Deep Learning–Based Ship Detection using SAR Imagery

Deep Learning for Computer Vision Mini-Project

Submitted by:

Omowonuola Molayosi Akintola
Rabina Twayana
Ethel Ogallo

Submitted To:

Minh-Tan Pham, IRISA-UBS

13 Jan 2026

Photo Source : SARScope.Kaggle Dataset, 2023

Co-funded by the European Union

Background

- Ship detection plays a vital role in maritime monitoring and security.
- Synthetic Aperture Radar (SAR) enables day-and-night, all-weather observation of ships, overcoming the limitations of optical sensors such as cloud cover and illumination dependence (Gupta et al., 2024).
- Deep learning enables end-to-end, noise-robust ship detection in SAR imagery by automatically learning discriminative features (Qiao et al., 2025).
- The main objective of this project is object detection using different deep learning models

Dataset

Data were accessed from Kaggle SARscope: Synthetic Aperture Radar Maritime Images. It has VV and VH channel.

Key Features

01 Dual-purpose dataset: Suitable for ship detection and instance segmentation

02 Image size: 640 × 640

03 17,708 ship instances

04 Dataset size: 398.17 MB

05 Standard format: Annotations provided in COCO (MMDetection-compatible) format

Data Splitting

Total Images		**Train (70%)**		**Validation (20%)**		**Test (10%)**
6735	=	4717	+	1346	+	672

Model : YOLOV8

- Released: 10 January 2023 by Ultralytics
- Single-stage, anchor-free detector for fast and robust object detection.
- Improves generalization across different object shapes
- Support multi-task such as object detection, instance segmentation, image classification pose estimation, etc.
- Model sizes: nano, small, medium, large, extra-large

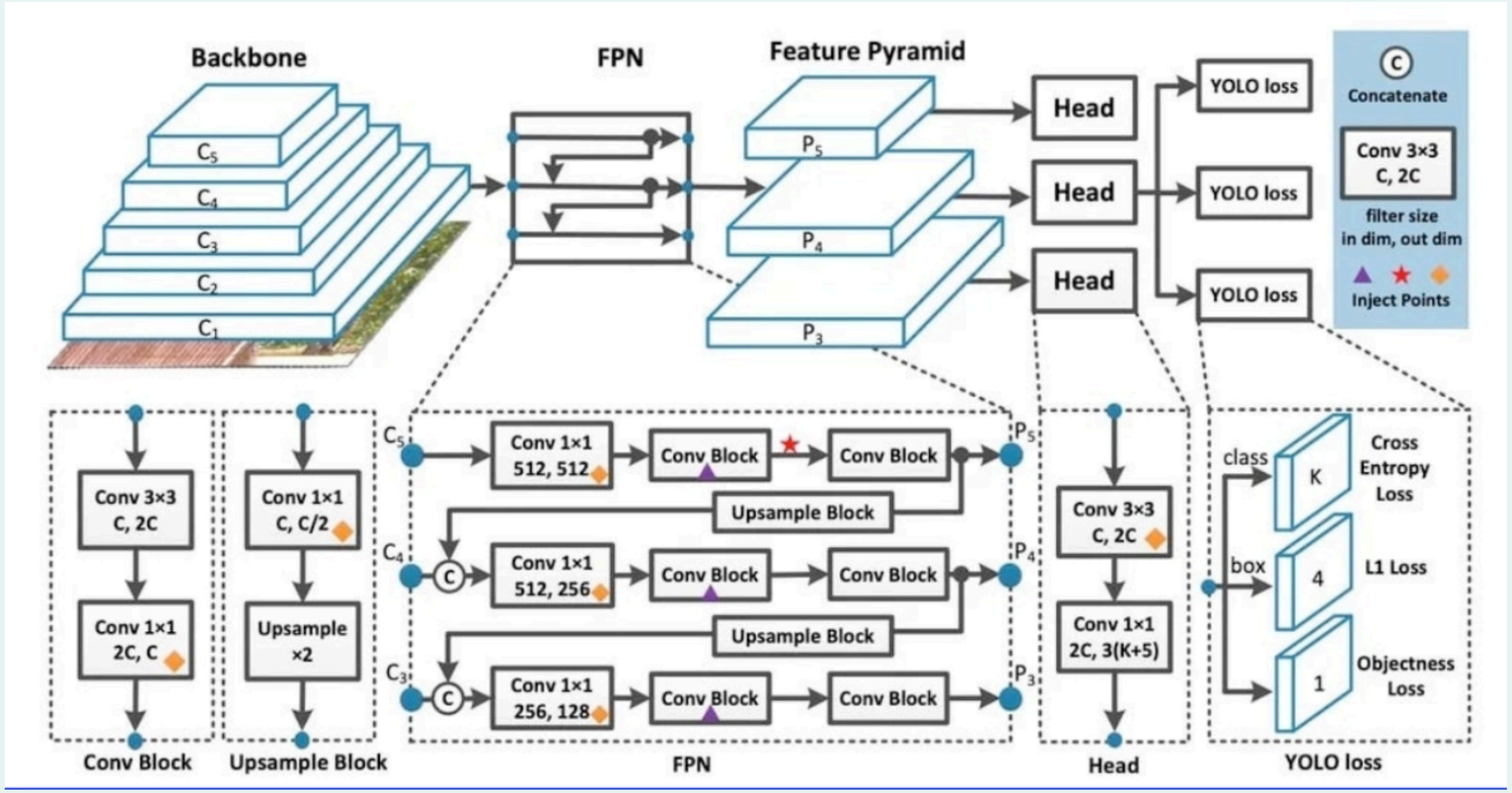


Figure 1: YOLOv8 Architecture (Source: YOLOv8, 2024)

