

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**VISION-BASED DETECTION AND RECOGNITION OF MARITIME OBJECTS
FOR AUTONOMOUS SURFACE NAVIGATION**



M.Sc. THESIS

Sevda SAYAN YONCA

Department of Computer Engineering

Computer Engineering Programme

JUNE 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**VISION-BASED DETECTION AND RECOGNITION OF MARITIME OBJECTS
FOR AUTONOMOUS SURFACE NAVIGATION**

M.Sc. THESIS

**Sevda SAYAN YONCA
(504211542)**

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Hazım Kemal Ekenel

JUNE 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**OTONOM DENİZ SEYRÜSEFERİ İÇİN GÖRÜNTÜ TABANLI
ENGEL TESPİTİ VE GEMİ SINIFLANDIRMA**

YÜKSEK LİSANS TEZİ

**Sevda SAYAN YONCA
(504211542)**

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hazım Kemal Ekenel

HAZİRAN 2025

Sevda SAYAN YONCA, a M.Sc. student of ITU Graduate School student ID 504211542 successfully defended the thesis entitled “VISION-BASED DETECTION AND RECOGNITION OF MARITIME OBJECTS FOR AUTONOMOUS SURFACE NAVIGATION”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Hazım Kemal Ekenel**
Istanbul Technical University

Jury Members : **Prof. Mine Elif Karşıl**
Yıldız Technical University

Dr. Öğr. Üyesi Yusuf Hüseyin Şahin
Istanbul Technical University

.....

Date of Submission : **13 May 2025**
Date of Defense : **13 June 2025**





To my family,



FOREWORD

First of all, I would like to express my gratitude to my valuable thesis advisor Prof. Dr. Hazım Kemal EKENEL, who has always helped me with his valuable knowledge, experience and time throughout my master's degree, from determining the study topic to completing the studies, whenever I needed it.

I am deeply grateful to my family, who has always supported me throughout my life, especially to my dear mother Ruhiye SAYAN and father Medain SAYAN, who raised me with love and never withheld their support. I would also like to thank my husband Evren YONCA, who has met this process with patience, love and understanding and has been my greatest supporter. His patience and calming presence helped me endure the stressful moments of this process with ease.

Furthermore, I am truly thankful to all my friends who shared both my professional and social life and contributed positively to my journey with their uplifting energy. In particular, I would like to express my special appreciation to Selin GÖK IŞIK and Alkın ALKAN, whose constant support and encouragement kept my motivation high whenever I needed help.

Finally, I would like to extend my gratitude to my company, ASELSAN Inc., for supporting this study through the Aselsan Academy program.

June 2025

Sevda SAYAN YONCA
(Software Engineer)

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xiii
SYMBOLS	xv
LIST OF TABLES	xvii
LIST OF FIGURES	xx
SUMMARY	xxi
ÖZET	xxiii
1. INTRODUCTION	1
1.1 Purpose of Thesis	3
1.2 Literature Review	4
1.2.1 Object detection	5
1.2.2 Ship classification	7
1.3 Hypothesis	10
2. BACKGROUND	11
2.1 YOLOv5	11
2.2 TPH-YOLOv5	12
2.3 ViT (Vision Transformers)	14
2.4 DeiT (Data-efficient Image Transformer)	16
2.5 Swin Transformer (Shifted Windows Transformer)	17
2.6 Attention Rollout	17
2.7 Pruning	18
3. METHODS	21
3.1 Datasets	21
3.1.1 MODS dataset	21
3.1.2 MARVEL dataset	23
3.2 Data Preprocessing and Adaptation	24
3.2.1 Adaption of MODS	24
3.2.2 Adaption of MARVEL	24
3.3 Training and fine-tuning	25
3.3.1 Object detection	26
3.3.2 Ship classification	28
3.4 Attention Rollout Extraction	28
3.5 Pruning	30
4. EXPERIMENTAL RESULTS	31
4.1 Evaluation Metrics	31
4.1.1 F1 score	31
4.1.2 mAP (Mean Average Precision)	32
4.1.3 Accuracy	33
4.1.4 FPS (Frames Per Second)	33
4.2 Experimental Setup	34
4.3 Object Detection Results	35
4.4 Ship Classification Results	39
5. CONCLUSIONS	49
REFERENCES	51
CURRICULUM VITAE	56



