

RESEARCH

Object detection and ship classification using YOLO and Amazon Rekognition

Alejandro Perez and Sikha Bagui*

Hal Marcus College of Science and Engineering, University of West Florida, Pensacola, FL, United States

***Correspondence:**Sikha Bagui,
bagui@uwf.edu**Received:** 26 July 2023; **Accepted:** 08 August 2023; **Published:** 31 August 2023

The need for reliable automated object detection and classification in the maritime sphere has increased significantly over the last decade. The increased use of unmanned marine vehicles necessitates the development and use of an autonomous object detection and classification systems that utilize onboard cameras and operate reliably, efficiently, and accurately without the need for human intervention. As the autonomous vehicle realm moves further away from active human supervision, the requirements for a detection and classification suite call for a higher level of accuracy in the classification models. This paper presents a comparative study using different classification models on a large maritime vessel image dataset. For this study, additional images were annotated using Roboflow, focusing on the types of subclasses that had the lowest detection performance, a previous work by Brown et al. (1). The present study uses a dataset of over 5,000 images. Using the enhanced set of image annotations, models were created in the cloud using Google Colab using YOLOv5, YOLOv7, and YOLOv8 as well as in Amazon Web Services using Amazon Rekognition. The models were tuned and run for five runs of 150 epochs each to collect efficiency and performance metrics. Comparison of these metrics from the YOLO models yielded interesting improvements in classification accuracies for the subclasses that previously had lower scores, but at the expense of the overall model accuracy. Furthermore, training time efficiency was improved drastically using the newer YOLO APIs. In contrast, using Amazon Rekognition yielded superior accuracy results across the board, but at the expense of the lowest training efficiency.

Keywords: YOLO, Amazon Rekognition, classification models, image processing, maritime vehicles

1. Introduction

The demand for automated object detection and classification in the maritime sphere has increased significantly over the last decade. Unmanned marine vehicle use has increased at a large rate as humans increasingly strive to automate previously dangerous or impossible-to-execute jobs in open waters. These tasks necessitate the development and use of autonomous object detection and classification systems that utilize onboard cameras and operate reliably, efficiently, and accurately without the need for human intervention. As the autonomous vehicle realm moves further away from active human supervision, the requirements for a detection and classification suite call for a higher level of accuracy in the classification models.

Brown et al. (1) created an object detection model for large maritime vessels using a publicly available dataset of maritime images from (2) and (3). This dataset was imported into Roboflow (4), where a subset was annotated with bounding boxes and labeled using five different classification labels for large seafaring vessels (“Container Ships,” “Cargo Ships,” “Military Ships,” “Tankers,” and Roll-On-Roll-Off ships or “RORO”). This annotated subset was used to train an object recognition and classification model by utilizing the open-source Ultralytics (5, 6) vision AI library YOLOv5 (7, 8). Their results produced a model that yielded an overall 0.892 average precision rate and 0.902 average recall rate; however, while the model proved accurate in detecting vessels with “Cruise Ship,” “Military Ship,” and

“RORO” labels with 93, 98, and 98% correct classification percentages, respectively, it performed considerably worse in classifications of “Container Ships” and “Tankers” with a correct classification percentage of 86 and 72%, respectively (1).

The aim of this paper is to explore methods to improve the efficiency and accuracy of this classification model, making special targets for the improvement of the two classification labels that had the lowest correct classification percentages. To achieve this:

First, Roboflow was used to improve the quality of the training data. This was done by enhancing the annotated subset with more bounded and labeled images from the Analytics Vidhya dataset. Special emphasis was placed on images that were labeled with the Container Ship and Tanker labels as those represented the largest target for improvement. Furthermore, the entire enhanced annotated subset would also be subject to preprocessing methods like auto orientation, image resizing, grayscale transformation, and image augmentation methods like brightness adjustments and noise reduction.

Second, in addition to training the model again with YOLOv5 (8) to establish a baseline and to test whether improvements were achieved by improving the training data quality, models were trained using YOLOv7 (9) and YOLOv8 (10) in an attempt to translate the improvements in the YOLO libraries over to the creation of more efficient and accurate models. Furthermore, the same enhanced annotated set would be used with Amazon Rekognition (11) computer vision API from Amazon Web Services (11) to train another detection and classification model, which would serve as a baseline comparison against the models created from the YOLO libraries.

Section 2 of this paper will explore some background on the YOLO family of algorithms and describe the key differences between the three versions, as well as the improvements achieved in the latest version as it pertains to the goals of this paper. Furthermore, this section also includes some background on Amazon Rekognition computer vision API and its uses as they relate to the work being done in this work.

This paper is unique in its comparative approach of different classification models using a large maritime vessel image database. In contrast to the original paper (1), where a single YOLO model was utilized, it creates three additional classification models and provides an efficiency and accuracy comparison between them to arrive at the best possible results. Furthermore, it utilizes a premium service in Amazon Rekognition to provide a different alternative to the YOLO approach. The “Related Works” Section 3 of this paper goes into further details about the difference of this paper with other published work. The rest of this paper is presented as follows: Section 4 presents the methodology; Section

5 explores the results; and finally, Section 6 details the conclusions.

2. Background on YOLO and Amazon Rekognition

YOLO is a family of compound-scaled object detection algorithms, which stands for “You Only Look Once” (5, 12). The YOLO object detection models are trained using the “Common Objects in Context” COCO (13) dataset of pre-trained objects and weights. These neural network-based models are used to detect objects in images or videos in real time by making predictions of bounding boxes and class probabilities with the use of a single fully contacted layer, in a single iteration. In essence, YOLO works by dividing the input image into a grid of cells and predicting bounding boxes and object scores for each cell. It then refines the predictions and applies non-maximum suppression to eliminate duplicate detections.

This paper uses three different evolutions of the YOLO family of algorithms as a comparative study on how the differences among them work to create more efficient and accurate object detection and classification models for large maritime vessels.

