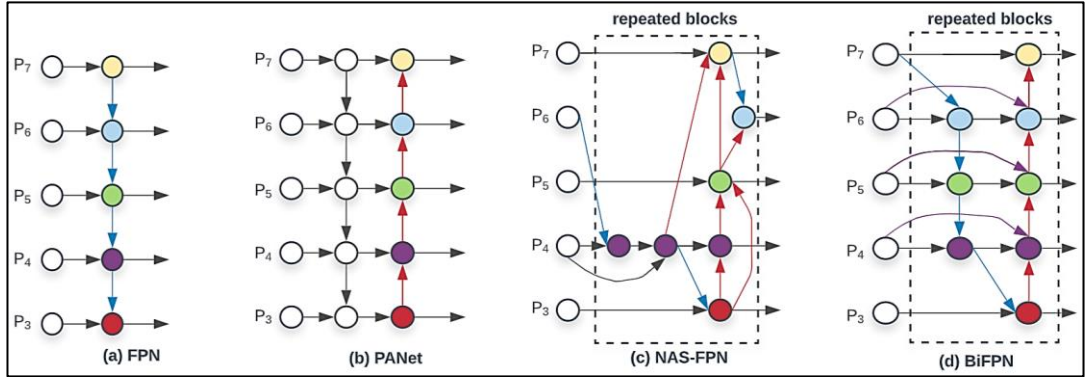


Figure 4.18: Comparison of different networks for multi-scale feature fusion.

In the compound scaling methodology, the same width/depth scaling coefficients of EfficientNet-B0 to B6 are reused for backbone network scaling. The BiFPN depth is increased linearly, and its width is increased exponentially. The width scaling factor is set at 1.35. In the box/class prediction network, the widths are fixed to be the same as BiFPN and the depth is increased linearly. Since feature level 3 - 7 is utilized in BiFPN, the input image resolution is linearly increased, dividable by $2^7 = 128$. With these compound scaling configurations, EfficientDet-D0 to D7 is developed. Of all of these, EfficientDet-D0 has the smallest input resolution with 512, while EfficientDet-D7 has the highest resolution with 1536.

4.3.4. CenterNet

CenterNet [Zhou et al., 2019] is a one-stage and keypoint-based object detection pipeline. Unlike conventional anchor-based object detection models, CenterNet represents objects as a single point at the center of their bounding box. In other words, it treats the center point of an object as a prerequisite for localizing a bounding box. Other information, including object size, pose, dimension, and orientation is calculated by regression from the image features at the center position. In this model, the input image is fed to a fully convolutional network (FCN). This FCN generates a heatmap whose peaks correspond to the object centers. The height and width of the object bounding boxes are predicted by the image features at each peak. It should be noted that no Intersection over Union-based (IoU) NMS or other post-processing is required for outputs, all are generated directly from keypoint prediction [Zhou et al., 2019].

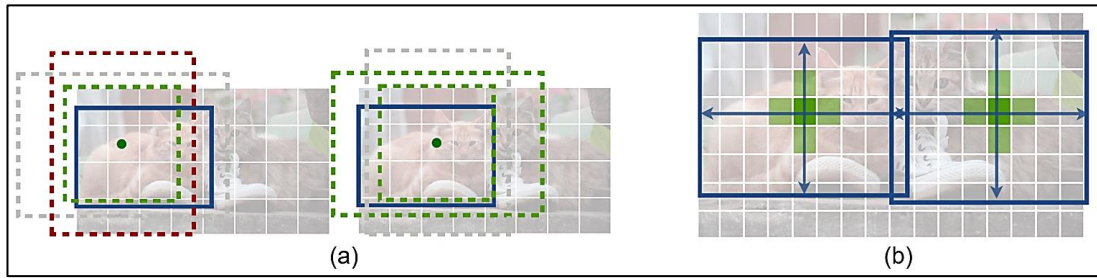


Figure 4.19: Comparison of anchor-based detectors and center point detector. (a) Standard anchor-based detection, (b) Center point-based detection.

A comparison of anchor-based detectors and center point detector is depicted in Figure 4.19 [Zhou et al., 2019]. Standard anchor-based detection considers the overlap of anchors to any objects at a specific IoU threshold. In such circumstance, if anchors provide $IoU > 0.7$ overlap to any object, they are considered positive. On the other hand, if anchors provide $IoU < 0.3$ overlap to any object, they are considered negative or ignored otherwise. However, in center point-based detection, the center pixel is assigned to the object and nearby points have a decreased negative loss. Object size is also calculated by regression. Anchorless center point-based detection has been developed to deal with bottlenecks of anchor-based detection. These bottlenecks include the manual design of anchor box sizes, anchor boxes are not always aligned

with ground truth boxes, and the high computational cost requirement due to a vast number of candidate boxes.

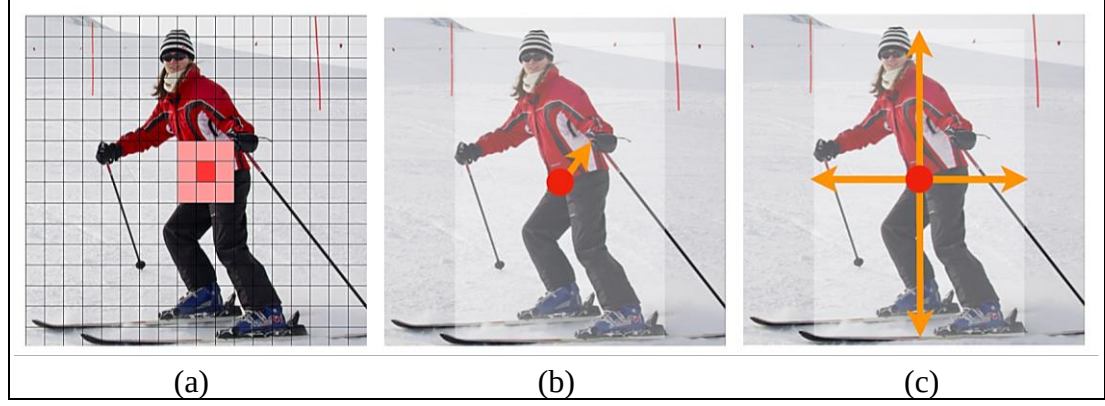


Figure 4.20: Overview of the CenterNet network outputs for the object detection task. (a) Keypoint heatmap, (b) Local offset, (c) Object size.

CenterNet utilizes a key point estimator $\hat{Y}$ to estimate all center points, and the object size for each object is calculated by regression. A single size estimation is employed for every object category to avoid the computational load. In addition, a local offset is estimated for each center point. An example of keypoint heatmap, local offset, and object size produced by the CenterNet network for object detection task is illustrated in Figure 4.20 [Zhou et al., 2019]. These keypoints ($\hat{Y}$), offsets ($\hat{O}$), and size ($\hat{S}$) are estimated using a single network. At inference time to determine all responses whose value is greater than or equal to its eight connected neighbors, first the peaks in the heatmap for each category are extracted independently, and the first one hundred peaks are kept. Afterward, each keypoint location is given by an integer coordinate (x_i, y_i) . These are combined with offset predictions and size predictions to generate the bounding box by equations 4.3 and 4.4.

$$(\hat{x}_i + \delta\hat{x}_i - \frac{\hat{w}_i}{2}, \hat{y}_i + \delta\hat{y}_i - \frac{\hat{h}_i}{2}) \quad (4.3)$$

$$(\hat{x}_i + \delta\hat{x}_i + \frac{\hat{w}_i}{2}, \hat{y}_i + \delta\hat{y}_i + \frac{\hat{h}_i}{2}) \quad (4.4)$$

In these formulas $(\delta\hat{x}_i, \delta\hat{y}_i) = \hat{O}_{\hat{x}_i, \hat{y}_i}$ represents offset predictions, $(\hat{w}_i, \hat{h}_i) = \hat{S}_{\hat{x}_i, \hat{y}_i}$ represents size predictions. Formula 4.3 corresponds to the top left corner of the bounding box and formula 4.4 corresponds to the bottom right corner of the bounding box [Zhou et al., 2019].

4.4. Transfer Learning

Transfer learning allows the weights learned by models pre-trained with large datasets (e.g., COCO, ImageNet) to be reused in the process of training the target model with a smaller number of datasets. Training the network from scratch, where the network weights of the model are randomly initialized, is a very time-consuming process. Instead, with transfer learning, using weights learned by a pre-trained network allows the model to converge faster. In addition, the use of a small number of datasets for the training of deep learning models causes the model to memorize the training dataset, (i.e., overfitting). In this respect, it is an important advantage of transfer learning that this model can be trained with a smaller number of datasets by using the weights learned by the models that were pre-trained with a large data set. Thus, features learned by the previously trained network are transferred and reused in another network designed to conduct a similar task. If this original dataset is general and large enough, the features that the pre-trained network learns can be useful for many computer vision tasks involving different classes from the original dataset. In this way, the target network can achieve high performance with a smaller dataset, and the training time of the target network is reduced.

Considering these characteristics of the transfer learning approach, deep learning algorithms utilized for ship object detection in this study were trained by applying the transfer learning method. State-of-the-art deep learning models, Faster R-CNN, SSD, EfficientDet, and CenterNet, used in the study were pre-trained on the Microsoft COCO 2017 dataset. Therefore, the training of these models on the ship dataset was initialized by using the weights that these models had previously learned in the Microsoft COCO 2017 dataset.

5. APPLICATION

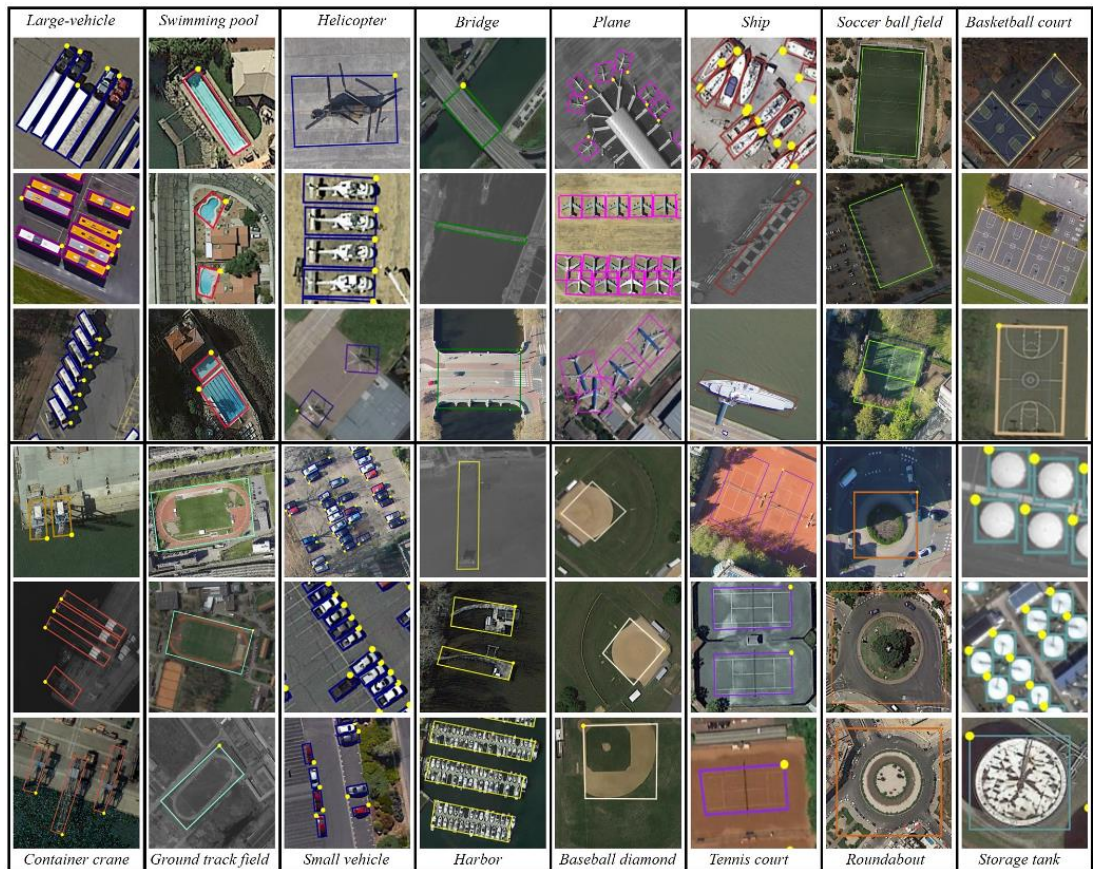
Within the scope of this thesis, the effectiveness of the state-of-the-art deep learning architectures for automatic ship recognition from remotely sensed images was investigated. For this purpose, images collected from DOTA and TGRS-HRRSD datasets were used to train and test deep learning models. The Faster R-CNN, SSD, EfficientDet-D1, and CenterNet architectures pre-trained in the TensorFlow deep learning library with the Microsoft Common Objects in Context (MS COCO) dataset were trained in the Python programming language using a transfer learning approach and the open-source TensorFlow library. Afterward, the performances of the trained models were analyzed using COCO metrics. Besides, to test the generalization capability of the models on different images from the training and testing datasets, GÖKTÜRK-1 satellite images were given to the models, and detection results were examined. The datasets utilized in the study, the training and testing process of the models, and the performance comparison are given in the next sections.

5.1. Dataset

The dataset utilized in this study was collected from open-source A Large-Scale Dataset for Object Detection in Aerial Images (DOTA) [Xia et al., 2018] and High-Resolution Remote Sensing Detection (TGRS-HRRSD) [Zhang et al., 2019b] datasets aimed at detecting objects from remotely sensed images. These datasets are widely utilized in many object detection research in the literature [Wang et al., 2019], [Liu et al., 2021], [Wu et al., 2021], [Hua et al., 2020], [Cheng et al., 2021]. They contain challenging and complicated images collected from distinct regions of the earth exhibiting the broad variety of shapes, orientations, scales, and backgrounds that object instances have. Moreover, these datasets consist of images collected at multiple resolutions from different sensors and platforms. For this reason, these two common datasets were utilized for ship detection purposes in this study. The essential cause behind combining these remote sensing public datasets is to produce a more enhanced and expanded dataset. Even if this perspective makes the dataset more arduous, it can help address the overfitting problem of the model and make it more robust and generalizable.

5.1.1. DOTA

DOTA-v1.5 is a large-scale dataset for object detection in aerial images consisting of 2806 images and 402,089 instances in 16 object categories containing ship, plane, helicopter, large vehicle, bridge, small vehicle, roundabout, harbor, storage tank, ground track field, swimming pool, basketball court, baseball diamond, tennis court, container crane, and soccer ball field. Samples of three annotated images per category in the DOTA are illustrated in Figure 5.1 [Ding et al., 2021]. The DOTA-v1.5 dataset is a revised and updated version of DOTA-v1.0. While DOTA-v1.5 contains the same images as DOTA-v1.0, the container crane category is added to the dataset in addition to the existing object categories. Also, tiny object instances (less than 10 pixels) are labeled in DOTA-v1.5. The images with sizes ranging from 800x800 to 4000x4000 and with spatial resolutions extending from 0.1 to 4.5 m were collected from different platforms including the Jilin-1, Gaofen-2 satellites and Google Earth provided by the China Centre for Resources Satellite Data and Application.



5.1.2. TGRS-HRRSD

TGRS-HRRSD is a high-resolution remote sensing image target detection dataset provided by the Optical Image Analysis and Learning Center of Xi'an Institute of Optics and Fine Mechanics, Chinese Academy of Sciences. The dataset contains 21,761 images with different sizes and a total of 55,740 object instances in 13 object categories, including ship, airplane, vehicle, harbor, storage tank, ground track field, T junction, parking lot, crossroad, tennis court, bridge, basketball court, and baseball diamond. The images were obtained from Google Earth and Baidu Map with spatial resolution varying from 0.15 to 1.2 m. Sample images of each object category in the TGRS-HRRSD dataset are shown in Figure 5.2.

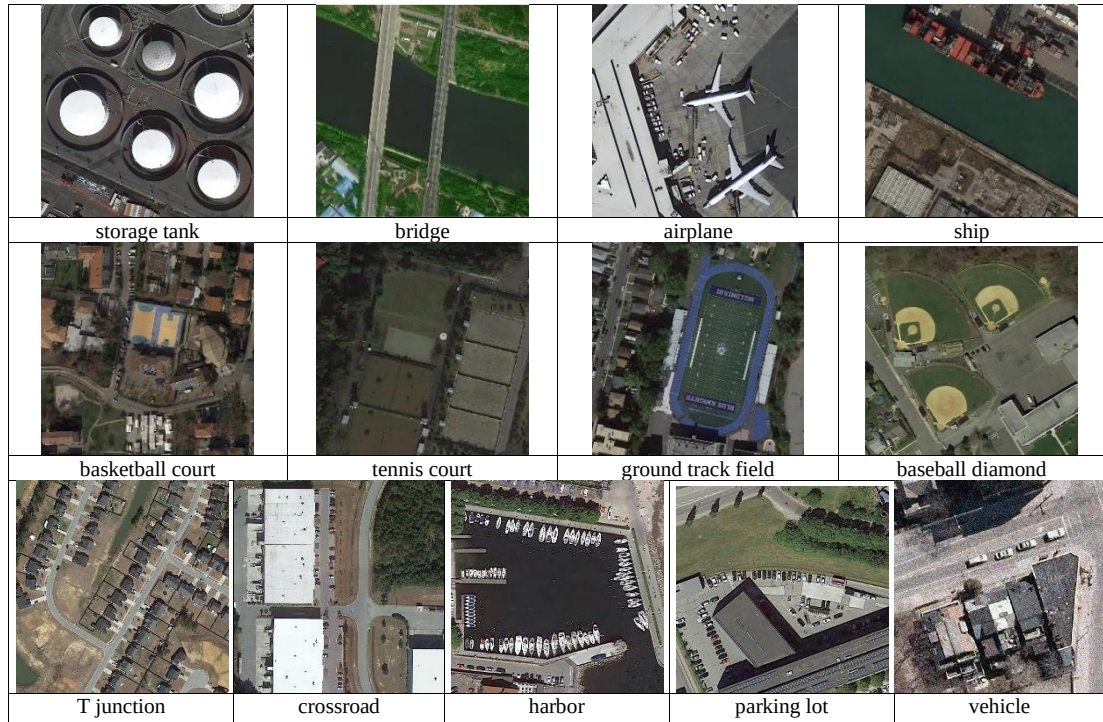


Figure 5.2: Example images of each object category from the TGRS-HRRSD dataset.

5.1.3. Preparation of the Dataset

For this work, a total of 2,136 high-resolution image samples containing ships were selected from these two datasets. The collected images have different lighting, background, occlusion conditions and include ship targets in various sizes, shapes, and orientations. Some examples of collected ship images are depicted in Figure 5.3.

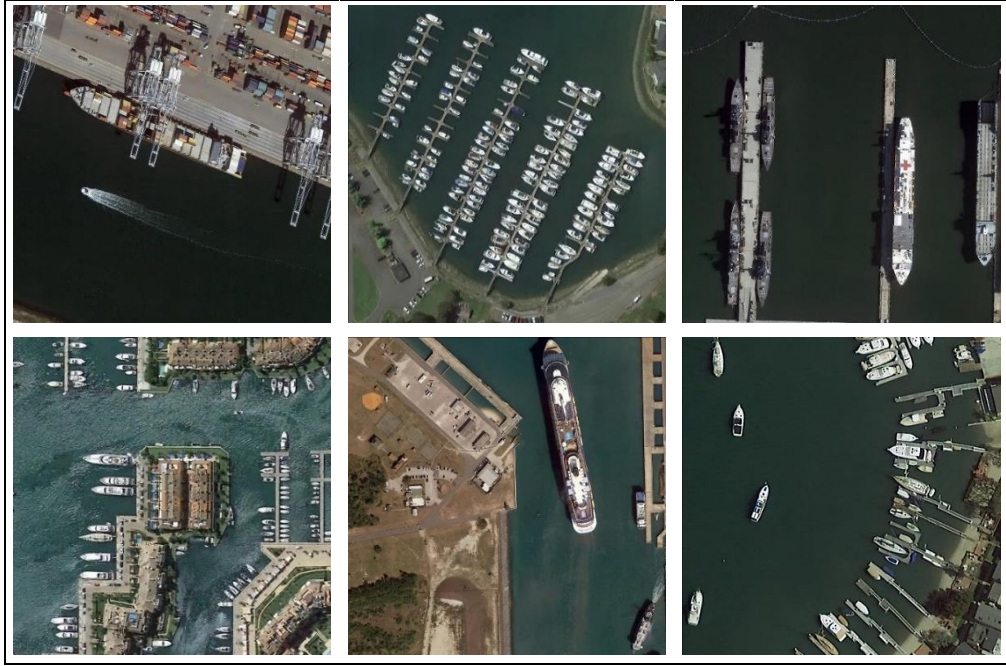


Figure 5.3: Representative images of ship dataset.

Although these datasets are already labeled and have annotation files for every image, the ground-truth boxes in some images are mislabeled and some small ships are unlabeled (Figure 5.4). Image Labeling is a crucial stage before training of the models to develop a robust and accurate model. The model learns from the labeled features during the training phase and produces outcomes based on the quality of these features. Therefore, the labeling preciseness of the features profoundly impacts on the accuracy of the model training. That is to say, ground-truth boxes should be annotated as correctly as possible and avoid ambiguity.