ABBREVIATIONS

AP	: Average Precision
CBAM	: Convolutional Block Attention Module
CNN	: Convolutional Neural Network
CSPNet	: Cross Stage Partial Network
DeiT	: Data-Efficient Image Transformer
FLOPs	: Floating Point Operations per Second
FPS	: Frame Per Second
FP	: False Positive
FP	: Feature Pyramid Network
GAN	: Generative Adversarial Networks
IMO	: International Maritime Organization
IMU	: Inertial Measurement Unit
IOU	: Intersection Over Union
LASR	: Large-Scale Panoptic Maritime Obstacle Detection Dataset
LR	: Learning Rate
LWIR	: Long-Wavelength Infrared
MARVEL	: Maritime Vessels
MLP	: Multilayer Perceptron
MODS	: Maritime Obstacle Detection System
MSA	: Multi-Head Self-Attention
NMS	: Non-Maximum Suppression
NIR	: Near-Infrared
NLP	: Natural Language Processing
PANet	: Path Aggregation Network
ResNet	: Residual Network
SENet	: Squeeze-and-Excitation Networks
SKNet	: Selective Kernel Networks
SMD	: Singapore Maritime Dataset
SVM	: Support Vector Machine
SwinT	: Shifted Window Transformer
SSD	: Single Shot MultiBox Detector
TP	: True Positive
TPH	: Transformer Prediction Heads
USV	: Unmanned Surface Vehicle
USVs	: Unmanned Surface Vehicles
VAIS	: Visible and Infrared Ship Dataset
ViT	: Vision Transformers
VIS	: Visual-Optical
YOLO	: You Only Look Once
WaSR	: Water Segmentation and Fine-Tuning



SYMBOLS

H	: Height
W	: Width
C	: Number of channels
x	: Input image tensor
$\mathcal{R}$	: Set of real numbers
P	: Patch size
$A^{(l)}$	: Attention matrix at layer l
$\hat{A}^{(l)}$	: Fused attention matrix via max operation at layer l
I	: Identity matrix, representing residual connections
$\tilde{A}^{(l)}$	: Attention rollout matrix up to layer l
$\cdot$	: Matrix multiplication operator
$\sum(\cdot)$	: Summation operator used for normalization
l	: Index of the transformer layer
k	: Index offset for selecting entries after the CLS token
$\dim$	: Dimension along which the mean is computed (e.g., batch axis)
$\max(\cdot)$	: Element-wise maximum function
$\cdot$	: Matrix multiplication
$[...]$	: Python-like slicing notation



LIST OF TABLES

	<u>Page</u>
Table 1.1 : Summary of maritime datasets with the number of classes, class names, and total number of images or frames.	9
Table 4.1 : MARVEL class distribution before (B) and after (A) applying augmentation to balance the number of samples per classes.	35
Table 4.2 : Selected key hyperparameters for YOLOv5 training.	36
Table 4.3 : Detailed comparison of performance metrics for various YOLOv5 models on the MODS [1] dataset.	36
Table 4.4 : Class-specific performance metrics for YOLOv5 models on the MODS dataset.	37
Table 4.5 : Hyperparameters used for training different models.	42
Table 4.6 : Performance Comparison of Various Neural Network Architectures on the MARVEL [2] Dataset.	44
Table 4.7 : Comparison of Image Classification Models' Computation Loads ...	44