YOLOv5 (8) is the fifth iteration of the YOLO family of object detection algorithms and is the natural extension of the YOLOv3 PyTorch repository (14). YOLOv5 ported the previous YOLO version Darknet weights into PyTorch (12) and made further improvements in speed, performance, and ease of use.

YOLOv7 (9) was released two years later and promised to deliver increased efficiency and model performance with key improvements in the way it handles the COCO (13) pre-trained weight object dataset (15). YOLOv7 uses more floating point operations than YOLOv5; hence, the increased arithmetic computational operations make it optimized to be used with state-of-the-art GPUs, delivering faster processing speeds than the YOLOv5 versions when training on high-processing GPUs.

YOLOv8 (10) is the latest iteration of the YOLO family of object detection algorithms. Developed by Ultralytics (5) as their official successor to YOLOv5 and released in the beginning of 2023, it was designed with a strong focus on speed, size, and accuracy. It introduces new features like a new backbone network, new loss function, and new anchor-free detection head. In direct benchmarks against YOLOv5 and v7 on high-end GPUs, it trains faster and produces more accurate models (16).

Amazon Rekognition is Amazon Web Services (AWS) (11, 17) solution to cloud-based image, video analysis, and object detection. It provides a simple and friendly API for users to introduce image and video detection into their applications without requiring a deep understanding of machine learning

and computer vision. Since its release in 2016, it has received several updates and it is now widely used in face recognition and detection, surveillance, content moderation, retail, and healthcare (17). Amazon Rekognition uses Custom Labels (11) to create classifications for objects and leverages the strong AWS cloud toolkit to create detection models.

3. Related works

Object detection and classification research has been done using a varied number of techniques, algorithms, and models. This paper is a continuation of the work started by Brown et al. (1) on maritime object detection and classification using YOLOv5. Brown et al. (1) used the same maritime dataset and created the five vessel subclasses used in this paper, using them to train a detection model with YOLOv5 and achieving an overall mAP (0.5) of 0.919. In this paper, we expand the scope of this research and introduce YOLOv7, YOLOv8, and Amazon Rekognition models to form a comparative study and attempt to arrive at an improved maritime detection and classification model. Kim et al. (18) also provide research on maritime object detection and classification using YOLOv5. Their research uses a different image dataset and different classification subcategories to train their model, achieving an overall mAP (0.5) value of 0.898. As with Brown et al. (1) however, they do not expand their study to different classification models.

Tang et al. (19) provide research on comparing different classification models; however, they do not use a traditional maritime image dataset, opting instead to focus on a GaoFen-3 dataset of synthetic aperture radar images and other satellite images of vessels. While technically it is a maritime image dataset, the top-down angle of their images makes the nature of their detection models completely different from the ones created by our research. Furthermore, their models are based on YOLOv3, YOLOv5, and their modified YOLO algorithm called N-YOLO (19), which makes for a different comparative study than the one undertaken in this paper. Zhao et al. (20) provide a similar comparative study using top-down maritime images in their detection algorithms, but instead of a satellite, they use a drone image dataset called SeaDronesee. Their comparative study is also based on their own custom YOLO model called YOLOv7-sea, which, as the name suggests, is based on YOLOv7 instead of YOLOv5 like Tang et al. (19).

Similarly, Olorunshola et al. (21) also provide a comparative study of object detection models using YOLO models, and this time, it is using YOLOv5 and YOLOv7. However, they do not use maritime images but a custom dataset for Remote Weapon Station with four classification subclasses of 'Persons', 'Handguns', 'Rifles,' and "Knives." Their models achieve an mAP (0.5) of 0.515 and 0.553 for YOLOv7 and YOLOv5, respectively.

Other comparative studies are done by Jiang et al. (22) using YOLOv7 and CBAM-YOLOv7 and Gillani et al. (23) using YOLOv5 and YOLOv7, but the former uses a Hemp Duck image dataset, while the latter uses video as their training data. Notably, Jiang et al. (22) use Convolutional Block Attention Module CBAM in their custom YOLO algorithm to improve the feature extraction rate of their model. Both these papers, however, do not explore the maritime vessel realm like some of the other papers mentioned.

Multiclass detector and recognition studies are done by Ghahremani et al. (24), using CNN algorithms to train maritime detection models on two multiclass vessel image datasets, and by Yang et al. (25), using KPE YOLOv5 to train a model using top-down drone images from the VisDrone-2020 dataset. Both these papers show results for object detection and classification models, but neither offers a comparative study on multiple algorithms.

Finally, Sharma (26) and Mohanta and Sethi (27) provide research on multiclass object detection and classification models using Amazon Rekognition. The former focuses on multiobject detection using custom labels, whereas the latter bases their study on using Amazon Rekognition for image pattern detection. Neither of these papers uses maritime image datasets, nor do they provide a comparative study with other detection models.

4. Methodology

The process of setting up the methods to implement our desired model requires several steps, summarized as follows:

- (i) The Roboflow environment of the original work performed by Brown et al. (1) was recreated. This means the (2) and (3) dataset of 8000 vessel images was uploaded and the original annotated subset consisting of roughly 1500 annotated images was imported into the Roboflow project. Following this, using Google Colab (28) cloud computing, the original YOLOv5 model was recreated to be used as baseline comparison against the efficiency and accuracy of the rest of the models used in this paper.
- (ii) To improve the quality of the training data, the original annotated subset was expanded with more annotated images using the remaining available images from the Analytics Vidhya dataset. This process consisted of copying over the original annotated subset in Roboflow and annotating another 1000 images, focusing on more annotations for the Container Ship and Tanker labels. Once the annotated subset was expanded, the preprocessing and image augmentation steps were taken to prepare the subset for model training.
- (iii) The prepared data were then independently imported into the Google Colab environment set up for each

of the YOLO models and imported into Amazon S3 (29) cloud storage to begin model training on each of the respective models. From here, each model was tuned and the data and results were collected for comparative analysis.