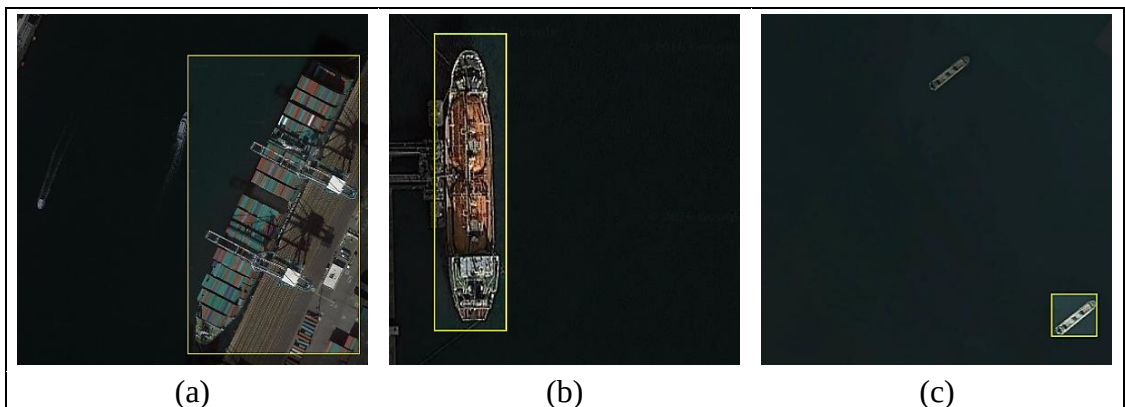


Figure 5.4: Sample ship images collected from the TGRS-HRRSD and DOTA datasets, with ground-truth bounding boxes depicting some annotation mistakes. (a) Two small unannotated ships, (b) Bounding box that cannot correctly delimit the ship, i.e., larger than the ship, (c) An unannotated ship.

To overcome these annotation mistakes, all ships in the selected images were manually relabeled with bounding boxes in the PASCAL VOC format (xmin-top left, ymin-top left, xmax-bottom right, ymax-bottom right) using the image annotation tool LabelImg [Web 15, 2022], and corresponding annotations were stored as an Extensible Markup Language (XML) file. The image annotation process is demonstrated in Figure 5.5. The annotation file contains the size of the image and the class and location information of each ship instance appearing in the image. As a result of data collection and labeling, a total of 13,735 targets were annotated in 2,136 images.

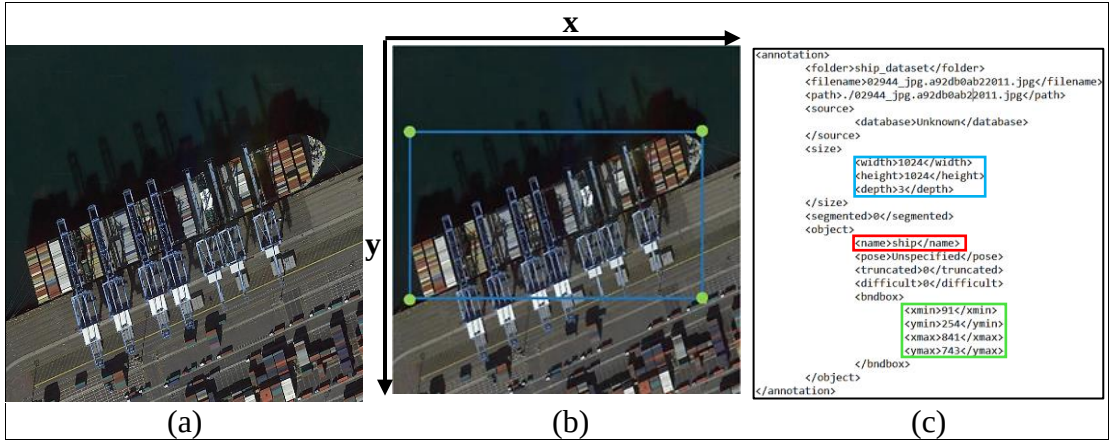


Figure 5.5: The details of the image annotation process. (a) Original image, (b) Labeled image, (c) Annotation file containing the size of the image (blue rectangle), the category of the target (red rectangle), and the position of the ship target (green rectangle).

In the experiment, the annotated ship dataset was randomly split into a training dataset and a testing dataset following the ratio of 8:2. The training and testing partition of the ship dataset is depicted in Table 5.1. The training dataset contains 1,709 images and 11,161 ship object instances. On the other hand, the testing dataset contains 427 images and 2,574 ship object samples.

Table 5.1: Number of images and ship instances in the dataset.

Model	Number of Images	Number of Ships
Training set	1709	11161
Testing set	427	2574
Total	2136	13735

5.2. Training Methodology of Deep Learning Models

In this thesis, all experiments for ship detection with deep learning-based models from remotely sensed images were carried out using TensorFlow Object Detection API, an open-source framework based on TensorFlow. It facilitates the creation, training, and use of object recognition algorithms. It includes many different models pre-trained on various datasets, including KITTI, COCO, and the Open Images Dataset. These various models with different architectures provide different execution speeds and accuracies. In accordance with the research to be done, practitioners can prefer convenient models. All deep learning models utilized in this study were trained using pre-trained weights on the MS COCO 2017 dataset. MS COCO [Lin et al, 2014] is a large-scale object detection, segmentation, and captioning dataset. It has 330,000 images (more than 200,000 tagged), 80 object classes, 1.5 million object instances, 91 stuff classes, 5 captions per image and 250,000 people with key points. It also has various features, including object recognition in context, object segmentation, and superpixel stuff segmentation. In this work, all models were trained using the Python programming language and TensorFlow framework. TensorFlow is a popular and open-source software library for deep learning and machine learning with high-performance computations designed by the Google Brain team. Intel i7 10870H CPU with NVIDIA GeForce RTX 3060 GPU hardware configurations were utilized for training the models. The hardware and software specifications utilized in this thesis study are given in Table 5.2 in detail.

Table 5.2: Hardware and software specifications used in this study.

Type	Model
Processor	Intel® Core™ i7-10870H CPU @ 2.20GHz (16 CPUs)
Memory (RAM)	32 GB
GPU	NVIDIA GeForce 3060 RTX GPU
Disk	477 GB SSD
Operating System	Windows 11 Pro (64 bit)
Programming Language	Python 3.8.11
Library	TensorFlow 2.4.0

5.3. Hyperparameters Used in Model Training

The optimal value of the hyperparameters of the models, which are predefined in the configuration files, was empirically adjusted after several experiments according to the used hardware configuration. These hyperparameters are backbone architecture, input size, batch size, learning rate, optimization algorithm, momentum, and the number of iterations. The values set for each architecture are given in Table 5.3.

Table 5.3: Training hyperparameters of the models.

	Model			
Hyperparameters	Faster R-CNN	SSD	EfficientDet-D1	CenterNet
Backbone Architecture	ResNet-101 V1	ResNet-50 V1 FPN	EfficientNet-B1-BiFPN	Hourglass-104
Input Size	1024 x 1024	1024 x 1024	640 x 640	512 x 512
Batch Size	1	1	2	2
Base Learning Rate (LR)	0.0003	0.0004	0.0001	0.00025
LR Policy	Cosine decay with warm up	Cosine decay with warm up	Cosine decay with warm up	Step decay
Momentum	0.9	0.9	-	-
Optimization	Momentum	Momentum	Adam	Adam
Number of Iterations	100000	100000	100000	100000

In the Faster R-CNN model, ResNet-101 V1 architecture was employed as feature extractor. ResNet-152 V1 and Inception ResNet V2 backbones could not be implemented due to high computational costs. The most appropriate input size for ResNet-101 V1 was determined as 1024×1024 according to the hardware capacity. The batch size was defined as 1 due to hardware limitations. For the learning rate, the cosine decay with warm up policy was used. In this policy, the learning rate was started with 0.0001 and tripled linearly until the end of the first two thousand iterations and then reduced to zero at the end of the training with the cosine decay rule. Also, the

Momentum optimizer was utilized to minimize the loss function of the model. The momentum optimizer value was set to 0.9 and the total number of training iterations of the model was set to 100,000.

In the SSD model, ResNet-50 V1 FPN architecture with an input size of 1024×1024 was implemented as feature extractor, in accordance with the hardware capacity. Similar to Faster R-CNN, batch size was defined as 1 due to hardware limitations, and cosine decay with warm up policy was utilized to define the learning rate. The learning rate was initialized with 0.0001 and quadrupled linearly until the end of the first two thousand iterations and then reduced to zero at the end of the training with the cosine decay rule. The Momentum optimizer was used to minimize the error function of the model. The momentum optimizer value and the total number of iterations of the model were set as 0.9 and 100,000, respectively.

Among the variants of the EfficientDet model to D0-D7, the EfficientDet D1 model with an input size of 640×640 , which was most convenient for the hardware capacity, was utilized. This high computational cost could not be met by the hardware capacity, as the trainable parameters in other variants of the model (D2 to D7) increased significantly. The EfficientDet D1 model uses EfficientNet-B1-BiFPN as the feature extractor. In this model, the batch size was defined as 2 and the cosine decay with warm up policy was utilized to define the learning rate. The learning rate was started with 0.00005 and doubled linearly until the end of the first two thousand five hundred iterations and then reduced to zero at the end of the training with the cosine decay rule. In this model, the Adam optimizer converged better than the Momentum optimizer for minimization of the loss function. The total number of iterations of the model was set to 100,000.

In the CenterNet model, Hourglass 104 architecture, which has the highest accuracy, was employed as feature extractor. Due to hardware limitations, the input size of 512×512 was used. 1024×1024 input resolution could not be met by hardware capacity. Similar to EfficientDet, the batch size was defined as 2. In this model, the step decay policy was used to define the learning rate. In this policy, the learning rate was initialized at 0.00025 and reduced to one-tenth every twenty-five thousand steps. Adam optimizer was used to minimize the loss function of the model. The total number of training iterations of the model was defined as 100,000.

5.4. Accuracy Assessment

In order to qualitatively evaluate the performance of the object detection models, MS COCO metrics, which are widely used to analyze the performance of the object detectors and adopt Average Precision (AP) as the primary metric, were utilized. MS COCO defines 12 metrics, including AP, AP_{0.5}, AP_{0.75}, AP_S, AP_M, AP_L, AR₁, AR₁₀, AR₁₀₀, AP_S, AP_M, AP_L to analyze the performance of an object detection model. In order to understand these metrics, some basic metrics should be examined. First, the Intersection over Union (IoU) measures the extent of overlap of the bounding box predicted by the model with the ground truth bounding box. As in Equation 5.1, it is calculated by the ratio of the overlapping area between the predicted bounding box B_p and the ground truth bounding box B_{gt} to the union area between them. A visual explanation of the IoU is illustrated in Figure 5.6. In this figure, the ground truth bounding box is represented in black and the predicted bounding box in red.

$$\text{IoU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} \quad (5.1)$$

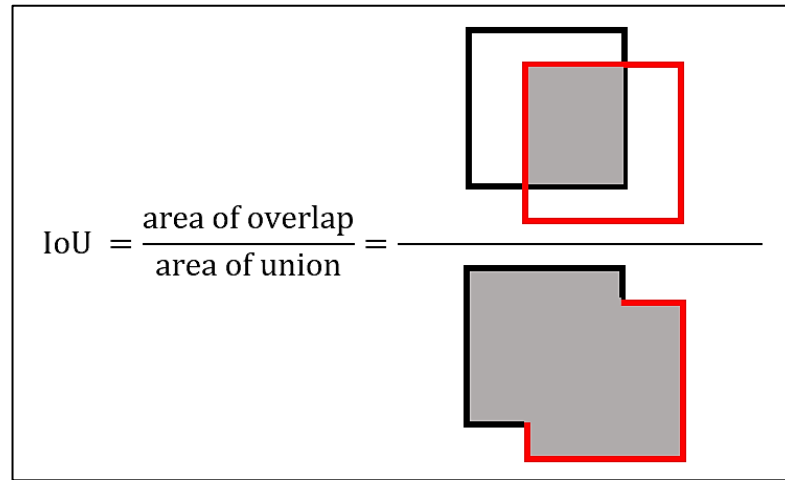


Figure 5.6: Visual representation of IoU.

Considering an IoU threshold, whether a predicted bounding box belongs to the desired object class or to the background can be determined. In general, the IoU threshold is considered to be 0.5, and predicted boxes with an IoU value below 0.5 for a given class are considered false positive (FP). Besides, predicted boxes with an IoU value above 0.5 are considered true positive (TP). The number of ground truth boxes

that do not match any predicted boxes ($\text{IoU} < 0.5$) indicates the false negative (FN). Precision and recall metrics are calculated using TP, FP, and FN. The precision is the capability of the model to identify only relevant objects. On the other hand, recall is the ability of the model to find all relevant cases (all ground truth bounding boxes). These metrics are calculated by equations 5.2 and 5.3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5.2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5.3)$$

The average precision (AP) summarizes the shape of the precision/recall curve, and it is explained as the mean precision at a set of eleven equally spaced recall levels $[0, 0.1, \dots, 1]$. The average recall (AR) is the maximum recall averaged over all $\text{IoU} \in [0.5, 1.0]$. The AP is calculated by equation 5.4, where $p_{\text{interp}}(r)$ represents the interpolated precision at the maximum precision, and $p(\tilde{r})$ represents the measured precision at recall $\tilde{r}$ [Everingham et al., 2010].

$$\text{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} p_{\text{interp}}(r) \quad (5.4)$$

$$p_{\text{interp}}(r) = \max_{\tilde{r}: \tilde{r} \geq r} p(\tilde{r}) \quad (5.5)$$

In COCO metrics, six different AP and AR metrics are defined to analyze the performance of an object detector. AP and AR are calculated by averaging over 10 different IoU values from 0.50 to 0.95, with a stride of 0.05. Traditionally, AP is called mean average precision (mAP). $\text{AP}_{0.5}$ and $\text{AP}_{0.75}$ indicate that the IoU threshold is set as 0.50 and 0.75 in the AP measurement, respectively. AP_S and AR_S are calculated for small objects covering areas smaller than 32^2 pixels. AP_M and AR_M are calculated for medium objects covering areas between 32^2 and 96^2 pixels, and AP_L and AR_L for large objects covering areas bigger than 96^2 pixels. AR_1 , AR_{10} , and AR_{100} are the maximum recalls averaged over all categories and IoUs, given 1, 10, and 100 detections per image, respectively. All these 12 COCO metrics are given in Table 5.4 [Lin et al., 2014].

Table 5.4: MS COCO Dataset Object Detection Evaluation Metrics.

Average Precision (AP)	
AP	AP at IoU=.50:.05:.95
AP _{0.5}	AP at IoU=.50
AP _{0.75}	AP at IoU=.75
AP Across Scales:	
AP _S	AP for small objects: area < 32 ²
AP _M	AP for medium objects: 32 ² < area < 96 ²
AP _L	AP for large objects: area > 96 ²
Average Recall (AR)	
AR ₁	AR given 1 detection per image
AR ₁₀	AR given 10 detections per image
AR ₁₀₀	AR given 100 detections per image
AR Across Scales:	
AR _S	AR for small objects: area < 32 ²
AR _M	AR for medium objects: 32 ² < area < 96 ²
AR _L	AP for large objects: area > 96 ²

5.5. Training and Testing Results of Deep Learning Models

All models used in this study were trained with the hyperparameters presented in Table 5.3 and on the GeForce NVIDIA GeForce 3060 RTX GPU with the constructed ship dataset. All models were trained for 100,000 iterations. Performance evaluations were made on the testing dataset by using the weights of the models saved at every 1,000 iterations. The training process of the models took about 28, 17, 14, and 34 hours for the Faster R-CNN, SSD, EfficientDet, and CenterNet models, respectively. EfficientDet model training took the shortest time and CenterNet model training took the longest time. In order to avoid underfitting and overfitting of the models, the training and testing processes were monitored on TensorBoard. In the subsections, the training and testing phases of all models were examined in detail. Performance analysis was made by evaluating loss curves and accuracy graphs.

5.5.1. Faster R-CNN

The loss graphs of the Faster R-CNN model are illustrated in Figure 5.7.

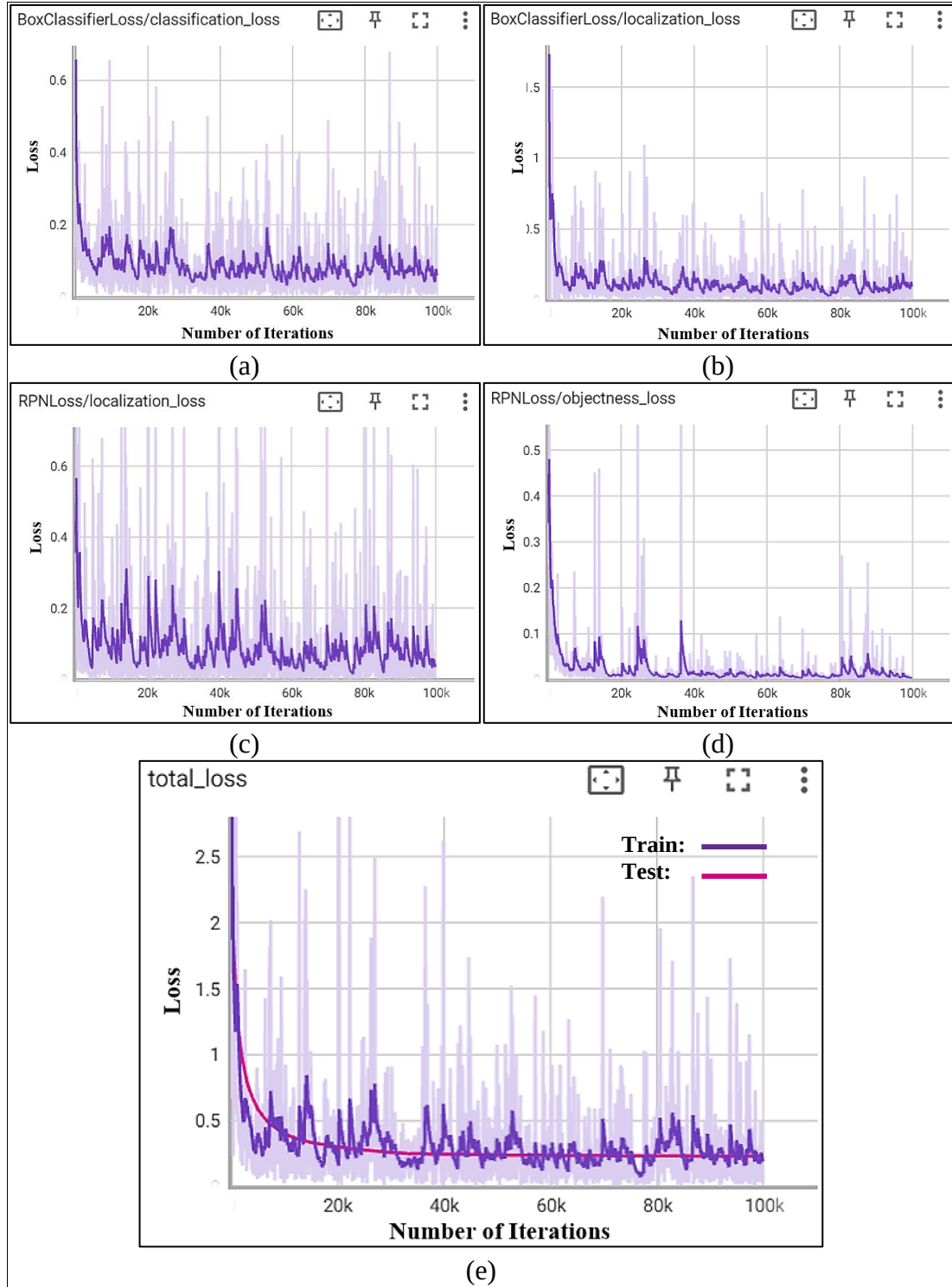


Figure 5.7: Loss curves of ResNet-101 V1 based Faster R-CNN. (a) Box classifier classification loss, (b) Box classifier localization loss, (c) RPN localization loss, (d) RPN objectness loss, (e) Total training and testing loss.

There are four types of losses generated during the training process of the Faster R-CNN model. These are box classifier classification loss, box classifier localization loss, RPN localization loss, and RPN objectness loss. The sum of these four losses represents the total loss. The loss values at the beginning of the training decreased as the number of iterations increased and converged after 5,000 iterations. Box classifier classification loss, box classifier localization loss, RPN localization loss, and RPN objectness loss during training fluctuated at 0.06, 0.07, 0.06, and 0.01 values, respectively, and they did not show a further decreasing trend. The total training loss value fluctuated around 0.2. Besides, the total training and testing losses of the model demonstrated a similar trend. In other words, as the training loss decreased, the testing loss also decreased, and this represented that the overfitting of the model was prevented. In fact, when model training was performed with higher learning rates, the model was overfitting. In other words, while the training loss was decreasing, the test loss started to increase after a certain iteration. This ideal training effect was achieved with the learning rate adjustments made.

5.5.2. SSD

The loss graphs of the SSD model are depicted in Figure 5.8. There are two types of losses produced in the training stage of the SSD model, namely classification loss and localization loss. The sum of these two losses represents the total loss. All loss values at the beginning of the training decreased as the number of iterations increased and converged after 30,000 iterations. Classification loss and localization loss values fluctuated at 0.08 and 0.04 values, respectively, and they did not show a further decreasing trend until the end of the training. The total training loss fluctuated around 0.25 and the total testing loss fluctuated around 0.55. Similar to the training loss, the testing loss also decreased as the number of iterations increased. It was observed that the model was underfitting at lower learning rates and overfitting at higher learning rates, and the training process of the model was completed by adjusting the optimum learning rate base of 0.0004 with the cosine decay with warm up policy. Actually, the fact that these models were pre-trained enabled them to learn even with the small learning rates.

