Turkish Journal of Engineering
https://dergipark.org.tr/en/pub/tuje
e-ISSN 2587-1366

Ship Detection from Optical Satellite Images Using Convolutional Neural Networks

Neslihan Toprak*[1], **Yıldıray Yalman** [2]

[1]*Piri Reis University, Department of Electrical and Electronics Engineering, Istanbul, TURKEY,* neslihantoprak23@gmail.com
[2]*Piri Reis University, Department of Computer Engineering, Istanbul, TURKEY,* yyalman@pirireis.edu.tr

Cite this study: Toprak, N. & Yalman, Y. (2025). Ship Detection from Optical Satellite Images Using Convolutional Neural Networks. Turkish Journal of Engineering, 9 (2), 342-353.

https://doi.org/10.31127/tuje.1529660

Keywords
Convolutional Neural Networks
Ship Detection
Optical Satellite Image
YOLOv8
YOLOv9

Research Article

Received: 07.08.2024
Revised: 14.09.2024
Accepted: 16.09.2024
Published: 01.04.2025

Abstract

Since most of the world is covered with oceans and seas, seas and oceans have aroused people's curiosity throughout history. Humans have used oceans and seas in versatile ways. The seas are critical areas for trade, transportation, fishing, tourism, energy resources, border security, defense, and intelligence operations. Today, the increasing use of maritime routes creates problems in terms of maritime security, maritime traffic, and management. It has become necessary to look for alternatives to solve such problems in the maritime industry, and deep learning techniques have been used to solve these problems. This paper presents ship detection method from optical satellite images using convolutional neural networks. The motivation of this paper is to produce solutions to the issues of detecting possible dangers in areas with heavy maritime traffic, preventing illegal fishing, preventing pirate attacks, human smuggling, country defense, security and tracking of maritime trade routes with ship detection systems. The convolutional neural network models used in the paper are based on YOLOv8 and YOLOv9 and include different packages of these models. The dataset used in the paper was created using the FGSCR-42 dataset. The dataset used in the paper includes 12 classes. The performance of the model results was compared, and the results are presented in this paper. The mAP50 value of our YOLOv8l model, which we use as a new approach to ship detection studies in the literature, is 98.9%. Compared to similar studies in the literature, our model obtains a higher mAP value.

1. Introduction

Oceans and seas make up a large part of the world and are therefore critical areas for trade, transportation, tourism, energy resources and border security. With developing technologies and increasing world population, people have started to use sea routes more. This situation has brought about problems in terms of maritime security, maritime traffic and management. Deep learning methods were used as one of the alternative methods that can be used to solve these problems in the maritime industry. The traffic density in the world has been shown in Figure 1.

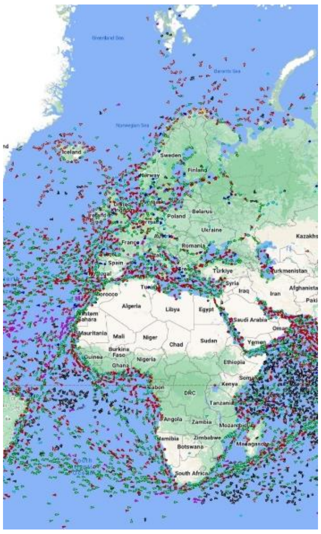


Figure 1. Ship traffic density [1]

In the maritime sector report published by the Turkish Chamber of Shipping (IMEAK) in 2023, it was stated that the world trade fleet had 60,004 (300 GT and above) ships at the beginning of 2023 [2]. This number of ships creates intense maritime traffic, and it is important to ensure navigation safety and detect possible dangers in areas with heavy maritime traffic. The International Maritime Organization (IMO), of which Turkey is a member, operates for a common regulation on issues such as navigation in international waters, marine protection, and security. The motivation of this paper is to produce solutions to the security and tracking issues of maritime trade routes with ship detection systems.

MARPOL, an international convention, aims to prevent the pollution of the seas by ships [2]. To solve the marine pollution problem, early detection of marine pollution using optical satellite images is a useful step towards this goal.

Using optical satellite imagery for ship detection can be beneficial in preventing pirate attacks and illegal fishing in regions of concern such as the Caribbean, West Africa, Southeast Asia, South and Central America, and the Indian Peninsula [2].

Illegal migration activities by sea also constitute an important problem all over the world. Turkey is located on the Eastern Mediterranean migration route and is a transit route for immigrants [2]. It is convenient to use ship detection applications with optical satellite images to detect and prevent human smuggling activities.

The issue of maritime traffic involving ships carrying explosives, toxic, and hazardous materials is a significant problem for the Turkish Straits in terms of national security. According to 2022 data, there were 77,486 ship transits through the straits, with 42,340 passing through the Dardanelles Strait and 35,146 through the Bosphorus Strait [2]. Utilizing optical satellite imagery for ship detection can provide fast results that facilitate the management of this traffic. If this ship traffic is not properly managed, potential accidents could have adverse economic, environmental, and public health consequences.

The use of convolutional neural networks is proposed as a solution method for the problems mentioned in this paper. Convolutional neural networks are a sub-branch of deep learning. Deep learning methods are used to quickly learn and apply complex data by eliminating the human factor [3]. The main reason for using CNN in this paper is that object detection can be performed with convolutional neural networks. Convolutional neural networks are basically modelled on the human visual system and are used to detect, identify, and classify objects [3]. According to this structure, the central cells are divided into subregions to cover the entire image. Simple cells focus on edge-like features. Complex cells, which have larger receptor fields, focus on the entire visual field. In this context, the mathematical convolution process can be considered as the response of a neuron to stimuli in its receptive field [4,5,6]. CNN is a type of multilayer perceptron. The CNN model consists of one or more convolutional layers, a subsampling layer, and one or more fully connected layers [7]. Deep learning methods are a powerful technique used in many current research areas [8]. Ciresan et al. have shown in their studies that CNN reduces the error rate [9] and provides faster results than traditional methods [10]. Examples of areas and studies where deep learning methods are used: groundwater problems [11], wheat yield forecasting [12], air quality forecasting and modeling [13], yoga pose classification [14], network traffic classification [15] studies and sentence modeling [16], semantic parsing [17], prediction [16], classification [18], image and sound processing various fields can be given as examples [19-33].

Remote sensing methods are preferred in ship detection due to their ability to scan large areas quickly and precisely. In the literature research, it has been seen that Synthetic Aperture Radar (SAR) and optical satellite images are used in ship detection studies. Optical sensors are used to reflect the energy emitted from the Sun from the Earth's surface; therefore, high-quality images cannot be provided in poor weather conditions. SAR images, unlike optical satellite images, can be used in all weather conditions [34]. SAR sensors send microwaves to the target with horizontal and vertical polarization and measure the intensity, polarization and phase of the waves reflected from the target [35]. The SAR sensor works by detecting reflections sent to the object and reflected.

