S

T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

COMPARISON OF YOLOV5 AND WASR NETWORK ON SURFACE VEHICLE DATASET

ÇAĞLA ŞAHİN

SUPERVISOR
PROF. DR. ERKAN ZERGEROĞLU

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

COMPARISON OF YOLOV5 AND WASR NETWORK ON SURFACE VEHICLE DATASET

ÇAĞLA ŞAHİN

SUPERVISOR
PROF. DR. ERKAN ZERGEROĞLU

2023
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 18/06/2023 by the following jury.

JURY

Member
(Supervisor) : Prof. Dr. ERKAN ZERGEROĞLU

Member : Prof. Dr. Erkan Zergeroğlu

Member : BS Çağla Şahin

ABSTRACT

The ability to accurately detect marine surface vehicles is a critical task in various applications, from surveillance and security to navigation and marine biology. This paper presents a comprehensive study on the use of image segmentation and object detection methodologies, specifically focusing on the WaSR algorithm and the YOLOv5 model, for this purpose. We introduce an innovative approach that employs artificial intelligence to manipulate the training images, with the aim of enhancing the distinctiveness of the target object within the images, thereby facilitating more accurate detection by the model. Despite the potential improvement in model sensitivity, this method also poses risks of overfitting, a challenge we address in detail. Our findings reveal complex interplays between these factors, providing valuable insights into the model's performance before and after the manipulation of images. We hope that our research will serve as a springboard for further exploration and development of robust techniques for the detection of marine surface vehicles.

Keywords: Marine Surface Vehicle Detection, Image Segmentation, Object Detection, WaSR Algorithm, YOLOv5, Image Manipulation, Artificial Intelligence, Overfitting, Performance Metrics, Precision and Recall, Mean Average Precision.

ÖZET

Deniz üstü araçların doğru bir şekilde tespiti, gözetim ve güvenlikten navigasyona ve deniz biyolojisine kadar çeşitli uygulamalar için kritik bir görevdir. Bu makale, bu amaç için görüntü segmentasyonu ve nesne tespiti metodolojilerinin kullanımı üzerine kapsamlı bir çalışmayı sunar, özellikle WaSR algoritması ve YOLOv5 modeline odaklanır. Hedef nesnenin görüntüler içindeki belirginliğini artırarak model tarafından daha doğru tespiti kolaylaştırmayı amaçlayan, eğitim görüntülerini yapay zeka ile manipüle etmek için yenilikçi bir yaklaşımı tanıtıyoruz. Model duyarlılığında potansiyel bir iyileşmeye rağmen, bu yöntem aşırı uyuma riskleri de oluşturur, bu zorluğu ayrıntılı bir şekilde ele alıyoruz. Bulgularımız, bu faktörler arasındaki karmaşık etkileşimleri ortaya çıkarır, görüntülerin manipülasyon öncesi ve sonrasında modelin performansına değerli içgörüler sağlar. Araştırmamızın, deniz üstü araçların tespiti için sağlam tekniklerin daha fazla keşfi ve gelişimi için bir sıçrama tahtası görevi görmesini umuyoruz.

Anahtar kelimeler: Deniz Üstü Araç Tespiti, Görüntü Segmentasyonu, Nesne Tespiti, WaSR Algoritması, YOLOv5, Görüntü Manipülasyonu, Yapay Zeka, Aşırı Uyma, Performans Metrikleri

ACKNOWLEDGEMENT

First and foremost, I express my deepest gratitude to my supervisor, Erkan Zergeroğlu. His unwavering support, infectious enthusiasm, and rigorous attention to detail have been instrumental to both this paper and the research that underlies it. His guidance has been invaluable, shepherding my work from our initial project discussions to the final draft of this paper.

I am also indebted to Sezai Burak Kantarcı and Yusuf Kunt, my project supervisors at HAVELSAN. Their assistance throughout all stages of this research has been crucial. They not only provided a conducive working environment but also imparted a broad perspective on the industry that significantly enriched my understanding and approach to the project.

My heartfelt thanks also go to my family. Their unwavering support, appreciation, and encouragement have been a constant source of motivation. Their keen interest in my academic pursuits has always propelled me to strive for excellence.

Lastly, but by no means least, I extend my sincere gratitude to my fiancé and his family. Their moral support throughout this research journey has been a beacon, illuminating even the most challenging phases of the work. Their belief in my capabilities and their constant encouragement played a significant role in the completion of this research.

This project is a testament to the collective effort of these individuals, and I am profoundly grateful for their contributions.

Çağla Şahin

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
YOLO	:	You Only Look Once
WaSR	:	Water Segmentation and Refinement Maritime Obstacle Detection Network
AP	:	Average Precision
mAP	:	Mean Average Precision

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In the field of maritime surveillance and navigation, the ability to accurately detect marine surface vehicles is of paramount importance. It forms the basis for a variety of applications, spanning from safety and security enforcement, navigation aid, to biodiversity studies in marine biology. With the rapid progression of technological advancements, there has been a surge in the development and deployment of computer vision techniques to perform this task efficiently and effectively.

This paper presents an exhaustive study of two such techniques, namely image segmentation, with a particular emphasis on the WaSR Network, and object detection, focusing on the YOLOv5 model. Image segmentation and object detection are cornerstones of computer vision, underpinning many of its applications. The WaSR Network is a well-known image segmentation method that separates an image into regions, while YOLOv5, a state-of-the-art object detection model, is renowned for its superior speed and accuracy.