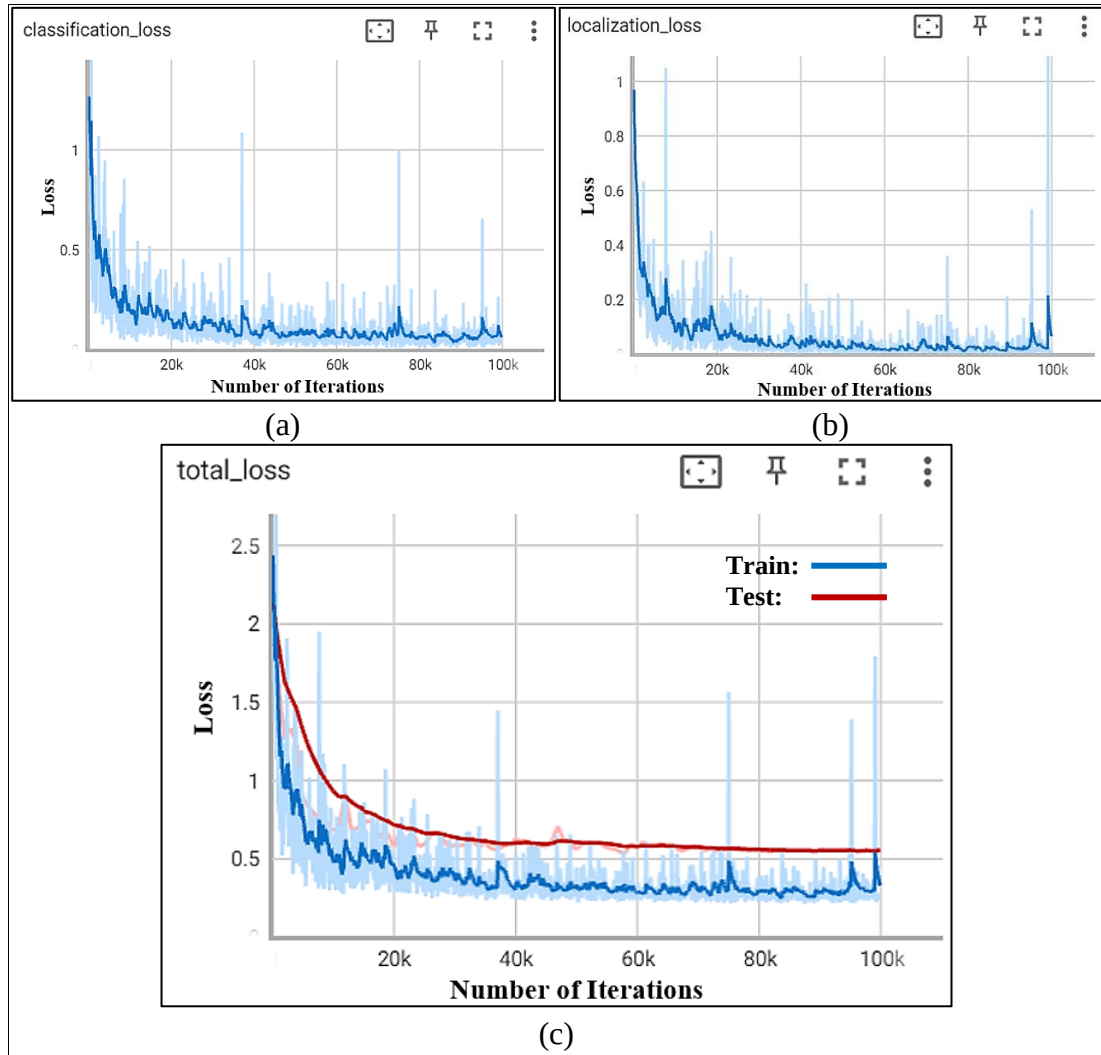


Figure 5.8: Loss curves of ResNet-50 V1 FPN based SSD. (a) Classification loss, (b) Localization loss, (c) Total training and testing loss.

5.5.3. EfficientDet

The loss graphs of the EfficientDet model are demonstrated in Figure 5.9. As in the SSD model, there are two types of losses produced in the training stage of the EfficientDet model, namely classification loss and localization loss. The sum of these mentioned two losses represents the total loss. In this model, the lost values at the beginning of the training decreased as the number of iterations increased, which means that the model was learning during the training stage. The loss values converged after 20,000 iterations. Classification loss and localization loss values fluctuated at 0.15 and 0.005 values, respectively. Total training and testing losses both fluctuated around 0.2. As in other models, similar to the training loss, the testing loss also decreased as the

number of iterations increased, and the overfitting of the model was prevented by convenient learning rate optimization.

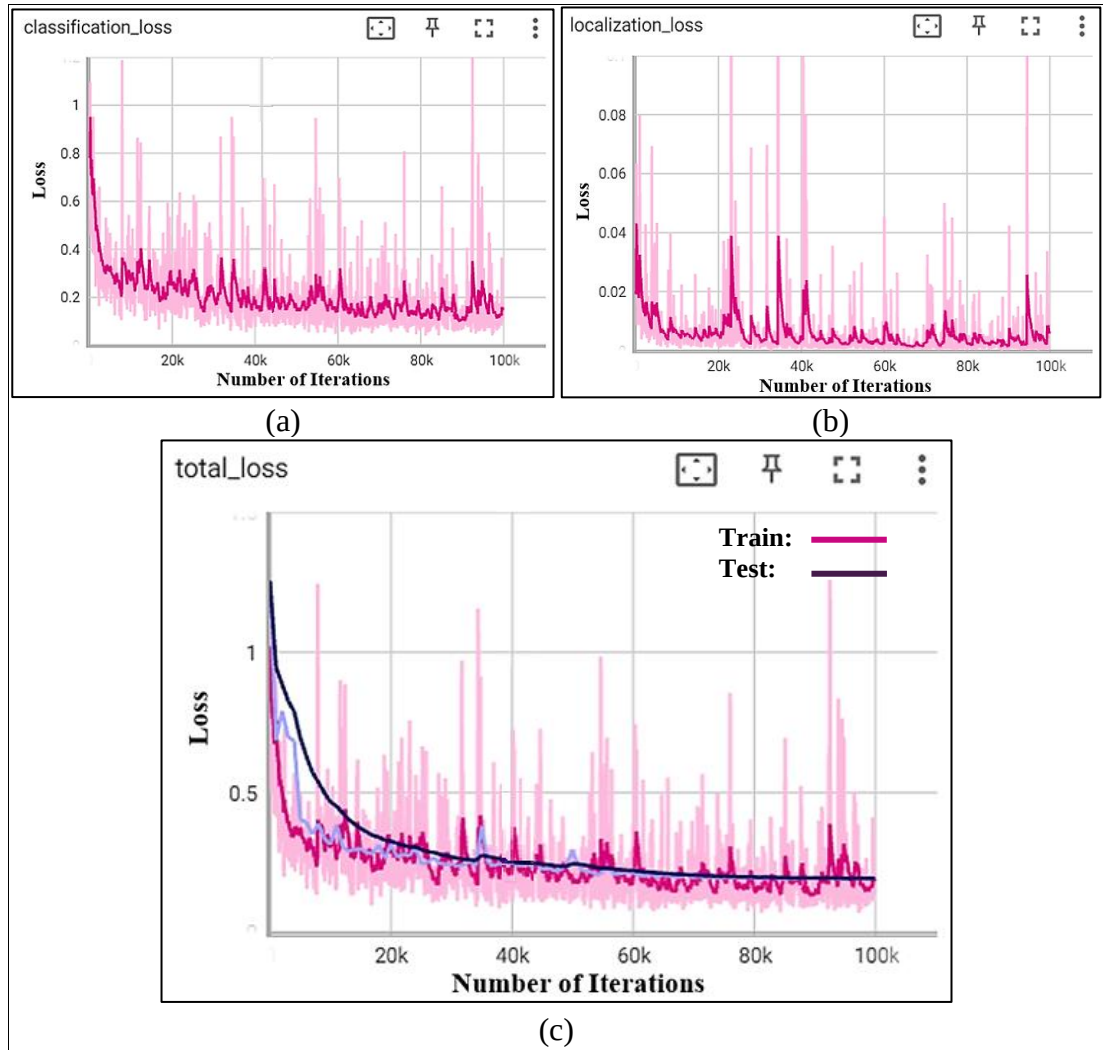


Figure 5.9: Loss curves of EfficientNet-B1-BiFPN based EfficientDet. (a) Classification loss, (b) Localization loss, (c) Total training and testing loss.

5.5.4. CenterNet

The loss graphs of the CenterNet model are shown in Figure 5.10. There are three types of losses generated during the training process of the CenterNet model: box offset loss, box scale loss, and object center loss. The sum of these three losses represents the total loss. In this model, similar to the EfficientDet model, the loss values converged after 20,000 iterations. Box offset loss, box scale loss, and object center loss values fluctuated at 0.25, 0.1, and 0.15 values, respectively. The total training and testing losses fluctuated around 0.50 and 0.75, respectively. In this model,

similar to the training loss, the testing loss also decreased and became stable as the number of iterations increased. Also, overfitting of the model was avoided by employing the step decay policy learning rate.

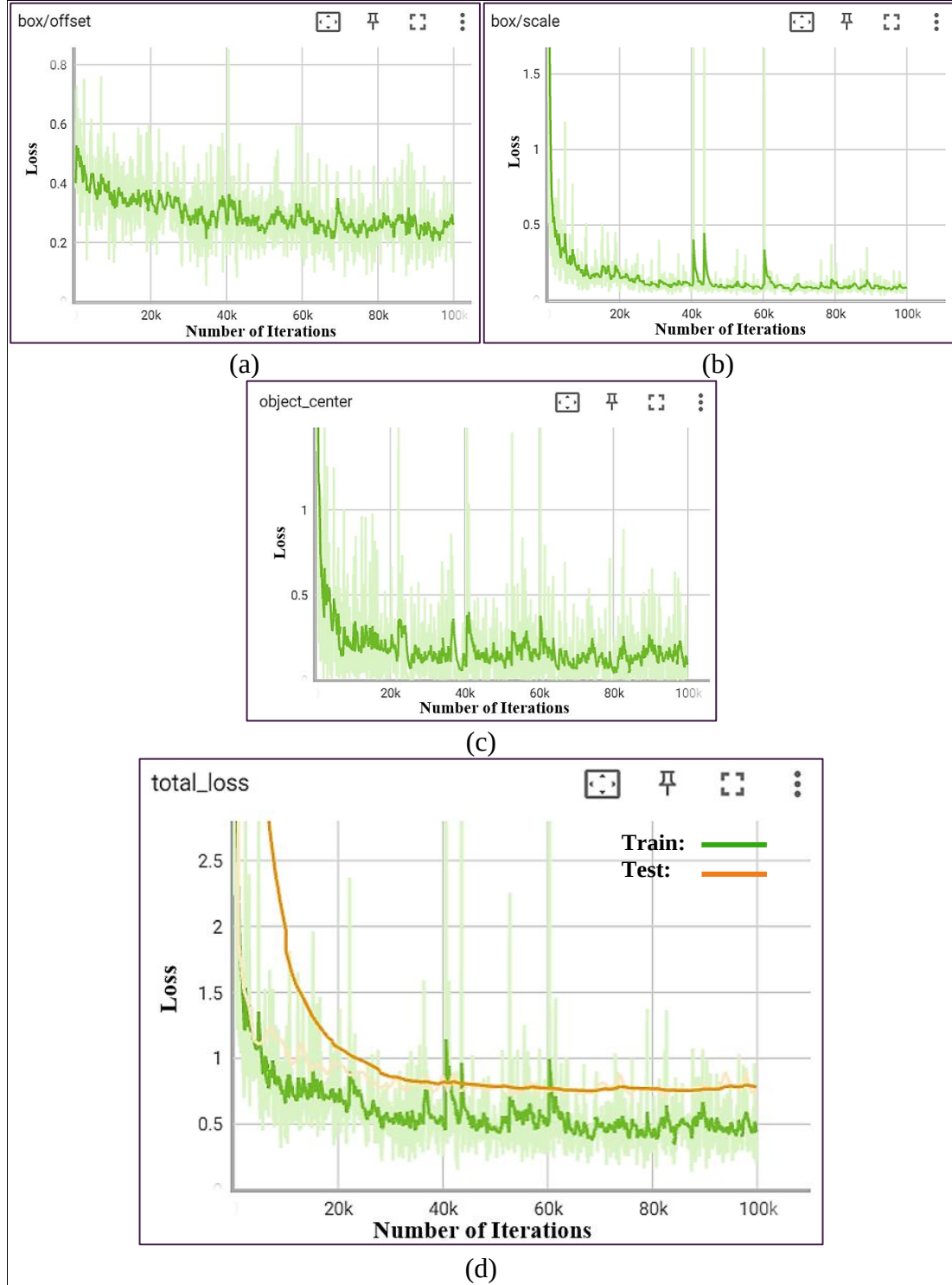


Figure 5.10: Loss curves of Hourglass-104 based CenterNet. (a) Box offset loss, (b) Box scale loss, (c) Object center loss, (d) Total training and testing loss.

5.5.5. Performance Comparison of Deep Learning Models

In this section, the performance analyses were made by examining the COCO metrics obtained after the testing process of all models. The accuracy values of the models started with low values and increased as the number of iterations increased and became stable after a certain iteration.

AP (called mAP) graphs for all models are illustrated in Figure 5.11. The highest increase in the mAP graph was observed in the CenterNet model, followed by the Faster R-CNN model. The lowest increase was seen in the EfficientDet model, and the SSD model showed a slightly greater increase from it. On the other hand, the highest increase in the mAP graph for large objects was observed in the CenterNet and then the Faster R-CNN model. However, this time, the lowest increase was seen in the SSD model. Although the highest increase in the mAP graph for medium objects was achieved in the CenterNet model, it was followed by the EfficientDet model. These were followed by the Faster R-CNN and SSD models with very slight differences. Besides, the highest increase in the mAP graph for small objects was obtained in the CenterNet model and followed by the EfficientDet model. The lowest increase was observed in the Faster R-CNN model. In the graph of mAP at 0.50 IoU, the general metric used for accuracy, the highest increase was achieved in the CenterNet, while the lowest increase was obtained in the SSD model. Finally, the highest value in the mAP at 0.75 IoU graph was seen in the CenterNet model, followed by the SSD model. The Faster R-CNN model came after them with a slight difference. Consequently, the highest increases in all AP graphs were observed in the CenterNet model.

AR graphs of all models are depicted in Figure 5.12. In the AR graph given 1 detection per image, the highest increase was observed in the CenterNet, followed by Faster R-CNN. The lowest increase was achieved in the SSD model. A similar increasing trend was seen in the AR graph, which was given 10 detections per image. However, in the AR graph given 100 detections per image, the highest increase was seen in the CenterNet model, followed by the Faster R-CNN model, while the lowest increase was seen in the EfficientDet model. When AR graphs were examined according to object scales, the highest increase in AR graph for large objects was seen in the CenterNet, Faster R-CNN, EfficientDet, and SSD models, respectively. Besides, the highest increase in the AR graph for medium objects was obtained in the CenterNet

model, followed by the EfficientDet model. These were followed by SSD and Faster R-CNN models, having close values. In the AR graph for small objects, the highest increase was seen in the CenterNet model, followed by EfficientDet model. The lowest increase was observed in the Faster R-CNN model. Similar to the trend in AP graphs, the highest increases in all AR graphs were achieved in the CenterNet, while the lowest values for small objects were achieved in the Faster R-CNN model.

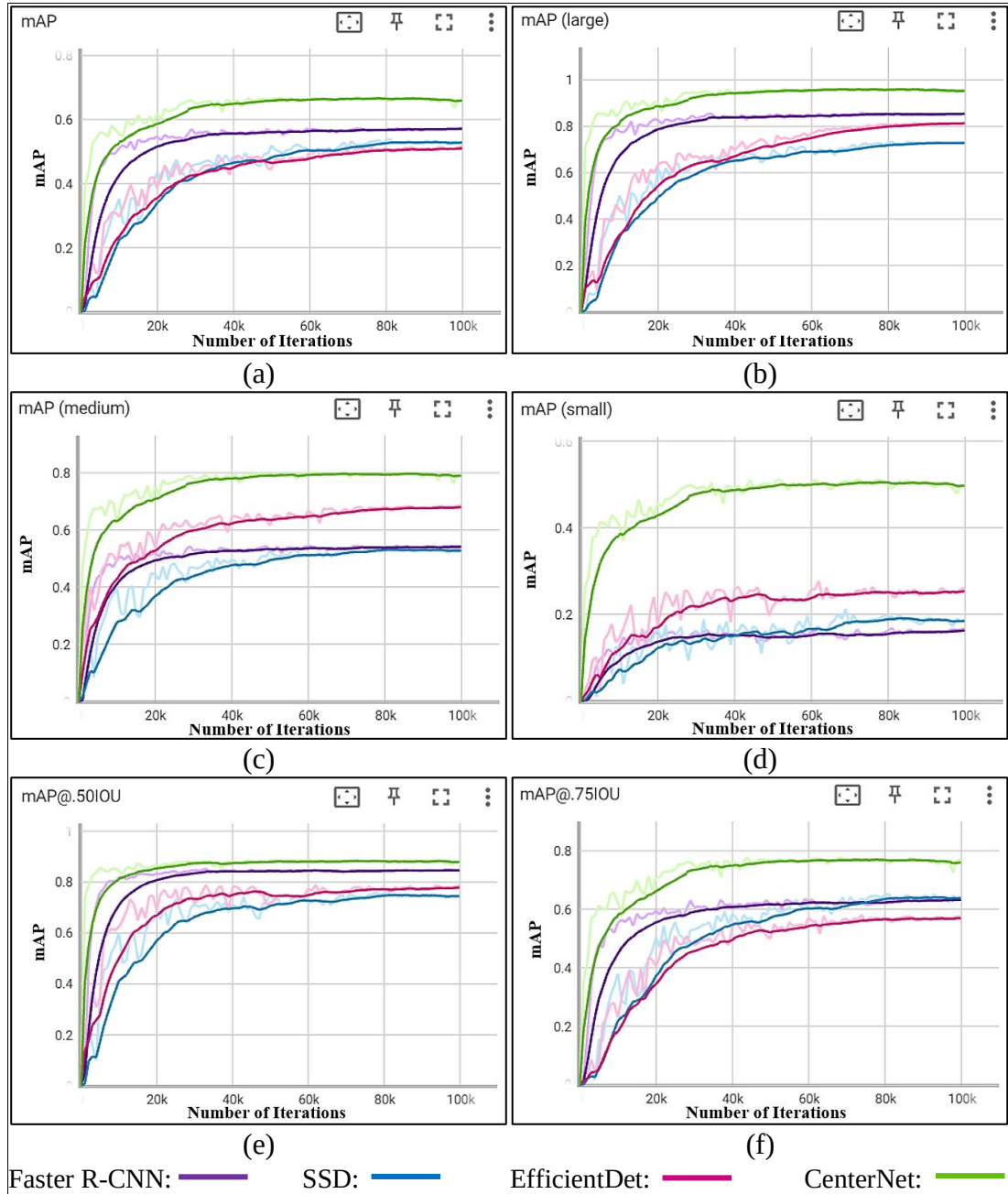


Figure 5.11: Mean Average Precision scores (mAP) of all models. (a) mAP at 0.50:0.95 IoU, (b) mAP for large objects, (c) mAP for medium objects, (d) mAP for small objects, (e) mAP at 0.50 IoU, (f) mAP at 0.75 IoU.

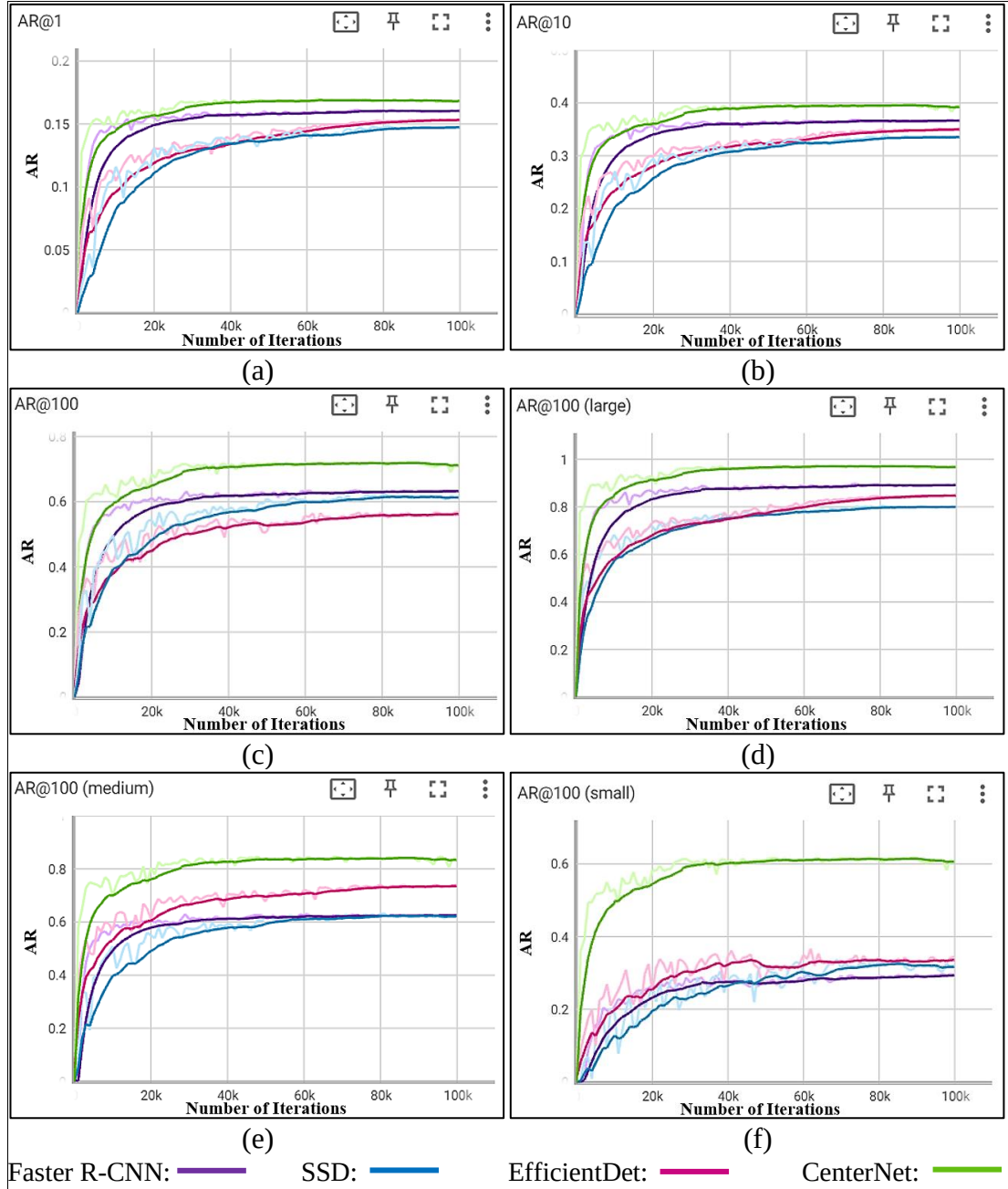


Figure 5.12: Average Recall (AR) scores of all models. (a) AR, given 1 detection per image, (b) AR, given 10 detection per image, (c) AR, given 100 detection per image, (d) AR for large objects, (e) AR for medium objects, (f) AR for small objects.