Many of the ship detection studies in the literature use SAR data. However, due to developing technology, increasing resolution and the increasing number of satellites, optical sensors have begun to be preferred for ship detection. High-resolution optical satellite images have become preferred due to their advantageous features such as color information, wide coverage area, continuous observation, low cost, ease of processing and short-term repeatability. The importance of optical satellite images in ships and ship type detection also contributes to the defense industry in various ways. This technology is useful in monitoring the movements of enemy ships, reacting quickly in ensuring maritime and coastal security, and collecting and analyzing important data. Additionally, these methods reduce the danger of electromagnetic detection during combat.

Papers in the literature on this subject have been examined. The results obtained by those using the YOLO architecture and the FGSCR-42 dataset are stated. Wang et al. [36] proposed their algorithm consisting of three different optimizations based on YOLOv4 to solve the real-time operation and accuracy problems of existing ship detection algorithms. The optimization processes they perform are K-means clustering, Mix-up and model configuration changes. Wang et al. [36] created their own dataset and performed the training and testing phases of their algorithm with this dataset on the GPU cloud server. They compared the performance of their algorithm with Faster R-CNN, SSD and YOLOv3 architectures. The datasets consisting of 4000 photographs consist of 9 categories in total, namely container ship, lifeboat, schooner, speedboat, ocean liner, yawl, bulk carrier, oil tanker and general cargo ship. In the experiment results mAP values are 93.55%.

Hong et al. [37] proposed a new improved model based on YOLOv3 architecture for ship detection from SAR and optical satellite images. They made three different improvements for their models. To obtain good results from anchor boxes, they performed linear scaling with the K-means++ algorithm. Gaussian parameters are added for positioning bounding boxes. While the default number of anchor boxes in the Gaussian-YOLO layer was three, they increased the number of anchor boxes to four in the model they created. The created model results were increased by 3% by applying YOLOv3-tiny and YOLOv3-spp. In their paper, they used the publicly available SAR dataset and the Kaggle "Airbus Ship Detection Challenge" dataset within the Optical ship dataset. In the model results, the mAP value was found to be 95.52% on average in SAR images and 93.56% on average in optical satellite images.

Si et al. [38] presented ship detection method on YOLOv5 and created the YOLO-remote sensing ship detection model by making four different optimizations in this model. In their proposed model, they performed four different optimizations: Used the K-means++ algorithm in the data preprocessing stage, used the bidirectional feature pyramid network structure in the feature fusion stage, used the EIoU loss function instead of the Regression loss function, and used the channel attention mechanism to increase the ship feature capture ability. The dataset used in the paper is satellite images of ships at sea in the Kaggle remote sensing image competition. The mAP in the experimental results is 96.1%.

Di et al. [39] made fine-grained ship classification from remote sensing images. Due to the scarcity of data sets in this field, they researched the datasets in the literature and created a new dataset in their articles. In addition to DSCR, the remote sensing ship classification dataset prepared by Di et al., they also added data from datasets used in the literature, NWPU VHR-10, DOTA, HRSC2016 and Google Earth. They created the FGSCR-42 dataset, which consists of 9320 photographs containing combat and civilian ships. The dataset they created consists of 10 main categories and 42 subcategories. They used 8 different architectures to determine the results of the datasets. They achieved an accuracy rate of 77.36% in VGG19 [40], which is a CNN classification network, 87.24% in ResNet-50 [41], 89.16% in ResNext - 50 [42], and 88.69% in DenseNet [43]. They achieved an accuracy rate of 89.53% in fine-grained classification network B-CNN [44], 91.63% in RA-CNN [45], 93.03% in DCL [46] and 93.51% in TASN. They also calculated the accuracy values separately for 42 subcategories using B-CNN.

FGSCR-42 dataset is used in this paper. The presented method's results obtained using this dataset were compared with the studies mentioned in the literature. Additionally, they were compared with the detailed results obtained using B-CNN and the final remarks are shared in the experimental results and conclusion section.

2. Materials and Methods

In this paper, YOLOv8 architecture and YOLOv9 architecture, which was newly launched in 2024, were used. Ship detections were performed from optical satellite images using these CNN architectures. Details of these are given in the following subsections.

2.1. YOLO Architecture

YOLO architecture was designed as an architecture that processes all image features through the Darknet. The created convolutional neural network architecture was evaluated with the PASCAL VOC dataset. YOLO architecture consists of 24 convolution layers, 4 pooling layers and 2 fully connected layers. The YOLO architecture basically divides the input image into a grid of cells and estimates the probability of an object within each cell and its bounding box. Except for the last layer where the linear activation function is used, the ReLU activation function is used in the YOLO architecture. In the YOLO architecture, the Non-Maximum Suppression algorithm is used for the accuracy of the confidence score in bounding boxes. YOLO architecture is shown in Figure 2.

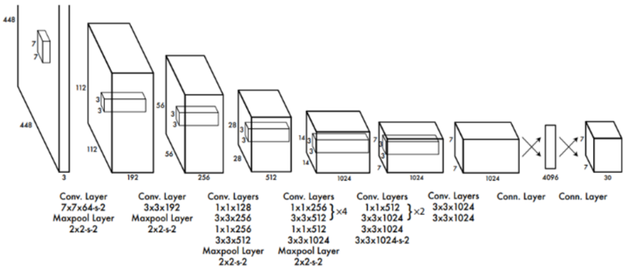


Figure 2. YOLO architecture [47]

The YOLO algorithm first divides the input image into NxN equal grids. All cells in the divided grids perform the task of predicting bounding boxes and predicting the class of objects. The YOLO architecture determines bounding boxes for each cell created, and bounding box can occur as many times as there are objects in the image. Basically, every bounding box has 5 parameters. The center coordinates of the box are "x" and "y", the height is "h", the width is "w", and the confidence score is "c". In the expression Y = [pc, bx, by, bh, bw, c1, c2] specified in the figure, the corresponding values of each parameter are given. Pc is the confidence score of the bounding box and is calculated by the Intersection over union (IoU) metric. Bx and by are the x and y coordinates of the center of the bounding box. Bh is the height value of the bounding box. Bw is the width value of the bounding box. Parameters c1 and c2 represent classes. Multiple bounding boxes may occur for an object during the detection process. IoU is used at this stage. Thanks to the determined IoU threshold value, the correct bounding boxes are found, and the others are ignored. If the number of detections obtained as a result of the IoU threshold value is more than one for an object, Non-Max suppression is applied at this stage. This process prevents the same image from being detected twice and the bounding box with the highest accuracy is used. A visual of the working principle of the YOLO model is given in Figure 3.

