



# Ship Detection in Sentinel-1 SAR Imagery using Deep Learning

**Deepesh Yadav**

27.01.2026

Assignment

## 1. INTRODUCTION

Monitoring ship activity is an important task for applications such as port management, maritime safety, and surveillance. Optical satellite images are often affected by cloud cover and lighting conditions, which limits their usefulness in many real-world scenarios. Synthetic Aperture Radar (SAR) overcomes these limitations by providing all-weather, day-and-night imaging capability.

This project focuses on detecting ships in SAR imagery using a deep learning-based approach. A YOLOv8 object detection model is trained on an open-source SAR ship dataset and then applied to real Sentinel-1 satellite imagery. The work demonstrates a complete end-to-end pipeline, starting from data preparation and model training to inference on real SAR scenes.

### 1.1. Reason for Choosing This Problem Statement

I chose the problem statement on ship detection in SAR imagery because it combines two areas I am genuinely interested in: computer vision and satellite-based remote sensing. Compared to standard image datasets, SAR imagery presents unique challenges such as speckle noise, different intensity distributions, and the absence of color information. These challenges make the problem more interesting and closer to real-world conditions.

This problem statement also allowed me to work with real satellite data from

Sentinel-1 and understand how deep learning models behave outside curated training datasets. The task required building a complete pipeline, including data preparation, model training, and inference on real SAR scenes, which helped me gain practical experience beyond model training alone.

Overall, I selected this problem statement because it reflects real operational challenges in space and maritime applications and provided an opportunity to apply computer vision techniques in a meaningful and practical context.

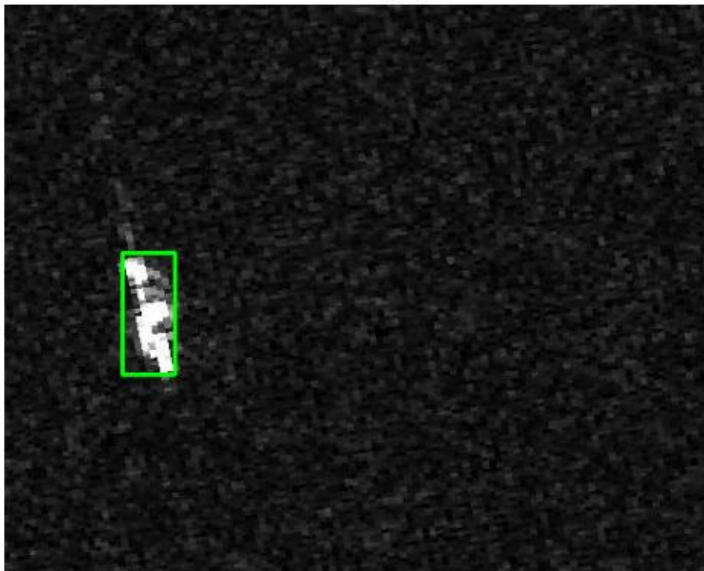
## 2. Dataset Description

### 2.1. SAR Ship Detection Dataset (SSDD)

The SAR Ship Detection Dataset (SSDD) is a publicly available dataset commonly used for ship detection research. It consists of SAR image chips collected from different sensors and resolutions, with bounding box annotations marking ship locations. The dataset includes both near-shore and offshore scenes, making it suitable for training a general ship detection model.

In this project, the bounding-box version of SSDD was used. Since the original annotations are not directly compatible with YOLO, the dataset was converted into YOLO format before training.

000003.jpg



## 2.2. Dataset Conversion and Preparation

The **SSDD** annotations were parsed and converted into YOLO format by normalizing bounding box coordinates with respect to image dimensions. The dataset was organized into training and validation splits using the official SSDD image lists.

This conversion step was essential to make the dataset compatible with the YOLOv8 training pipeline while preserving the original annotation quality.

## 3. Sentinel-1 Data Acquisition

Sentinel-1 SAR imagery was obtained programmatically using the **Google Earth Engine (GEE) Python API**, ensuring reproducibility and open-access data usage.