4.1. Original project setup

Establishing the original model as a baseline is key for the goal of this paper. Having a baseline of efficiency and performance will give the new models a comparison point with which to tangibly establish improvement. The original project must be recreated to retain not only the original annotated subset but also the original classification labels. Importing the Analytics Vidhya image dataset and the annotated subset provided by Brown et al. (1) into a new Roboflow environment allows for the recreation of the annotation environment, the splitting of the annotated subset into training, test, and validation sets, and provides the export tools necessary to export the data over to the model training environment. **Figure 1** presents an example of the bounding box annotation and labeling of a Container using Roboflow.

With the data now available, the Google Colab (28) cloud computing environment provided by Brown et al. (1) can now be used to import the annotated data and begin training the model using the YOLOv5 library. As with Brown et al. (1), training is started by passing the arguments shown in **Figure 2**.

The training parameters are as follows:

- img: input image size in pixels length/width
- batch: batch size, hardware dependent on the available GPU memory
- epochs: number of training epochs

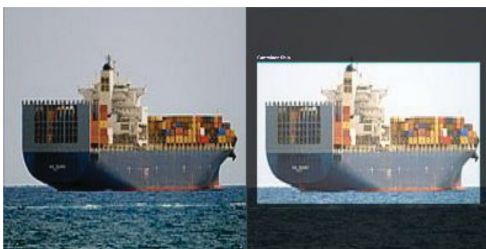


FIGURE 1 | Bounding Box Annotation and Labeling of Container using Roboflow.

```
!python train.py
--img 416 --batch 16
--epochs 150 --data {dataset.location}/data.yaml
--weights yolov5s.pt --cache
```

FIGURE 2 | YOLOv5 model training arguments.

- data: location where training data are saved
- weights: pre-trained weights provided by YOLO
- cache: cache images for faster training

4.2. Data quality enhancement

Having the baseline data and model established, the process now shifts to enhancing the annotated subset to improve the training data. For this, the annotated subset is duplicated, and more annotations are performed on the available images from the dataset. Special focus is made to annotate more images with the Container Ship and Tanker labels as those represent the lowest-performing classifications in the original model. In total, an additional 1000 images were annotated and added to the extended annotated subset. **Table 1** presents the original annotations versus the extended annotations.

The extended annotated subset is then further enhanced by applying image transformations and augmentation preprocessing techniques. The goal here is to standardize all the images and remove errant instances that might cause the model to train slowly and produce inaccurate results.

All annotations are subject to the following preprocessing:

- Auto-orientation: Discards EXIF rotations and standardizes pixel ordering
- Resize: Downsizes all images to 416×416 pixels
- Grayscale Transformation: Merges color channels to make images insensitive to color.

Figure 3 presents an example of grayscale transformation.

Brightness: Apply -25%/25% brightness variability to standardize images against lighting and camera setting changes. **Figure 4** presents an example of brightness variability.

Noise smoothing: Image smoothing transformation to remove image noise. **Figure 5** demonstrates the noise reduction.

4.3. Model training

For YOLO model training, the steps are very similar across all three libraries. First, the annotated sets of train, test,

TABLE 1 | Original annotated subset [created by (1)] vs. extended dataset.

Class	Original annotations	Extended annotations
All	1,460	2,507
Container ship	292	581
Cruise ship	292	443
Military ship	292	448
RORO	292	451
Tanker	292	584

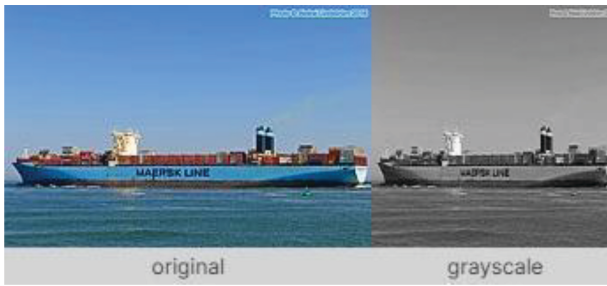


FIGURE 3 | Grayscale transformation.



FIGURE 4 | Brightness variability.



FIGURE 5 | Noise reduction.

```
!python train.py
--img 416 --batch 16
--epochs 150 --data {dataset.location}/data.yaml
--weights yolov5s.pt --cache
```

FIGURE 6 | YOLOv5 model training with an extended dataset.

and validation data are imported into the respective training environment; then the training is started by executing the train command with the familiar arguments. YOLOv5 model training is as in Brown et al. (1). **Figure 6** presents the parameters used to train the extended dataset with YOLOv5.

YOLOv7 model training is similar; however, there are two key differences, that is, the specifications using YOLOv7 training weights with the `-weights` argument and the key argument of `-device 0` to specify which CUDA (30) device to utilize for training to utilize YOLOv7 greater CUDA optimization [(5), (21), and (26)]. **Figure 7** presents the model training parameters for the extended dataset with YOLOv7.

The YOLOv8 (10) model train command syntax changed slightly from the previous two versions; however, the core

```
!python train.py
--img 416 --batch 16
--epochs 150 --data {dataset.location}/data.yaml
--weights 'yolov7_training.pt' --device 0 --cache
```

FIGURE 7 | YOLOv7 model training with an extended dataset.

```
!yolo task=detect mode=train
model=yolov8s.pt data={dataset.location}/data.yaml
epochs=150 imgsz=416 batch=16
```

FIGURE 8 | YOLOv8 model training with an extended dataset.

meaning behind the argument remains the same. YOLOv8 has standardized CUDA use, so there is no need to specify CUDA device in the train command. We specify the weights as YOLOv8 weights. **Figure 8** presents YOLOv8 model training with the extended dataset.

For Amazon Rekognition (11), the setup is a little different, requiring manual Roboflow export to Amazon S3 bucket (29) cloud storage as Amazon Custom Labels and setting the S3 bucket permissions to allow Rekognition access. Once the S3 bucket is set with the labeled images, the Rekognition classification project can be created and pointed to that S3 bucket, and the model training can start. This work was performed on cloud-based infrastructure provided by AWS, using dedicated runners on AWS servers that provide premium CPUs and GPUs for training on machine learning models.