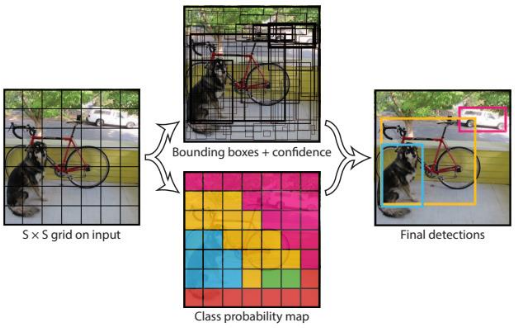


Figure 3. YOLO model [47]

2.2. YOLOv8 Algorithm

YOLOv8 architecture was developed in 2023. It has more accurate and faster detection ability compared to older models. It uses EfficientDent and PAN architectures. It is a model pre-trained with the COCO dataset. In Figure 4, YOLOv8 architecture is shown.

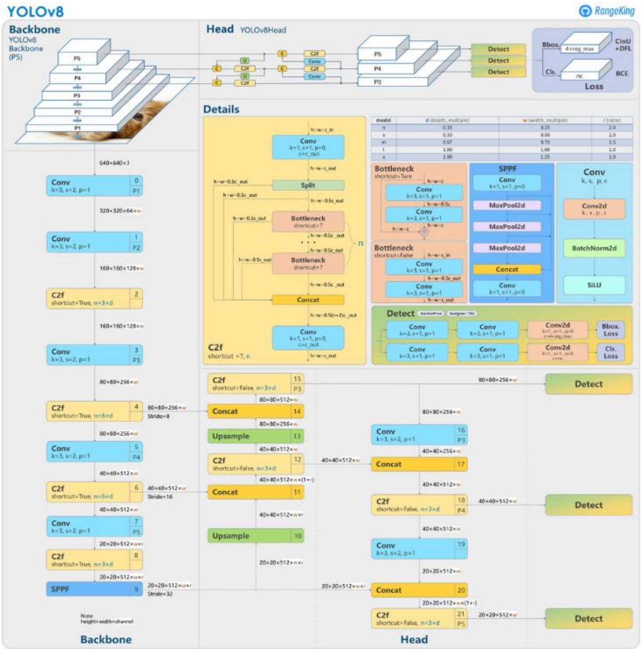


Figure 4. YOLOv8 architecture [48]

2.3. YOLOv9 Algorithm

YOLOv9 was developed by Chien-Yao Wang and colleagues in 2024. It is the newest of YOLO architectures. Wang and his colleagues introduced innovations to the YOLO architecture to increase computational efficiency and reduce information loss. This new architecture uses Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN). In the YOLOv9 architecture, PGI is used to prevent information loss in deep neural networks. In YOLOv9 architecture, GELAN is used for optimization. PGI architecture is shown in Figure 5.

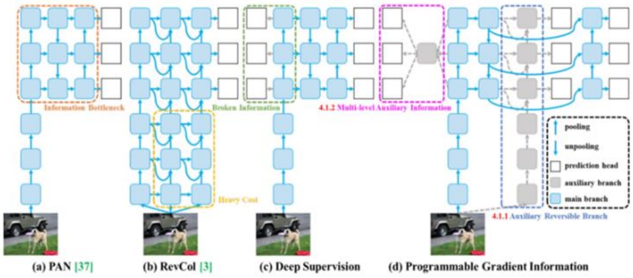


Figure 5. PGI architecture [49]

GELAN architecture is shown in Figure 6.

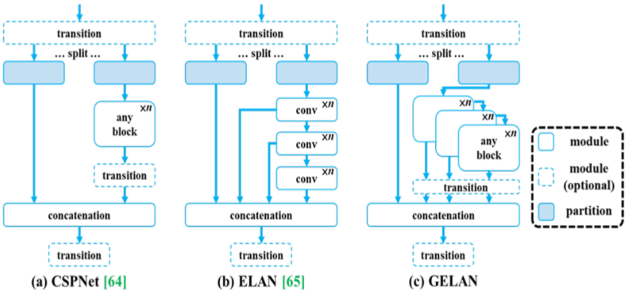
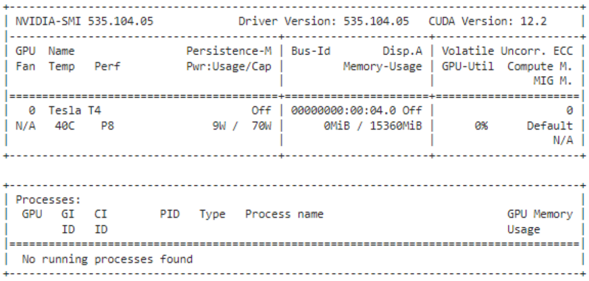


Figure 6. GELAN architecture [49]