LIST OF FIGURES

	<u>Page</u>
Figure 2.1 : The architecture and components of YOLOv5 [3] model.	12
Figure 2.2 : The architecture of TPH-YOLOv5 model. Taken from [4].	13
Figure 2.3 : Schematic representation of the Vision Transformer (ViT) pipeline. Taken from [5].	14
Figure 3.1 : Example images from the MODS dataset representing various environmental conditions. (a) a harbor scene with partially distorted visibility due to sun reflection, (b) a wavy sea surface close to the shore, (c) a foamy water trail created by a moving boat, (d) a small boat moving on a calm water surface in the open sea, (e) sailboats positioned in different directions, (f) a harbor scene with buoys and the sun directly reflected on the camera near sunset.	22
Figure 3.2 : Examples of raw and preprocessed images from the MARVEL dataset. Original images as collected from the dataset and the corresponding images after preprocessing, including operations such as horizontal flipping, translation, brightness and contrast adjustments, and scaling.	24
Figure 3.3 : Post-training analysis from YOLOv5 for MODS. (a) The bar chart shows the distribution of object classes in the dataset. (b) The scatter plot represents object positions in the image space. (c) The scatter plot visualizes the width-height distribution of detected objects.	25
Figure 3.4 : Sample images from the MODS dataset, highlighting annotations for small-sized objects.	27
Figure 3.5 : Graphical representation of training process of the vision transformers on MARVEL dataset.	28
Figure 4.1 : Comparison of object detection performance across three YOLO model variants (YOLOv5-s , YOLOv5-l, TPH-YOLOv5) on the MODS dataset, illustrating detection outcomes in various maritime scenes.	38
Figure 4.2 : Comparative analysis of learning rate optimization for different transformer-based and convolutional models. Each row represents the performance of a specific model—ViT, DeiT, Swin-T, Swin-B, and ConvNext v2—trained with different learning rates. The left column shows the training loss curves, while the right column presents the evaluation accuracy over training steps.	41

Figure 4.3 : Training loss and evaluation accuracy curves for different models. The training loss graph illustrates the reduction in loss over training steps, while the evaluation accuracy graph shows the accuracy progression throughout the training process. **43**

Figure 4.4 : Attention rollout maps for true and false predicted samples on vision transformer models. **45**

Figure 4.5 : Sparsity vs Accuracy and Inference time change when applying pruning to our fine-tuned DeiT model. **46**



VISION-BASED DETECTION AND RECOGNITION OF MARITIME OBJECTS FOR AUTONOMOUS SURFACE NAVIGATION

SUMMARY

Unmanned Surface Vehicles (USVs) are robotic systems capable of performing operations in various marine environments autonomously or via remote control, without the need for direct human intervention. Their use in fields such as environmental monitoring, military surveillance, reconnaissance, and search and rescue missions has driven a significant transformation in the maritime industry. The critical role of USVs in these operations increases the demand for advanced technologies to further enhance their capabilities.

One of the key challenges encountered by these vehicles, which possess autonomous route planning capabilities, is the accurate detection and classification of objects that may pose obstacles on the water surface. This requirement is vital for mission success, navigational safety, and traffic management. However, achieving high accuracy in object detection and ship classification is highly challenging due to environmental factors like changing weather conditions, wave movements, reflections, and varying light levels. These variables can cause image distortions that reduce the reliability of detection systems. In addition, the diversity in the shape, size and motion of the ships adds further complexity to the classification process.

Recent advances in deep learning offer effective solutions to these challenges. Among them, object detection models like YOLO (You Only Look Once) stand out for real-time applications thanks to their speed and accuracy. YOLOv5, an advanced version of YOLO, with its lightweight architecture and optimized performance, is a suitable model for obstacle detection. It enables the fast and reliable identification of dynamic maritime objects, such as ships and buoys, even under challenging conditions. To improve the detection performance of small-scale objects, the TPH-YOLOv5 model integrates Transformer Prediction Heads (TPH) and Convolutional Block Attention Modules (CBAM), enhancing feature representations through self-attention mechanisms.

In the deep learning field, recent breakthroughs—particularly Vision Transformers (ViT)—have revolutionized image classification tasks. Unlike traditional Convolutional Neural Networks (CNNs), Vision Transformers apply the Transformer architecture which is originally designed for Natural Language Processing (NLP), to image classification tasks. ViTs divide images into patches and process them as tokens, similar to NLP workflows, by applying self-attention to model the dependencies among different patches. Variants such as Data-Efficient Image Transformers (DeiT), Swin Transformers, and ConvNeXt have further optimized this approach. DeiT

offers data efficiency through techniques like distillation, Swin Transformers improve computational efficiency by using a hierarchical structure with shifted windows, and ConvNeXt modernizes CNN architectures by integrating transformer-like features into neural networks.

This study presents a two-stage strategy for tackling the difficulties of real-time object detection and classification on USVs and to improve their performance in the maritime domain. Two large datasets, MODS and MARVEL, are utilized that provide comprehensive benchmarking opportunities for maritime object detection and ship classification.