In our endeavor to further enhance the performance of these techniques, we introduce an innovative approach that involves manipulating the training images using artificial intelligence. By enhancing the distinctiveness of the target object within the images, we aim to facilitate more accurate detection by the model. However, this approach is not without its challenges. In particular, the risk of overfitting – where a model learns the training data too well and struggles to generalize to unseen data – presents a significant hurdle.

In this paper, we discuss our experimental process and findings in detail. We present a thorough analysis of the model's performance before and after the image manipulation, providing valuable insights into the impact of this technique on model sensitivity and generalizability. We believe our research will serve as a solid foundation for future work in this area, contributing to the ongoing development of robust techniques for the detection of marine surface vehicles.

1.1. Aim Of This Project

The primary objective of this project is to detect objects on the water surface, with particular application in defense industry contexts. This task will be accomplished by analyzing imagery obtained from the camera of a marine vessel, utilizing cutting-edge artificial intelligence techniques to identify any objects present on the water's surface. It is hoped that the successful execution of this project will provide a significant contribution to defense technologies.

My personal interest in this project lies in its intersection with object detection, which represents one of the most advanced frontiers of current technology. The idea of combining this technology with artificial intelligence, thereby pushing the boundaries of object detection into new realms, is an exhilarating prospect. By engaging with this topic, I hope to be part of the ongoing dialogue that is shaping the future of AI technology and its application across a broad spectrum of industries, including defense.

1.2. Edge Detection

Edge detection, a key component of visual perception, is shaped by our understanding of the brain's visual processing. Insights have primarily been drawn from studying the primary visual cortex of monkeys, which suggest a similar mechanism in humans and animals that suppresses edge detection. [1] [2]

This mechanism operates in a way that the response from an edge detector is dimmed by the reactions happening outside the focus area of the tool or function being used. [1]

Researchers like Shi et al [3] have shed light on the importance of early recurrence, or feedback, in edge detection. Our visual system has two pathways—dorsal and ventral—that send visual information at different rates and are extensively interconnected. [3]

Understanding how these pathways and their interconnections work has paved the way for a computational model inspired by our biological processes. This model aids in further understanding and improvement of edge detection processes.

1.3. Image Segmentation

Image segmentation is a process of dividing an image into multiple segments or sub-regions based on certain features like texture, color, and intensity. It is basically coming from detecting edges on general manner. It is used in a wide range of real-life applications such as object recognition, tracking, and detection, and medical imaging. In the context of water surface detection, image segmentation is useful for detecting obstacles on the surface of water, identifying the shoreline, and extracting areas of potential pollution.

One of the challenges of water surface detection is the scattering and absorption of light rays by the water and particles present in it. To overcome these challenges, researchers have proposed various AI-based methods that use deep learning to segment images and identify objects such as obstacles, buoys, and other navigational aids in the water.

On my project, for image segmentation, WaSR is a great network to work on. It is specified for captured marine images. It will be elaborated later in this document.

1.4. Object Detection

Object detection, a key aspect of computer vision, has greatly advanced over time. Initially, basic techniques were used to detect objects. However, as machine learning evolved, so did object detection.

The introduction of convolutional neural networks (CNNs) marked a major turning point. Recently, more advanced methods like deep learning, including CNNs and vision transformers, have been used. This has led to the creation of high-performing models like YOLO and Faster R-CNN, which are excellent at detecting objects in real time.

Moreover, new strategies such as computational intelligence, and techniques like SSD and MobileNet, have increased object detection's efficiency and accuracy. Also, collaborations between machines and humans have furthered these advancements.

CNN-based Object Detectors are most useful in recommendation systems. YOLO (You Only Look Once) models are used for high-performance Object detection. YOLO divides an image into grids, and each grid detects objects contained within itself. They can be used for object detection in real time based on data streams. They require a minimal amount of computational resources. YOLOv5 will be explained in detail later in this document.

2. WASR – A WATER SEGMENTATION AND REFINEMENT MARITIME OBSTACLE DETECTION NETWORK

The WaSR (Water Segmentation and Refinement) maritime obstacle detection network proposed in the literature is an excellent example of how edge detection and segmentation can be improved using artificial intelligence (AI). The proposed WaSR network leverages deep encoder-decoder architecture and uses a novel decoder that fuses inertial information from the IMU sensor with visual features to improve detection and water edge accuracy. Additionally, a novel loss function is designed to augment early separation between water and obstacle features. These improvements have led to a better performance of the WaSR network compared to other methods in detecting maritime obstacles. However, the original WaSR architecture also has some limitations, such as underestimating water edges and false positives. [4] [5]

2.1. Acquiring Effective Models

2.1.1. Quality Data

Image segmentation, a key task in computer vision, aims at dividing an image into multiple segments. This is intended to simplify the image representation or to identify areas of interest. The effectiveness of these models, especially those based on deep learning methods, largely depends on the quality of the data used during model training.

1. **Model Performance and Generalization:** High-quality data is a cornerstone of robust performance. Training a model on clean, diverse, and representative data allows the model to more effectively comprehend and learn the underlying patterns. The diversity in the data ensures that the model can generalize well to unseen data, improving its predictive accuracy in a wide range of real-world situations.

2. **Overfitting Mitigation:** A large amount of quality data can prevent overfitting, a common pitfall in machine learning. Overfitting happens when a model learns from the noise or random fluctuations in the training data, which affects its

ability to generalize. With sufficient high-quality data, a model can discern the underlying trends without succumbing to overfitting.