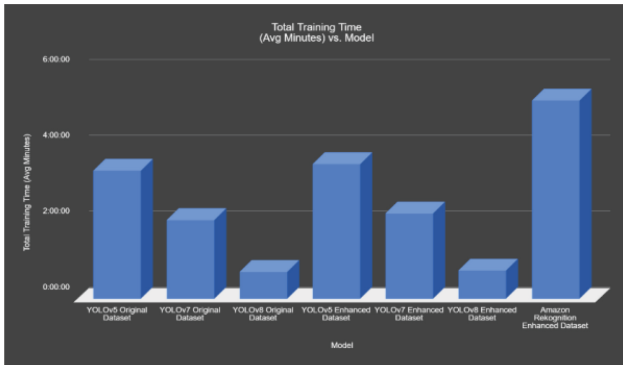
5. Results and discussion

The results are split into three distinct sections:

- (i) Model training efficiency is analyzed across all the models and iterations. Seven total training time data points are captured representing the three YOLO models done with the original annotated subset, three YOLO models with the enhanced annotated subset, and the Amazon Rekognition model with the enhanced annotated subset.
- (ii) YOLO models' accuracy utilizing the original annotated subset is analyzed. This is the true baseline comparison as the original model created by Brown et al. (1) trained with the original annotated subset is compared against the other YOLO models also trained with the original annotated subset.
- (iii) The results of all three YOLO models and Amazon Rekognition model utilizing the enhanced annotated subset are analyzed. This represents the true comparison point as all improvement steps are represented in this part and should provide the most interesting results.

TABLE 2 | Training time efficiency per model.

Model	Runs	Epochs	Training time/epoch (Avg)	Total training time (Avg)
YOLOv5 original	5	150	0:01:21	3:22:30
YOLOv7 original	5	150	0:00:50	2:05:00
YOLOv8 original	5	150	0:00:17	0:42:30
YOLOv5 enhanced	5	150	0:01:25	3:32:30
YOLOv7 enhanced	5	150	0:00:54	2:15:00
YOLOv8 enhanced	5	150	0:00:18	0:45:00
Rekognition enhanced	5			5:13:00

**FIGURE 9** | Training efficiency graph (smaller bars are better).**TABLE 3** | YOLOv5 model performance metrics.

Class	Images	Labels	Precision	Recall	mAP @.5	mAP @ 5.95
All	292	322	0.897	0.887	0.919	0.591
Container ship	292	69	0.903	0.841	0.916	0.569
Cruise ship	292	59	0.896	0.879	0.923	0.57
Military ship	292	65	0.884	0.908	0.961	0.584
RORO	292	60	1	0.994	0.995	0.712
Tanker	292	69	0.802	0.812	0.798	0.518

5.1. Model training efficiency

Steps were taken to standardize the training environment used for all the YOLO models. The Google Colab (28) premium cloud computing environment was set to create a stable training environment with access to the same baseline computing resource across all Colab notebooks. This included premium GPU and RAM access and enough computing units to permit uninterrupted training for all instances of runs and epochs.

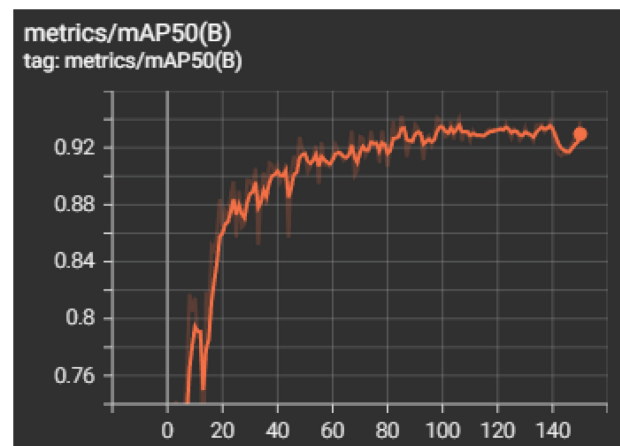
Overall, each YOLO model was run 5 times for 150 epochs and training times were recorded for all models, split by training time per epoch. Amazon Rekognition does not display epoch data, so only run data are recorded along with the recorded training time for this model. [Table 2](#) presents the training time efficiency per model.

TABLE 4 | YOLOv7 model performance metrics.

Class	Images	Labels	Precision	Recall	mAP @.5	mAP @ 5.95
All	292	322	0.801	0.749	0.824	0.437
Container ship	292	69	0.816	0.71	0.826	0.41
Cruise ship	292	59	0.682	0.712	0.73	0.454
Military ship	292	65	0.822	0.708	0.837	0.361
RORO	292	60	0.949	0.933	0.978	0.613
Tanker	292	69	0.734	0.78	0.747	0.348

TABLE 5 | YOLOv8 model performance metrics.

Class	Images	Labels	Precision	Recall	mAP @.5	mAP @ 5.95
All	292	322	0.908	0.885	0.934	0.621
Container ship	292	69	0.897	0.882	0.952	0.609
Cruise ship	292	59	0.863	0.847	0.91	0.6
Military ship	292	65	0.934	0.908	0.95	0.586
RORO	292	60	0.983	0.992	0.995	0.761
Tanker	292	69	0.864	0.854	0.887	0.575

**FIGURE 10** | YOLOv5 mAP (0.5) graph.

As shown in [Table 2](#), YOLOv8 had the most efficient training time among all seven data points, followed by the YOLOv7 model, the YOLOv5 model, and last the

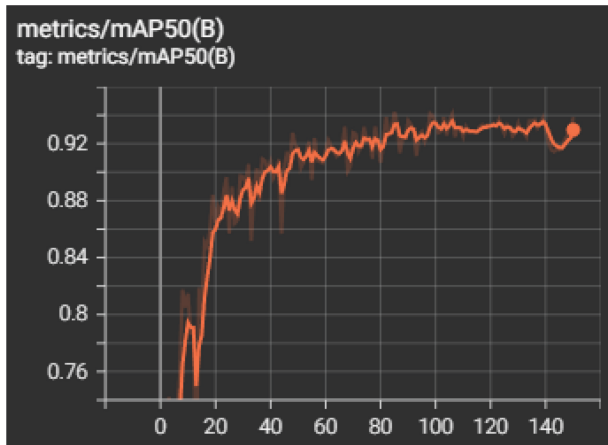


FIGURE 11 | YOLOv7 mAP (0.5) graph.