In the first stage, MODS dataset was used to train and evaluate YOLOv5 based object detection models. This dataset contains bounding box annotations of dynamic obstacles (e.g. ships and people) and static obstacles (e.g. buoys). The study focuses on the detection of small and dynamic objects, which are critical to avoid collisions. For this purpose, it was necessary to fit the data for the model. Therefore, a preprocessing step was applied by converting the annotations into a YOLO compatible format. Among the models tested, TPH-YOLOv5, which is optimized for small object detection, demonstrated superior performance in precision metrics (0.894%), recall (0.83%) and average precision (0.863%), proving its effectiveness in detecting small obstacles.

In the second stage, the MARVEL dataset which includes 26 subtypes of ships divided into five main categories was used for the classification of ships. Because of the size and class imbalance of the dataset, data preprocessing techniques such as rotation and scaling were applied to maintain the balance and improve model generalization. In this stage, the implementations of ViT models and their variants (DeiT, Swin Transformers, ConvNeXt-v2) were investigated. These models outperformed traditional CNNs in terms of accuracy; for example, DeiT achieved the highest classification accuracy of 92.87%, followed by Swin Transformer with 90.44% and ConvNext with 90.07%. It presented an accuracy of 91.76%, which shows the best performance among the CNN architectures used to investigate the impact of ViT-based models. However, despite their strong accuracy, the average inference times of transformer-based models (e.g., DeiT: 1.31 fps, Swin Transformer: 0.51 fps) were significantly higher than those of CNNs (ResNet50: 52.66 fps), which limits their practicality for real-time use.

In this study, various pruning techniques were applied to enhance the computational efficiency of transformer-based models, and their effects were observed. Inference times were reduced by selectively removing low-impact weights and neurons using structural and non-structural pruning methods. This process was carefully applied to ensure minimal impact on classification accuracy and made the models better adapted for real-time deployment on USVs, where speed and precision are critical. However, the inference time of the pruned transformers still lag behind the frames-per-second (FPS) performance of traditional CNNs. These observations highlights the need for further advances in transformer optimization, especially in dynamic marine environments, where rapid decision-making is crucial.

Another key contribution of this work is the development of a general heatmap generation method for vision transformers. This method produces attention flow maps that visually represent the areas on which the model focuses.

OTONOM DENİZ SEYRÜSEFERİ İÇİN GÖRÜNTÜ TABANLI ENGEL TESPİTİ VE GEMİ SINIFLANDIRMA

ÖZET

İnsansız Deniz Araçları (İDA), deniz ortamlarında doğrudan insan müdahalesine gerek kalmadan otonom yada bir yer kontrol istasyonu aracılığıyla uzaktan kumandalı olarak görev yapabilen robotik sistemlerdir. Çevresel izleme, askeri gözetim, keşif ve arama-kurtarma gibi alanlardaki kullanımları, denizcilik sektöründe önemli bir dönüşüme yol açmıştır. Bu araçların söz konusu operasyonlardaki kritik rolü, yeteneklerini daha da geliştirecek ileri teknolojilere olan talebi artırmaktadır.

İDA'ların sahip olduğu otonom rota planlama ve hareket kabiliyetlerine rağmen, karşılaştıkları en büyük zorluklardan biri, su yüzeyinde engel oluşturabilecek nesnelerin doğru ve tutarlı bir şekilde algılanması ve sınıflandırılmasıdır. Bu gereklilik, görev başarısı, seyir güvenliği ve deniz trafiği yönetimi açısından yüksek bir öneme sahiptir. Ancak deniz ortamı, nesne tespiti ve sınıflandırma süreçlerinde yüksek doğruluk elde edilmesini zorlaştıran çok sayıda çevresel değişkene sahiptir. Değişken hava koşulları, dalga hareketleri, güneş yansımaları ve değişken ışık seviyeleri gibi faktörler, görüntü kalitesini doğrudan etkileyerek algılama sistemlerinin güvenilirliğini düşürebilmektedir. Ayrıca, gemilerin boyut, şekil, malzeme yapısı ve hareket modelleri açısından çeşitlilik göstermesi, sınıflandırma görevini daha da karmaşık hale getirmektedir.