The MS COCO evaluation metrics obtained for all models in the testing dataset are illustrated in Table 5.5. According to these results, the CenterNet achieved more successful results in all accuracy metrics. It reached 88.80 AP with 0.50 IoU threshold and achieved robust performance with more than 95% AP and AR values over large objects. If a comparison is made between the rest of the other models, the Faster R-CNN had the highest performance, except for the $AP_{0.75}$, AP_M , AP_S , AR_M , and AR_S .

metrics. Considering the 0.75 IoU threshold, the SSD model outperformed the Faster R-CNN and EfficientDet models in terms of AP. Moreover, the EfficientDet model was more successful than the SSD and Faster R-CNN models in medium and small ship objects. SSD model was superior to Faster R-CNN for AP_S and AR_S . Given this phenomenon, it can be said that the SSD model was more robust than the Faster R-CNN when considering small ship objects. These results vary depending on many parameters such as the architecture of the models and the backbone network utilized. For example, in the SSD model thanks to the usage of the ResNet50-FPN backbone obtained by embedding the Feature Pyramid Network (FPN) into ResNet50, it was successful in detecting small objects that were difficult. This relies on FPN's ability to produce high-quality multi-scale feature maps for accuracy and speed. If a whole performance evaluation was made, it can be said that the CenterNet achieved the most successful results for all metrics among all models.

Table 5.5: Performance comparison of all models with MS COCO metrics on the testing dataset.

	Faster R-CNN	SSD	EfficientDet	CenterNet
AP	57.34	53.89	51.65	66.97
AP_{0.5}	85.30	75.98	78.91	88.80
AP_{0.75}	63.57	65.23	57.62	77.82
AP_S	16.82	21.17	26.39	51.24
AP_M	54.54	54.09	68.50	80.22
AP_L	85.88	73.33	81.72	96.44
AR₁	16.12	14.84	15.39	17.00
AR₁₀	36.75	33.89	35.24	39.60
AR₁₀₀	63.49	62.31	56.89	72.15
AR_S	29.79	33.74	36.70	61.70
AR_M	63.06	63.29	74.05	84.75
AR_L	89.72	80.59	85.12	97.50

In order to better analyze the performance difference among all models, the graphical comparison in terms of COCO metrics is also illustrated in Figure 5.13.

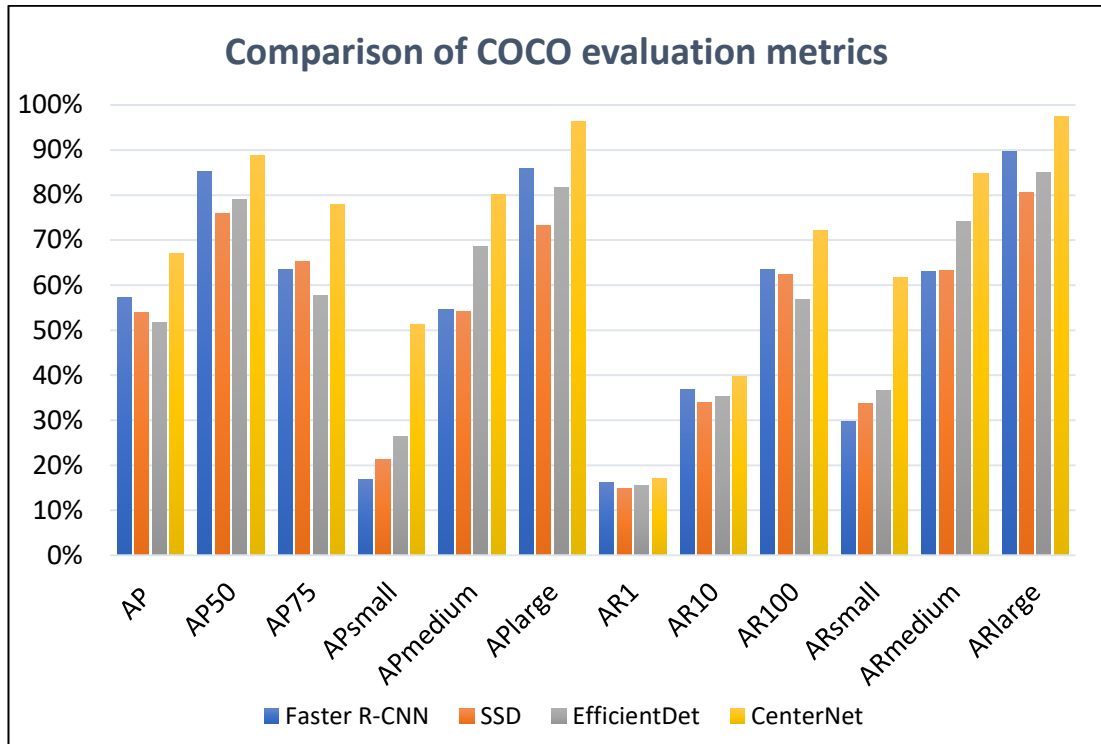


Figure 5.13: Graphical comparison of all models with COCO metrics.

In all object detection applications, processing time is a crucial factor as well as accuracy. In this context, along with the accuracy of the models, the processing times should also be analyzed. According to the produced results, CenterNet was better than other models in terms of AP and AR, but more time was required for CenterNet. The Faster R-CNN, the model that gave the second-best result in terms of AP with 0.50 threshold (with a difference of 3.5 from CenterNet), took 6 hours less than CenterNet to train. The Faster R-CNN was followed by the EfficientDet, which achieved an AP value of 78.91% with 0.50 threshold. This model required 11 hours less training time than the Faster R-CNN. The SSD, which produced the worst result in terms of the same metric (75.98%), was trained half as much as time than the Faster R-CNN.

5.6. Visual Evaluation of Results

Representative ship detection results of all models on some test images were visually investigated in this section.



Figure 5.14: Qualitative ship detection results of all models on test image 1. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

Test image 1 in Figure 5.14 includes a ship on shore with a complex background. This ship in the image was detected by all deep learning models. However, if the bounding box produced by the models for the ship was examined in detail, it was observed that the most appropriate bounding box was drawn by the Faster R-CNN, CenterNet, and SSD models. On the other hand, the EfficientDet model could not accurately delimit the boundaries of the ship. That is, the bounding box was not fitted to the object. Besides, all models recognized the ship object with a prediction score of more than 95%.

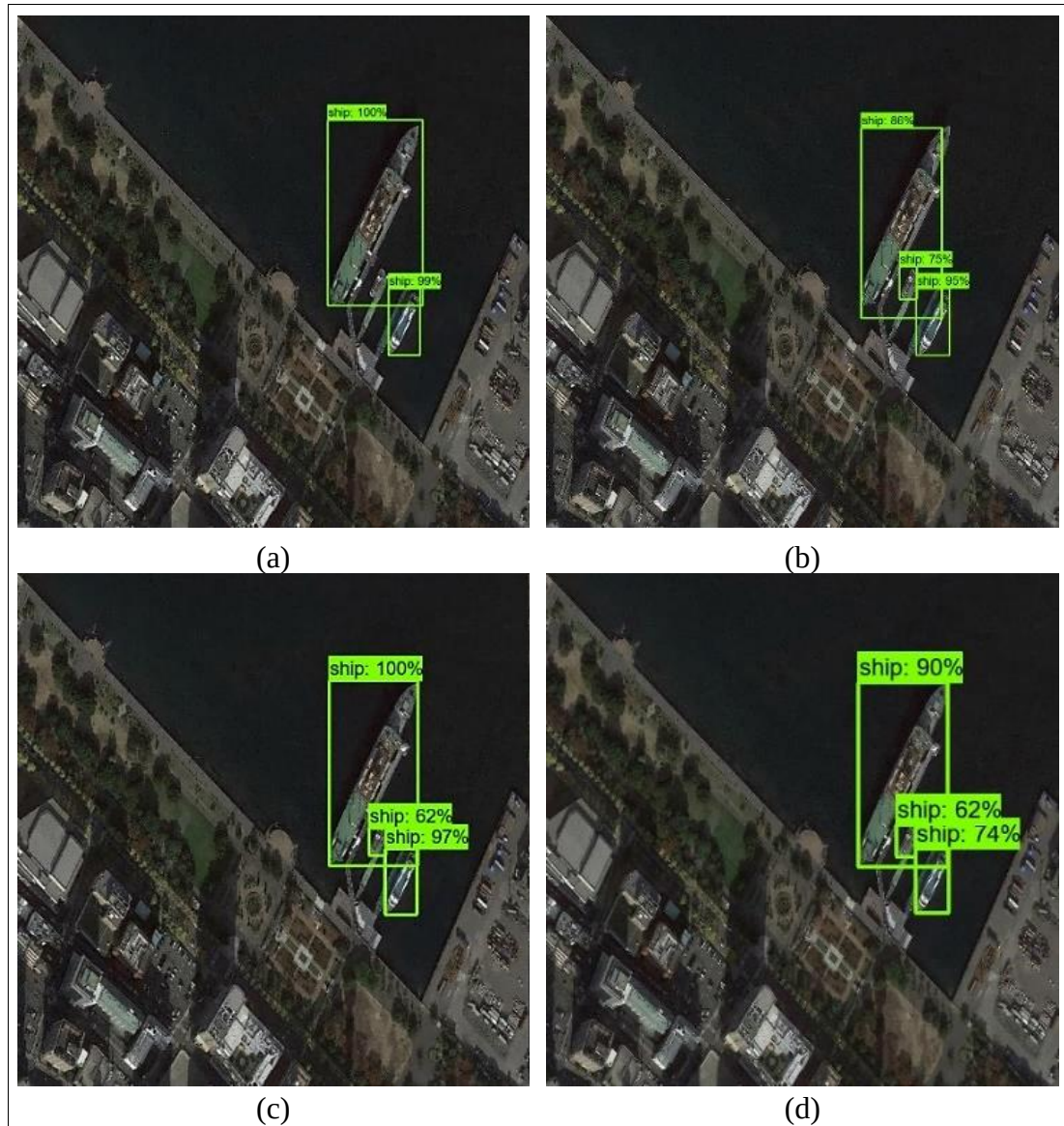


Figure 5.15: Qualitative ship detection results of all models on test image 2. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

In test image 2 given in Figure 5.15, there were three ship objects very close to each other on the shore. Similar to test image 1, this image also had a complex background that included various objects such as buildings, trees, and cars on the shore. All models except for the Faster R-CNN recognized these three ships. Faster R-CNN missed the smallest ship, meaning it produced a false negative. Despite this, it was able to detect the other two ships with a high prediction score. On the other hand, although the SSD model recognized all ships, it could not properly predict the bounding box of the large ship.

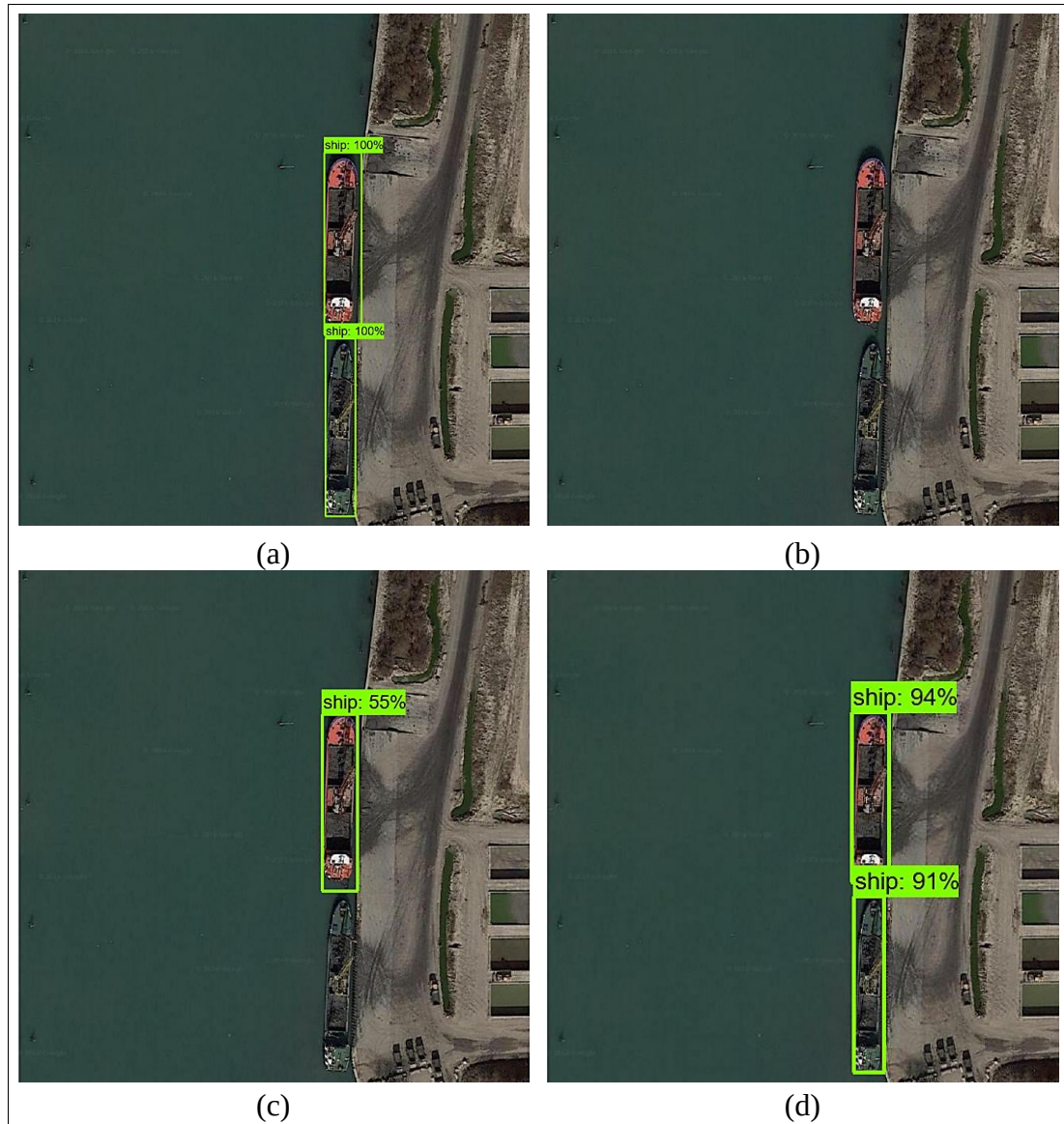


Figure 5.16: Qualitative ship detection results of all models on test image 3. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

Another test image, given in Figure 5.16, included two ship objects of similar sizes, distinct colors, and shapes, located one after the other, on the shore. Faster R-CNN and CenterNet models successfully detected and accurately localized these two ships. The prediction scores of both models are more than 90%. The EfficientDet model recognized one ship with a low prediction score (55%), but missed the other ship, producing a false negative. In addition, the localization of this detected ship was not satisfactory. Worse still, the SSD model had the worst result, missing both ship targets. Thus, the SSD model produced two false negatives.

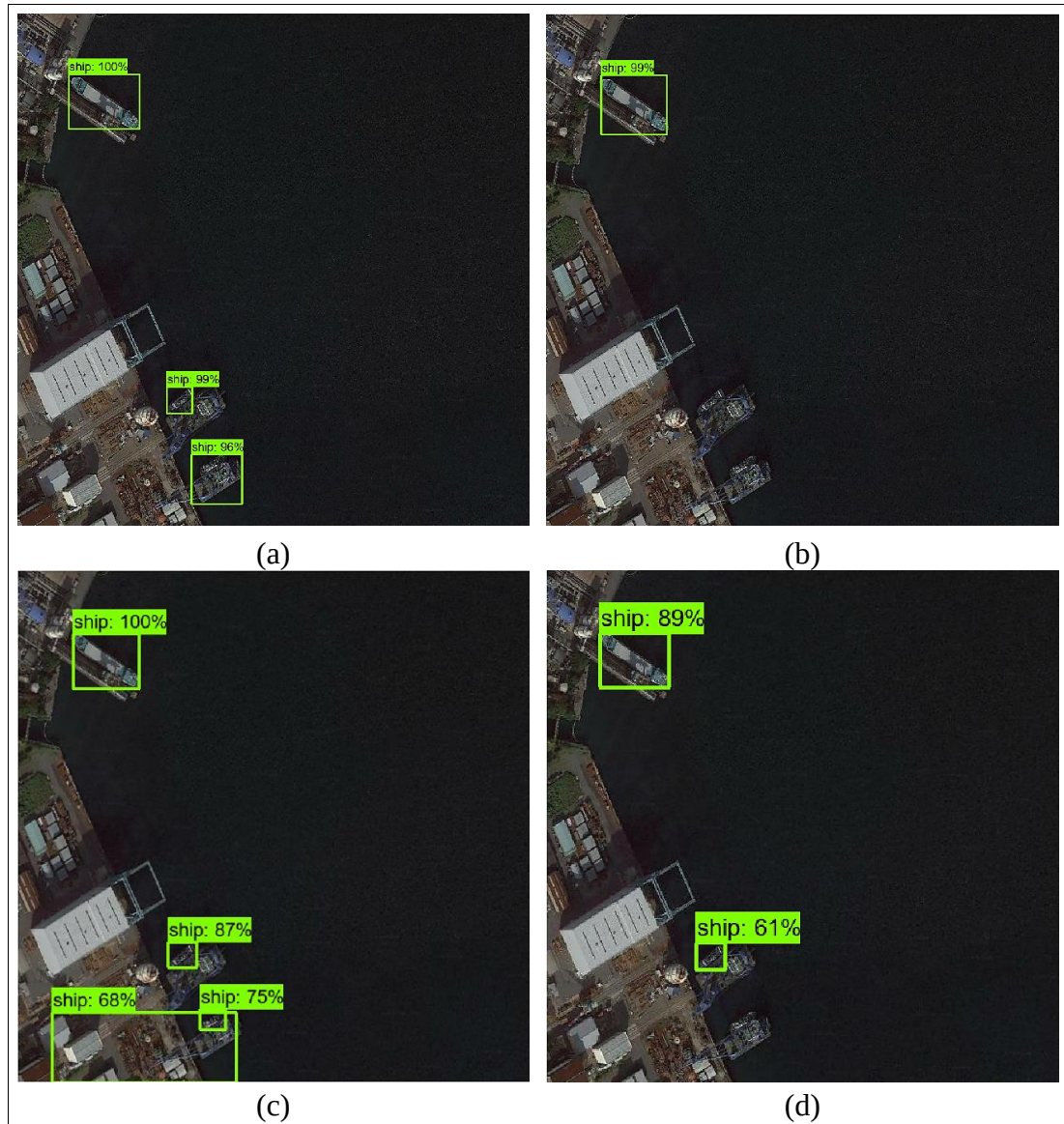


Figure 5.17: Qualitative ship detection results of all models on test image 4. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

In the test image in Figure 5.17, there were three ships on the shore and one small ship attached to each two of these ships. In this image, the Faster R-CNN model recognized an independent large ship, a small ship, and detected two adjacent ships together. The SSD model only recognized the large ship, which was independent, and missed the others. The EfficientDet model recognized an independent large ship and two small ships. In addition, it detected a false alarm, which included the ship to which the small ship was adjacent, along with objects on land. That is, it produced a false positive. On the other hand, the CenterNet recognized an independent large ship and a small ship.

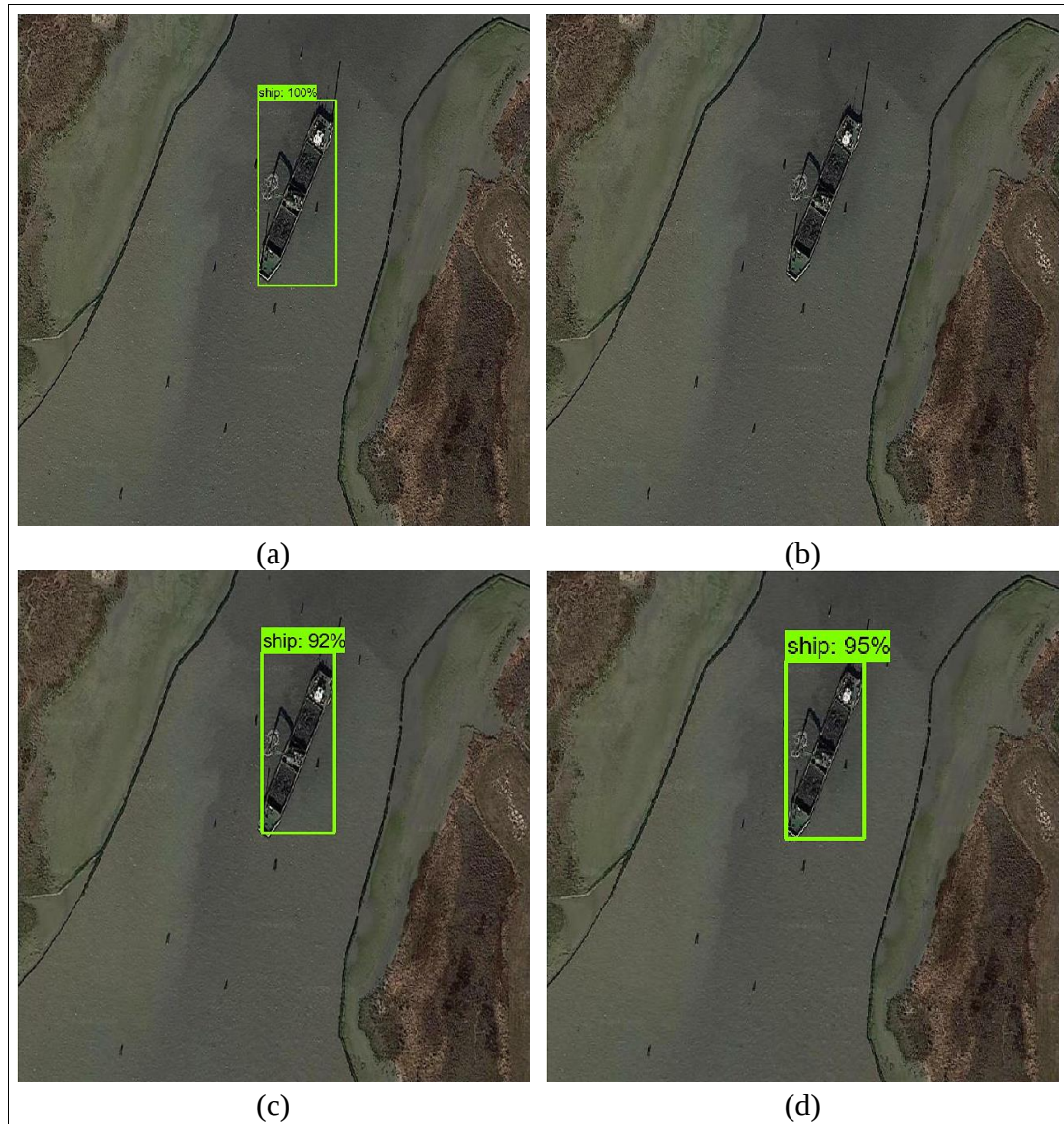


Figure 5.18: Qualitative ship detection results of all models on test image 5. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

The final test image given in Figure 5.18 has a ship target with a different background than the other test images. This ship demonstrated different object characteristics compared to the ship targets in other images and appeared in a river. In the testing phase, this ship was recognized by all models except for the SSD model. However, the CenterNet model was the most successful in terms of object localization. When the identification results of Faster R-CNN and EfficientDet models are examined closely, it was seen that the bounding box they estimated for the target did not fit the object precisely.