3. **Class Imbalance Resolution:** Quality datasets ensure balanced representation across different classes. In image segmentation, some classes might be underrepresented, which can lead to a model bias towards the majority class. A well-curated, high-quality dataset can address this issue by ensuring a balanced distribution across classes.

4. **Ground Truth Establishment:** In image segmentation, ground truth annotations are crucial for both model evaluation and learning. High-quality data includes accurate annotations that provide a reliable benchmark for model training and performance evaluation.

5. **Enhanced Feature Learning:** High-quality data allows for better feature learning. The granularity and diversity in quality data enable the model to learn intricate features and differences between classes, resulting in improved segmentation outcomes.

2.1.2. Annotation Of The Segments

Data annotation or labeling is a critical step in developing image segmentation models. The process involves manually identifying and labeling each segment in an image or video to train supervised learning models. Annotated data is essential to develop accurate computer vision models for a wide range of applications.

Accurate and efficient data annotation and labeling methods are crucial for developing robust segmentation models that can improve feature extraction, classification, and recognition. Therefore, researchers are focusing on developing new techniques to improve the efficiency of the annotation process and further improve the quality of segmentation models.

2.1.3. Using Different-Sized Datasets

The size and quality of datasets are crucial factors to consider in deep learning-based image segmentation. Different techniques, such as artificial image generation, mixed-sized image segmentation, and label augmentation, are effective in improving segmentation performance on small or challenging datasets. Therefore, careful consideration of dataset size and quality is essential to develop accurate and efficient deep learning-based segmentation models for various applications.

2.2. Literature Review on Water Segmentation

The literature showcases numerous AI-based systems designed to improve edge detection and segmentation in the maritime environment. The WODIS (Water Obstacle Detection network based on Image Segmentation) proposes using an encoder decoder network to extract high-level features and capture high-level information. Attention Refine Module improves the detection of obstacles at sea-sky-line areas. Similarly, ShorelineNet leverages a novel deep learning approach to detect shoreline environments with high frames-per-second performance for autonomous navigation for Unmanned Surface Vehicles. Multi-Scale Object Detection Model proposed by Shao et al. and A HED-optimized Automatic Detection and Tracking Algorithm for marine moving targets based on YOLO V3 are other advanced AI-based systems that offer different approaches to improve edge detection and segmentation in a maritime environment. [6] [7] [8] [9]

The studies provide a wide array of AI-based systems and techniques that can be used to improve edge detection and segmentation performance in a maritime environment, contributing to the development of more efficient and reliable detection networks for maritime navigation and safety purposes.

3. YOLO V5

The YOLO (You Only Look Once) algorithm was first proposed by Joseph Redmon in May 2016, and since then, the algorithm has continuously evolved, with new versions released almost every year. YOLO uses a neural network-based approach for object detection and introduces the region-based concept to the network architecture, where the input image is divided into grid cells, and bounding boxes enclosing objects are predicted for each cell. The predicted bounding boxes include the center coordinate of the object, confidence scores, and class probabilities.

Since its inception, the YOLO algorithm has been widely studied and applied in object detection and related fields. New versions, YOLOv2, YOLOv3, YOLOv4, and even tiny versions of each standard model have been developed to improve its performance and efficiency. YOLO has been used in various fields, including wildlife monitoring, driver identification, satellite imagery analysis, and human pose estimation. [10]

3.0.1. Differences of YOLOv5

1. **Architecture Modifications:** While the original YOLO used Darknet-53 as its backbone, YOLOv5 implements a variant of CSPDarknet-53, which is a significantly improved and more efficient architecture. These architectural modifications lead to better performance and efficiency, making YOLOv5 more suitable for real-time object detection in more diverse applications.

2. **Model Sizes:** YOLOv5 introduces multiple variants of the model - small (YOLOv5s), medium (YOLOv5m), large (YOLOv5l), and extra-large (YOLOv5x). This tiered approach provides greater flexibility in deployment, as users can select the model best suited to their computational capabilities and performance requirements.

3. **Improved mAP:** The Mean Average Precision (mAP), a crucial measure for object detection algorithms, has shown a significant improvement in YOLOv5 as compared to the original YOLO model. These advancements have resulted from architectural changes, refined training strategies, and hyperparameter tuning.

4. **Training Efficiency:** YOLOv5 has a more efficient and faster training process, due to refinements in the loss function and adoption of strategies such as Mosaic augmentation and the Cosine learning rate scheduler. These additions enable YOLOv5 to achieve better performance with shorter training times.

5. **PyTorch Implementation:** While the original YOLO was implemented using Darknet, an open-source neural network framework written in C and CUDA, YOLOv5 is implemented in PyTorch. PyTorch is a more popular and widely supported framework in the deep learning community, making it easier to integrate and use YOLOv5 for object detection tasks.

6. **Integrated Tooling:** YOLOv5 comes with a set of integrated tools for tasks such as model training, inference, and testing. It also includes support for ONNX (Open Neural Network Exchange) for easier model porting and deployment on various platforms, enhancing the usability of YOLOv5.

3.1. Tests on Different-Sized Datasets

The test results below are made on private data provided by HAVELSAN.