Radar ve LIDAR gibi sensörler çeşitli otonom sistemlerde yaygın olarak kullanıma rağmen, deniz ortamlarında çoğunlukla etkin performans gösterememektedir. Radar sistemleri, özellikle küçük ve düşük profilli nesneleri tespit etmekte zorlanmakta ve dalga kaynaklı yansımalar nedeniyle yanlış alarm üretebilmektedir. LIDAR ise, su yüzeyinin düşük yansıtıcılığı ve değişken doğası nedeniyle güvenilir veri sağlayamamakta, ayrıca açık deniz gibi menzil açısından zorlu ortamlarda performans kaybı yaşamaktadır. Bu nedenlerle, kamera tabanlı algılama sistemleri deniz ortamları için daha avantajlı bir alternatif sunmaktadır. Kameralar, yüksek çözünürlükte detaylı görsel bilgi sağlayarak küçük nesnelerin, insan figürlerinin ve diğer potansiyel tehlikelerin daha hassas bir şekilde tespit edilmesini mümkün kılmaktadır. Ayrıca, görüntü verileri üzerine makine öğrenmesi ve derin öğrenme yöntemlerinin uygulanabilmesi, nesne tespiti ve sınıflandırma süreçlerinde esnekliği ve doğruluğu artırmaktadır.

Kamera tabanlı derin öğrenme çözümleri, deniz ortamında karşılaşılan bu zorluklara karşı etkili sonuçlar sunmaktadır. Özellikle YOLO (You Only Look Once) gibi hızlı ve doğruluk odaklı nesne tespiti algoritmaları, gerçek zamanlı uygulamalar için ön plana çıkmaktadır. YOLO'nun optimize edilmiş bir versiyonu olan YOLOv5, hafif yapısı, hız ve doğruluk dengesi ile dinamik deniz ortamlarında küçük nesne tespitine uygun

bir modeldir. Ayrıca, TPH-YOLOv5 modeli, küçük nesne tespit performansını daha da artırmak amacıyla Transformer Prediction Head (TPH) ve Convolutional Block Attention Module (CBAM) mekanizmalarını entegre ederek, dikkat mekanizmaları aracılığıyla öznitelik temsillerini zenginleştirmiştir. Bu sayede, düşük çözünürlüklü ve karmaşık deniz ortamlarında küçük nesne tespiti daha güvenilir hale getirilmiştir.

Derin öğrenme alanındaki bir diğer önemli gelişme ise Görüntü Dönüştürücüler (ViT) mimarisidir. Geleneksel Evrimsel Sinir Ağlarından (CNN) farklı olarak ViT'ler, görüntüleri küçük parçalara (patch) bölerek bu parçaları doğal dil işleme alanındaki kelimelere benzer token'lar gibi işler. Bu tokenlar arasındaki ilişkiler, self-attention mekanizmasıyla modellenir. Böylece, görüntünün farklı bölgeleri arasındaki uzun menzilli bağlantılar öğrenilebilir. Veri verimliliğini artırmak için geliştirilmiş olan Data-Efficient Image Transformer (DeiT), hesaplama verimliliği odaklı Swin Transformer ve CNN ile Transformer yaklaşımlarını harmanlayan ConvNeXt gibi mimariler, görüntü dönüştürücü ailesini daha geniş bir kullanım alanına yaymıştır.

Bu tezde, İDA'ların deniz ortamında nesne tespiti ve gemi sınıflandırması problemlerine çözüm getirmek amacıyla iki aşamalı bir yaklaşım geliştirilmiştir. Çalışmada, hem nesne tespiti hem de sınıflandırma görevleri için kapsamlı veri setleri olan MODS ve MARVEL kullanılmıştır.