5.7. Generalization Ability Testing

In the last phase of the study, the generalization capability of all trained object detection models on independent satellite images different from the training and testing dataset was evaluated. For this aim, the GÖKTÜRK-1 satellite image taken on 24 June 2021 was utilized. The technical specifications of the GÖKTÜRK-1 satellite were shown in Table 5.6. This image is pan-sharpened (0.5 m spatial resolution), and covers part of the Marmara Sea, along with the coasts of Istanbul and Kocaeli. Besides, the sea mucilage impact was observed in the image, unlike the images used in the training and testing of the models. Image patches of 1024 x 1024 size were cropped from the areas with ship objects from this image to be given to the trained models. The testing results of all models on these images are depicted in Figures 5.19 to 5.28.

Table 5.6: Technical specifications of GÖKTÜRK-1 satellite.

Specifications	GÖKTÜRK-1 Satellite
Launch Date	December 5th, 2016
Orbit Type	Sun synchronous
Orbit Altitude	681 km
Inclination Angle	98,11°
Period	98 min.11 sec.
Spot Size	15 km x 15 km
Swath Width	15 km
Strip Length	780 km / 14.300 km
Spatial Resolution	0,5 m PAN – 2 m RGB
Radiometric Resolution	12-Bit
Positional Accuracy	Horizontal: 10 m (without GCP), 2 m (with GCP) Vertical: 20 m (without GCP), 3 m (with GCP)
Spectral Bands	PAN, RGB, NIR
Number of Orbits per Day	14-15 (14,7)
Revisit Time	2-3 days ($\pm 30^\circ$ incidence) 11 days ($\pm 5^\circ$ incidence)

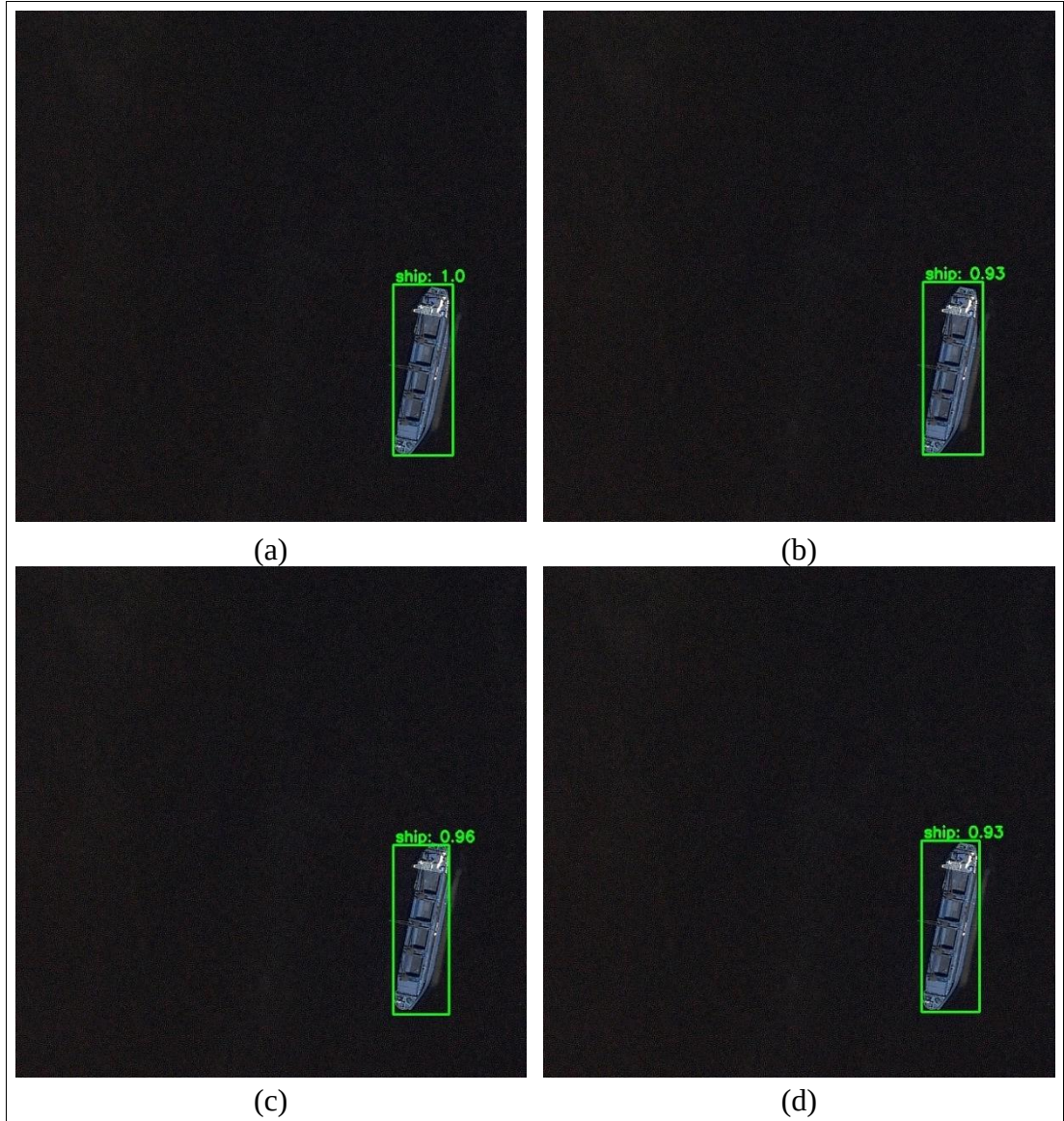


Figure 5.19: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

In the GÖKTÜRK-1 satellite image patch demonstrated in Figure 5.19, there is a large ship in the offshore background. All object detectors were able to successfully recognize and localize the ship in these background conditions. In addition, each model had a prediction score of more than 90% for this ship target.

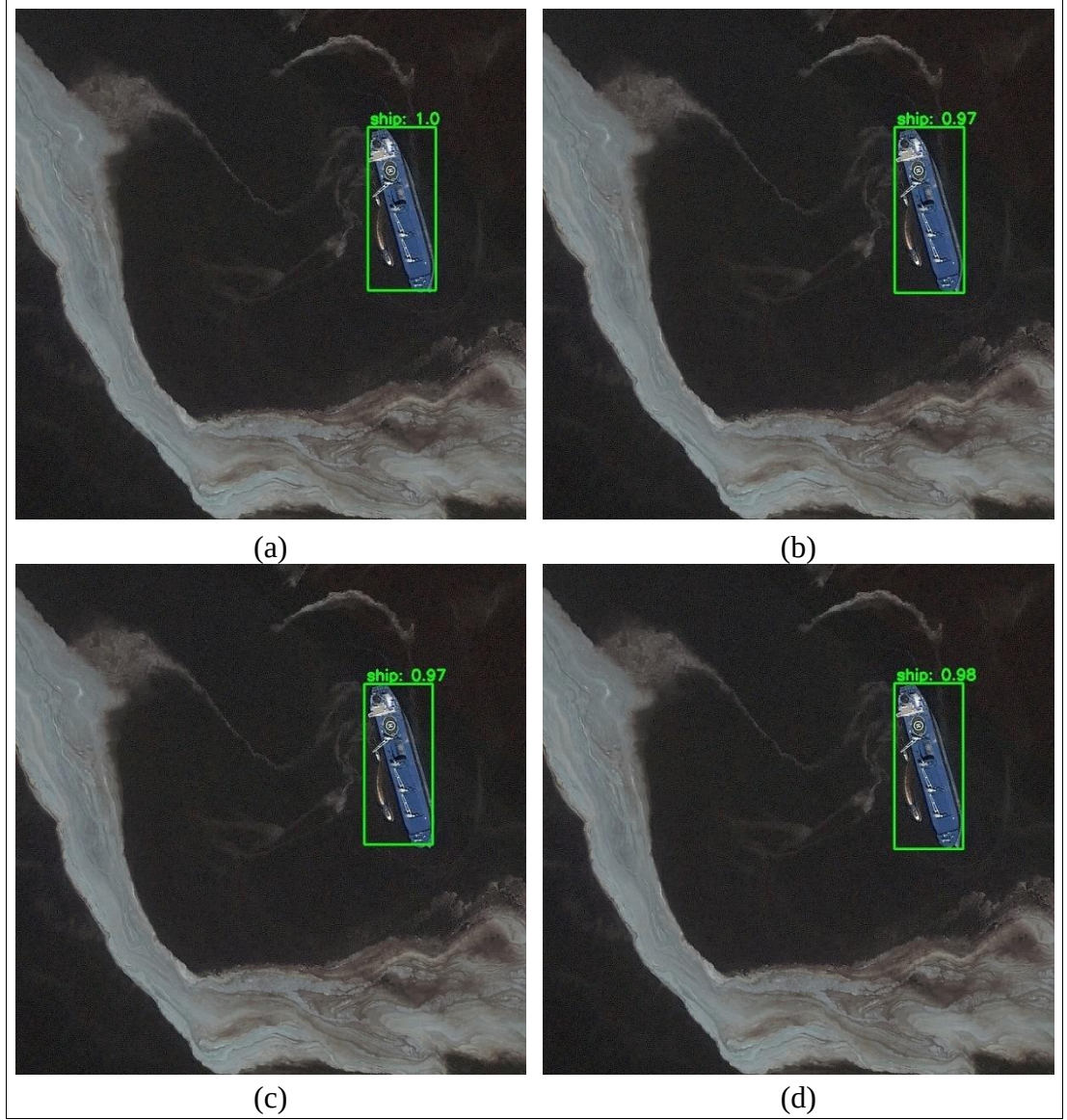


Figure 5.20: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

An example of a ship in offshore and two tiny ships very close to it is observed in Figure 5.20. In addition, the image shows an intense mucilage impact on the sea. All models were also evaluated with these images to examine their detection capabilities under these complex background conditions. As can be seen from Figure 5.20, all object detection models were able to detect one large and two very small ships together. Besides, each model had a prediction score of more than 95% for this detection.

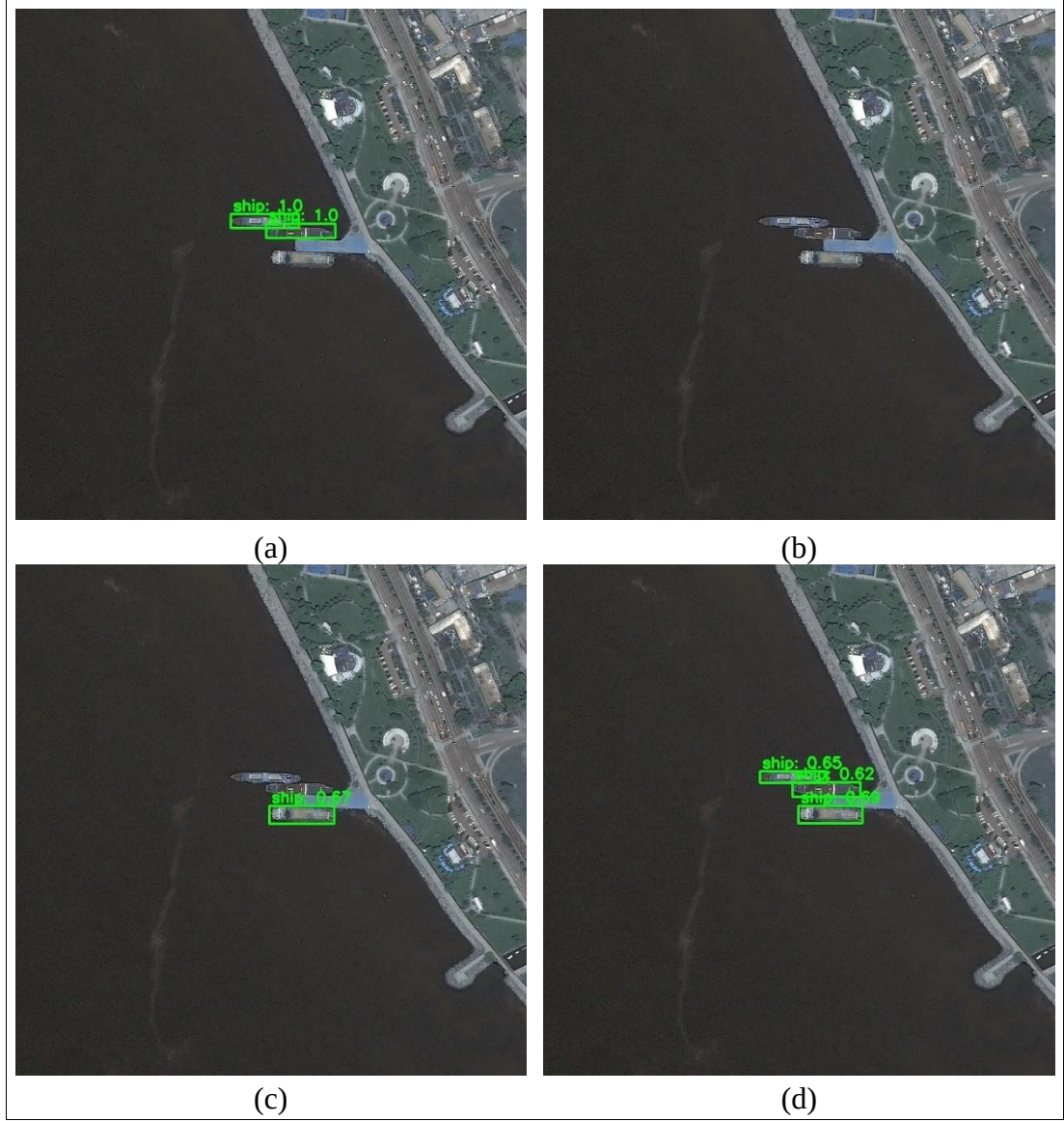


Figure 5.21: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

In the image given in Figure 5.21, there are three ship objects close to each other on the shore. Among all the models, the CenterNet model was able to recognize and localize these three ships. Although the Faster R-CNN model was able to recognize two ships with 100% prediction score, it missed one ship. On the other hand, although the EfficientDet model was able to detect this ship, missed by the Faster R-CNN model, it missed the other two ships. Worse still, the SSD model failed to detect any of these three ships.

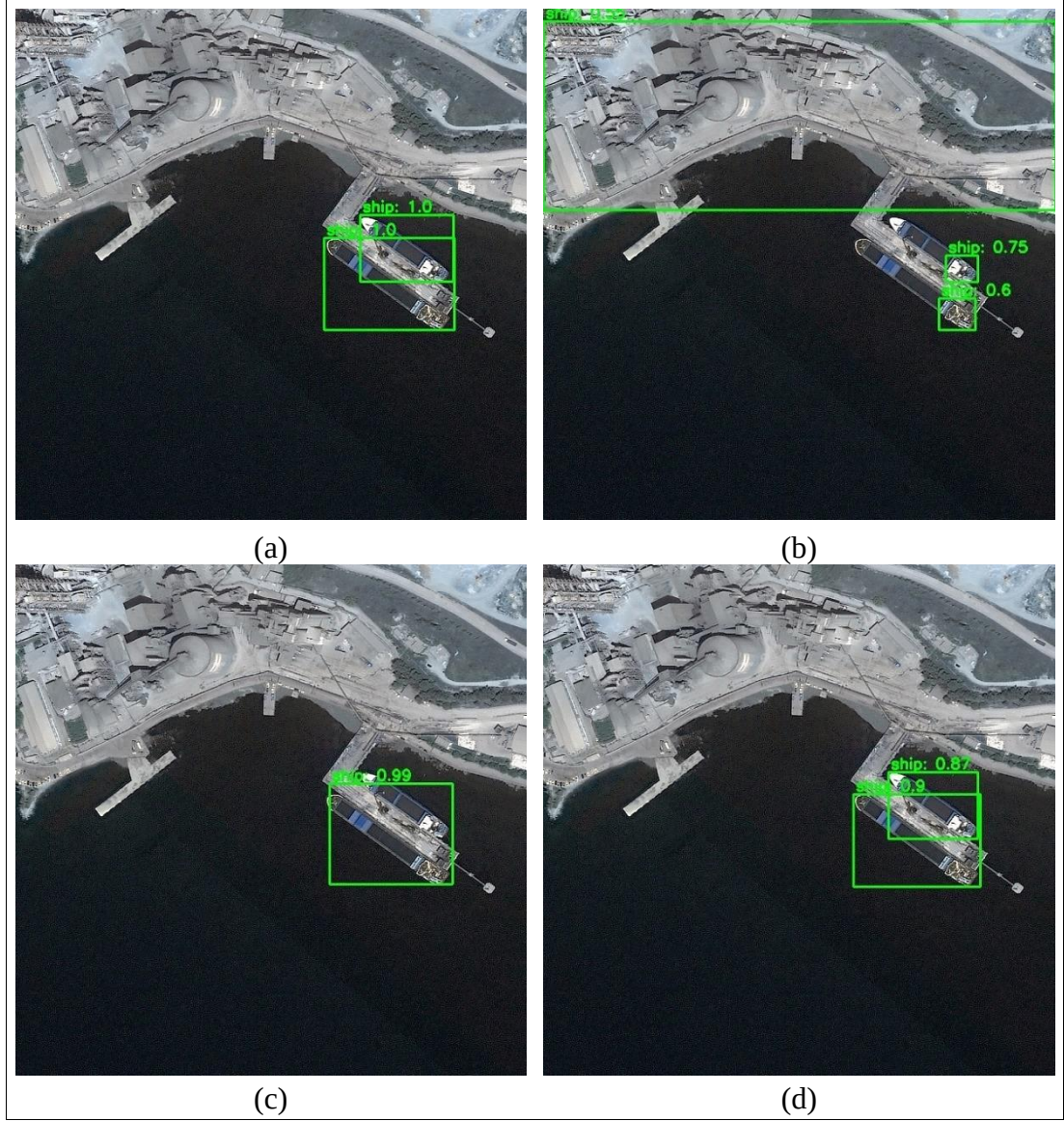


Figure 5.22: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

In the satellite image patch given in Figure 5.22, there are two ship targets on shore. As can be seen from the detection results, the CenterNet and Faster R-CNN models were able to successfully recognize both ships. The EfficientDet model could not detect these two ships separately and detected them together. On the other hand, the SSD model failed to recognize both ships. The model identified the rear part of the ships as ships. It also detected the shore as a ship. As a whole, the SSD model detected three false alarms and produced false positives.

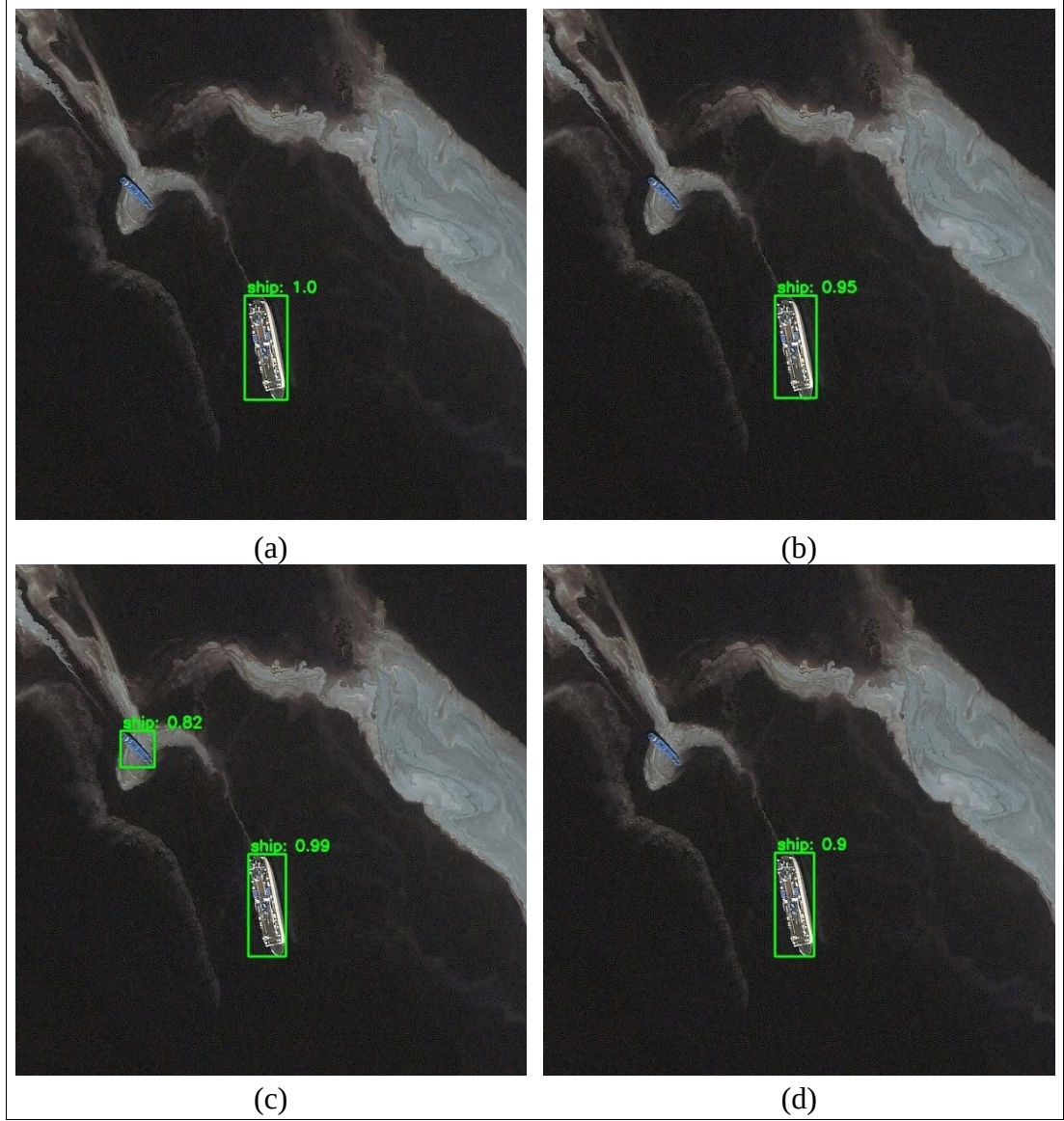


Figure 5.23: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

The image given in Figure 5.23 shows an example of a large and a small ship in offshore. Also, intensive mucilage event is evident throughout the image. Especially the mucilage spread around the small ship is quite high. If the detection results of all models for these two ship objects were examined, it was obvious that all models could detect the large ship with a prediction score of more than 90%. However, considering the small ship, it was observed that only the EfficientDet model successfully recognized and located the ship. Even under this complex background condition, the ship was detectable by this model.