2.4. Computer Environment

The presented study has been carried out using Tesla T4 GPU on the Google Colab platform. Information about the hardware used is given in Figure 7.



```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   40C    P8               9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

Figure 7. Computer environment

3. The Proposed Ship Detection Approach

The methods and process steps we used in our paper are stated in the new block diagram we created in Figure 8. We can divide our experimental preparation process and experiment into 2 stages.

In the first stage of our experimental preparation process, a literature review and dataset research were conducted. Deep learning models and datasets used in the literature were examined. To introduce a new approach in this field, studies have been carried out on deep learning models that have not been used before. Currently, the most recently published YOLOv8 and YOLOv9 models have become our convolutional neural network model preferences. In addition, real-time detection with YOLO architectures and the ability to detect multiple objects separately in the same image are the advantages of YOLO over other architectures in the literature. In many datasets in the literature, there is more than one ship type in the same photograph. Under all these conditions, the most appropriate model was determined to be used in our article. In terms of dataset selection, the FGSCR-42 dataset stands out due to the high number of photographs and ship types it contains. As a result of all evaluations, the FGSCR-42 dataset was found suitable to be used in our article. FGSCR-42 dataset consists of 10 main and 42 subcategories. Each category in the dataset was examined one by one. As a result of the review, incorrectly classified photographs were detected. Additionally, some of images were found to be inadequate in terms of clarity and information. A new dataset was created based on the FGSCR-42 dataset.

There are 8 subheadings under the main heading of aircraft carrier in the FGSCR-42 dataset. These 8 subheadings are gathered under a single heading named aircraft carrier. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photographs in the aircraft carrier category we have newly created are divided into 70% training, 15% testing, 15% validation.

The cruiser category in the FGSCR-42 dataset consists of a single main heading. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photos in the cruiser category we newly created are divided into 70% training, 15% testing, 15% validation.

In the FGSCR-42 dataset, there are 10 subheadings under the destroyer main heading. These 10 sub-headings are gathered under a single heading under the name of destroyer. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photos in the destroyer category we newly created are divided into 70% training, 15% testing, 15% validation.

Moreover, there are 3 subheadings under the main heading of assault ship in the FGSCR-42 dataset. These 3 sub-headings are gathered under a single heading under the name of assault ship. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photographs in the newly created assault ship category are divided into 70% training, 15% testing, 15% validation. There are 4 subheadings under the main heading of landing ship in the FGSCR-42 dataset. These 4 subheadings are gathered under a single heading called a landing ship. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photos in the landing ship category we newly created are divided into 70% training, 15% testing, 15% validation.

Besides there are 2 subheadings under the main heading of transport dock in the FGSCR-42 dataset. These 2 subheadings are gathered under a single heading called transport dock. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photos in the transport dock category we newly created are divided into 70% training, 15% testing, 15% validation.

In addition, there are 3 subheadings under the main heading of the support ship in the FGSCR-42 data set. Among these 3 subheadings, the medical ship class was taken. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photographs in the newly created medical ship category are divided into 70% training, 15% testing, 15% validation.

On the other hand, there are 2 subheadings under the main heading of the combat ship. These 2 subheadings are gathered under a single heading under the name of a combat ship. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photos in the newly created combat ship category are divided into 70% training, 15% testing, 15% validation.

In the FGSCR-42 dataset, there are 2 subheadings under the main heading of frigate. These 2 subheadings are gathered under a single heading under the name of frigate. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. The photos in the newly created frigate category are divided into 70% training, 15% testing, 15% validation.