Architecture

- **Backbone** (Feature Extraction Layer): Integrates an advanced CSP (Cross Stage Partial) bottleneck module to reduce computation and improve feature reuse
- **Neck** (Feature Aggregation for Detection): Fuse features at multiple scales using PANet-like structure for more accurate detection of objects of all sizes
- **Head** (Final Detection Output): Fully anchor-free head for faster, more robust training without predefined boxes

Model : RT-DETR

- **R**eal-Time **DE**tection **TR**ansformer (RT-DETR) , released in 2023, was developed by Baidu.
- Based on the DETR which is a NMS free framework
- Model sizes: large and extra-large

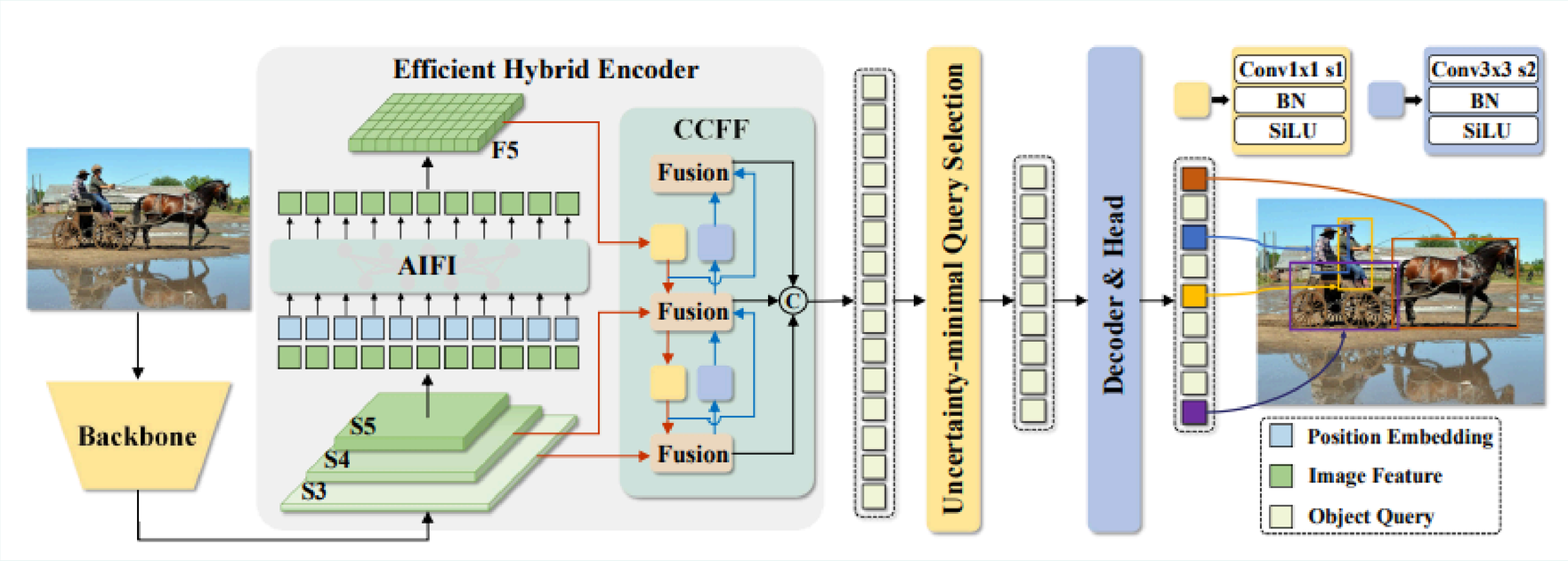


Figure 2: RT-DETR Architecture (Zhao et al., 2024)

Architecture:

- **Backbone**: It is built on the **Vi**sion **T**ransformer architecture (ViT)
- **Hybrid Encoder** : consists of two modules, the Attentionbased Intra-scale Feature Interaction (AIFI) which enhances local feature representations within each scale while the CNNbased Cross-scale Feature Fusion (CCFF) fuses features across scales using CNN operations to generate a sequence of image features.
- **Uncertainity-minimal Query Selection**: selects a fixed number of encoder features to serve as initial object queries for the decode
- **Decoder&Head**: NMS free and Anchor-free head that iteratively refines queries to directly output categories and bounding boxes eliminating the need for post-processing

Model : Faster-RCNN

- Released: 2015 by Ren, He, Girshick, and Sun (Microsoft Research)
- Two-stage object detection: first propose regions, then classify
- Uses Region Proposal Network (RPN) to generate candidate boxes
- Shares convolutional features between stages for efficiency
- Original backbone: VGG16