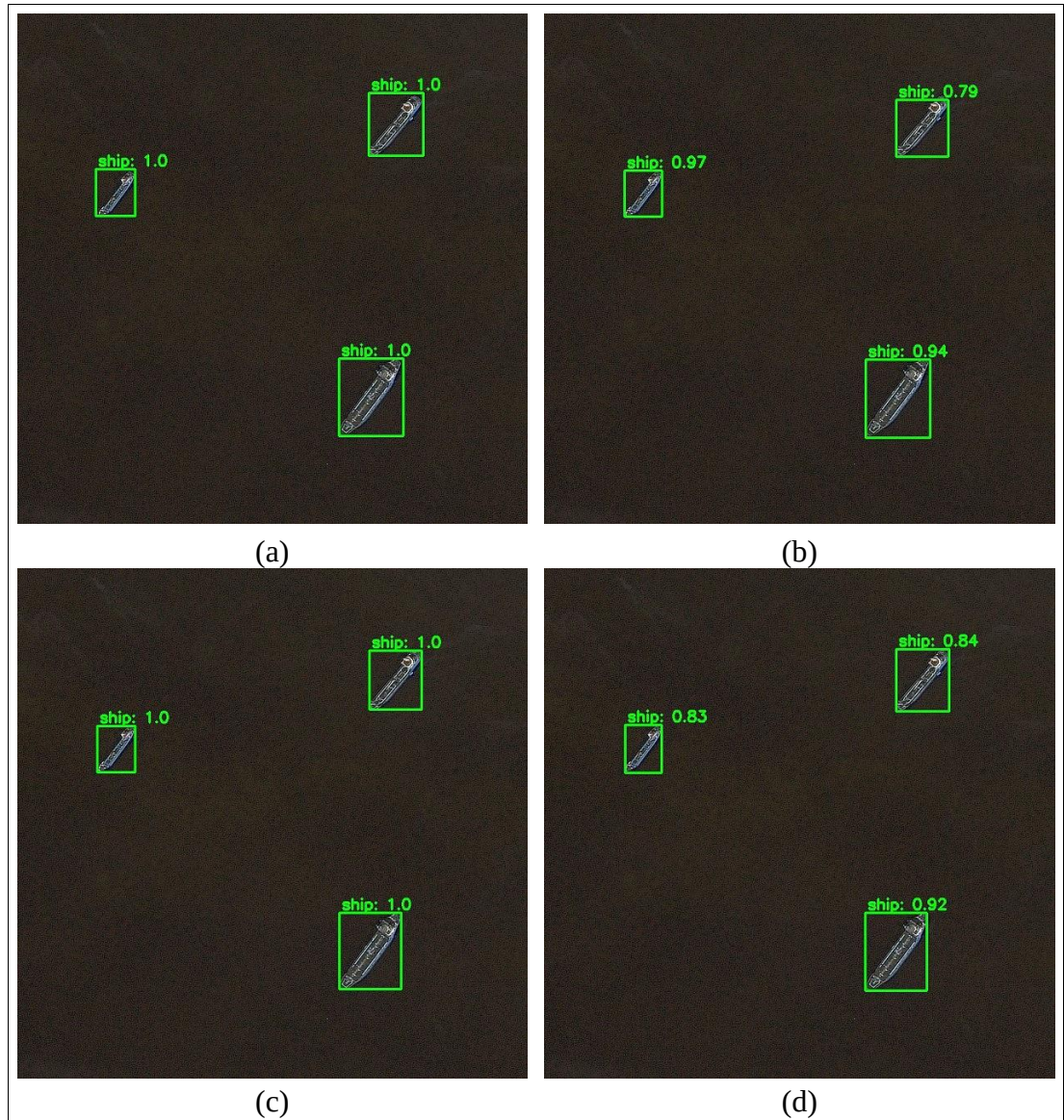


Figure 5.24: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

The image given in Figure 5.24 demonstrates three examples of ships with an offshore background. When the detection results of all models are investigated, it was observed that all models were able to detect these three ships accurately. All of them were also successful in the localization of the ship instances.

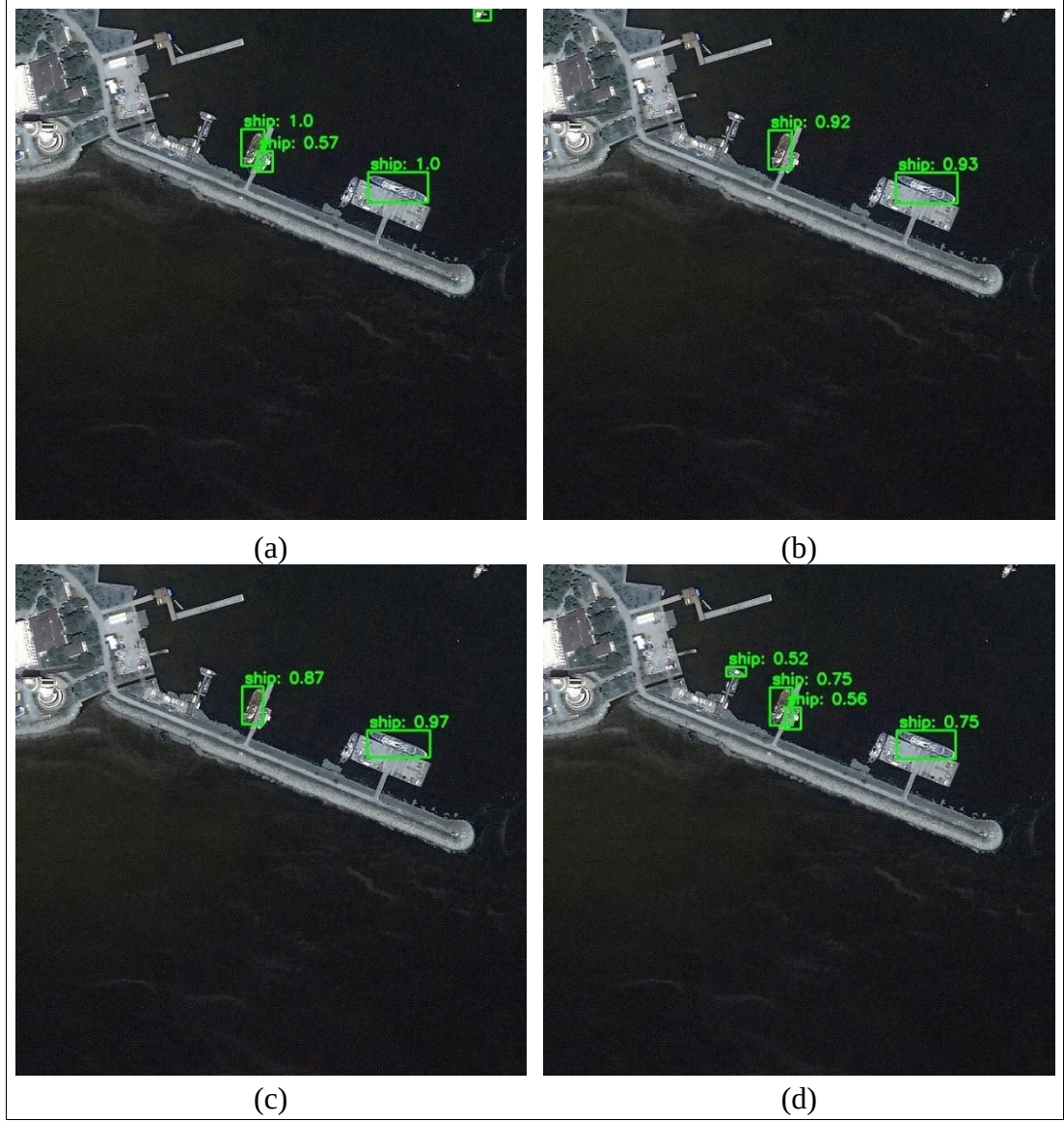


Figure 5.25: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

In another case, there are many large and small ship targets on the shore (Figure 5.25). Among these ship objects, the EfficientDet and SSD models recognized two ships that were larger than the other ships. The Faster R-CNN model detected a small ship in addition to these two ships. It was also able to recognize a small, truncated ship in the upper right part of the image. On the other hand, the CenterNet model detected two smaller ship targets in addition to the two ships detected by the EfficientDet and SSD models.

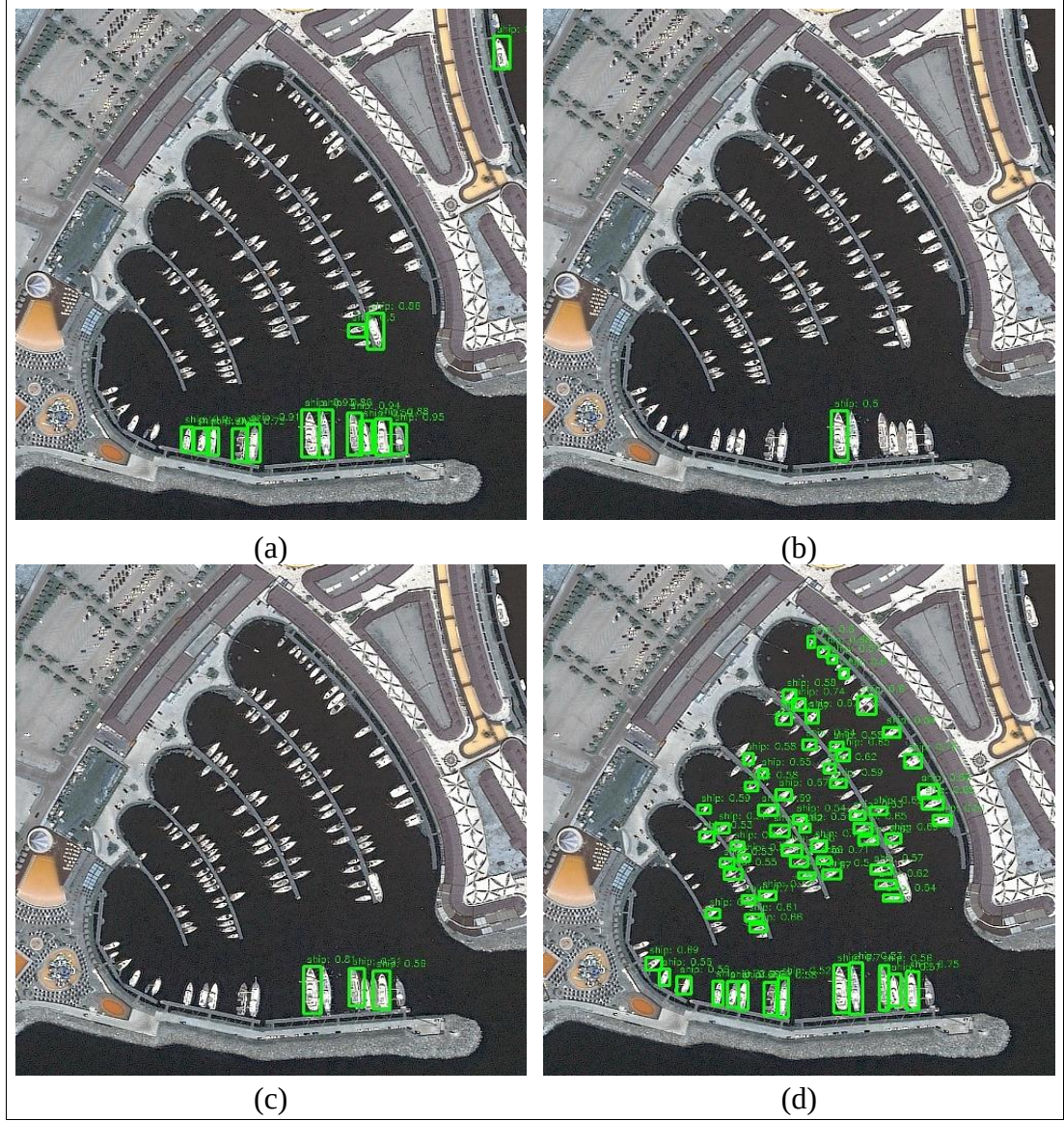


Figure 5.26: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

A GÖKTÜRK-1 satellite image patch of the Tuzla marina is given in Figure 5.26. This image depicts numerous small ship instances located close together. In this complex image patch, the CenterNet model performed best, recognizing the vast majority of the ship targets. The Faster R-CNN model was the second-best model, detecting a very small number of these ships, especially the large ones. On the other hand, the EfficientDet model was able to recognize only three ships. Finally, the SSD was the most unsuccessful model for this image, recognizing only one ship object.

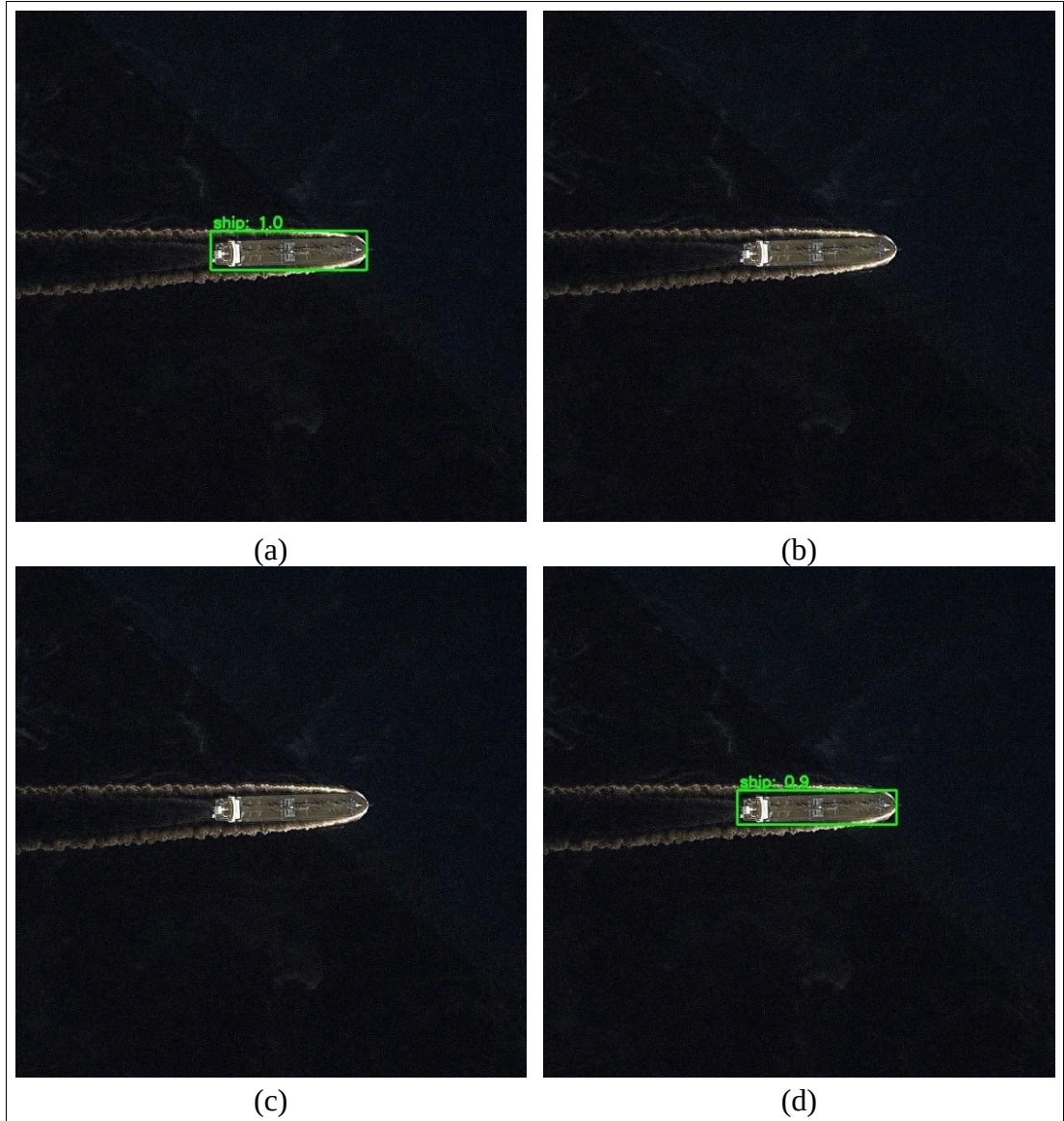


Figure 5.27: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

An example of a ship target in the offshore background is observed in Figure 5.27. When the detection results of all models were examined, only the Faster R-CNN and CenterNet models were able to recognize and locate this ship accurately. In fact, the difficulty in detecting this ship stems from the waves left by the ship in the sea.

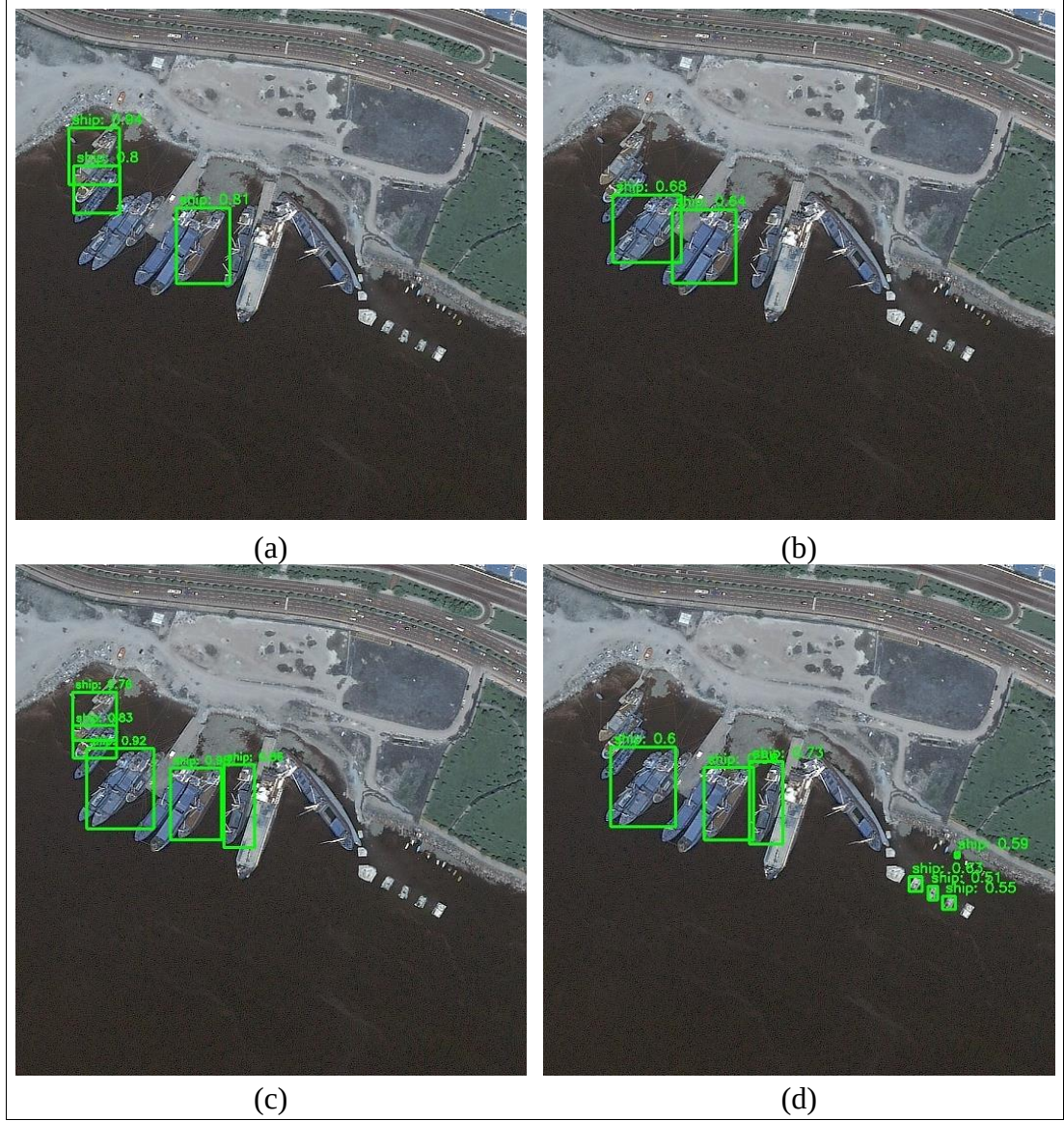


Figure 5.28: Ship detection results on GÖKTÜRK-1 images of all models. (a) Faster R-CNN, (b) SSD, (c) EfficientDet, (d) CenterNet.