Moreover, there are 7 subheadings under the main heading of civil vessels in the FGSCR-42 dataset. Container ship, towing vessel and tanker ship classes were taken from these 7 subheadings. The photographs were examined and images that were deemed inaccurate and inadequate for education were eliminated. Three new categories have been created: container ships, towing vessels, and tanker ships. Within these 3 new categories, photographs were divided into 70% training, 15% testing, 15% validation. A new dataset consisting of 7008 photographs and 12 categories was created. The categories are Aircraft carrier, cruiser, destroyer, assault ship, landing ship, transport dock, medical ship, combat ship, frigate, container ship, towing vessel, and tanker ship respectively. Each photo in the new dataset was tagged using the MakeSenseAI application in accordance with the YOLO format.

In the second stage, we move on to the experimental part. The CNN architectures chosen to evaluate the results of the dataset are YOLOv8 and YOLOv9. For the YOLOv8 model, YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l models, which are sub-packages of YOLOv8, were used. Since YOLOv9 architecture is a newly released architecture, not all packages are available at the time of the experiment. Therefore, YOLOv9-e and YOLOv9-c were used as subpackages in the YOLOv9 architecture. The training parameters of the models were chosen the same. 35 epoch, 640 image size, 8 workers, 8 batch parameters were used and they were trained using Tesla T4 GPU in the Google Colab environment.

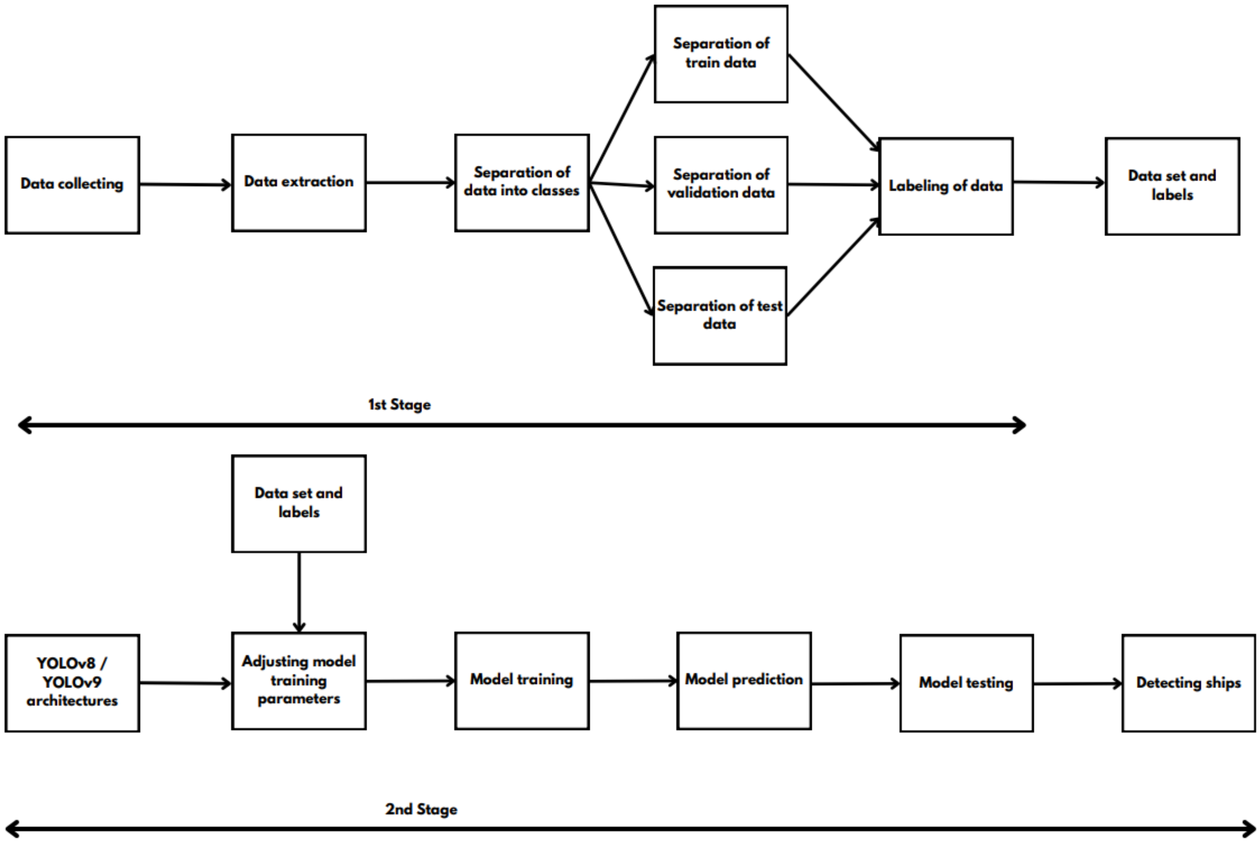


Figure 8. The proposed approach's block scheme

3.1. Dataset Preparation

During the preparation of the paper, all datasets in the literature were examined. Considering the small amount of data in this field and the difficulty of accessing the data, the FGSCR-42 dataset prepared by Di et al [39]. was chosen to be used in our paper. In addition to the DSCR, which is a remote sensing ship classification dataset they had previously prepared, they also added the data they received from the datasets used in the literature, NWPU VHR-10, DOTA, HRSC2016 and Google Earth, to create a 42 category. They created FGSCR-42 datasets consisting of 9320 photographs, including combat and civilian ships. They made the FGSCR-42 dataset they created publicly available. Make Sense AI, an open-source website, was used to label the data to be used in our paper. This site is an object tagging tool. The dataset we use in our paper is based on FGSCR-42. This dataset was examined and reduced to 12 classes to be used in our paper. A total of 7008 photographs were used in our paper. Figure 9 shows the classes in the dataset we created.

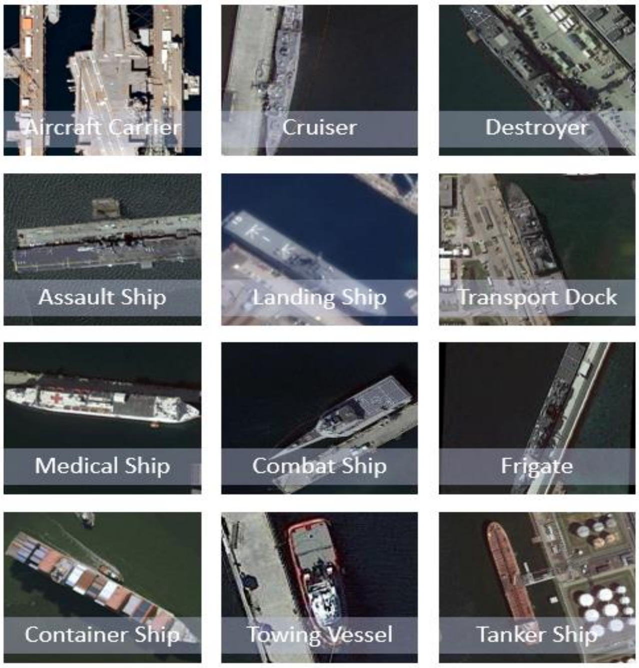


Figure 9. Ship classes in the dataset

4. Experimental Results

Table 1 shows the results obtained when ship detection was performed using different CNN models. Apart from the approach we recommend in the table, there are 12 models in total. Among these 12 models, the most successful algorithms are TASN with an accuracy rate of 93.51% and YOLO-RSSD with a mAP value of 96.1%.

VGG19, ResNet-50, DenseNet and ResNext are used in image classification, B-CNN, RA-CNN, DCL and TASN are used in fine-grained classification, and YOLO is used in object detection tasks. While the accuracy performance metric is used in image classification algorithms, the mAP performance metric is used in YOLO. Accuracy shows the ratio of correct predictions to all predictions in classification problems.

The mAP, on the other hand, is used to measure the performance of the model in object detection tasks. While image classification algorithms predict the class of a particular image, YOLO can detect both the class and location of the object in the image. Additionally, YOLO can detect multiple objects at the same time. When our data set is examined, there are photographs containing more than one number or type of ship in a single photograph. The use of YOLO was preferred in the article due to its suitability for these conditions and real-life scenarios. When we compare it with the other model results given in Table 1, our model has a better performance with a mAP value of 98.9%.

In the dataset we used in our article, Di et al. performed ship detection for all classes separately with the B-CNN model. The accuracy value they obtained with B-CNN is 89.53%.

Table 1. Algorithms Comparison Results

Model	Accuracy	Model	mAP
VGG19 [39]	77.36%	Improved – YOLOv3 [36]	93.56%
RESNET-50 [39]	87.24%	YOLOv4 [35]	93.55%
DenseNet [39]	88.69%	YOLO-RSSD [37]	96.1%
ResNext [39]	89.16%	YOLOv5 [37]	91.8%
B-CNN [39]	89.53%	YOLOv8l (**The Proposed approach**)	**98.9%**
RA-CNN [39]	91.63%		
DCL [39]	93.03%		
TASN [39]	93.51%		