**Sentinel-1 Ground Range Detected (GRD)** products were selected, as they are widely used for maritime and ship detection applications. The image collection was filtered using the following criteria:

- Instrument mode: **IW (Interferometric Wide)**
- Polarization: **VV**
- Orbit direction: **Descending**
- Spatial extent: Defined Area of Interest (AOI)
- Temporal range: Multiple acquisitions to support averaging

A **small maritime Area of Interest (AOI)** near the **Panama Canal** was selected, with approximate center coordinates:

- **Latitude:** 8.81° N
- **Longitude:** -79.49°

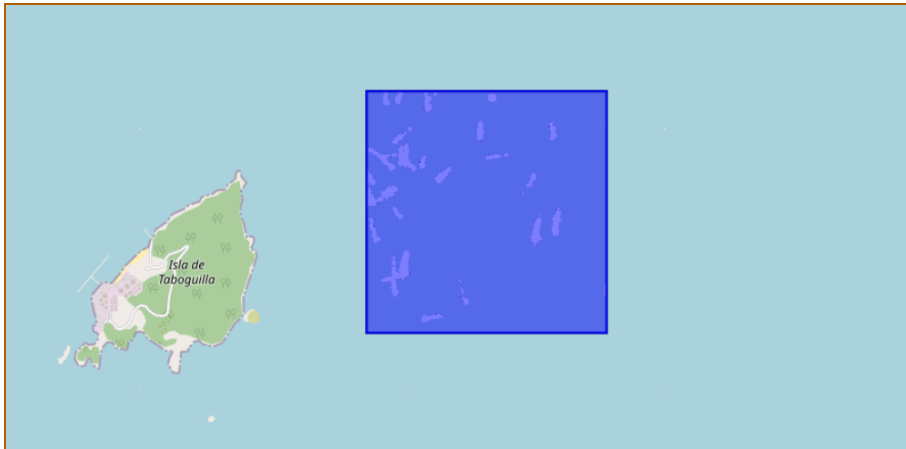
This AOI was chosen for the following reasons:

- The region experiences **continuous and dense ship traffic**, increasing confidence that ships would be present within the Sentinel-1 revisit period.
- Selecting a known high-traffic region is more reliable than choosing a random open-ocean area for qualitative evaluation.
- A compact AOI helps **reduce computational load** during inference, as

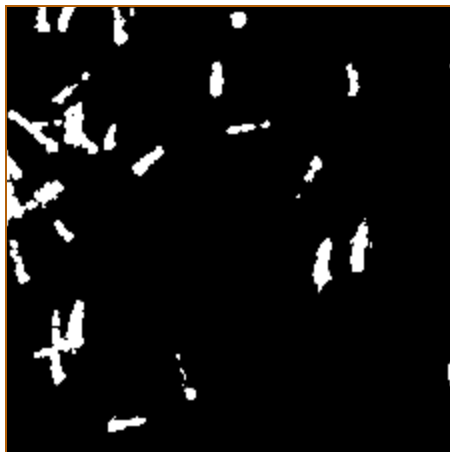
larger areas require extensive tiling and higher processing time.

To reduce speckle noise and stabilize ship signatures, a **temporal mean composite** of the filtered Sentinel-1 images was generated before exporting the data.

The final processed SAR image was then exported from GEE for downstream preprocessing and inference using the trained deep learning model.



*fig: AOI on GEE*



*fig: This is the SAR Image from the Sentinel-1*

#### 4. SAR Preprocessing

SAR imagery requires specialized preprocessing before it can be used in deep learning models. The exported Sentinel-1 data was **first converted to decibel (dB) scale**. Pixel values were then **clipped to a range of -25 dB to 0 dB**, which

represents typical ocean and ship backscatter values.

Invalid values such as NaN and infinite pixels were handled explicitly to avoid numerical instability. The image was normalized to a 0–1 range and converted to 8-bit format. Since YOLO expects three-channel input, the single-channel SAR image was replicated across three channels.

These preprocessing steps ensured that the SAR data could be safely and effectively used for inference.

## **5. Model Architecture and Training**

### **5.1. YOLOv8 Detector**

I have selected YOLOv8 as the object detection model due to its strong performance and efficiency. The YOLOv8-small variant was used as it provides a good balance between accuracy and computational cost. The model was initialized with pretrained weights and fine-tuned on the SSDD dataset for ship detection.

### **5.2. Training Setup**

Model training was carried out on Google Colab using an T4 GPU. The training configuration included an input image size of  $640 \times 640$  pixels, a batch size of 16, and training for 100 epochs using the AdamW optimizer with an initial learning rate of 0.001. Early stopping with a patience of 20 epochs was enabled to prevent overfitting by terminating training when validation performance no longer improved.

Standard data augmentation techniques provided by the YOLO framework were applied automatically during training.

### **5.3. Training Results**

The training and validation curves show stable convergence of the model. Both training and validation losses decrease consistently, indicating that the model learned meaningful ship features without severe overfitting. Precision and recall values increase steadily, and the final mean Average Precision (mAP) scores

indicate strong detection performance.