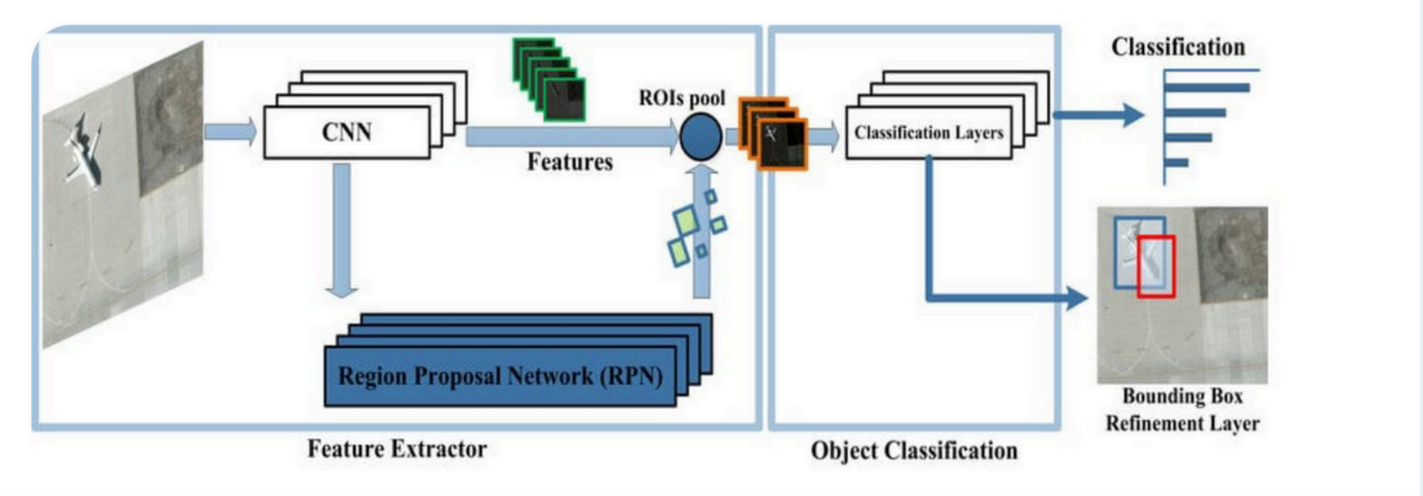


Figure 2: Faster R-CNN Architecture (Boesch, 2024)

Architecture:

- **Backbone (Feature Extraction Layer)**: Originally VGG16; extracts feature maps from input image
- **Region Proposal Network (RPN)**: Generates candidate object regions using anchor boxes of multiple scales and aspect ratios; predicts objectness score and box coordinates
- **RoI Pooling**: Converts variable-sized region proposals into fixed-size feature vectors
- **Head (Detection Network)**: Classifies each proposal and refines bounding box location

Model Training: YOLOv8

- Trained on YOLOv8m (Medium) and YOLOv8l (large)
- Hyperparameters :
  - batch size = 16
  - number of epochs = 20

Notes:

- Batch size value was fixed considering GPU memory and resources limit in kaggle
- Number of epoch choice due to time limitations