An image containing many examples of adjacent large ships and small ships on the shore was also analyzed with the models (Figure 5.28). In this image, the CenterNet model detected three large and four small ships. The Faster R-CNN model detected three ships but could not accurately delimit one ship. The EfficientDet model detected five ships but failed to correctly locate one. Finally, the SSD model detected two ships with bounding boxes that did not fit the objects.

6. CONCLUSIONS

Parallel with the developments in remote sensing technologies and satellite systems, the number and quality of images are increasing day by day, allowing many artificial objects to be distinguished. The detection and identification of objects from remotely sensed images have received great attention in the field of aerial and satellite image processing. It is possible to detect even small objects, with these high-resolution images allowing for more precise classification and locating the target objects. Nowadays, studies for the recognition of various objects such as aircraft, ships, buildings, and vehicles have been intensively studied. In particular, the detection of ship objects is of great interest due to its importance for military and civilian applications, including maritime traffic monitoring, marine pollution monitoring, and maritime rescue. On the other hand, due to the use of large amounts of data and increased computational power, deep learning demonstrates high performance in image processing applications such as object detection, image classification, and image segmentation. It can outperform traditional object detection methods, especially with its ability to adapt and automatically extract features from the data given with a self-learning approach.

Within the scope of this thesis, ship objects were handled within the scope of object detection by using deep learning methods from very high-resolution satellite images. For this aim, ship detection performances of deep learning-based object detection meta-architectures were evaluated and compared. For the application, Faster R-CNN, SSD, EfficientDet-D1, CenterNet object detection models combined with ResNet-101 V1, ResNet-50 V1 FPN, EfficientNet-B1-BiFPN, and Hourglass-104 feature extractors were employed. All the models were convolutional neural network-based architectures pre-trained on the Microsoft COCO 2017 dataset. For the training and testing of the models, images containing ship targets from A Large-Scale Dataset for Object Detection in Aerial Images (DOTA) and High-Resolution Remote Sensing Detection (TGRS-HRRSD) datasets, which are widely used in object detection research in the literature, were selected and utilized. The collected images were relabeled with the image annotation tool LabelImage. As a result of data collection and labeling, a total of 13,735 targets were annotated in 2136 images. The collected images were randomly divided into a training and a testing dataset following the 8:2 ratio.

In the training phase of the models, the transfer learning approach was applied to all models. The training of these models on the ship dataset was initialized by using the weights that the models had previously learned in the MS COCO dataset. The optimal value of the hyperparameters of the models, which are predefined in the configuration files, was empirically adjusted after several experiments according to the used hardware configuration. All models were trained for 100,000 iterations and the training process of the models took 28, 17, 14, and 34 hours for the Faster R-CNN, SSD, EfficientDet, and CenterNet models, respectively. The performance of the trained models was analyzed using Common Objects in Context (COCO) evaluation metrics.

Experimental results revealed that the Hourglass-104 based CenterNet model gave the best results for ship detection in terms of all COCO metrics with the data set utilized. This model reached 88.80% AP with 0.50 IoU threshold. Among the remaining models, the ResNet-101 V1 based Faster R-CNN model became the second-best model, reaching 85.30% AP with 0.50 IoU threshold. For the same metric, EfficientNet-B1-BiFPN based EfficientDet and ResNet-50 V1 FPN based SSD models reached 78.91% and 75.98% values, respectively. However, considering medium and small ship objects, the EfficientDet model achieved higher AP and AR values than the Faster R-CNN and SSD models. Moreover, the SSD model obtained higher AP and AR for small objects than the Faster R-CNN.

When the detection results of all object detection models on the testing dataset were examined visually, it was observed that CenterNet and Faster R-CNN models produced more accurate detection results in terms of locating ship objects. In other words, the bounding boxes produced by these models limited the ship targets more precisely. Also, these models tended to produce fewer false negatives. The Faster R-CNN model had higher prediction scores than the other models. Most detected ship objects had a prediction score of more than 95%. On the other hand, the EfficientDet model was successful in detecting small ships. In addition, this model also detected some false alarms. The SSD model failed to detect ship targets located one after the other and produced more false negatives than other models. However, considering the training times of the models, the SSD model was the fastest converging model. On the other hand, the CenterNet model was the slowest converging model with the longest training time.

In order to test the generalization capability of the models on different images from the training and testing dataset, GÖKTÜRK-1 satellite images were given to the trained models and the detection results were investigated. Compared to the training and testing datasets, these images were taken from different satellites and belonged to different regions on earth. In addition, these images demonstrated different ship object characteristics and had different background conditions. Despite all this, the performance of the trained models on these images was also desired to be exhibited. Besides, marine mucilage event could be observed in these images. Under these challenging conditions, the ship detection performances of all models were tested on the selected ten GÖKTÜRK-1 image patches in the RGB bands. Experimental results revealed that all models were able to detect ship targets even offshore and in the presence of mucilage impact. It was observed that a small ship object in intensive mucilage background condition and offshore could only be recognized by the EfficientDet model. Also, it was observed that CenterNet and Faster R-CNN models could detect a ship leaving wave traces offshore. On the other hand, it was analyzed that the CenterNet model performed better when there were multiple ship objects of varied sizes on the shore. Even in the image of a marina with many small ships close together, the CenterNet model recognized most ships. This model was also more successful in locating ships. Followed this model, the Faster R-CNN model was the second-best performing model with fewer false negatives. The SSD model resulted in a vast number of false negatives and false positives. Especially when there are many ship objects close to each other on the shore, this model gave poor results. Under similar conditions, the EfficientDet model produced more promising results than the SSD model, but still produced a large number of false negatives.

As a result, CenterNet model provided the best results among all deep learning-based object detection models utilized in the study for ship detection from remotely sensed images. However, this model was trained longer than the others and the slowest converging model. If a better trade-off between time and accuracy is desired, the Faster R-CNN model with lower accuracy and less training time can be preferred. For future studies, a masking process that separates the sea and land is planned to avoid the false alarm problem that the models generally produce on the shores. Moreover, in addition to the R, G, and B bands used in the study, images containing other spectral bands can be used in model training. The contribution of these bands to accuracy is another possibility to improve the performance of deep learning models.

REFERENCES

- Abbass H. A., (2002), "An evolutionary artificial neural networks approach for breast cancer diagnosis", *Artificial Intelligence in Medicine*, 25(3), 265-281.
- Abileah R., (2009), "Surveying coastal ship traffic with LANDSAT", *OCEANS 2009*, 1-6, Biloxi, Mississippi, USA, 26-29 October.
- Acharya S., Pant A. K., Gyawali P. K., (2015), "Deep learning based large scale handwritten Devanagari character recognition", *Proceedings of the 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 1-6, Kathmandu, Nepal, 15-17 December.
- Ahmadi N., Constandinou T. G., Bouganis C. S., (2019), "Decoding hand kinematics from local field potentials using long short-term memory (LSTM) network", *Proceedings of the 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, 415-419, San Francisco, California, USA, 20-23 March.
- Alaloul W. S., Qureshi A. H., (2020), "Data processing using artificial neural networks". In: D. G. Harkut, Editor, "Dynamic data assimilation-beating the uncertainties", London, UK: IntechOpen.
- Anderson J. A., (1972), "A simple neural network generating an interactive memory", *Mathematical Biosciences*, 14(3-4), 197-220.
- Badrinarayanan V., Kendall A., Cipolla R., (2017), "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481-2495.
- Bastin L., (1997), "Comparison of fuzzy c-means classification, linear mixture modelling and MLC probabilities as tools for unmixing coarse pixels", *International Journal of Remote Sensing*, 18(17), 3629-3648.
- Benediktsson J. A., Sveinsson J. R., (2003), "Multisource remote sensing data classification based on consensus and pruning", *IEEE Transactions on Geoscience and Remote Sensing*, 41(4), 932-936.
- Bochkovskiy A., Wang C. Y., Liao H. Y. M., (2020), "Yolov4: Optimal speed and accuracy of object detection", *arXiv preprint arXiv: 2004.10934*, Available at <https://arxiv.org/abs/2004.10934>.
- Bodla N., Singh B., Chellappa R., Davis L. S., (2017), "Soft-NMS--improving object detection with one line of code", *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 5562-5570, Venice, Italy, 22-29 October.
- Brivio P. A., Colombo R., Maggi M., Tomasoni R., (2002), "Integration of remote sensing data and GIS for accurate mapping of flooded areas", *International Journal of Remote Sensing*, 23(3), 429-441.

Broomhead D. S., Lowe D., (1988), “Multivariable functional interpolation and adaptive networks”, *Complex Systems*, 2, 321- 355.

Carpenter G. A., Grossberg S., (1988), “The ART of adaptive pattern recognition by a self-organizing neural network”, *Computer*, 21(3), 77-88.

Chen L., Shi W., Deng D., (2021), “Improved YOLOv3 Based on attention mechanism for fast and accurate ship detection in optical remote sensing images” *Remote Sensing*, 13(4), 660.

Cheng B., Li Z., Xu B., Dang C., Deng J., (2021), “Target Detection in Remote Sensing Image Based on Object-and-Scene Context Constrained CNN”, *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5.

Cheng G., Han J., (2016), “A survey on object detection in optical remote sensing images”, *ISPRS Journal of Photogrammetry and Remote Sensing*, 117, 11-28.

Chu Y., Zhao X., Zou Y., Xu W., Han J., Zhao Y., (2018), “A decoding scheme for incomplete motor imagery EEG with deep belief network”, *Frontiers in Neuroscience*, 12, 680.

Ciregan D., Meier U., Schmidhuber J., (2012), “Multi-column deep neural networks for image classification”, *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3642-3649, Providence, Rhode Island, USA, 16-21 June.

Colkesen I., Kavzoglu T., (2019), “Comparative evaluation of decision-forest algorithms in object-based land use and land cover mapping”. In: H.R. Pourghasemi, C. Gokceoglu, Editors, “Spatial Modeling in GIS and R for Earth and Environmental Sciences”, 499-517, Amsterdam: Elsevier.

Corbane C., Najman L., Pecoul E., Demagistri L., Petit M., (2010), “A complete processing chain for ship detection using optical satellite imagery”, *International Journal of Remote Sensing*, 31(22), 5837-5854.

Cortes C., Vapnik V., (1995), “Support-vector networks”, *Machine Learning*, 20(3), 273-297.

Coşkun M., Yıldırım Ö., Uçar A., Demir Y., (2017), “An overview of popular deep learning methods”, *European Journal of Technique (EJT)*, 7(2), 165-176.

Çölkesen İ., (2009), “Uzaktan algılamada ileri sınıflandırma tekniklerinin karşılaştırılması ve analizi”, *Yüksek Lisans Tezi, Gebze Teknik Üniversitesi*.

Çölkesen İ., (2015), “Yüksek çözünürlüklü uydu görüntüleri kullanarak benzer spektral özelliklere sahip doğal nesnelerin ayırt edilmesine yönelik bir metodoloji geliştirme”, *Doktora Tezi, Gebze Teknik Üniversitesi*.

Dahikar S. S., Rode S. V., (2014), “Agricultural crop yield prediction using artificial neural network approach”, *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 2(1), 683-686.

Dalal N., Triggs B., (2005), “Histograms of oriented gradients for human detection”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 886-893, San Diego, California, USA, 20-25 June.

Datta D., David P. E., Mittal D., Jain A., (2020), “Neural machine translation using recurrent neural network”, *International Journal of Engineering and Advanced Technology*, 9(4), 1395-1400.

Dave V. S., Dutta K., (2014), “Neural network based models for software effort estimation: a review”, *Artificial Intelligence Review*, 42(2), 295-307.

Deng J., Dong W., Socher R., Li L. J., Li K., Fei-Fei L., (2009), “Imagenet: A large-scale hierarchical image database”, *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 248-255, Miami, Florida, USA, 20-25 June.

Ding J., Xue N., Xia G. S., Bai X., Yang W., Yang M., Belongie S., Luo J., Datcu M., Pelillo M., Zhang L., (2021), “Object detection in aerial images: A large-scale benchmark and challenges”, *arXiv preprint arXiv: 2102.12219*, Available at <https://arxiv.org/abs/2102.12219>.

Elman J. L., (1990), “Finding structure in time”, *Cognitive Science*, 14(2), 179–211.

Everingham M., Van Gool L., Williams C. K., Winn J., Zisserman A., (2010), “The pascal visual object classes (voc) challenge”, *International Journal of Computer Vision*, 88(2), 303-338.

Fei-Fei L., Perona P., (2005), “A Bayesian hierarchical model for learning natural scene categories”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 524-531, San Diego, California, USA, 20-25 June.

Fernandes R., Fraser R., Latifovic R., Cihlar J., Beaubien J., Du Y., (2004), “Approaches to fractional land cover and continuous field mapping: A comparative assessment over the BOREAS study region”, *Remote Sensing of Environment*, 89(2), 234-251.

Foody G. M., (1995a), “Using prior knowledge in artificial neural network classification with a minimal training set”, *International Journal of Remote Sensing*, 16(2), 301-312.

Foody G. M., (1995b), “Land cover classification by an artificial neural network with ancillary information”, *International Journal of Geographical Information Systems*, 9(5), 527-542.

Foody G. M., Lucas R. M., Curran P. J., Honzak M., (1997), “Non-linear mixture modelling without end-members using an artificial neural network”, *International Journal of Remote Sensing*, 18(4), 937-953.

Freund Y., Schapire R. E., (1997), “A decision-theoretic generalization of on-line learning and an application to boosting”, *Journal of Computer and System Sciences*, 55(1), 119-139.

Frizzelle B. G., Moody A., (2001), “Mapping continuous distributions of land cover: a comparison of maximum-likelihood estimation and artificial neural networks”, *Photogrammetric Engineering & Remote Sensing*, 67(6), 693–705.

Fukushima K., (1980), “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, *Biological Cybernetics*, 36(4), 193-202.

Gibney E., (2016), “Google AI algorithm masters ancient game of Go”, *Nature*, 529(7587), 445-446.

Girshick R., (2015), “Fast r-cnn”, *Proceedings of the IEEE International Conference on Computer Vision*, 1440-1448, Santiago, Chile, 7-13 December.

Girshick R., Donahue J., Darrell T., Malik J., (2014), “Rich feature hierarchies for accurate object detection and semantic segmentation”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580-587, Columbus, Ohio, USA, 23-28 June.

Glorot X., Bordes A., Bengio Y., (2011), “Deep sparse rectifier neural networks”, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 315-323, Fort Lauderdale, Florida, USA, 11-13 April.

Goldberg D. E., (1989), “Genetic Algorithms in Search, Optimization and Machine Learning”, Reading, Massachusetts: Addison-Wesley.

Goodfellow I., Bengio Y., Courville A., Bengio Y., (2016), “Deep learning”, Cambridge: MIT Press.

Goswami A. K., Gakhar S., Kaur H., (2014), “Automatic object recognition from satellite images using artificial neural network”, *International Journal of Computer Applications*, 95(10), 33-39.

Graves A., Mohamed A. R., Hinton G., (2013), “Speech recognition with deep recurrent neural networks”, *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 6645-6649, Vancouver, British Columbia, Canada, 26-31 May.

Gu J., Wang Z., Kuen J., Ma L., Shahroudy A., Shuai B., Liu T., Wang X., Wang G., Cai J., Chen T., (2018), “Recent advances in convolutional neural networks”, *Pattern Recognition*, 77, 354-377.

Han X., Jiang T., Zhao Z., Lei Z., (2020), “Research on remote sensing image target recognition based on deep convolution neural network”, *International Journal of Pattern Recognition and Artificial Intelligence*, 34(5), Article ID 2054015.

Han Y. S., Min S. S., Choi W. H., Cho K. B., (1992), “A learning pattern recognition system using neural network for diagnosis and monitoring of aging of electrical motor”, *Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation*, 1074-1077, San Diego, California, USA, 13-13 November.

He H., Lin Y., Chen F., Tai H. M., Yin Z., (2017a), “Inshore ship detection in remote sensing images via weighted pose voting”, *IEEE Transactions on Geoscience and Remote Sensing*, 55(6), 3091-3107.

He K., Gkioxari G., Dollár P., Girshick R., (2017b), “Mask R-CNN”, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980-2988, Venice, Italy, 22-29 October.

He K., Zhang X., Ren S., Sun J., (2015), “Spatial pyramid pooling in deep convolutional networks for visual recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904-1916.

He K., Zhang X., Ren S., Sun J., (2016), “Deep residual learning for image recognition”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778, Las Vegas, Nevada, USA, 27-30 June.

He L., Hou W., Zhen X., Peng C., (2006), “Recognition of ECG patterns using artificial neural network”, *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, 477-481, Jian, China, 16-18 October.

Hebb D.O., (1949), “The organization of behavior”, New York: John Wiley & Sons.

Hewitson B. C., Crane R. G., (1994), “Looks and uses”. In: B. C. Hewitson, R. G. Crane, Editors, “*Neural Nets: Applications in Geography*”, 1-9, Springer, Dordrecht.

Hinton G. E., (2009), “Deep belief networks”, *Scholarpedia*, 4(5), 5947.

Hinton G. E., Salakhutdinov R. R., (2006), “Reducing the dimensionality of data with neural networks”, *Science*, 313(5786), 504-507.

Hinton G. E., Sejnowski T. J., (1983), “Optimal perceptual inference”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 448-453, Washington, DC, June 1983.

Hochreiter S., Schmidhuber J., (1997), “Long short-term memory”, *Neural Computation*, 9(8), 1735-1780.

Holland J., (1975), “*Adaptation in Natural and Artificial Systems*”, Ann Arbor: University of Michigan Press.

Hopfield J. J., (1982), “Neural networks and physical systems with emergent collective computational abilities”, *Proceedings of the National Academy of Sciences*, 79(8), 2554-2558.

Hopfield J. J., (1984), “Neurons with graded response have collective computational properties like those of two state neurons”, *Proceedings of the National Academy of Sciences*, 81(10), 3088-3092.

Hua X., Wang X., Rui T., Zhang H., Wang D., (2020), “A fast self-attention cascaded network for object detection in large scene remote sensing images”, *Applied Soft Computing*, 94, 106495.

Ivakhnenko A. G., (1971), “Polynomial theory of complex systems”, *IEEE Transactions on Systems, Man, Cybernetics*, 1(4), 364–378.

Ivakhnenko A. G., Lapa V.G., (1965), “Cybernetic predicting devices”, *School of Electrical Engineering, Purdue University, Lafayette, Indiana*.

Jain A. K., Mao J., Mohiuddin K. M., (1996), “Artificial neural networks: A tutorial”, *Computer*, 29(3), 31-44.

Ji C. Y., (2000), “Land-use classification of remotely sensed data using Kohonen self-organizing feature map neural networks”, *Photogrammetric Engineering and Remote Sensing*, 66(12), 1451-1460.

Jiang J., Trundle P., Ren J., (2010), “Medical image analysis with artificial neural networks”, *Computerized Medical Imaging and Graphics*, 34(8), 617-631.

Jin X., Davis C. H., (2007), “Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks”, *Image and Vision Computing*, 25(9), 1422-1431.

Jordan M. I., (1986), “Attractor dynamics and parallelism in a connectionist sequential machine”, *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, 531–546, Amherst, Massachusetts, USA, 15-17 August.

Kang M., Leng X., Lin Z., Ji K., (2017), “A modified faster R-CNN based on CFAR algorithm for SAR ship detection”, *Proceedings of the 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, 1-4, Shanghai, China, 18-21 May.

Kavzoglu T., (2001), “An investigation of the design and use of feed-forward artificial neural networks in the classification of remotely sensed images”, *PhD Thesis, The University of Nottingham, United Kingdom*.

Kavzoglu T., (2009), “Increasing the accuracy of neural network classification using refined training data”, *Environmental Modelling & Software*, 24(7), 850-858.

Kavzoglu T., Mather P. M., (2001), “Effective use of artificial neural networks for remotely sensed image classification”, Proceedings of the First Annual Conference of Remote Sensing & Photogrammetry Society, 92-98, London, UK, 12-14 September.

Kavzoglu T., Tonbul H., Colkesen İ., Sefercik U. G., (2021), “The Use of Object-Based Image Analysis for Monitoring 2021 Marine Mucilage Bloom in the Sea of Marmara”, International Journal of Environment and Geoinformatics, 8(4), 529-536.

Kavzoglu T., Vieira C. A. O., (1998), “An analysis of artificial neural network pruning algorithms in relation to land cover classification accuracy”, Proceedings of the Remote Sensing Society Student Conference, 53-58, Oxford, UK, 23 April.

Kavzoğlu T., Çölkesen İ., (2011), “Uzaktan algılama teknolojileri ve uygulama alanları”, Türkiye’de Sürdürülebilir Arazi Yönetimi Çalıştayı, İstanbul, Türkiye, 26-27 Mayıs.

Kohonen T., (1972), “Correlation matrix memories”, IEEE Transactions on Computers, 100(4), 353-359.

Kohonen T., (1982), “Self-organized formation of topologically correct feature maps”, Biological Cybernetics, 43(1), 59-69.

Kohonen T., (1989), “Self-organization and associative memory”, 3rd Edition, Berlin, Germany: Springer-Verlag.

Kononenko I., Kukar M., (2007), “Machine learning and data mining: Introduction to Principles and Algorithms”, Horwood Publishing Limited, Chichester, UK.

Krenker A., Bešter J., Kos A., (2011), “Introduction to the artificial neural networks”. In: K. Suzuki, Editor, “Artificial Neural Networks-Methodological Advances and Biomedical Applications”, 3-18, InTech, Rijeka, Croatia.

Krizhevsky A., Sutskever I., Hinton G. E., (2012), “ImageNet classification with deep convolutional neural networks”, Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), 1097-1105, Lake Tahoe, Nevada, USA, 3-6 December.