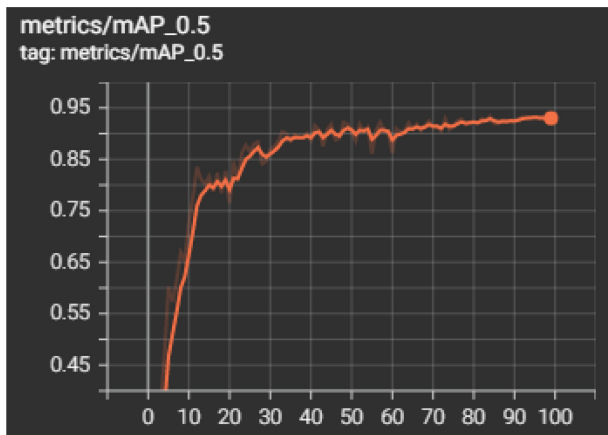


FIGURE 12 | YOLOv8 mAP (0.5) graph.

Amazon Rekognition model. YOLOv8 shows a blazing 18 seconds of training time per epoch, which is quite a fast result. Not surprisingly, the training of the models with the enhanced annotated subset took longer given the fact that it introduced approximately 700 more images into the training subset. Interestingly, however, the increased training time is minimal, especially in the YOLOv8 model, which in part must be attributed to our steps to better prepare the data with different preprocessing techniques. **Figure 9** presents a graphical representation of the training time efficiency per model.

5.2. YOLO baseline model results

For setting up the baseline, all three YOLO models were trained using the original annotated dataset. This was set up to produce a true comparison baseline with which to rate the models, keeping the training data in the same state as they were when Brown et al. (1) trained their YOLOv5 model.

Metrics used are True Positive, True Negative, False Positive, False Negative, Precision, Recall, Mean Average

TABLE 6 | YOLOv5 performance metrics for enhanced datasets.

Class	Images	Labels	Precision	Recall	mAP @.5	mAP @ 5.95
All	584	389	0.898	0.89	0.921	0.592
Container ship	581	95	0.907	0.84	0.917	0.571
Cruise ship	443	64	0.898	0.880	0.928	0.572
Military ship	448	74	0.89	0.906	0.965	0.586
RORO	451	62	0.991	0.993	0.995	0.721
Tanker	584	94	0.811	0.812	0.811	0.528

TABLE 7 | YOLOv7 performance metrics for enhanced datasets.

Class	Images	Labels	Precision	Recall	mAP @.5	mAP @ 5.95
All	584	389	0.9141	0.92	0.915	0.601
Container ship	581	95	0.89	0.889	0.898	0.569
Cruise ship	443	64	0.917	0.922	0.923	0.57
Military ship	448	74	0.923	0.928	0.961	0.584
RORO	451	62	0.987	0.994	0.995	0.712
Tanker	584	94	0.854	0.867	0.854	0.568

TABLE 8 | YOLOv8 performance metrics for enhanced datasets.

Class	Images	Labels	Precision	Recall	mAP @.5	mAP @ 5.95
All	584	389	0.912	0.916	0.944	0.648
Container ship	581	95	0.911	0.905	0.951	0.621
Cruise ship	443	64	0.938	0.922	0.923	0.688
Military ship	448	74	0.919	0.917	0.961	0.632
RORO	451	62	0.968	0.967	0.995	0.712
Tanker	584	94	0.864	0.861	0.891	0.601

Precision, and F1 score. Using the Tanker label as an example, these metrics are defined as:

- True Positive (TP): Data points labeled as Tanker, which are actually Tanker.
- True Negative (TN): Data points labeled as not Tanker, which are actually not Tanker.
- False Positive (FP): Data points labeled as Tanker, which are actually not Tanker.
- False Negative (FN): Data points labeled as not Tanker, which are actually Tanker.

Precision: Number of true positives divided by the total number of positive labels.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Number of true positives divided by true positives plus false negatives.

$$Recall = \frac{TP}{TP + FN}$$

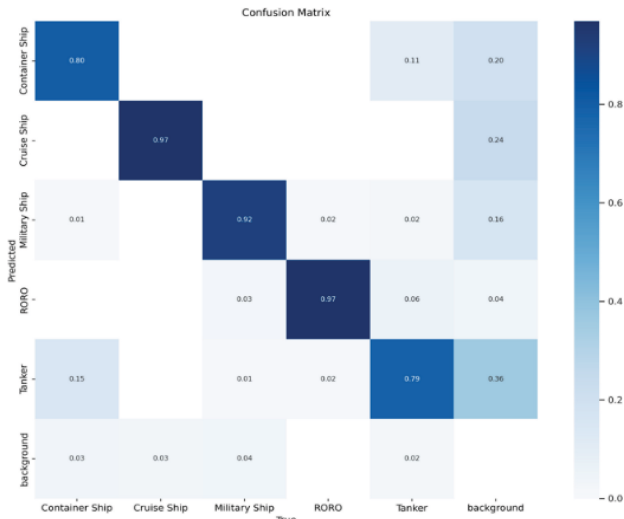


FIGURE 13 | YOLOv5 confusion matrix for enhanced datasets.