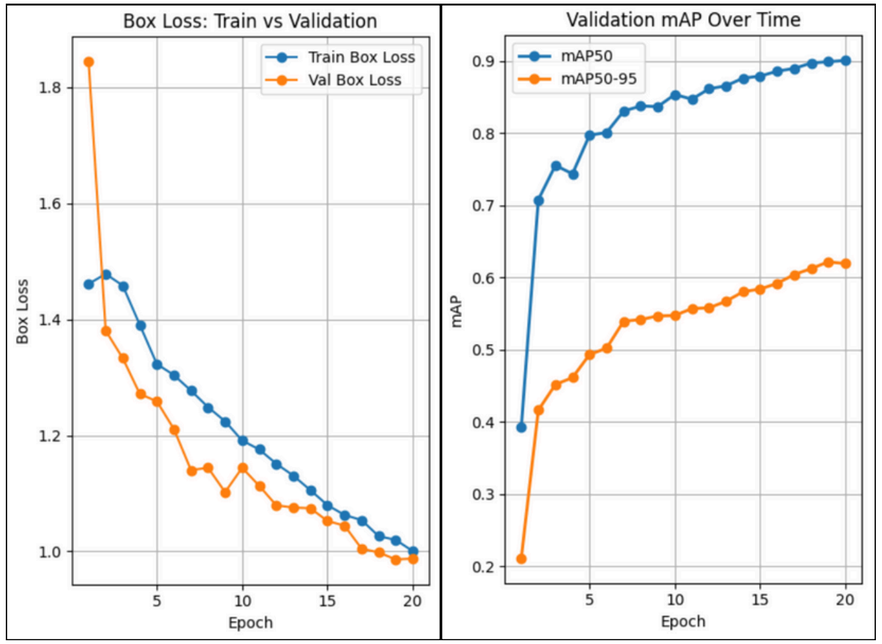


Figure 1: YOLOv8m Training loss and mAP

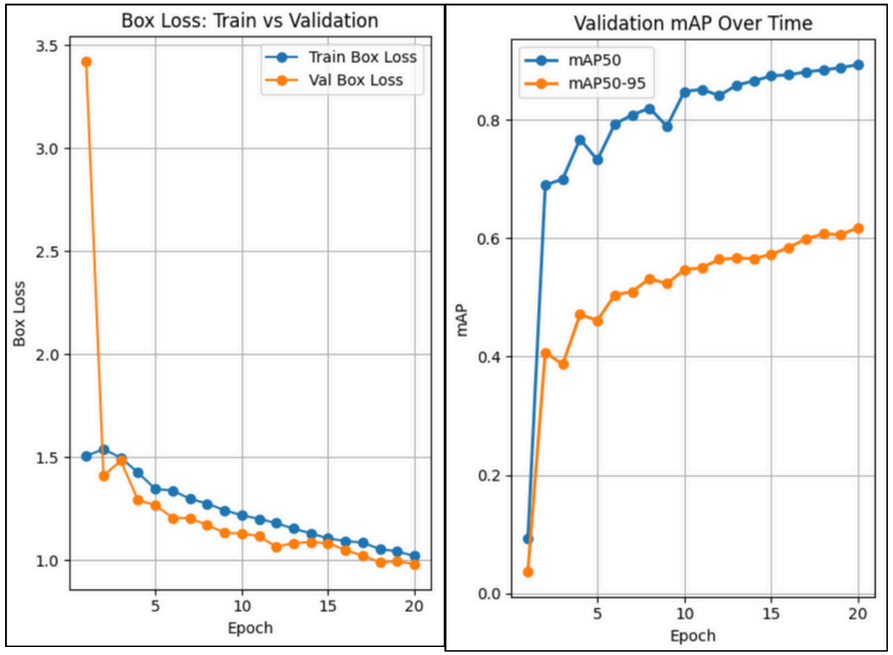


Figure 2: YOLOv8l Training loss and mAP

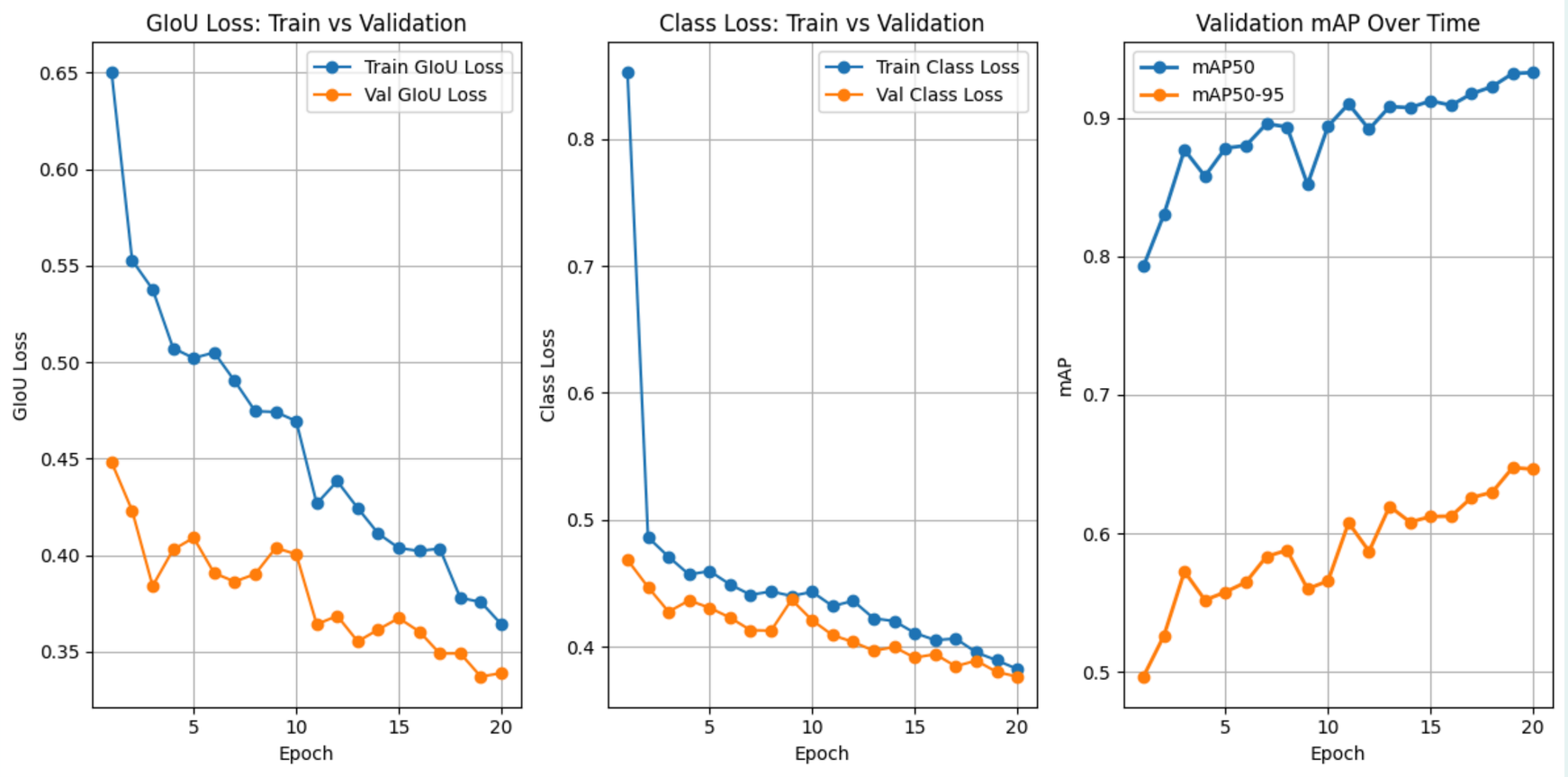
- Box Loss
  - Both models show smooth decrease in train & val loss
  - YOLOv8l starts with slightly higher val loss but drops marginally in 2nd epoch
- mAP Behavior:
  - YOLOv8m has slightly higher mAP50 as compared YOLOv8l
  - Much faster early gains of mAP50-95 in YOLOv8l but nearly same in 20 epoch
- Both validation and training loss continues decreasing, therefore more training epoch is necessary for better result