## 6. Quantitative Evaluation Metrics

### 6.1. Evaluation Metrics Used

To evaluate the performance of the ship detection model, standard object detection metrics were used. These metrics measure both the accuracy of localization and the correctness of ship detection.

The following metrics were computed on the validation dataset:

- **Precision (P):** Measures how many detected ships are actually ships.
- **Recall (R):** Measures how many true ships were successfully detected.
- **mAP@0.5:** Mean Average Precision at 50% IoU threshold, commonly used to measure detection accuracy.
- **mAP@0.5:0.95:** Mean Average Precision averaged over multiple IoU thresholds, providing a stricter and more comprehensive evaluation.

These metrics together provide a balanced view of the model's detection capability.

### 6.2. Validation Results

After training for 100 epochs, the model achieved strong performance on the validation set. Precision and recall values remained consistently high, indicating that the model detects ships reliably while producing very few false positives.

The high mAP@0.5 score demonstrates accurate ship localization, while the mAP@0.5:0.95 score shows that the model maintains good performance even under stricter overlap conditions.

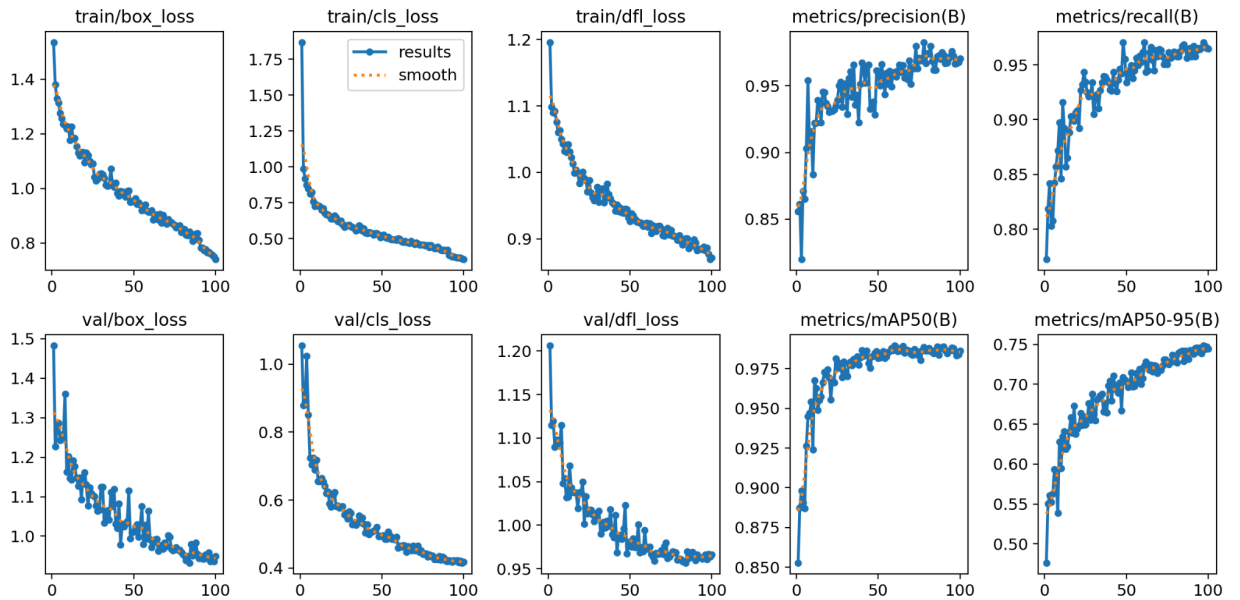
*Table 1: Validation Performance Metrics*

Metric	Value
Precision (P)	0.967
Recall (R)	0.971
mAP@0.50	0.986
mAP@0.50:0.95	0.749

### 6.3. Training and Validation Behavior

The training and validation loss curves show smooth convergence over epochs. Both box loss and classification loss decrease steadily, with validation curves closely following training curves. This indicates stable learning and no significant overfitting.

Precision and recall curves increase rapidly during early epochs and stabilize toward the end of training. This behavior suggests that the model learns ship features efficiently and generalizes well to unseen data.



*Fig: YOLOv8 training and validation curves*

#### **6.4. Discussion of Results**

The evaluation results indicate that the trained YOLOv8 model performs very well on SAR ship detection tasks. The high precision suggests that most detections correspond to actual ships, while the high recall indicates that very few ships are missed.

The slightly lower mAP@0.5:0.95 score compared to mAP@0.5 is expected in SAR imagery due to ship orientation, varying ship sizes, and speckle noise. Despite these challenges, the model demonstrates robust detection performance.