Test 1 :

```
Model summary: 213 layers, 7029004 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 3/3
                 all         85        580      0.789      0.427      0.387      0.211
             class_1         85         27          1          0          0          0
             class_2         85        228       0.56      0.491       0.41      0.199
             class_3         85         38      0.439      0.842      0.473      0.149
             class_4         85         69      0.734      0.797      0.862       0.67
             class_5         85         64          1      0.433      0.578       0.25
             class_6         85        154          1          0          0          0
```

This model was trained by labeling the data collected from the SIDA tool into 7 classes. 1 class has not been validated. The performance of the model is far below expectations.

Test 2:

```
Model summary: 213 layers, 7034398 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 3/3
                 all         92       1151      0.861      0.543      0.559      0.301
             class_1         92         75      0.919      0.753      0.775      0.683
             class_2         92        339      0.748      0.838      0.832      0.396
             class_3         92         36      0.947          1      0.992      0.495
             class_4         92         82      0.758      0.354      0.339     0.0816
             class_5         92        468          1          0    0.00102   0.000136
             class_6         92        151      0.796      0.311      0.413      0.149
```

This model has been trained with data collected from the SIDA platform and labeled with SUIT trainees (Feyza Nur - Çağla) to date. There are 9 different classes. 3 classes have not been validated.

Test 3:

```
Model summary: 213 layers, 7042489 parameters, 0 gradients, 15.9 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 3/3
                 all         73        539      0.904      0.696      0.748      0.476
             class_1         73         41      0.996          1      0.995      0.933
             class_2         73        171        0.9       0.83      0.932       0.57
             class_3         73         25          1      0.859      0.892      0.451
```

```
            class_4        73         11      0.997          1      0.995      0.647
            class_5        73         57      0.854      0.614      0.706      0.268
            class_6        73        205      0.618       0.26      0.448      0.136
            class_7        73         16      0.982          1      0.995      0.732
            class_8        73          2      0.766          1      0.995      0.796
            class_9        73         10      0.924        0.4      0.526      0.224
            class_10        73          1          1          0          0          0
```

This model has been trained with SIDA platform data and data collected from the internet until this date. There are 12 different classes. 2 classes have not been validated.

Test 4:

```
Model summary: 213 layers, 7037095 parameters, 0 gradients, 15.8 GFLOPs
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 3/3
                  all         73        577      0.869      0.792      0.825      0.542
              class_1         73         41      0.993          1      0.995      0.958
              class_2         73        173      0.879       0.89      0.934      0.577
              class_3         73         31      0.954      0.677      0.763      0.414
              class_4         73         11          1      0.956      0.995      0.605
              class_5         73         57      0.796      0.753      0.719      0.294
              class_6         73        205      0.625      0.337      0.445       0.13
              class_7         73         16      0.978          1      0.995      0.765
              class_8         73          2      0.918          1      0.995      0.821
              class_9         73         41      0.681      0.512      0.584      0.311
```

This model has been trained with SIDA platform data and swimming man data collected and labeled from the internet so far. There are 10 different classes. 1 class has not been validated.

3.1.1. Conclusion and Review Of Real-Time Tests

In the tests, it was observed that, although the model's performance initially improved as the number of training data increased, it eventually stabilized and showed no further development.

To improve the model further, methodologies were investigated. The research findings will be provided below.

1. **Data Quality:** While increasing the size of the dataset, also ensuring that the data is high quality and diverse enough to represent the problem. In our content, data quality is ensured. All data is captured frame by frame on 1080p camera.

2. **Data Augmentation:** Applying data augmentation techniques to create variations of your existing data. These techniques can include rotations, translations, rescaling, horizontal flips, etc. This is also a research topic of one of the undergraduate students with same data. Because of that, it will be passed on this research.

3. **Hyperparameter Tuning:** Experiment with different hyperparameters. YOLOv5 has numerous hyperparameters like learning rate, number of epochs, optimizer, batch size, etc. These parameters are adjusted during each test.

4. **Model Ensemble:** Using multiple models and combining their results to improve accuracy. Overall, this method can be a successful approach. however, due to the constraints of the environment in which the project is carried out, the project should be progressed on YOLOV5.

5. **Additional Data Sources:** If the dataset is imbalanced or lacks certain representative samples, it could be considered incorporating additional data sources. This method has been already tried on test 4 and 5.

4. METHODOLOGY FOR YOLOV5 MODEL IMPROVEMENT THROUGH IMAGE MANIPULATION

In our pursuit of improving object detection with the YOLOv5 model, we adopted a novel approach focused on manipulating the training images via artificial intelligence. The primary intent behind this procedure was to enhance the distinctiveness of the target object within the images, thereby facilitating more accurate detection by the model.

This technique was applied to approximately 50 images within our dataset. Our expectation was that by refining the quality of these images, we would in turn enhance the model's learning capability, an assumption grounded in the fundamental principle that the quality of input data is a critical determinant of a model's performance.

Nevertheless, this approach has its potential advantages and disadvantages. On the positive side, it could contribute to improving the model's sensitivity to the target object, which could potentially lead to more accurate object detection. On the other hand, the model might develop an overfitted bias towards the manipulated images, thereby affecting its generalization ability on unseen data or test sets.

In this context, it becomes crucial to ascertain the overall impact of this approach, considering both its potential benefits and drawbacks. It is imperative to understand how these two factors balance each other out and influence the model's performance. Thus, the following sections will present and discuss the results obtained from this experiment.

4.1. Without Manipulating Data

In this section, we present the results obtained from the YOLOv5 model before applying the image manipulation technique.

Below, it can be seen all the experiment details. Note that same images will be used on manipulated version of training set.

