Ship Object Detection: Preliminary Report

Kerem Adalı

May 9, 2025

1 Introduction

This preliminary report outlines the approach for the Ship Object Detection project as part of the Introduction to Artificial Intelligence course (Summer 2025). The objective is to develop a solution for detecting ships and boats from aerial images, which falls under the category of object detection tasks in computer vision.

2 Dataset Description

2.1 Overview

The dataset[1] consists of aerial images containing ships and boats of various sizes, types, and orientations. Based on initial exploration, the following observations can be made:

2.2 Descriptive Statistics

- **Image count:** 661 images in the dataset
- **Image resolution:** Varies, but typically around 500x500 pixels
- **Color format:** RGB (3 channels)
- **Annotation format:** PASCAL VOC format
- **Object categories:** Single class (”boat”)
- **Objects per image:** Ranges from 1 to 15+ ships per image

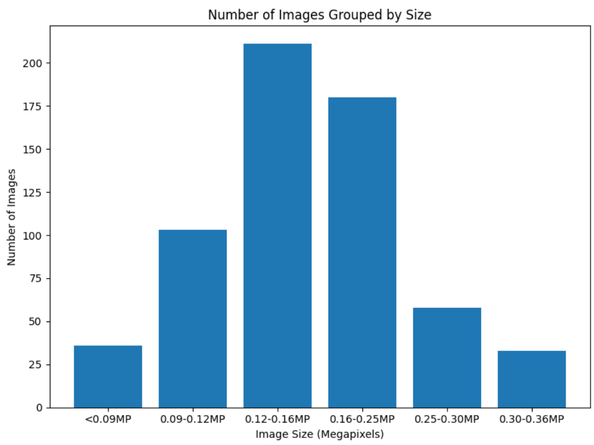
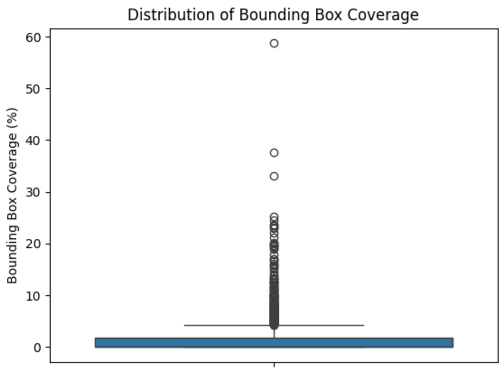


Figure 1: Number of Images grouped by size

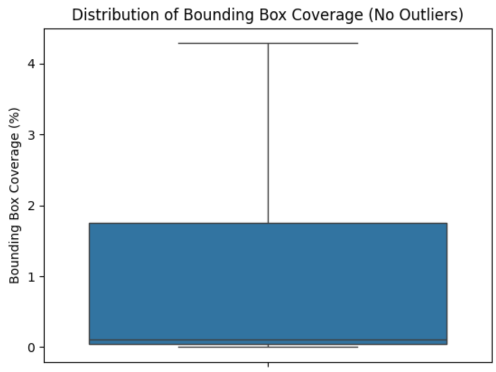
2.3 Sample Visualizations

Initial visual analysis reveals several interesting patterns:

- Ships vary in size, from small boats to large vessels
- Images contain diverse backgrounds including open water, harbors, and coastal areas
- Some images contain tightly clustered ships (port scenarios)
- Varying angles of capture (directly overhead vs. angled aerial views)
- Each ship bounding box covers approximately 2-4% of the image for the majority of the time, although having outliers



((a)) Bounding box coverages, all ships



((b)) Bounding box coverages, without outliers

Figure 2: Bounding box coverages of ships on the image

2.4 Challenges Identified

- Small objects: Some ships appear very small in images taken from high altitude
- Occlusion: Ships in harbors may partially occlude each other
- Background confusion: Some coastal structures or waves may resemble ships
- Varying lighting conditions: Reflections on water surfaces
- Different orientations: Ships appear at various angles

3 Proposed Approach

3.1 Dataset Splitting

The dataset will be split as follows:

- Training set: 70% of the data (approximately 435 images)
- Validation set: 15% of the data (approximately 93 images)
- Test set: 15% of the data (approximately 93 images)

The splitting will be performed using random sampling to ensure a balanced distribution of scenarios (open water, harbor, etc.) and ship densities across all sets.

3.2 Selected Algorithms and Models

For this project, I plan to implement and compare multiple object detection algorithms:

3.2.1 YOLOv8

YOLOv8[3] is the probably the most famous in the YOLO (You Only Look Once) family, known for its excellent balance between speed and accuracy. Usually used for real-time solutions.

3.2.2 RetinaNet with ResNet50 backbone

RetinaNet model with ResNet50 backbone fine-tuned on COCO in 800x800 resolution with FPN features created from P5 level.

3.2.3 DETR model with ResNet-50 backbone

DETR (DEtection TRansformer)[2] is an end-to-end object detection architecture that uses a transformer encoder-decoder with a ResNet-50 backbone.

3.3 Implementation Tools and Framework

- **Primary framework:** PyTorch with Torchvision
- **Object detection libraries:**
  - Ultralytics for YOLOv8 or Keras with YOLOv8 backbone
  - Keras for RetinaNet with ResNet50 bacbkbone
  - transformers library from HuggingFace for DETR
- **Data preprocessing:** OpenCV
- **Evaluation metrics:** Torchmetrics, pycocotools
- **Visualization:** Matplotlib, seaborn
- **Training environment:** My computer with NVIDIA RTX 4050 Laptop GPU (45W), no notebooks
- **Version control:** GitHub

3.4 Data Preprocessing Pipeline

- **Image normalization:** Standard scaling (mean subtraction and division by standard deviation)
- **Data augmentation:**
  - Random horizontal and vertical flips
  - Random rotation ($\pm 15^\circ$)
  - Random brightness and contrast adjustments
  - Random cropping (maintaining all objects)
- **Image resizing:** Resize to 640×640 for YOLOv8, 800×800 for other models

4 Evaluation Methods

4.1 Performance Metrics

- **Primary metrics:**
  - Mean Average Precision (mAP) at IoU thresholds of 0.5 (mAP@0.5)
  - Mean Average Precision (mAP) at IoU thresholds from 0.5 to 0.95 with a step of 0.05 (mAP@0.5:0.95)
- **Secondary metrics:**
  - Precision and Recall curves
  - F1-score at various confidence thresholds
  - Inference speed (ms or FPS)

4.2 Model Comparison

The models will be compared across the following dimensions:

- Detection accuracy (using the metrics defined above)
- Performance across different ship sizes (small, medium, large)
- Computational efficiency (inference time etc.)

5 Experimental Setup

5.1 Hyperparameter Tuning

For each model, the following hyperparameters will be explored:

- Learning rate (initial and scheduling)
- Batch size
- Number of training epochs
- Confidence threshold for detection

5.2 Training Strategy

- Initial training with default parameters
- Performance analysis
- Iterative hyperparameter tuning
- Final model training with optimal parameters

6 Conclusion

This preliminary report outlines the approach for developing a ship object detection system using various state-of-the-art algorithms. The focus will be on understanding the dataset characteristics, implementing and comparing different detection architectures, and optimizing the models for the specific challenge of detecting ships in aerial imagery.

The project will provide insights into the strengths and weaknesses of different object detection approaches for this specific domain, as well as practical experience in developing and evaluating computer vision systems for real-world applications.

References

[1] Ships dataset.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[3] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.