Table 2 shows the mAP values we obtained with the YOLOv8 and YOLOv9 models we used in our article. When the YOLO architectures used in the article were compared, it was seen that the YOLOv8l model achieved the best results.

Table 2. Algorithms Comparison Results

The Proposed Model	mAP50	mAP50-95
YOLOv8-n	99%	86.1%
YOLOv8-s	98.9%	87.1%
YOLOv8-m	99%	87.8%
YOLOv8-l	**98.9%**	**88.1%**
YOLOv9-c	98.8%	85.9%
YOLOv9-e	98.9%	87.5%

In Table 3, the accuracy results obtained by Di et al. when they performed ship detection using B-CNN on FGSCR-42 data are given separately for each class. The results we obtained using the YOLOv8l model are given separately for each class in Table 3. When Table 3 is examined, it is seen that our model has high mAP values in all 12 classes. The result graph, which includes train loss, validation loss, precision, recall, mAP50, mAP50-95 graphs, is given in Figure 10. Figure 10 shows that the precision and recall metrics of our YOLOv8l model are like the mAP50 and mAP50-95 metrics and converge to unity, which shows that the models give successful results.

In addition, it is seen that the training loss graphs of the models are like the validation loss graphs and converge to zero, which is important for model success.

Table 3. B-CNN and YOLOv8l (The proposed approach) Algorithms' Performance Results

	Accuracy B-CNN Method [39]	mAP- YOLOv8l (**The proposed approach**)
Aircraft carrier	94.84%	98.9%
Cruiser	98.84%	97.8%
Destroyer	92.14%	94.9%
Assault ship	93.45%	99.4%
Landing ship	91.88%	99.5%
Transport dock	98.03%	99.5%
Medical ship	88.61%	99.5%
Combat ship	93.6%	99.5%
Frigate	87.56%	99.5%
Container ship	91.50%	99.5%
Towing vessel	61.70%	99.5%
Tanker ship	96.43%	99.5%

Figure 11 shows the number of correct and incorrect identifications according to classes in the confusion matrices of our YOLOv8l model. By using this matrix, it is possible to find accuracy, precision, recall and f1 score.

In the confusion matrix, the rows show the classes predicted by the model and the columns show the real classes. When the confusion matrix is examined, the concentration of predictions on a diagonal line shows the goodness of the model performance. It is possible to evaluate class-based performance by looking at the confusion matrix.

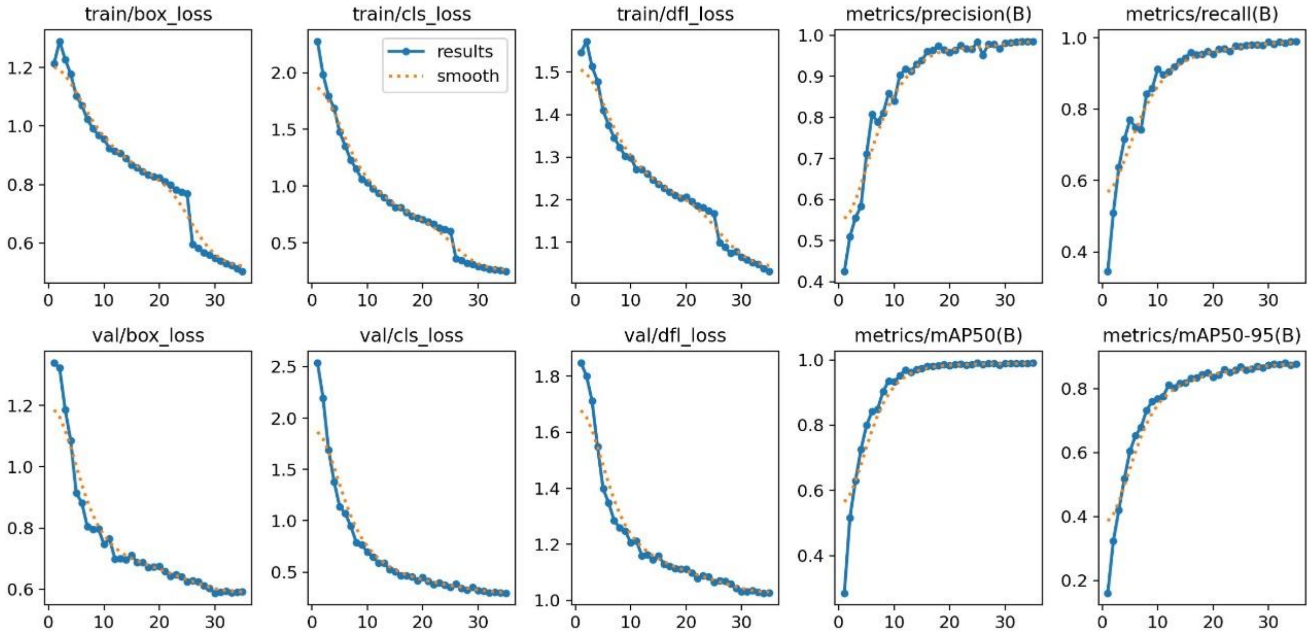


Figure 11. YOLOv8l confusion matrix

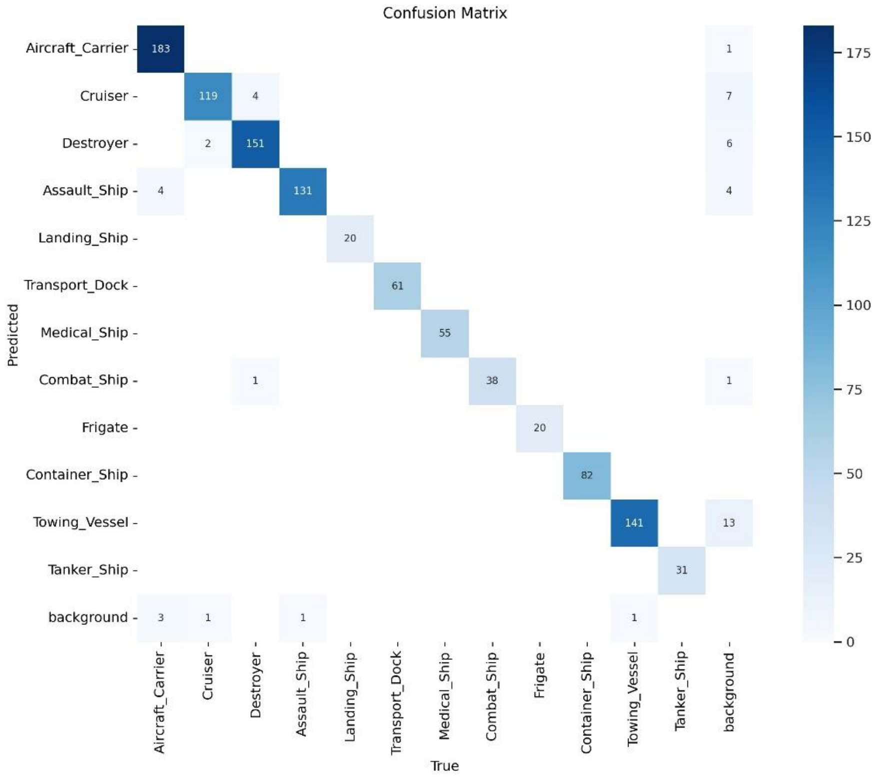


Figure 10. YOLOv8l result graphs

Using the YOLOv8l model, a study was conducted on test data that our model had not seen before in the dataset, we created. Figure 12, Figure 13 and Figure 14 show sample images from the results of the study we carried out on the test dataset using our YOLOv8l model. The pictures and results in these figures are an example of our model success.

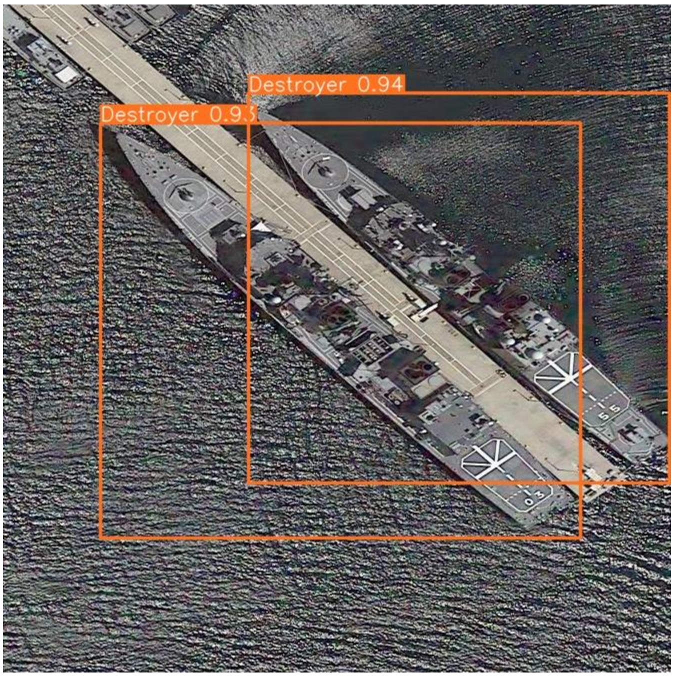


Figure 12. YOLOv8l test image

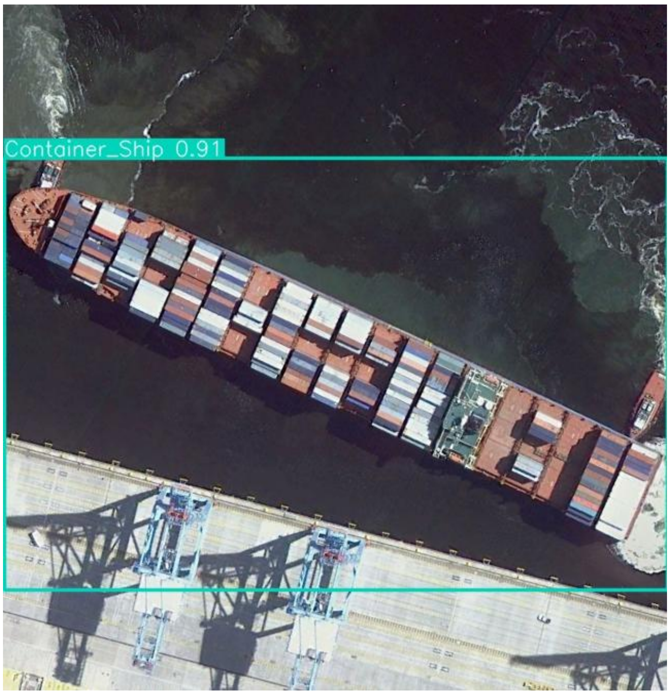


Figure 13. YOLOv8l test image

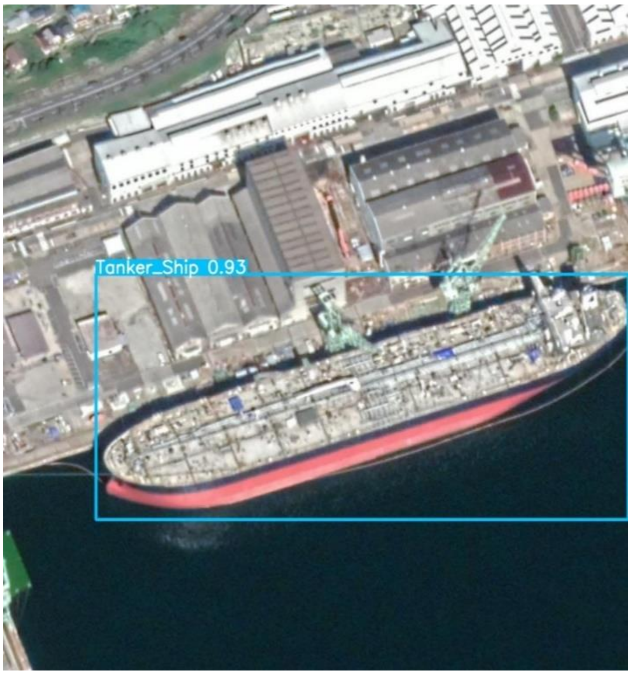


Figure 14. YOLOv8l test image

5. Discussion

In this paper, ship detection from optical satellite images was performed using YOLO models. When compared to similar projects, it was seen that our approach gave better results. A new dataset was created based on the FGSCR-42 data set. Misclassification errors in this dataset were eliminated. In this case, it contributed to the increase in the accuracy rate. In addition, since YOLO was developed for real-time object detection projects, it is possible to obtain more successful results by using YOLO in such projects. Ship detection methods from satellite images with CNN may face some challenges in real-world conditions, such as changing weather conditions and low resolution of satellite images.

6. Conclusion

The aim of this paper is to ship detection in optical satellite images using the CNN algorithm. A new dataset was created based on the FGSCR-42 dataset in this paper. A new dataset consists of 7008 photographs and 12 categories. The categories are Aircraft carrier, cruiser, destroyer, assault ship, landing ship, transport dock, medical ship, combat ship, frigate, container ship, towing vessel, and tanker ship respectively. Experiments have been conducted on YOLOv8 and YOLOv9 algorithms and packages of these algorithms. The necessary optimizations have been made for the YOLOv8 and YOLOv9 architectures to work in accordance with our dataset. All models were trained and compared with the same parameters. The mAP values of the results were compared among themselves and with other papers. In the experimental results, the highest accuracy rate was obtained using the YOLOv8l model. When we use our YOLOv8l model, the mAP value is 98.9%. It has been observed that we achieved a higher accuracy rate than ship detection studies using CNN algorithms in the literature.

Author contributions

Neslihan Toprak: Conceptualization, Methodology, Software, Field study, Writing-Original draft preparation. **Yıldıray Yalman**: Data curation, Methodology, Software, Visualization, Writing, Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