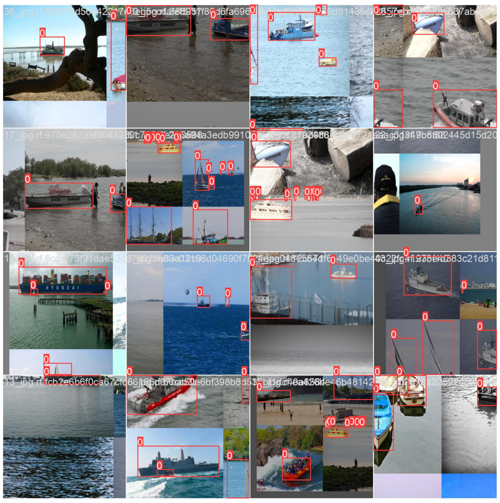

Figure 4.1: Train Batch 0

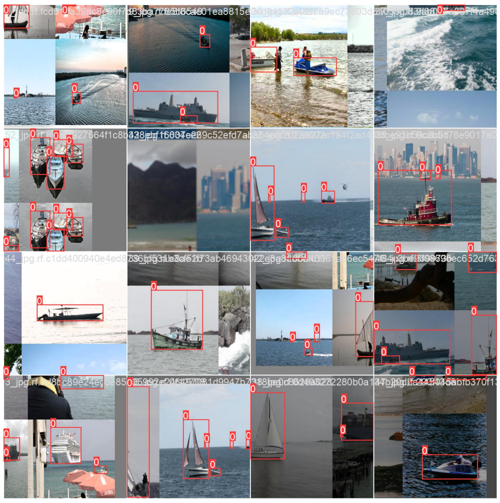

Figure 4.2: Train Batch 1

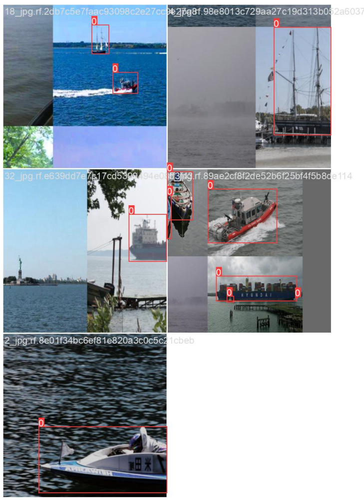

Figure 4.3: Train Batch 2

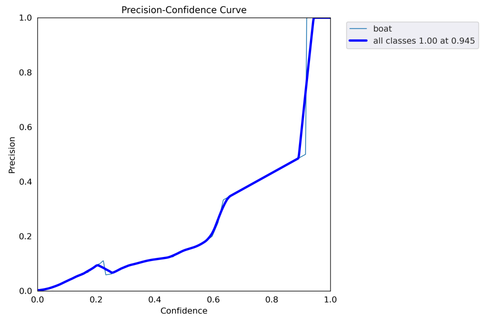

Figure 4.4: Precision-Confidence Curve

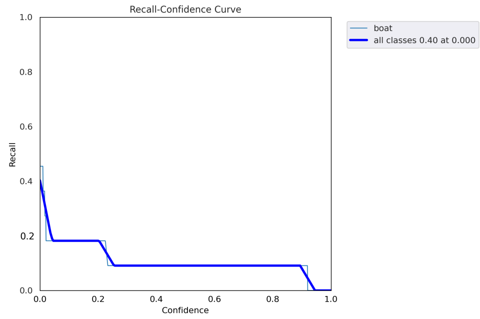

Figure 4.5: Recall-Confidence Curve

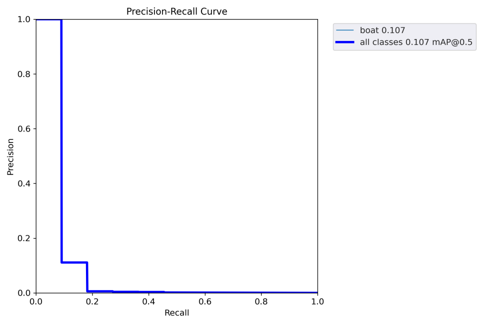

Figure 4.6: Precision-Recall Curve

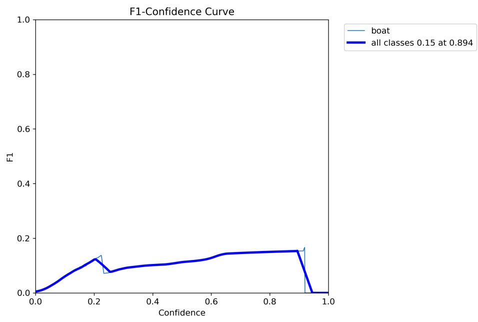

Figure 4.7: Precision-Recall Curve

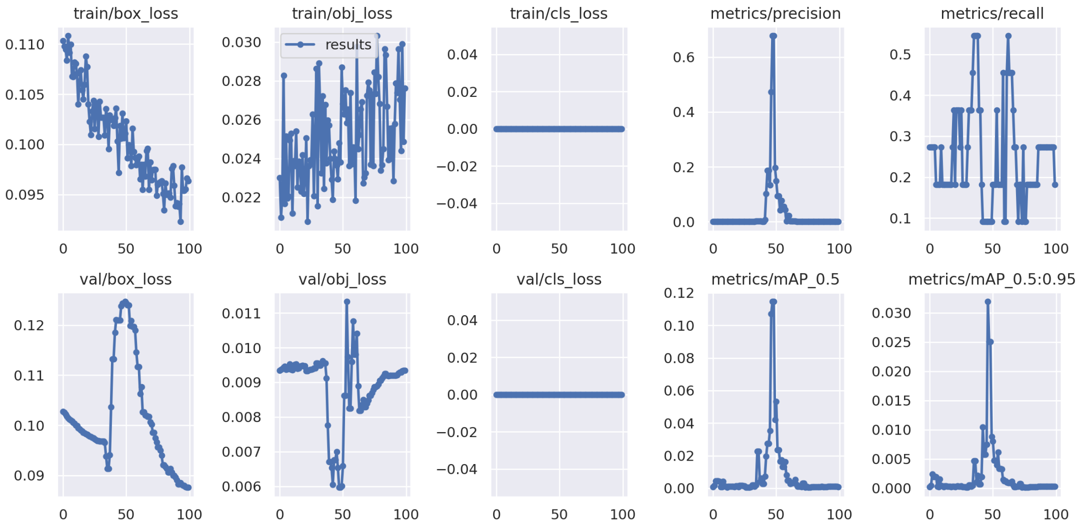

Figure 4.8: Obtained Graphs

Figure 4.9: Ground Truth

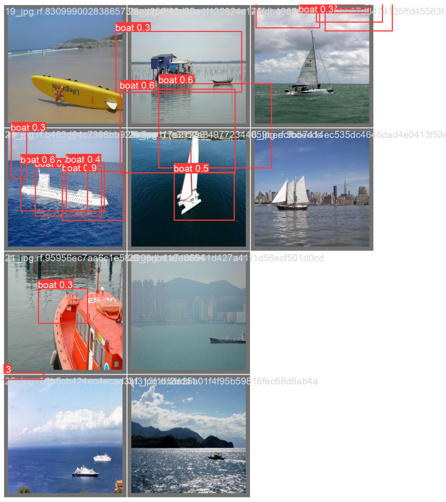

Figure 4.10: Prediction Scores

4.2. With Manipulating Data

In this section, we present the results obtained from the YOLOv5 model after applying the image manipulation technique.