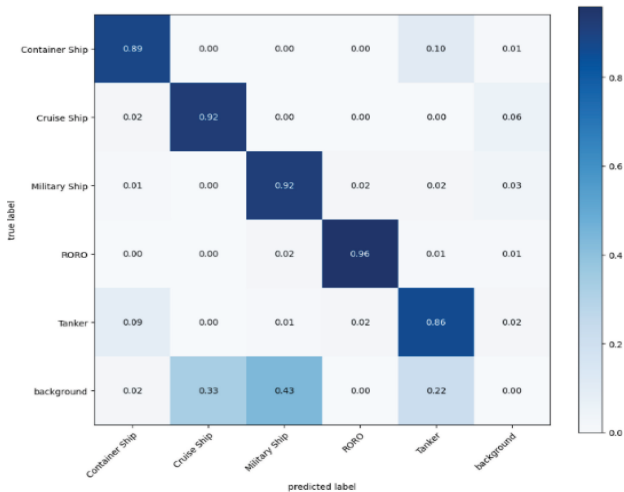


FIGURE 14 | YOLOv8 confusion matrix for enhanced datasets.

Mean average precision (mAP): Average precision of each class and average over all classes.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

F1 score: The harmonic mean of the precision and recall.

$$F1 = 2 \frac{precision * recall}{precision + recall}$$

The following tables show the performance metrics of the baseline models using the original annotated dataset. Tables 3–5 show the training results for the YOLOv5, YOLOv7, and YOLOv8 models, respectively.

These results already offer some interesting observations, even without introducing our enhanced subset into the equation. First, it can be observed that YOLOv5 still performed better than YOLOv7 on every single classification

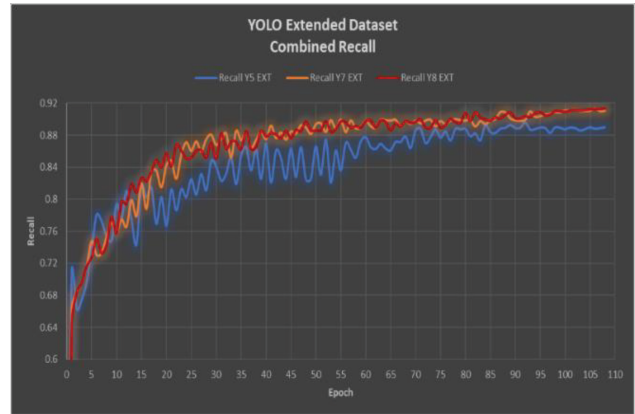


FIGURE 15 | Combined recall score of YOLO models.

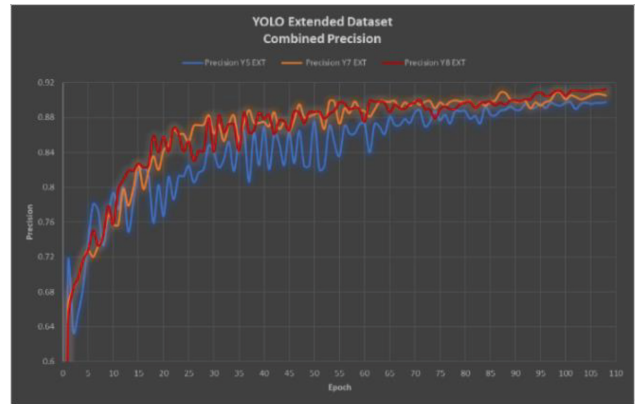


FIGURE 16 | Combined precision score of YOLO models.

label. Even though YOLOv7 trained its model 60% faster than YOLOv5, the superior accuracy of the YOLOv5 model more than makes up for that difference.

In contrast, YOLOv8 showed improvement in half of the classification labels and notably showed marked improvement in the two labels that the YOLOv5 model performed the weakest in the original model. The YOLOv8 model also showed improvement in accuracy of the overall model, with a mean average precision in the intersection over union threshold 0 to 0.5 (mAP@.5) of 0.934 versus 0.919 on the original model. Furthermore, as seen in section 5.1, the YOLOv8 model offered approximately a 440% training efficiency improvement over the YOLOv5 model. These two facts clearly set the YOLOv8 model as the clear candidate model out of the three YOLO models.

Figures 10–12 show the graphical representation of the mAP@.5 for the YOLOv5, YOLOv7, and YOLOv8 models, respectively.

5.3. Enhanced data models

This section shows the true results of the paper. Using the enhanced annotated subset along with both the three YOLO

TABLE 9 | Amazon Rekognition performance metrics for enhanced datasets.

Class	Images	Labels	Precision	Recall	F1 score
All	584	389	0.972	0.97	0.971
Container ship	581	95	0.984	0.968	0.976
Cruise ship	443	64	1	0.983	0.991
Military ship	448	74	0.967	0.951	0.959
RORO	451	62	0.984	0.968	0.976
Tanker	584	94	0.925	0.98	0.951

TABLE 10 | Comparison of metric scores for container ship and tanker classification labels.

Class	YOLOv5 original (mAP@.5 score)	YOLOv8 (mAP@.5 score)	Amazon Rekognition (F1 score)
All	0.919	0.944	0.971
Container ship	0.916	0.951	0.976
Tanker	0.798	0.891	0.951

models and introducing the Amazon Rekognition model, the results here should clearly identify the superior model.

First, we explore the training results of the YOLO models. As before, **Tables 6–8** show the training results using the enhanced annotated subset for the YOLOv5, YOLOv7, and YOLOv8 models, respectively.

It can be observed that the YOLOv5 model achieved a slight improvement over the original model by using the enhanced annotated set. A nominal improvement of mAP@.5 from 0.919 to 0.921 in the overall model is minor but not insignificant. Even while minor, the improvement is across the board on every single classification label as well as the overall model. This shows the efforts undertaken to improve the training data with preprocessing techniques had a positive effect on the overall model accuracy, even while still using the same underlying model training library.

Second, it can also be observed that the YOLOv7 model achieved a significant improvement using the enhanced annotated set over the same model using the original annotated set. An improvement of 0.824 to 0.915 mAP@.5 is quite a substantial improvement. It still falls short to the YOLOv5 model on overall model accuracy, but notably, the Tanker classification label achieves a big improvement bump over the results from the original model by achieving 0.854 versus the original 0.798 mAP@.5. This shows the YOLOv7 model to be more receptive to data quality improvements and makes us postulate that further efforts in this area should make this model pull ahead of the original model.