1. Marine Traffic. (n.d.). Live map. Retrieved August 7, 2024, from https://help.marinetraffic.com/hc/en-us/articles/204062548-Live-Map
2. IMEAK. (2023). Maritime sector report Istanbul 2023. Istanbul & Marmara, Aegean, Mediterranean, Black Sea Regions Chamber of Shipping.
3. Kayaalp, K., & Süzen, A. A. (2018). Derin öğrenme ve Türkiye'deki uygulamaları. IKSAD International Publishing House.
4. Fukushima, K. N. (1980). A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4), 193–202. https://doi.org/10.1007/BF00344251
5. Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology, 195(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455
6. Le Cun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791
7. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436–444. https://doi.org/10.1038/nature14539
8. Aydın, V. A. (2024). Comparison of CNN-based methods for yoga pose classification. Turkish Journal of Engineering, 8(1), 65–75. https://doi.org/10.31127/tuje.1275826
9. Cireşan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3642–2649. https://doi.org/10.48550/arXiv.1202.2745
10. Cireşan, D., Meier, U., Masci, J., & Gambardella, L. M. (2012). Flexible high-performance convolutional neural networks for image classification. Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 1237–1242. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210
11. Othman, M. M. (2023). Modeling of daily groundwater level using deep learning neural networks. Turkish Journal of Engineering, 7(4), 331–337. https://doi.org/10.31127/tuje.1169908
12. Meghraoui, K., Sebari, I., Bensiali, S., & Ait El Kadi, K. (2022). On behalf of an intelligent approach based on 3D CNN and multimodal remote sensing data for precise crop yield estimation: Case study of wheat in Morocco. Advanced Engineering Science, 2, 118–126.
13. Çubukçu, E. A., Demir, V., & Sevimli, M. F. (2023). Carbon Monoxide forecasting with artificial neural networks for Konya (Case Study of Meram). Engineering Applications, 2(1), 69–74.
14. Jain, S., Rustagi, A., Saurav, S., Saini, R., & Singh, S. (2021). Three-dimensional CNN-inspired deep learning architecture for Yoga pose recognition in the real-world environment. Neural Computing and Applications, 33, 6427–6441. https://doi.org/10.1007/s00521-020-05405-5
15. Singh, A. P., Singh, M., Bhatia, K., & Pathak, H. (2024). Encrypted malware detection methodology without decryption using deep learning-based approaches. Turkish Journal of Engineering, 8(3), 498–509. https://doi.org/10.31127/tuje.1416933
16. Grefenstette, E., Blunsom, P., Freitas, N. de, & Hermann, K. M. (2014). A deep architecture for semantic parsing. https://doi.org/10.48550/arXiv.1404.7296
17. Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL). https://doi.org/10.3115/v1/P14-1062
18. Kim, Y. (2014). Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751. https://doi.org/10.3115/v1/D14-1181
19. Kayıran, H. F. (2022). The function of artificial intelligence and its sub-branches in the field of health. Engineering Applications, 1(2), 99–107. Retrieved September 14, 2024, from https://publish.mersin.edu.tr/index.php/enap/article/view/328
20. Pajaziti, A., Basholli, F., & Zhaveli, Y. (2023). Identification and classification of fruits through robotic system by using artificial intelligence. Engineering Applications, 2(2), 154–163. Retrieved September 14, 2024, from https://publish.mersin.edu.tr/index.php/enap/article/view/974
21. Ertuğrul, Özgür L., & İnal, F. (2022). Assessment of the artificial fiber contribution on the shear strength parameters of soils. Advanced Engineering Science, 2, 93–100. Retrieved September 14, 2024, from https://publish.mersin.edu.tr/index.php/ades/article/view/172
22. Meghraoui, K., Sebari, I., Bensiali, S., & Ait El Kadi, K. (2022). On behalf of an intelligent approach based on 3D CNN and multimodal remote sensing data for precise crop yield estimation: Case study of wheat in Morocco. Advanced Engineering Science, 2, 118–126. Retrieved September 14, 2024, from https://publish.mersin.edu.tr/index.php/ades/article/view/329
23. Naumov, A., Khmarskiy, P., Byshnev, N., & Piatrouski, M. (2023). Methods and software for estimation of total electron content in ionosphere using GNSS observations. Engineering Applications, 2(3), 243–253. Retrieved September 14, 2024, from

https://publish.mersin.edu.tr/index.php/enap/article/view/1165

24. Mirbakhsh, A., Lee, J., Jagirdar, R., Kim, H., & Besenski, D. (2023). Collective assessments of active traffic management strategies in an extensive microsimulation testbed. Engineering Applications, 2(2), 146–153. Retrieved September 14, 2024, from https://publish.mersin.edu.tr/index.php/enap/article/view/929
25. Mema, B., Basholli, F., & Hyka, D. (2024). Learning transformation and virtual interaction through ChatGPT in Albanian higher education. Advanced Engineering Science, 4, 130–140. Retrieved September 14, 2024, from https://publish.mersin.edu.tr/index.php/ades/article/view/1509