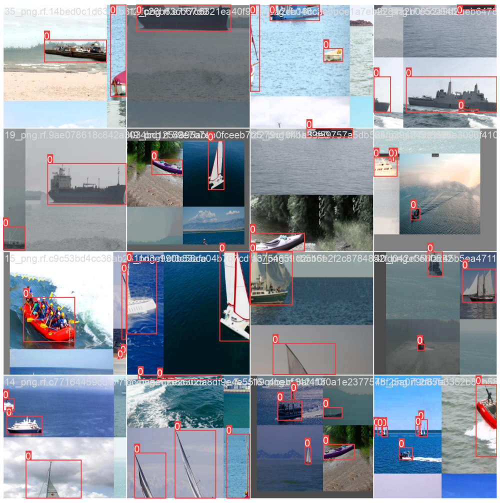

Figure 4.11: Train Batch 0

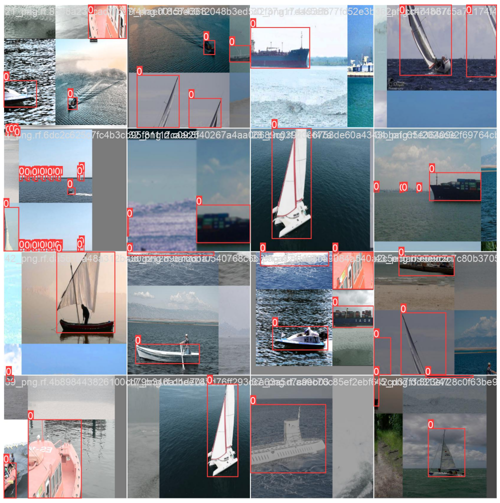

Figure 4.12: Train Batch 1

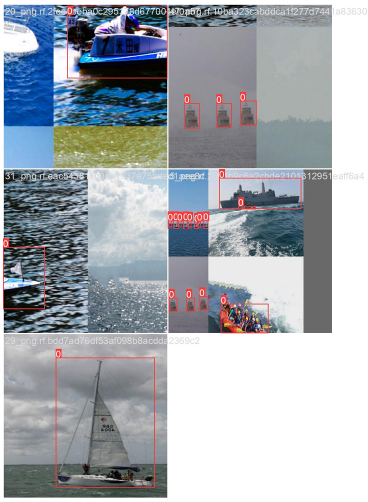

Figure 4.13: Train Batch 2

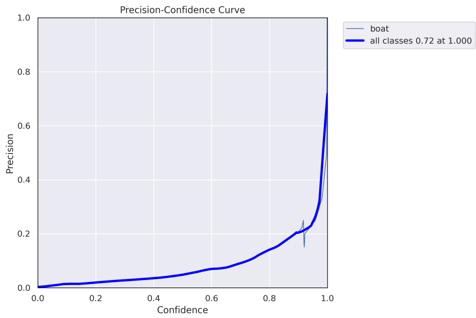

Figure 4.14: Precision-Confidence Curve

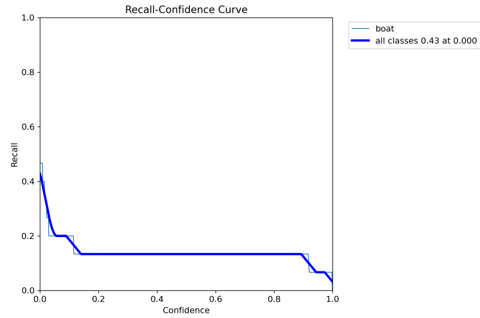

Figure 4.15: Recall-Confidence Curve

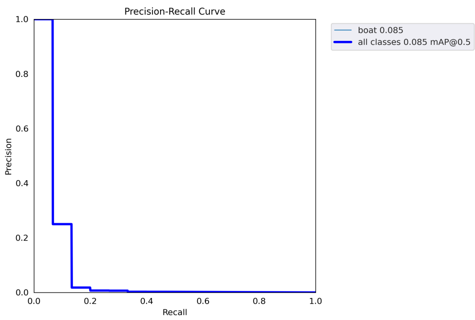

Figure 4.16: Precision-Recall Curve

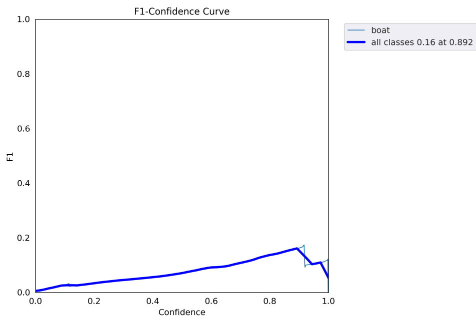

Figure 4.17: Precision-Recall Curve

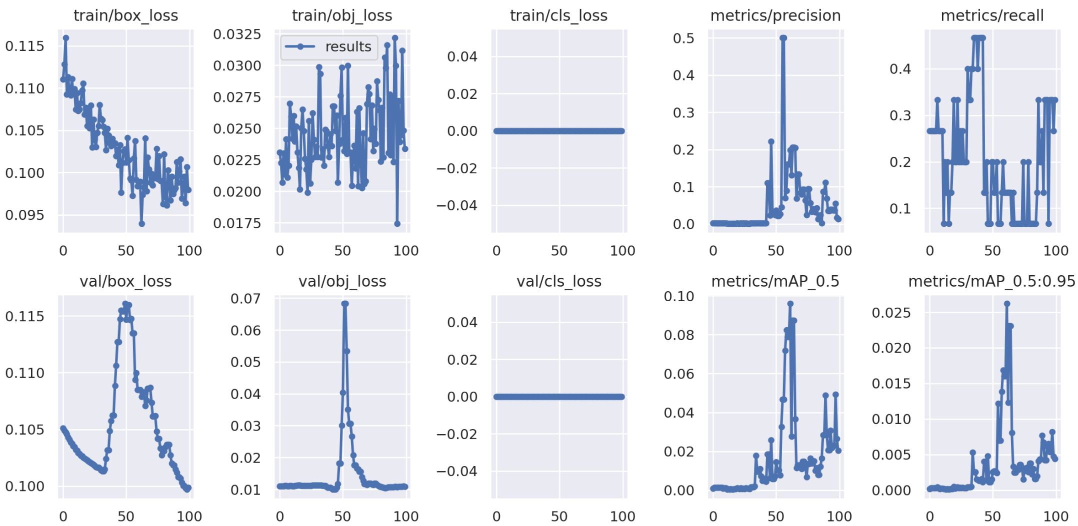

Figure 4.18: Obtained Graphs

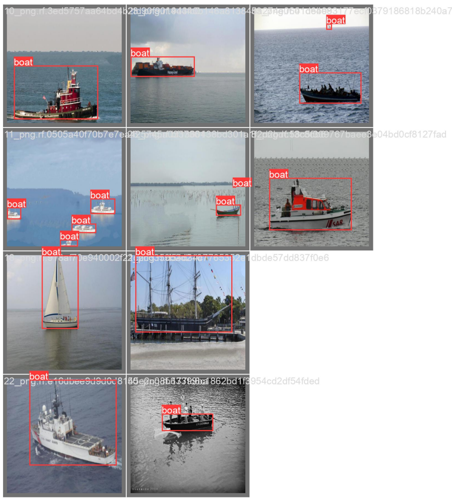

Figure 4.19: Ground Truth

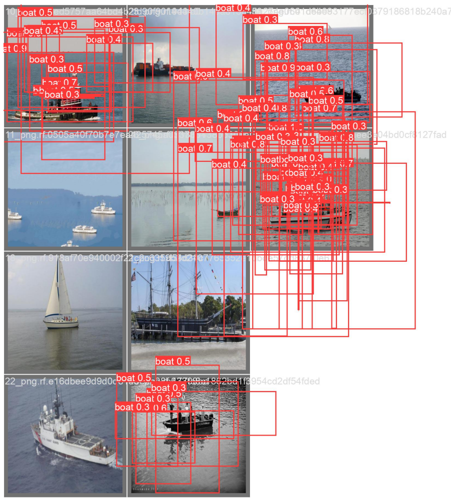

Figure 4.20: Prediction Scores

5. CONCLUSION

In this project, we have conducted an in-depth investigation of methods that can be used for the detection of marine surface vehicles. Our research has particularly focused on the areas of image segmentation, with an examination of WaSR Network, and object detection, where we have explored the use of the YOLOv5 model.

In order to improve the results from our real-time tests, we experimented with a method that involved manipulating the images used for training the model. This process was executed as follows:

With the assistance of artificial intelligence, we manipulated the images used for training the model to more sharply highlight the target objects. This technique was applied to approximately 50 data points. We hypothesized that this would improve the model's performance, a theory based on the principle that the quality of input data is a critical determinant of model performance.