Finally, as also observed in section 5.2, the YOLOv8 model again achieved the largest improvement over the original model. Not only does it achieve significant improvement on the overall model accuracy with a 0.944 versus 0.919 mAP@.5



Labels	Confidence
Tanker False positive	80.2%
Container Ship False negative	19.7%

FIGURE 17 | Amazon Rekognition classification model: mislabel of Tanker instead of Container Ship.

improvement but also it achieves noticeable improvement in our targeted classification labels of Container Ship and Tanker, particularly the Tanker classification label which sees an improvement to 0.891 from 0.798 mAP@.5. This model also showed improvement when trained with the enhanced annotated set versus the original, showing, just like with the YOLOv7 model, to be more receptive to training data quality improvements.

Figures 13, 14 show the confusion matrix for the YOLOv5 and YOLOv8 models, which are the two best performing models at this stage of the results.

Confusion matrices for both models show the biggest area of mislabeling occurs between Container Ships and Tankers, which coincides with the original conclusions reached by Brown et al. (1). Notably, the YOLOv8 model manages to reduce the amount of incorrectly labeled images between these two labels by a significant amount, hence the reason for the overall improvement in this model against the YOLOv5 model.

Figures 15, 16 show the combined recall and precision graphs of all three YOLO models trained with the annotated enhanced set.

The combined figures again reinforce the results showing the YOLOv8 model superior performance even at the raw precision and recall levels. Interestingly, YOLOv7 manages to keep up with the other models with this raw score, despite lower overall mAP@.5 performance.

Amazon Rekognition does not provide epoch data as part of the result set; thus, the results are shown as averages of the five training runs. Furthermore, it uses the F1 score as the



Labels	Confidence
Container Ship False positive	86.0%
Tanker False negative	13.6%

FIGURE 18 | Amazon Rekognition classification model: mislabel of Container Ship instead of Tanker.

overall model accuracy score instead of mAP. **Table 9** shows these results averaged out across the five training runs.

The Amazon Rekognition model managed to achieve improved results across the board. A top overall model F1 score of 0.971 compared to the best YOLO model overall mAP@.5 of 0.944 was achieved by the YOLOv8 model. Furthermore, it also managed to significantly increase the accuracy on both our target classification labels of Container Ship and Tanker substantially, compared to both the original YOLOv5 model and our best performing YOLOv8 model. On the other hand, as shown in section 5.1, this model offered the lowest training efficiency. This shows a clear tradeoff in model classification accuracy improvement at the expense of training efficiency and speed.

Table 10 shows the overall scores for our target labels for the original YOLOv5 model, the best performing YOLOv8 model, and the Amazon Rekognition model.

Interestingly, the Amazon Rekognition model had similar mislabeling issues as the YOLO models when it came to mislabeling Container Ships and Tankers. Obviously given the score, the mislabeling issues were on a smaller scale, but it does offer an interesting data point. **Figures 17, 18** show an example of a mislabeling from the Rekognition model.

This continued mislabeling issue across all trained models tested exposes a potential issue with the ground truth labels of the vessels. Either there is cross labeling happening between the Container Ship and Tanker images or these particular

vessels are too similar for the models to make distinctions on some special cases. We noticed, while annotating images, that some vessels had profiles of both classifications despite ground truth labeling them with a specific label.

6. Conclusion and future work

Using the previous work developed by Brown et al. (1), the annotated set of Analytics Vidhya marine vessel images and the original annotated and labeled subset of images was recreated in a Roboflow environment. This allowed for their original YOLOv5 model to be recreated to serve as a comparison baseline for the research conducted in this paper. The original annotated set was used with two additional YOLO models, YOLOv7 and YOLOv8, to compare the performance against the original model. This model comparison began to yield interesting results. First, an overall significant increase in training efficiency for the new YOLO models was achieved, with the YOLOv8 model achieving a nearly 440% training efficiency improvement over the original YOLOv5 model (1). Second, while the YOLOv7 model failed to achieve similar levels of classification accuracy, falling short in both overall model mAP scores and single-label scores, the YOLOv8 achieved classification accuracy improvements across the board in both overall and single-label scores.

Further steps were taken to explore ways to improve these models and obtain an improved object detection and classification model. To achieve this, the original annotated set was enhanced by adding an additional 1000 image annotations from images extracted from the same Analytics Vidhya image dataset, and the entire expanded annotated set was subject to several image preprocessing steps to improve the quality of the training data. These new annotated data were used to retrain the same three YOLO models, but additionally, they were also used to train an Amazon Rekognition classification model to introduce a non-YOLO classification model library for comparison.

Results from the training of these models showed improvements across the board in classification accuracy for all three YOLO models. Improvements ranged from a modest 2% classification accuracy increase for the YOLOv5 model to a significant 10% improvement for the YOLOv8 model. This showed the steps taken to improve the quality of the training data had significant effects on the accuracy of each of the models. Furthermore, the Amazon Rekognition model achieved the highest improvement in classification accuracy. With a nearly 15% improvement in overall classification over the original model, nearly 20% improvement in classification of the lowest performing classification, Tanker, was achieved by the original model. The Rekognition model, however, had the lowest training efficiency of them all, clearly showing a conscious tradeoff of model classification accuracy over training speed and efficiency.

Interestingly, all models struggled mostly with mislabels between Container Ships and Tankers, illustrating a potential issue with the ground truth labels for these two vessel types. There seems to be a high overlap in image labels between these two types, leading models to significantly mislabel these types above any other. Future work on enhancing these models will most definitely need to delve into this issue. Both YOLOv7 and YOLOv8 models showed good improvement with the enhanced annotated set, giving good evidence of the model's receptiveness to improved training data. This, together with the possible ground truth issues with Container Ship and Tanker labels, gives an excellent starting point for potential improvement with these specific image classifications, which, if improved further, would push the models to a very high level of accuracy across the board.