Model Training: RT-DETR

- Trained on RT-DETR large
- Hyperparameters :
  - batch size = 16
  - number of epochs = 20

Notes:

- Batch size value was fixed considering GPU memory and resources limit in kaggle while choice on number of epochs limited by time constraints.



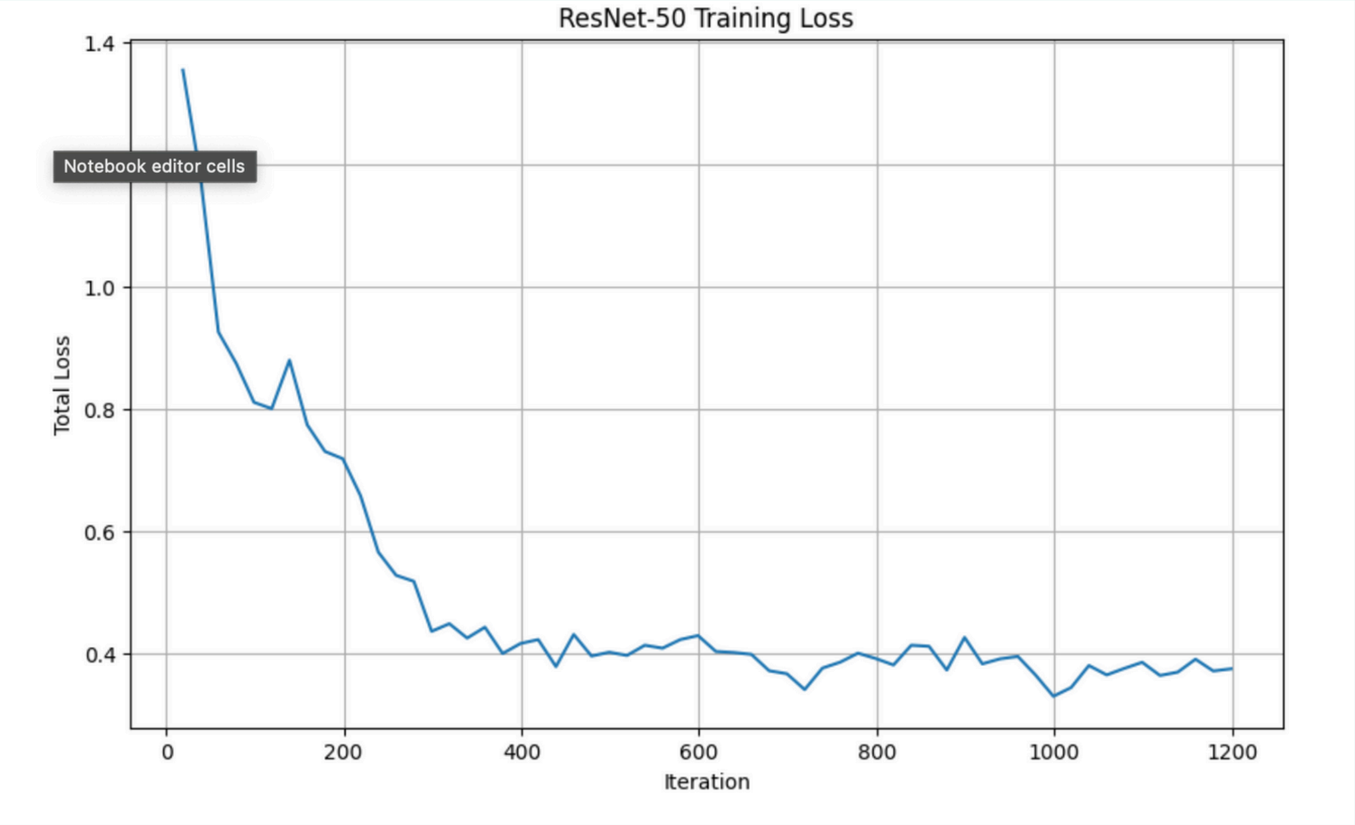
- **Loss**: Both validation and training loss continue decreasing indicating model performance is improving.
- **mAP** :
  - mAP50 = ~90% indicates strong detection at IoU threshold 0.5
  - mAP50-95 = ~60% shows good performance across stricter IoU thresholds
- Continued decrease of loss suggests increasing the number of epochs to enhance model performance

Model Training: Faster R-CNN

- Framework: Detectron2
- Config: R50-FPN-3x (ResNet50 backbone with Feature Pyramid Network)
- Hyperparameters :
  - batch size = 16
  - number of iterations = 1200
  - ROI: 128

Notes:

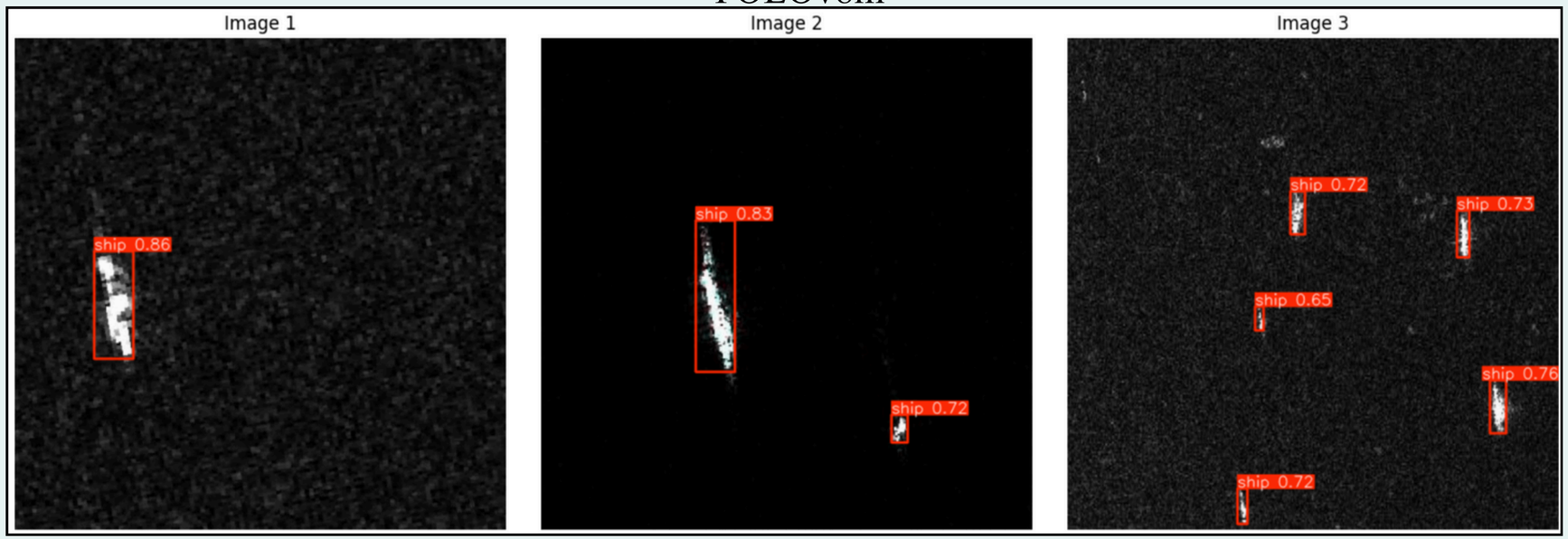
- Batch size value was fixed considering GPU memory and resources limit in kaggle. We initially started with 32 batch size but kaggle ran out memory.
- Number of iterations choice due to time limitations



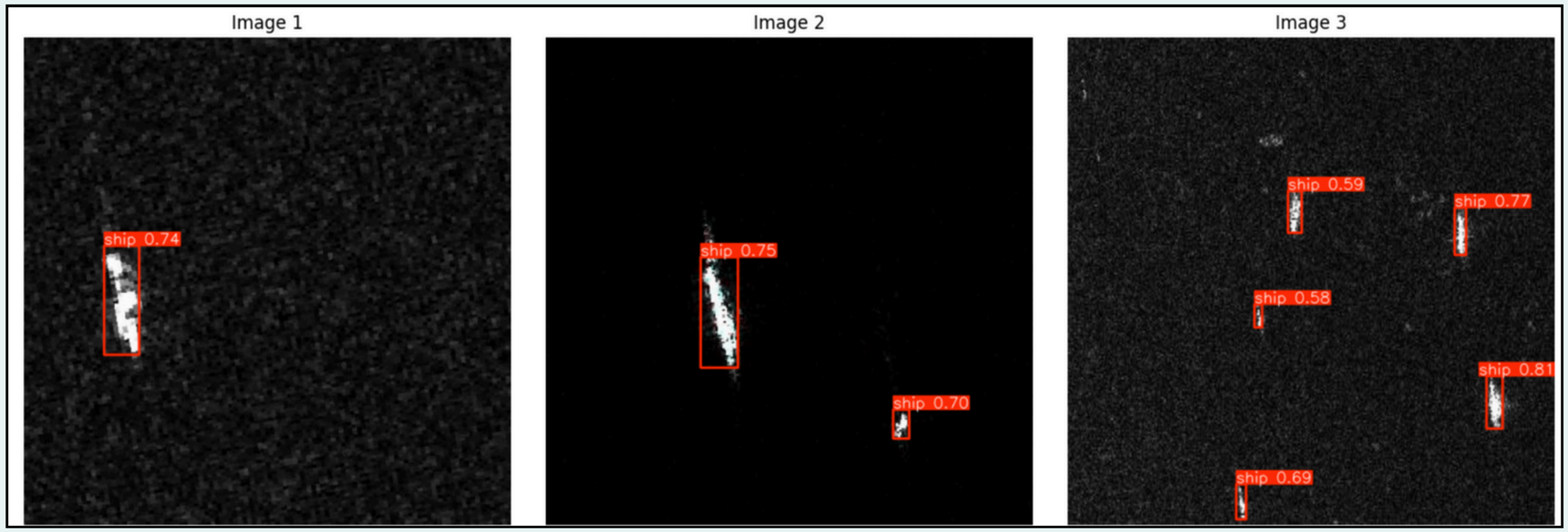
- Total Loss
  - Sharp drop from ~1.4 to ~0.8 as the model rapidly learns ship features from COCO pre-trained weights
  - Loss stabilizes around 0.35-0.45 with minor fluctuations, indicating the model is fine-tuning.
- Unlike YOLO (Ultralytics) which computes validation after each epoch, Detectron2 uses iteration-based training with validation/testing performed separately after training completes.
- The slight downward trend at iteration 1200 suggests additional training could yield further improvements.