İlk aşamada, MODS veri seti kullanılarak YOLOv5 tabanlı nesne tespiti modelleri eğitilmiş ve değerlendirilmiştir. MODS, stereo kamera görüntüleri üzerinden hem dinamik (gemi, insan) hem de statik (şamandıra gibi) engelleri içeren zengin anotasyonlara sahiptir. Veri setindeki etiketler YOLO formatına dönüştürülerek modelin eğitimi için uygun hale getirilmiştir. Nesne tespiti için TPH-YOLOv5 modeli seçilmiş ve özellikle küçük nesne algılama performansını artıracak şekilde yapılandırılmıştır. TPH (Transformer Prediction Head) yapısı, farklı ölçeklerdeki nesneleri daha iyi öğrenebilmek için kestirim katmanlarına dönüştürücü modülleri entegre etmiştir. Buna ek olarak, CBAM (Convolutional Block Attention Module) mekanizması kullanılarak, ağır dikkatini görsel olarak önemli bölgelere yönlendirmesi sağlanmıştır. Geliştirilen TPH-YOLOv5 modeli, yapılan değerlendirmelerde yüksek doğruluk (%89.4), yüksek geri çağırma (%83) ve ortalama doğruluk (%86.3) değerleri elde ederek küçük ve zor tespit edilen nesnelerde önemli başarı sağlamıştır.

İkinci aşamada, MARVEL veri seti kullanılarak gemi sınıflandırması gerçekleştirilmiştir. MARVEL, beş ana kategoriye ayrılmış toplam 26 farklı gemi alt tipini içermektedir. Veri setindeki dengesizlikler, çeşitli veri artırma teknikleriyle giderilmiştir. Sınıflandırma aşamasında ViT, DeiT, Swin Transformer ve ConvNeXt-v2 gibi farklı görüntü dönüştürücü tabanlı modeller eğitilmiş ve test edilmiştir. Bu modellerin doğrulukları CNN tabanlı ResNet50 ve ResNet101 gibi modellerle karşılaştırılmıştır. Sonuçlar, DeiT modelinin %92.87 doğruluk oranı ile en iyi performansı sergilediğini göstermiştir. Ancak, FPS (Frames Per Second) değerleri dikkate alındığında, ViT tabanlı modellerin çıkarım hızlarının CNN mimarilerine göre daha düşük olduğu gözlemlenmiştir (örneğin, ResNet50: 52.66 FPS, DeiT: 1.31 FPS). Bu durum, ViT tabanlı modellerin, gerçek zamanlı uygulamalarda kullanılabilmesi için, optimizasyon gerektirdiğini ortaya koymuştur.

Denizcilik uygulamalarında, özellikle İDA'ların çevresel farkındalık ve engelden kaçınma görevlerinde, kararların milisaniyeler içinde verilmesi hayati öneme sahiptir. Bu nedenle, kullanılan algılama ve sınıflandırma modellerinin sadece yüksek doğruluk sunması yeterli değildir; aynı zamanda düşük gecikmeli, yani yüksek çıkarım hızına (FPS) sahip olması da kritik bir gerekliliktir. Dönüştürücü tabanlı modellerin yüksek doğruluk sağlamalarına rağmen çıkarım sürelerinin (inference time) gerçek zamanlı uygulamalar için yetersiz olduğu belirlenmiştir. Bu sorunu aşmak amacıyla hem yapısal hem de yapısal olmayan budama (structured ve unstructured pruning) yöntemleri uygulanmıştır. Özellikle DeiT modeli üzerinde yapılan budama çalışmaları sonucunda, model parametrelerinde anlamlı bir azalma sağlanmış ve çıkarım hızında yaklaşık %11 oranında iyileşme elde edilmiştir. Bu süreçte model doğruluğunda sadece %0.39'luk küçük bir kayıp yaşanmıştır, bu da performans-hız dengesinin başarılı bir şekilde optimize edildiğini göstermektedir. Bununla birlikte, dönüştürücü tabanlı modellerin çıkarım hızlarının hâlen CNN tabanlı modellere göre daha düşük olduğu ve gerçek zamanlı İDA operasyonları için daha ileri düzey optimizasyonların gerekli olduğu sonucuna varılmıştır.

Tez çalışmasında yalnızca performans artırımı değil, aynı zamanda model kararlarının açıklanabilirliğine yönelik çalışmalar da yapılmıştır. Görüntü dönüştürücü tabanlı modellerin karar verme süreçlerini şeffaf hale getirmek amacıyla genel bir dikkat akışı (attention rollout) görselleştirme yöntemi geliştirilmiştir. Bu yöntem, modelin sınıflandırma yaparken hangi görüntü bölgelerine odaklandığını göstererek doğru ve yanlış sınıflandırmaların sebeplerinin daha iyi anlaşılmasını sağlamıştır. Doğru tahminlerde modelin hedef nesne üzerinde odaklandığı, yanlış tahminlerde ise genellikle su yüzeyi ya da arka plan gibi ilgisiz alanlara yöneldiği tespit edilmiştir.