Author contributions

AP prepared the dataset by annotating the images, worked on applying the machine learning algorithms, and wrote the initial draft of the manuscript. SB provided guidance and oversight on the whole project and did the overseeing and final edits of the manuscript to bring it to publication form. Both authors contributed to the article and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Brown S, Hall C, Galliera R, Bagui S. Object detection and ship classification using YOLOv5. *BOHR Int J Comput Sci.* (2022) 1:124–133.
- Analytics Vidhya. *Game of deep learning: computer vision Hackathon.* (n.d.). Available online at: <https://datahack.analyticsvidhya.com/contest/game-of-deep-learning/> (accessed January 22, 2023).
- Kaggle. *Game of deep learning: ship datasets.* (n.d.). Available online at: <https://www.kaggle.com/datasets/arpitjain007/game-of-deep-learning-ship-datasets> (accessed January 22, 2023).
- Roboflow. *Roboflow: go from raw images to a trained computer vision model in minutes.* (n.d.). Available online at: <https://roboflow.com/> (accessed January 22, 2023).
- Joseph R, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. *arxiv [preprint]* (2016): arXiv:1506.02640
- Ultralytics. *Ultralytics | Revolutionizing the world of vision AI.* Available online at: <https://ultralytics.com/yolov8> (accessed January 22, 2023).
- Jocher G, Chaurasia A, Stoken A, Borovec J, Kwon Y, Michael K, et al. *Ultralytics/yolov5: v7.0 - YOLOv5 SOTA realtime instance segmentation.* (2022). Available online at: <https://zenodo.org/record/7347926> (accessed February 28, 2023).
- Ultralytics. *Ultralytics/yolov5.* (2020). Available online at: <https://github.com/ultralytics/yolov5> (accessed 2023).
- Wong KY. *Official YOLOv7.* (2022). Available online at: <https://github.com/WongKinYiu/yolov7> (accessed 2023).
- Ultralytics. *YOLOv8 by Ultralytics.* (2023). Available online at: <https://github.com/ultralytics/ultralytics> (accessed 2023).
- Amazon. *Amazon rekognition – video and image - AWS.* Seattle, WA: Amazon Web Services, Inc (2017).
- PyTorch. (n.d.). Available online at: https://pytorch.org/hub/ultralytics_yolov5/ (accessed January 22, 2023).
- Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T editors. *Computer vision – ECCV 2014.* Lecture notes in computer science. (Vol. 8693), Cham: Springer (2014). p. 740–755. doi: 10.1007/978-3-319-10602-1_48
- Ultralytics/Yolov3. *ultralytics/yolov3.* (2020). Available online at: <https://github.com/ultralytics/yolov3> (accessed 2023).
- Wang C-Y, Bochkovskiy A, Liao H-YM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arxiv [preprint]* (2022): arXiv:2207.02696
- Gabriel. *Performance benchmark of YOLO v5, v7 and v8.* Stereolabs (2023). Available online at: <https://www.stereolabs.com/blog/performance-of-yolo-v5-v7-and-v8/> (accessed 2023).
- Mishra A. *Machine learning in the AWS cloud.* Hoboken, NJ: John Wiley and Sons (2019).
- Kim J-H, Kim N, Park YW, Won CS. Object detection and classification based on YOLO-V5 with improved maritime dataset. *J Mar Sci Eng.* (2022) 10:377. doi: 10.3390/jmse10030377
- Tang G, Zhuge Y, Claramunt C, Men S. N-YOLO: a SAR ship detection using noise-classifying and complete-target extraction. *Remote Sens.* (2021) 13:871. doi: 10.3390/rs13050871
- Zhao H, Zhang H, Zhao Y. YOLOv7-sea: object detection of maritime UAV images based on improved YOLOv7. *Proceedings of the IEEE/CVF winter conference on applications of computer vision workshops (WACVW).* Waikoloa, HI: (2023). doi: 10.1109/WACVW58289.2023.00029
- Olorunshola OE, Irhebhude ME, Ewwiekpaefe AE. A comparative study of YOLOv5 and YOLOv7 object detection algorithms. *J Comput Soc Inf.* (2023) 2:1–12. doi: 10.33736/jcsi.5070.2023
- Jiang K, Xie T, Yan R, Wen X, Li D, Jiang H, et al. An attention mechanism-improved YOLOv7 object detection algorithm for hemp duck count estimation. *Agriculture.* (2022) 12:1659. doi: 10.3390/agriculture12101659
- Gillani IS, Munawar MR, Talha M, Azhar S, Mashkoor Y, Uddin MS, et al. Yolov5, Yolo-x, Yolo-r, YOLOv7 performance comparison: a survey. *Comput Sci Inf Technol (CS and IT).* (2022) 12:17. doi: 10.5121/csit.2022.121602
- Ghahremani A, Kong Y, Bondarau Y, de With PHN. Multi-class detection and orientation recognition of vessels in maritime surveillance. *Paper presented at IS&T international symposium on electronic imaging 2019, image processing: algorithms and systems XVII.* Eindhoven University of Technology Research Portal (2019). doi: 10.2352/ISSN.2470-1173.2019.11.IPAS-266
- Yang R, Li W, Shang X, Zhu D, Man X. KPE-YOLOv5: an improved small target detection algorithm based on YOLOv5. *Electronics.* (2023) 12:817. doi: 10.3390/electronics12040817
- Sharma V. Object detection and recognition using Amazon Rekognition with Boto3. *Proceedings of the 6th international conference on trends in electronics and informatics (ICOEI).* Tirunelveli: (2022). doi: 10.1109/icoei53556.2022.9776884
- Mohanta R, Sethi B. Amazon rekognition for pattern recognition. *J Eng Sci.* (2020) 11:197–200.
- Google. *Google colab.* (2019). Available online at: <https://colab.research.google.com/> (accessed 2023).
- AWS. *Cloud object storage | Store and retrieve data anywhere | Amazon simple storage service.* Seattle, WA: Amazon Web Services, Inc (2018).
- CUDA Toolkit. *CUDA toolkit, NVIDIA developer.* (2013). Available online at: <https://developer.nvidia.com/cuda-toolkit> (accessed 2023).