Results

YOLOv8m



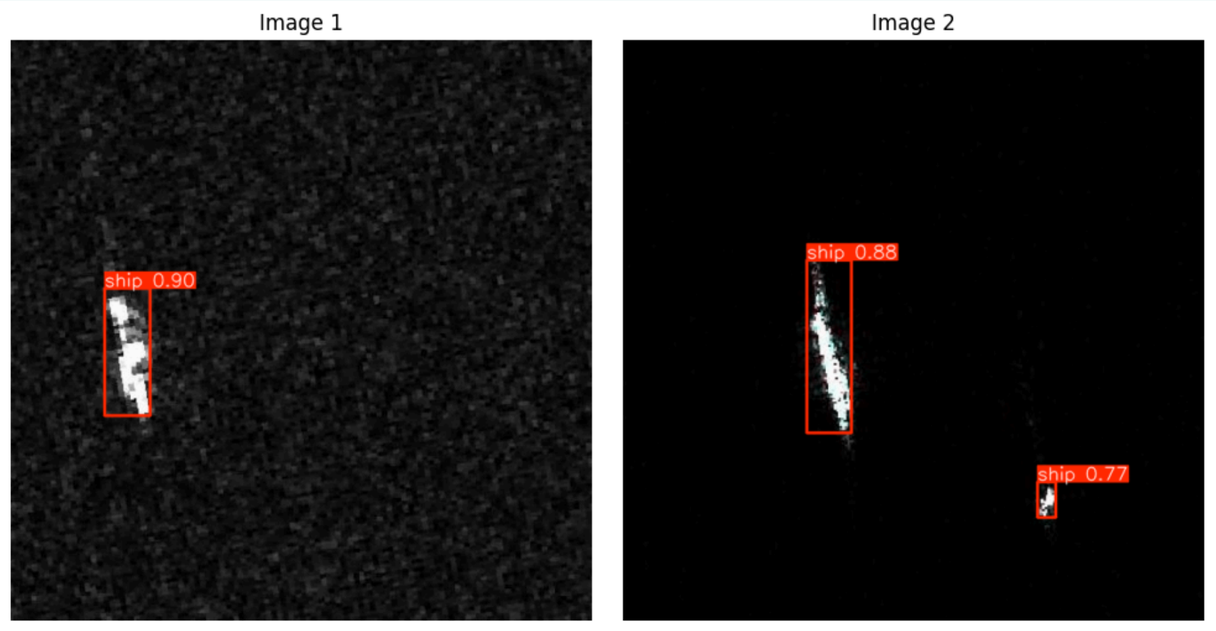
YOLOv8l



Inference results on test dataset

- Confidence Threshold of 0.5 was set to filter detections to reduce false positives.
- YOLOv8m generally outputs slightly higher confidence scores compared to YOLOv8l
- Both models accurately identify ships of various sizes but YOLOv8m tends to make more confident predictions.