Sonuç olarak, bu tez, İDA'lar için deniz ortamında hem güvenli hem de etkin bir şekilde çalışabilecek yüksek doğruluklu ve yorumlanabilir yapay zeka tabanlı sistemlerin geliştirilmesine katkı sağlamaktadır. Elde edilen bulgular, gelecekte daha hızlı ve daha optimize edilmiş dönüştürücü tabanlı sistemlerin geliştirilmesi için önemli bir temel oluşturmaktadır. Her ne kadar geliştirilen modeller eğitim ve test aşamalarında yüksek doğruluk değerleri elde etmiş olsa da, gerçek deniz ortamlarında çalışacak sistemler için modelin genelleme yeteneği kritik bir faktördür. Eğitimde kullanılan MODS ve MARVEL veri setleri farklı deniz koşullarını kapsasa da, tüm olası çevresel değişkenleri temsil etmek mümkün değildir. Özellikle aşırı dalgalı denizler, yoğun güneş yansımaları veya düşük görüş koşulları gibi ekstrem senaryolar, model performansında düşüşe neden olabilir. Gerçek dünya koşullarında tam güvenilirlik için saha testlerinin yapılması ve modellerin farklı deniz ortamlarında yeniden değerlendirilmesinin gerekliliği açıktır.



1. INTRODUCTION

Unmanned Surface Vehicles (USVs) have become indispensable in both military and civilian fields, ensuring human safety during remote-controlled and autonomous operations. These vehicles, expected to play crucial role in future fleets, eliminate the risks of injury or loss of life while enabling advanced capabilities in research, reconnaissance, observation, and military missions. Their modularity and ability to operate autonomously or in specialized configurations make USVs highly adaptable and advantageous, especially in unpredictable maritime conditions.

Unlike manned vessels, they operate without direct human intervention, making their obstacle detection systems critical for mission success and safety. For a fully autonomous ship voyage, the vessel must progress from one point to the target point without human intervention, think independently to make decisions in the face of potential problems that may arise. A critical component of these decisions is the ability to accurately detect and classify maritime objects, such as ships, buoys and person, to ensure navigation safety and enhance situational awareness.

USVs utilize a variety of sensor technologies to detect and classify obstacles. Radar stands out as one of the primary technologies in maritime navigation, providing long-range detection capabilities and functions well in low-visibility conditions such as fog or rain. However, radars cannot detect all types of floating objects on the water. The magnetic waves they emit need to bounce back by striking structures, particularly those made of materials such as metal. For this reason, boats made of fiberglass, wooden vessels, floating platforms, and similar objects may not reflect these waves and therefore may not appear on radar screens. Lidar systems are another tools, which use laser beams to generate high-resolution 3D maps, excel at detecting obstacles above the water surface but can struggle in harsh weather. Optical cameras, both RGB and thermal, provide detailed visual data for classification tasks. Unlike radar or lidar, optical cameras offer rich contextual information such as color, texture, and

object type. It enables precise identification of vessels, buoys, debris, or swimming animals. This capability is particularly important in complex scenarios where visual confirmation of an object's nature is required, such as distinguishing between a floating log and a small boat.

The maritime environment presents unique challenges for obstacle detection. It is inherently dynamic, with changing weather, and wave patterns. These factors demand robust systems capable of operating in unpredictable and noisy environments. To list challenges specific to object detection and classification:

- **Environmental Factors:** Weather variability, including fog, rain, and storms, can obscure visibility, while lighting changes, such as sun glare or low light conditions at night, further complicate detection. The constant motion of waves creates a shifting background that blends with objects, and water reflections introduce noise and distortions in images.
- **Object Diversity:** Variation in size, shape, color, and small object detection issues.
- **Scene Complexity:** Dynamic backgrounds, cluttered scenes, and occlusions.
- **Technological Constraints:** Limited sensor resolution, real-time processing demands, and low-quality datasets.