    26. Yüksek, G., Muratoğlu, Y., & Alkaya, A. (2022). Modelling of supercapacitor by using parameter estimation method for energy storage system. Advanced Engineering Science, 2, 67–73. Retrieved September 14, 2024, from https://publish.mersin.edu.tr/index.php/ades/article/view/98
27. Kaya, Y., Şenol, H.İ., Yiğit, A.Y., & Yakar, M. (2023). Car detection from very high-resolution UAV images using deep learning algorithms. Photogrammetric Engineering & Remote Sensing, 89(2), 117-123. https://doi.org/10.14358/PERS.22-00101R2
28. Akar, Ö, Saralioğlu, E., Güngör, O., & Bayata, H. F. (2024). Semantic segmentation of very-high spatial resolution satellite images: A comparative analysis of 3D-CNN and traditional machine learning algorithms for automatic vineyard detection. International Journal of Engineering and Geosciences, 9(1), 12-24. https://doi.org/10.26833/ijeg.1252298
29. Mahdavifard, M., Ahangar, S. K., Feizizadeh, B., Kamran, K. V., & Karimzadeh, S. (2023). Spatio-Temporal monitoring of Qeshm mangrove forests through machine learning classification of SAR and optical images on Google Earth Engine. International Journal of Engineering and Geosciences, 8(3), 239-250. https://doi.org/10.26833/ijeg.1118542
30. Demirgül, T., Demir, V., & Sevimli, M. F. (2024). Farklı makine öğrenmesi yaklaşımları ile Türkiye'nin solar radyasyon tahmini. Geomatik, 9(1), 106-122. https://doi.org/10.29128/geomatik.1374383
31. Hazer, A., Bozdağ, A., & Atasever, Ü. H. (2024). Hiper-optimize edilmiş makine öğrenim teknikleri ile taşınmaz değerlemesi, Yozgat kenti örneği. Geomatik, 9 (3), 299-312. https://doi.org/10.29128/geomatik.1454915
32. Günen, M. A., & Beşdok, E. (2023). Effect of denoising methods for hyperspectral images classification: DnCNN, NGM, CSF, BM3D and Wiener. Mersin Photogrammetry Journal, 5(1), 1-9. https://doi.org/10.53093/mephoj.1213166
33. Demirel, Y., & Türk, N. (2024). Automatic detection of active fires and burnt areas in forest areas using optical satellite imagery and deep learning methods. Mersin Photogrammetry Journal, 6(2), 66-78. https://doi.org/10.53093/mephoj.1575877
34. Gharechelou, S., Tateishi, R., Sri Sumantyo, J. T., & Johnson, B. A. (2021). Soil moisture retrieval using polarimetric SAR data and experimental observations in an arid environment. ISPRS International Journal of Geo-Information, 10(10), 711. https://doi.org/10.3390/ijgi1010071
35. Sakshaug, S. E. H. (2013). Evaluation of polarimetric SAR decomposition methods for tropical forest analysis. University of Tromsø.
36. Wang, B., Han, B., & Yang, L. (2021). Accurate real-time ship target detection using YOLOv4. 2021 6th International Conference on Transportation Information and Safety (ICTIS), 222–227. https://doi.org/10.1109/ICTIS54573.2021.9798495
37. Hong, Z. H., et al. (2021). Multi-scale ship detection from SAR and optical imagery via a more accurate YOLOv3. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 6083–6101. https://doi.org/10.1109/JSTARS.2021.3087555
38. Si, J., Song, B., Wu, J., Lin, W., Huang, W., & Chen, S. (2023). Maritime ship detection method for satellite images based on multiscale feature fusion. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 16, 6642–6655. https://doi.org/10.1109/JSTARS.2023.3296898
39. Di, Y., Jiang, Z., & Zhang, H. (2021). A public dataset for fine-grained ship classification in optical remote sensing images. Remote Sensing, 13(4), 747. https://doi.org/10.3390/rs13040747
40. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1409.1556
41. 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 (CVPR), 770–778.
42. Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5987–5995. https://doi.org/10.1109/CVPR.2017.634
43. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2261–2269. https://doi.org/10.1109/CVPR.2017.243
44. Lin, T. Y., Chowdhury, A. R., & Maji, S. (2015). Bilinear CNN models for fine-grained visual recognition. Proceedings of the International Conference on Computer Vision (ICCV).
45. Fu, J., Zheng, H., & Mei, T. (2017). Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4476–4484. https://doi.org/10.1109/CVPR.2017.476
46. Chen, Y., Bai, Y., Zhang, W., & Mei, T. (2019). Destruction and construction learning for fine-grained image recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and

Pattern Recognition (CVPR), 5152–5161. https://doi.org/10.1109/CVPR.2019.00530

47. 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 (CVPR), 779–788. https://doi.org/10.1109/CVPR.2016.91
48. GitHub. Ultralytics. Retrieved August 7, 2024, from https://github.com/ultralytics/ultralytics?ref=blog.roboflow.com
49. Wang, C.-Y., Yeh, I.-H., & Hong-Yuan, M. L. (2024). YOLOv9: Learning what you want to learn using programmable gradient information. https://doi.org/10.48550/arXiv.2402.13616