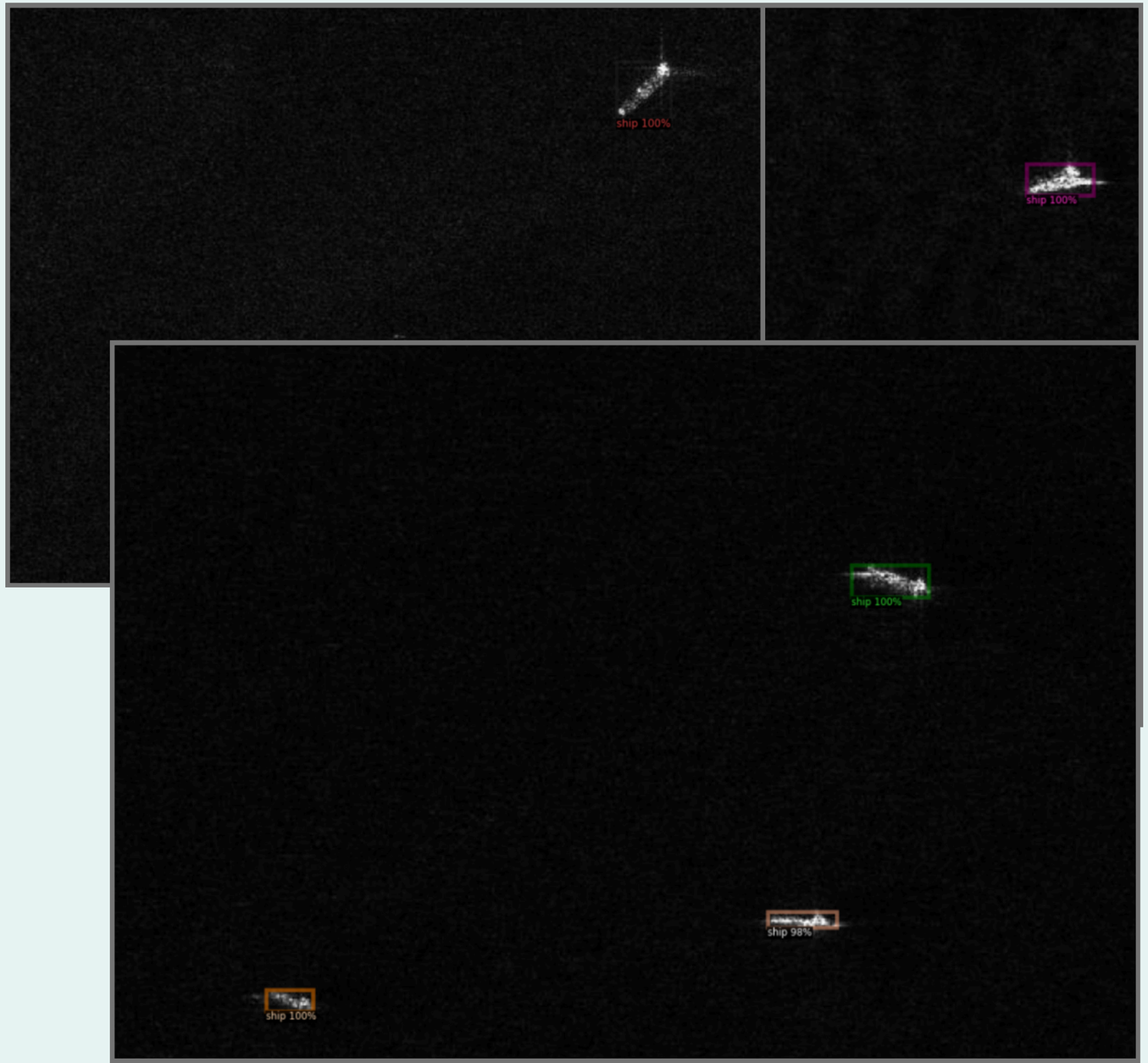
Results: RT-DETR



Inference Results on Test Dataset:

- Confidence Threshold of 0.5 was set during inference to filter detections reducing false positives.
- The model achieving overall precision of 0.90 indicating fewer false positives, with a recall of 0.85 indicating that the model is making correct detections.
- The model demonstrates a strong performance on inference with mAP50 = 91.5% and mAP50-95 = 62.5%

Results: Faster R-CNN



Inference Results:

- Confidence Threshold of 0.5 was set during inference to filter detections reducing false positives.
- The model demonstrates good generalization, as evidenced by comparable validation and test performance.
- The model also performs well across object scales, with strongest performance on medium-sized objects, APm = 64%, large-sized objects of APl = 59% ,though slightly reduced, performance on small objects, with 46% on the test set.

Model Performance Comparison

Model	#Parameters (Million)	GFLOPs	Epoch/Iterations	Batch Size	Training Time (min)	Inference Time Per Image (ms)	AP (%)		AP_{50} (%)	
							Val	Test	Val	Test
YOLOv8m	25.90	79.3	20	16	58	**24.7**	62.2	60.5	89.9	88.4
YOLOv8l	43.69	165.7	20	16	87	42.9	61.8	60	89.3	87.7
RT-DETR-l	32.81	108	20	16	122	48.5	64.7	**62.5**	93.2	**91.5**
Faster-RCNN	41.07	-	1200 (Iter)	16	43	87.2	51.9	51.6	79.2	80.1

- RT-DETR achieves the highest detection accuracy on both validation and test sets, outperforming YOLOv8 models at the cost of longer training and inference time.
- YOLOv8m and YOLOv8l provide faster training and inference, making them suitable for speed-essential applications.
- Faster R-CNN shows lower accuracy and efficiency, despite comparable model size, highlighting the advantage of single-stage and transformer-based detectors.

Conclusion

- In our experiment, the Transformer-based RT-DETR model outperforms both YOLO and Faster R-CNN in object detection performance.

- Current training was limited to a few epochs due to time and resource constraints; training for more epochs could provide a better understanding of model performance.

- Hyperparameter tuning can further improve the model's accuracy and efficiency.

Reference

- Gupta, H., Verma, O.P., Sharma, T.K. et al. Ship detection using ensemble deep learning techniques from synthetic aperture radar imagery. Sci Rep 14, 29397 (2024). https://doi.org/10.1038/s41598-024-80239-y
- Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs Beat YOLOs on Real-time Object Detection (No. arXiv:2304.08069). arXiv. https://doi.org/10.48550/arXiv.2304.08069
- Qiao, S., Zhang, Q., & Wang, Z. (2025). A Review of Deep-Learning-Based SAR Image Ship Interpretation Technology: The Latest Advances. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18, 26152–26185. https://doi.org/10.1109/JSTARS.2025.3614504
- YOLOv8. (2024, January 13). What is YOLOv8? Exploring its cutting-edge features. Retrieved from https://yolov8.org/what-is-yolov8/
- Boesch, G. (2024, October 5). The fundamental guide to Faster R-CNN. Viso.ai. https://viso.ai/deep-learning/faster-r-cnn-2/