LeCun Y., Bengio Y., Hinton G., (2015), “Deep learning”, Nature, 521(7553), 436-444.

LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., (1989), “Backpropagation applied to handwritten zip code recognition”, Neural Computation, 1(4), 541-551.

LeCun Y., Bottou L., Bengio Y., Haffner P., (1998), “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, 86(11), 2278-2324.

Lek S., Guégan J. F., (1999), “Artificial neural networks as a tool in ecological modelling, an introduction”, Ecological Modelling, 120(2-3), 65-73.

Li Z., Khananian A., Fraser R. H., Cihlar J., (2001), “Automatic detection of fire smoke using artificial neural networks and threshold approaches applied to AVHRR imagery”, *IEEE Transactions on Geoscience and Remote Sensing*, 39(9), 1859-1870.

Licciardi G. A., Del Frate F., (2011), “Pixel unmixing in hyperspectral data by means of neural networks”, *IEEE Transactions on Geoscience and Remote Sensing*, 49(11), 4163-4172.

Lillesand T., Kiefer R. W., Chipman J., (2015), “Remote sensing and image interpretation”, 7th Edition, John Wiley & Sons, Hoboken, N.J., USA.

Lin T. Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Bourdev L., Girshick R., Dollar P., Zitnick C. L., (2014), “Microsoft coco: Common objects in context”, *Proceedings of the European Conference on Computer Vision (ECCV)*, 740-755, Zurich, Switzerland, September 6-12.

Liu G., Zhang Y., Zheng X., Sun X., Fu K., Wang H., (2014), “A new method on inshore ship detection in high-resolution satellite images using shape and context information”, *IEEE Geoscience and Remote Sensing Letters*, 11(3), 617-621.

Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C. Y., Berg A. C., (2016), “SSD: Single shot multibox detector”, *Proceedings of the European Conference on Computer Vision*, 21–37, Amsterdam, The Netherlands, 11–14 October.

Liu W., Wang Z., Liu X., Zeng N., Liu Y., Alsaadi F. E., (2017), “A survey of deep neural network architectures and their applications”, *Neurocomputing*, 234, 11-26.

Liu Y., Sun P., Wergeles N., Shang Y., (2021), “A survey and performance evaluation of deep learning methods for small object detection”, *Expert Systems with Applications*, 172, 114602.

Ma L., Liu Y., Zhang X., Ye Y., Yin G., Johnson B. A., (2019), “Deep learning in remote sensing applications: A meta-analysis and review”, *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 166-177.

Mani N., Srinivasan B., (1997), “Application of artificial neural network model for optical character recognition”, *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, 2517-2520, Orlando, Florida, USA, 12-15 October.

Mas J. F., Flores J. J., (2008), “The application of artificial neural networks to the analysis of remotely sensed data”, *International Journal of Remote Sensing*, 29(3), 617-663.

McClelland J. L., Rumelhart D. E., (1986), “Parallel distributed processing: Explorations in the microstructure of cognition”, Vol 2: Psychological and Biological Models, Cambridge, MA: MIT Press.

McCulloch W. S., Pitts W., (1943), “A logical calculus of the ideas immanent in nervous activity”, *Bulletin of Mathematical Biophysics*, 5, 115–133.

Mei Y., Chen H., Yang S., (2021), “Research on Building Target Detection Based on High-Resolution Optical Remote Sensing Imagery”, *Algorithms*, 14(10), 300.

Mikolov T., Karafiát M., Burget L., Cernocký J., Khudanpur S., (2010), “Recurrent neural network based language model”, *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, 1045-1048, Makuhari, Chiba, Japan, 26-30 September.

Minsky M., Papert S. A., (1969), “Perceptrons: An Introduction to Computational Geometry”, MIT Press: Cambridge, MA, USA.

Moghaddam A. H., Moghaddam M. H., Esfandyari M., (2016), “Stock market index prediction using artificial neural network”, *Journal of Economics, Finance and Administrative Science*, 21(41), 89-93.

Mokhtarzade M., Zoej M. V., (2007), “Road detection from high-resolution satellite images using artificial neural networks”, *International Journal of Applied Earth Observation and Geoinformation*, 9(1), 32-40.

Mosavi A., Ardabili S., Varkonyi-Koczy A. R., (2019), “List of deep learning models”, In: A.R. Varkonyi-Koczy, Editor, “Engineering for Sustainable Future”, 202-214. Springer, Cham.

Mountrakis G., Im J., Ogole C., (2011), “Support vector machines in remote sensing: A review”, *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3), 247-259.

Nagarkar P., Rege P. P., (2019), “Integrating Artificial Neural Network and Taguchi Method for Land Use and Land Cover Classification of Multispectral Images”, *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1-6, Kanpur, India, 06-08 July.

Nallapati R., Zhai F., Zhou B., (2017), “SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents”, *Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence (AAAI-17)*, San Francisco, California, USA, 4-9 February.

Nazeer S. A., Omar N., Khalid M., (2007), “Face recognition system using artificial neural networks approach”, *Proceedings of the 2007 International Conference on Signal Processing, Communications and Networking*, 420-425, Chennai, India, 22-24 February.

Ni R., Hua B., (2021), “Ship Recognition Technology of High Precision Remote Sensing Images Based on YOLO”, *Proceedings of the 2020 International Conference on Guidance, Navigation and Control (ICGNC 2020)*, 443-451, Tianjin, China, 23–25 October.

Nie X., Duan M., Ding H., Hu B., Wong E. K., (2020), “Attention mask R-CNN for ship detection and segmentation from remote sensing images”, *IEEE Access* 2020, 8, 9325-9334.

Noda K., Yamaguchi Y., Nakadai K., Okuno H. G., Ogata T., (2015), “Audio-visual speech recognition using deep learning”, *Applied Intelligence*, 42(4), 722-737.

Oancea B., Ciucu Ş. C., (2014), “Time series forecasting using neural networks”, *arXiv preprint arXiv: 1401.1333*, Available at <https://arxiv.org/abs/1401.1333>.

Öztemel E., (2006), “Yapay Sinir Ağları”, 2. Baskı, Papatya Yayıncılık, İstanbul, Türkiye.

Öztemel E., (2012), “Yapay Sinir Ağları”, 3. Baskı, Papatya Yayıncılık, İstanbul, Türkiye.

Pekel E., Soner Kara S., (2017), “A Comprehensive Review for Artificial Neural Network Application to Public Transportation”, *Sigma Journal of Engineering and Natural Sciences*, 35(1), 157-179.

Peterson L. E., (2009), “K-nearest neighbor”, *Scholarpedia*, 4(2), 1883.

Pham D. T., Packianather M. S., Afify A. A., (2007), “Artificial neural networks”. In: D. Andina, D. T. Pham, Editors, “Computational Intelligence”, 67-92, Springer, Boston, MA.

Pradhan B., (2009), “Groundwater potential zonation for basaltic watersheds using satellite remote sensing data and GIS techniques”, *Central European Journal of Geosciences*, 1(1), 120-129.

Qi S., Ma J., Lin J., Li Y., Tian J., (2015), “Unsupervised ship detection based on saliency and S-HOG descriptor from optical satellite images”, *IEEE Geoscience and Remote Sensing Letters*, 12(7), 1451-1455.

Ou C., Yang J., Du Z., Zhang X., Zhu D., (2019), “Integrating cellular automata with unsupervised deep-learning algorithms: A case study of urban-sprawl simulation in the Jingjintang urban agglomeration, China”, *Sustainability*, 11(9), 2464.

Redmon J., Divvala S., Girshick R., Farhadi A., (2016), “You only look once: Unified, real-time object detection”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788, Las Vegas, Nevada, USA, 27-30 June.

Redmon J., Farhadi A., (2017), “YOLO9000: better, faster, stronger”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7263-7271, Honolulu, Hawaii, 21-26 July.

Redmon J., Farhadi A., (2018), “Yolov3: An incremental improvement”, *arXiv preprint arXiv: 1804.02767*, Available at <https://arxiv.org/abs/1804.02767>.

Ren S., He K., Girshick R., Sun J., (2015), “Faster r-cnn: Towards real-time object detection with region proposal networks”, *Proceedings of the Advances in Neural Information Processing Systems*, 91-99, Montréal, Quebec, Canada, 7-12 December.

Ribeiro M., Lazzaretti A. E., Lopes H. S., (2018), “A study of deep convolutional auto-encoders for anomaly detection in videos”, *Pattern Recognition Letters*, 105, 13-22.

Rosenblatt F., (1958), “The perceptron: a probabilistic model for information storage and organization in the brain”, *Psychological Review*, 65(6), 386.

Rumelhart D. E., McClelland J. L., (1986), “Parallel distributed processing: Explorations in the microstructure of cognition”, Vol 1: Foundations, Cambridge, MA: MIT Press.

Salami A. A., Ajavon A. A., Dotche K. A., Bedja K. S., (2018), “Electrical load forecasting using artificial neural network: the case study of the grid inter-connected network of Benin electricity community (CEB)”, *American Journal of Engineering and Applied Sciences*, 11(2), 471-481.

Sarker I. H., (2021), “Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions”, *SN Computer Science*, 2(6), 1-20.

Shah S. A. A., ul Asar A., Shaukat S. F., (2009), “Neural network solution for secure interactive voice response”, *World Applied Sciences Journal*, 6(9), 1264-1269.

Shi Z., Yu X., Jiang Z., Li B., (2014), “Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature”, *IEEE Transactions on Geoscience and Remote Sensing*, 52(8), 4511-4523.

Simonyan K., Zisserman A., (2014), “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv: 1409.1556*, Available at <https://arxiv.org/abs/1409.1556>.

Smolensky P., (1986), “Information processing in dynamical systems: Foundations of harmony theory”, Department of Computer Science, University of Colorado Boulder, Colorado.

Socher R., (2014), “Recursive deep learning for natural language processing and computer vision”, PhD Thesis, Stanford University, Stanford, California, USA,

Specht D. F., (1988), “Probabilistic neural networks for classification, mapping or associative memory”, *Proceedings of the IEEE Conference on Neural Networks*, 525-532, San Diego, California, 24-27 July.

Specht D. F., (1991), “A general regression neural network”, *IEEE Transactions on Neural Networks*, 2(6), 568-576.

Srivastava N., Hinton G. E., Krizhevsky A., Sutskever I., Salakhutdinov R., (2014), “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *Journal of Machine Learning Research* 15(1), 1929-1958.

Sublime J., Kalinicheva E., (2019), “Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the Tohoku tsunami”, *Remote Sensing*, 11(9), 1123.

Sudheer K. P., Gosain A. K., Mohana Rangan D., Saheb S. M., (2002), “Modelling evaporation using an artificial neural network algorithm”, *Hydrological Processes*, 16(16), 3189-3202.

Sun X. M., Zhang Y. J., Wang H., Du Y. X., (2022), “Research on ship detection of optical remote sensing image based on Yolo V5”, *Journal of Physics: Conference Series*, 2215(1), 012027.

Sunar F., Özkan C., Osmanoğlu B., (2011), “Uzaktan Algılama”, Anadolu Üniversitesi Yayını, Eskişehir. Türkiye.

Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A., (2015), “Going deeper with convolutions”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9, Boston, Massachusetts, USA, 7-12 June.

Tan M., Le Q., (2019), “EfficientNet: Rethinking model scaling for convolutional neural networks”, *Proceedings of the 36th International Conference on Machine Learning (PMLR)*, 6105-6114, Long Beach, California, USA, 9-15 June.

Tan M., Pang R., Le Q. V., (2020), “EfficientDet: Scalable and efficient object detection”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10781-10790, Seattle, Washington, USA, 14-19 June.

Tang J., Deng C., Huang G. B., Zhao B., (2015), “Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine”, *IEEE Transactions on Geoscience and Remote Sensing*, 53(3), 1174-1185.

Tatem A. J., Lewis H. G., Atkinson P. M., (2001), “Super-resolution target identification from remotely sensed images using a Hopfield neural network”, *IEEE Transactions on Geoscience and Remote Sensing*, 39(4), 781–796.

Temurtas H., Yumusak N., Temurtas F., (2009), “A comparative study on diabetes disease diagnosis using neural networks”, *Expert Systems with Applications*, 36(4), 8610-8615.

Tonbul H., Colkesen I., Kavzoglu T., (2020), “Classification of poplar trees with object-based ensemble learning algorithms using Sentinel-2A imagery”, *Journal of Geodetic Science*, 10(1), 14-22.

Tonbul H., Colkesen I., Kavzoglu T., (2022), “Pixel-and Object-Based Ensemble Learning for Forest Burn Severity Using USGS FIREMON and Mediterranean Condition dNBRs in Aegean Ecosystem (Turkey)”, *Advances in Space Research*, 69(10), 3609-3632.

Viola P., Jones M., (2001), “Rapid object detection using a boosted cascade of simple features”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, 511-518, Kauai, Hawaii, USA, 8-14 December.

Wang B., Valentine S., Plasencia M., Raghuraman S., Zhang X., (2010), “Artificial neural networks for the prediction of peptide drift time in ion mobility mass spectrometry”, BMC Bioinformatics, 11(1), 182.

Wang Y., Zhang Y., Zhang Y., Zhao L., Sun X., Guo Z., (2019), “SARD: Towards scale-aware rotated object detection in aerial imagery”, IEEE Access, 7, 173855-173865.

Web 1, (2021), <https://github.com/ultralytics/yolov5>, (Accessed on: 25/12/2021).

Web 2, (2022), <https://www.rfcafe.com/references/electrical/electromagnetic-wave-physics.htm>, (Accessed on: 17/03/2022).

Web 3, (2022), https://saylordotorg.github.io/text_general-chemistry-principles-patterns-and-applications-v1.0/s10-01-waves-and-electromagnetic-radi.html, (Accessed on: 18/03/2022).

Web 4, (2022), <https://seos-project.eu/classification/classification-c01-p05.html>, (Accessed on: 24/03/2022).

Web 5, (2022), <https://earthdata.nasa.gov/learn/backgrounders/remote-sensing>, (Accessed on: 25/03/2022).

Web 6, (2022), <https://subscription.packtpub.com/book/big-data-and-business-intelligence/9781787121393/5/ch05lv1sec38/the-biological-neuron>, (Accessed on: 09/04/2022).

Web 7, (2022), <https://bdtechtalks.com/2019/08/05/what-is-artificial-neural-network-ann/artificial-neuron/>, (Accessed on: 09/04/2022).

Web 8, (2022), <https://developer.nvidia.com/blog/deep-learning-nutshell-history-training/>, (Accessed on: 24/04/2022).

Web 9, (2022), https://www.hihstore.com/?product_id=87736409_44, (Accessed on: 29/04/2022).

Web 10, (2022), <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>, (Accessed on: 30/04/2022).

Web 11, (2022), <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>, (Accessed on: 30/04/2022).

Web 12, (2022), <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>, (Accessed on: 01/05/2022).

Web 13, (2022), <https://medium.com/@Aj.Cheng/convolutional-neural-network-d9f69e473feb>, (Accessed on: 01/05/2022).

Web 14, (2022), <https://github.com/AshokBhat/ml/wiki/Fully-Connected-Layer>, (Accessed on: 02/05/2022).

Web 15, (2022), LabelImg. Available online: <https://github.com/tzutalin/labelImg>, (Accessed on: 17/05/2022).

Weng Q., (2002), “Land use change analysis in the Zhujiang Delta of China using satellite remote sensing, GIS and stochastic modelling”, *Journal of Environmental Management*, 64(3), 273-284.

Widrow B., Hoff M. E., (1960), “Adaptive switching circuits”, 1960 IRE WESCON Convention Record, 4, 96–104, New York, USA.

Wu K., Bai C., Wang D., Liu Z., Huang T., Zheng H., (2021), “Improved Object Detection Algorithm of YOLOv3 Remote Sensing Image”, *IEEE Access*, 9, 113889-113900.

Wu Q., Feng D., Cao C., Zeng X., Feng Z., Wu J., Huang Z., (2021), “Improved mask R-CNN for aircraft detection in remote sensing images”, *Sensors*, 21(8), 2618.

Xia G. S., Bai X., Ding J., Zhu Z., Belongie S., Luo J., Datcu M., Pelillo M., Zhang L., (2018), “DOTA: A large-scale dataset for object detection in aerial images”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3974-3983, Salt Lake City, Utah, USA, 18-23 June.

Xie G., Li J., Xiao F., Li T., (2021), “Optical Remote Sensing Image Ship Detection Method for Improving EfficientDet”, *Proceedings of the IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 340-343, Greenville, South Carolina, 12-14 November.

Yang C., Rottensteiner F., Heipke C., (2018a), “Classification of land cover and land use based on convolutional neural networks”, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(3), 251-258.

Yang G., Li B., Ji S., Gao F., Xu Q., (2014), “Ship detection from optical satellite images based on sea surface analysis”, *IEEE Geoscience and Remote Sensing Letters*, 11(3), 641-645.

Yang X., Sun H., Fu K., Yang J., Sun X., Yan M., Guo Z., (2018b), “Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks”, *Remote Sensing*, 10(1), 132.

Yang X., Ye Y., Li X., Lau R. Y., Zhang X., Huang X., (2018c), “Hyperspectral image classification with deep learning models”, *IEEE Transactions on Geoscience and Remote Sensing*, 56(9), 5408-5423.

Yılmaz Z., Bozkurt M., (2012), “Determination of women iron deficiency anemia using neural networks”, *Journal of Medical Systems*, 36(5), 2941-2945.

Yildirim E., Kavzoglu T., (2021), “Performance Comparison of Deep Learning Techniques for Ship Detection Using Optical Satellite Images”, *International Symposium on Applied Geoinformatics (ISAG 2021)*, Riga, Latvia, 2-3 December.

Yildirim E., Kavzoglu T., (2022), “Ship Detection in Optical Remote Sensing Images Using YOLOv4 and Tiny YOLOv4”. In: M. Ben Ahmed, A. A. Boudhir, İ. R. Karaş, V. Jain, S. Mellouli, Editors, “Innovations in Smart Cities Applications Volume 5”, 913-924, Springer, Cham.

Yildirim E., Nazar M., Sefercik U. G., Kavzoglu T., (2022), “Stone Pine (*Pinus Pinea* L.) Detection from High-Resolution UAV Imagery Using Deep Learning Model”, International Geoscience and Remote Sensing Symposium (IGARSS 2022), Kuala Lumpur, Malaysia, 17-22 July.

Yuan H., Van Der Wiele C. F., Khorram S., (2009), “An automated artificial neural network system for land use/land cover classification from Landsat TM imagery”, Remote Sensing, 1(3), 243-265.

Zeiler M. D., Fergus R., (2014), “Visualizing and understanding convolutional networks”, Proceedings of the European Conference on Computer Vision, 818-833, Zurich, Switzerland, 6-12 September.

Zhang D., Zhan J., Tan L., Gao Y., Župan R., (2021), “Comparison of two deep learning methods for ship target recognition with optical remotely sensed data”, Neural Computing and Applications, 33(10), 4639-4649.

Zhang G., Patuwo B. E., Hu M. Y., (1998), “Forecasting with artificial neural networks: The state of the art”, International Journal of Forecasting, 14(1), 35-62.

Zhang S., Wu R., Xu K., Wang J., Sun W., (2019a), “R-CNN-based ship detection from high resolution remote sensing imagery”, Remote Sensing, 11(6), 631.

Zhang Y., Yuan Y., Feng Y., Lu X., (2019b), “Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection”, IEEE Transactions on Geoscience and Remote Sensing, 57(8), 5535-5548.

Zhou L., Yang X., (2008), “Use of neural networks for land cover classification from remotely sensed imagery”, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37, 575-578.

Zhou X., Wang D., Krähenbühl P., (2019), “Objects as points”, arXiv preprint arXiv: 1904.07850, Available at <https://arxiv.org/abs/1904.07850>.

Zhu C., Zhou H., Wang R., 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.

Zou Z., Shi Z., (2016), “Ship detection in spaceborne optical image with SVD networks. IEEE Transactions on Geoscience and Remote Sensing”, 54(10), 5832-5845.

Zurada J. M., (1992), “Introduction to Artificial Neural Systems”, West Publishing Company.

BIOGRAPHY

Esra YILDIRIM completed his primary, secondary, and high school education in Trabzon, and received her bachelor's degree in Geomatics Engineering as the second and high honor student in her class from Karadeniz Technical University (Turkey) in 2019. In the same year, she started her master's degree education at the Department of Geomatics Engineering, Graduate Institute of Natural and Applied Sciences, Karadeniz Technical University. In 2021, she continued her master's degree at Gebze Technical University, Geodesy and Geographical Information Technologies Program with a postgraduate transfer program. In the same year, she started working as a Research Assistant in the Department of Geomatics Engineering at Gebze Technical University. She is currently working in this position.

APPENDICES

Appendix A: Publications Made within the Scope of the Thesis Study

Yildirim E., Kavzoglu T., (2022), “Ship Detection in Optical Remote Sensing Images Using YOLOv4 and Tiny YOLOv4”. In: M. Ben Ahmed, A. A. Boudhir, İ. R. Karaş, V. Jain, S. Mellouli, Editors, “Innovations in Smart Cities Applications Volume 5”, 913-924, Springer, Cham.

Yıldırım E., Kavzoğlu T., (2022), “CenterNet ve EfficientDet Derin Öğrenme Modelleri ile Yüksek Çözünürlüklü Uydu Görüntülerinden Obje Tespiti”, XI. Türkiye Ulusal Fotogrametri ve Uzaktan Algılama Birliği (TUFUAB) Teknik Sempozyumu, Mersin, Türkiye, 12-14 Mayıs.

Yildirim E., Nazar M., Sefercik U. G., Kavzoglu T., (2022), “Stone Pine (*Pinus Pinea* L.) Detection from High-Resolution UAV Imagery Using Deep Learning Model”, International Geoscience and Remote Sensing Symposium (IGARSS 2022), Kuala Lumpur, Malaysia, 17-22 July.

Yildirim E., Kavzoglu T., (2021), “Performance Comparison of Deep Learning Techniques for Ship Detection Using Optical Satellite Images”, International Symposium on Applied Geoinformatics (ISAG 2021), Riga, Latvia, 2-3 December.