However, this approach has potential pros and cons. On the positive side, it could contribute to increasing the model's sensitivity to the target object, possibly leading to more accurate object detection. Conversely, a potential downside could be that the model becomes overly sensitive to the manipulated images, affecting its ability to generalize when tested with different data. We were interested in observing how these two factors would intersect and impact the model's performance.

The subsequent sections present the test results obtained following this experiment. [11]

Without manipulating data, here what we get from experiment.

```
100 epochs completed in 0.029 hours.
Optimizer stripped from runs/train/yolov5s_results2/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results2/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results2/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 182 layers, 7246518 parameters, 0 gradients
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%
                   all         10         11      0.473     0.0909      0.107     0.0319
Results saved to runs/train/yolov5s_results2
CPU times: user 1.3 s, sys: 139 ms, total: 1.44 s
Wall time: 2min 7s
```

Figure 5.1: Outputs of Training

After manipulating data, the training outputs are shown in below.

```
100 epochs completed in 0.031 hours.
Optimizer stripped from runs/train/yolov5s_results3/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results3/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results3/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 182 layers, 7246518 parameters, 0 gradients
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 1/1
                   all         10         15      0.186      0.133     0.0855     0.0241
Results saved to runs/train/yolov5s_results3
CPU times: user 1.36 s, sys: 159 ms, total: 1.52 s
Wall time: 2min 16s
```

Figure 5.2: Outputs of Training

Our experiments yielded the following performance metrics before and after the image manipulation process.

Before manipulating images, the model achieved a precision of 0.473, indicating that approximately 47.3% of the objects detected by the model were indeed the target objects. The recall value was 0.0909, suggesting that the model detected about 9.09% of the total actual target objects present in the images. The Mean Average Precision (mAP) at Intersection over Union (IoU) of 50% (mAP50) was 0.107, and the mAP from IoU of 50% to 95% (mAP50-95) was 0.0319. These metrics offer insight into the model's ability to correctly identify objects and its robustness against different IoU thresholds.

Following the image manipulation, the model's precision decreased to 0.186, implying that the model's accuracy in identifying true positives dropped. Interestingly, the recall value increased slightly to 0.133, suggesting that the model was able to detect a larger portion of the actual target objects. However, this might be attributed to a higher number of false positives, as suggested by the lower precision. The mAP50 decreased to 0.0855, and the mAP50-95 decreased to 0.0241, both indicating a decrease in the model's overall object detection performance across different IoU thresholds.

These results suggest that while the image manipulation technique might have increased the model's sensitivity towards detecting the target objects (evident from the increase in recall), it might have also caused the model to incorrectly classify more non-target objects as target objects (evident from the decrease in precision). The decline in mAP50 and mAP50-95 scores further validates this interpretation. Therefore, while the image manipulation technique did lead to certain changes in the model's performance, its overall impact on the object detection task may need further investigation.

BIBLIOGRAPHY

[1] C. Grigorescu, N. Petkov, and M. Westenberg, "Contour detection based on nonclassical receptive field inhibition," *IEEE Trans. on Image Process.*, vol. 7, no. 12, pp. 729–739, 2003. DOI: 10.1109/tip.2003.814250.

[2] C. Berg, J. Endler, D. Papinczak, and K. Cheney, "Using colour pattern edge contrast statistics to predict detection speed and success in triggerfish (rhinecanthus aculeatus)," *Journal of Experimental Biology*, vol. 23, no. 225, 2022. DOI: 10.1242/jeb.244677.

[3] X. Shi, B. Wang, and J. Tsotsos, "Early recurrence improves edge detection," 2013. DOI: 10.5244/c.27.22.

[4] W. Zhou, X. Huang, and X. Zeng, "Obstacle detection for unmanned surface vehicles by fusion refinement network," *IEICE Trans. Inf. Syst.*, vol. 8, no. E105.D, pp. 1393–1400, 2022. DOI: 10.1587/transinf.2021edp7254.

[5] B. Borja and K. Matej, "A water-obstacle separation and refinement network for unmanned surface vehicles," 2020. DOI: 10.48550/arxiv.2001.01921.

[6] X. Chen, Y. Liu, and K. Achuthan, "Wodis: Water obstacle detection network based on image segmentation for autonomous surface vehicles in maritime environments," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021. DOI: 10.1109/tim.2021.3092070.

[7] L. Yao, D. Kanoulas, Z. Ji, and Y. Liu, "Shorelinenet: An efficient deep learning approach for shoreline semantic segmentation for unmanned surface vehicles," 2021. DOI: 10.1109/iros51168.2021.9636614.

[8] Z. Shao, H. Lyu, Y. Yin, *et al.*, "Multi-scale object detection model for autonomous ship navigation in maritime environment," *JMSE*, vol. 11, no. 10, p. 1783, 2022. DOI: 10.3390/jmse10111783.

[9] Z. Wang and Y. Zhou, "A hed-optimized automatic detection and tracking algorithm for marine moving targets based on yolo v3," *J. Phys.: Conf. Ser.*, vol. 1, no. 1449, p. 012 126, 2020. DOI: 10.1088/1742-6596/1449/1/012126.

[10] G. Jocher, *Yolov5 by ultralytics*, version 7.0, May 29, 2020. DOI: 10.5281/zenodo.3908559. [Online]. Available: https://github.com/ultralytics/yolov5.

[11] Caglalogy, *Gradproject*, GitHub repository, 2023. [Online]. Available: https://github.com/caglalogy/GradProject.