Overall, the quantitative metrics confirm that the trained model is suitable for real-world SAR-based ship detection applications.

### **7. Inference on Real Sentinel-1 SAR Data**

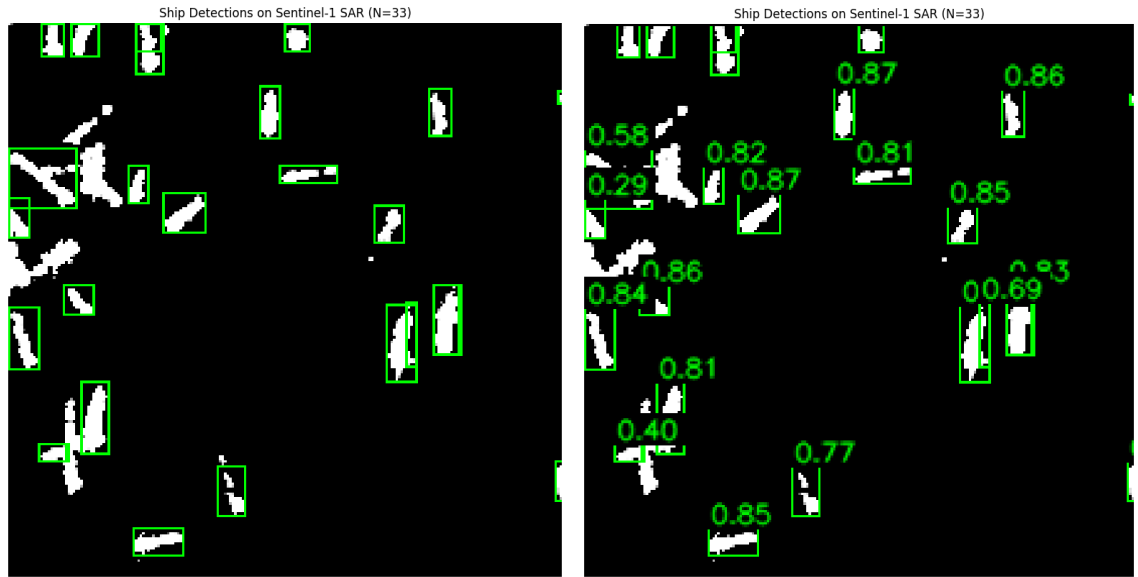
Inference on real Sentinel-1 SAR imagery presents additional challenges compared to curated training datasets. The exported SAR image was divided into overlapping tiles using a sliding-window approach to handle large image sizes. Padding was applied when necessary to ensure sufficient spatial context for detection.

YOLO inference was performed independently on each tile. The resulting detections were mapped back to the original image coordinates. To remove duplicate detections caused by overlapping tiles, a global Non-Maximum Suppression (NMS) step was also applied.

#### **7.1. Final Detection Results**

The final detection results show that the trained model successfully identifies ships in real Sentinel-1 SAR imagery. Ships appear as bright scatterers on the ocean surface and are accurately localized using bounding boxes.

In the selected Panama Canal region, approximately 33 ships were detected, which is consistent with expected maritime traffic density in this area.



*fig: Final ship detections on Sentinel-1 SAR imagery*

## 8. Challenges and Observations

Several challenges were encountered during this project. First, SAR data is fundamentally different from optical imagery, and improper normalization or handling of invalid values can lead to unstable model behavior. Careful preprocessing was required to make Sentinel-1 data usable for deep learning inference.

Second, training the YOLO model required significant computational resources. Training on large SAR datasets was not feasible locally, so cloud-based GPU resources were used. Managing training time, memory usage, and dataset organization required multiple iterations.

Finally, applying the trained model to real Sentinel-1 scenes introduced domain shift issues. The SSDD dataset consists of cropped image chips, while real Sentinel-1 images contain varying resolutions, speckle noise, and land clutter. Sliding-window inference and global NMS were essential to obtain reliable results.

## 9. Conclusion and Future Work

This project demonstrates a complete end-to-end pipeline for ship detection in SAR imagery using deep learning. A YOLOv8 detector trained on SSDD was successfully applied to real Sentinel-1 data, producing realistic and accurate ship detections.

Future work could include rotated bounding box detection for elongated ships, multi-temporal ship tracking, and integration with AIS data for quantitative validation. Overall, the results show that deep learning models trained on curated SAR datasets can generalize effectively to real satellite imagery when appropriate preprocessing and inference strategies are applied.