In this work, advancements were made in addressing obstacle detection and classification challenges in marine settings by utilizing the robust feature learning capabilities of deep CNN architectures. To adapt pre-trained models, originally designed for different tasks, to the problem at hand, fine-tuning was applied to these models. All stages of the training process were thoroughly analyzed. In addition, Vision Transformers (ViTs) [5] have been investigated as a modern alternative to CNNs. While CNNs excel in efficiency and speed, ViTs offer improved accuracy and interpretability through global attention mechanisms, providing a fresh perspective on solving maritime detection challenges. The proposed strategy was evaluated on benchmark datasets commonly used in the field of marine obstacle detection. Various

preprocessing methods were applied to the images in these datasets, and the deep convolutional neural network architectures were fed with different types of input data.

1.1 Purpose of Thesis

This thesis aims to address critical challenges in object detection and ship classification within maritime environments, focusing on Unmanned Surface Vehicles (USVs). USVs rely on accurate detection and classification of objects such as vessels, humans, and other maritime features for safe navigation, effective monitoring, and operational efficiency in dynamic and challenging situations.

This thesis focuses on achieving the following goals:

- Improve real-time object detection accuracy by leveraging YOLOv5-based [3] models, with a focus on small and dynamic object detection using TPH-YOLOv5 [4].
- Enhance maritime object classification using transformer-based models, such as ViTs and their variants (Data-Efficient Image Transformer (DeiT) [6], Shifted Window Transformer (Swin Transformer) [7], and ConvNeXt-v2 [8]). For accurate ship classification on the MARVEL dataset, leveraging attention mechanisms to capture global context effectively.
- Optimize model deployment to employ model pruning techniques and analyze attention mechanisms, enhancing computational efficiency and interpretability for real-time applications on USVs.

By leveraging state-of-the-art datasets (MODS [1] and MARVEL [2]), this study provides comprehensive evaluations, addressing small object detection, class imbalance, and computational constraints. The thesis contributes to advancing USV technology by integrating robust, scalable, and interpretable models into maritime operations, improving both accuracy and speed in object detection and classification. This work is essential for enhancing the safety, efficiency, and functionality of USVs in modern maritime environments.

1.2 Literature Review

When we examine the recent literature on object detection and classification in marine environments, we observe that the base of the methods is generally focused on CNN learning algorithms. Studies often complain about difficulties (adverse environmental conditions in marine environments, water reflections, changing weather conditions, etc.) that negatively affect the detection and classification accuracy. This section reviews the important studies and datasets in the field and highlights their methodologies, results, and contributions.

With the increasing use of USVs in recent years, object detection in the maritime domain has emerged as a key research focus for these vehicles to safely avoid obstacles. To achieve this goal, datasets with different environmental conditions and various coverages were created and some of them were used as challenge platforms to provide benchmarks to researchers (see Table 1.1). While LaRS (Large-Scale Panoptic Maritime Obstacle Detection Dataset) [9] focuses on pattern segmentation with annotations suitable for anomaly detection, SeaShips [10] is designed for object detection on six ship classes under varying conditions. SMD (Singapore Maritime Dataset) [11] offers multi-spectral sensing capabilities with both visual-optical and infrared videos, making it suitable for day and night surveillance scenarios. In contrast, McShips (Multi-Category Ship Dataset) [12] emphasizes classification and retrieval by providing detailed metadata such as ship type and hull dimensions. VAIS (Visible and Infrared Ship Dataset) [13] stands out by supporting multi-spectral learning, which is especially valuable for low-light maritime operations, by combining visible and infrared imaging.

Compared to these datasets, MARVEL [2] offers the largest scale with a comprehensive class diversity, focusing on the classification of ships in 26 categories, thus providing a comprehensive benchmark for large-scale maritime ship recognition tasks. On the other hand, MODS [1] provides real-time obstacle detection scenarios for USVs, featuring diverse maritime objects and challenging environmental conditions.

1.2.1 Object detection

Unlike terrestrial environments, maritime scenes are characterized by unstable horizons, dynamic waves, sun glare, and strong reflections, all of which degrade object visibility. Small targets may be camouflaged against complex water textures or partially submerged, further reducing contrast. Moreover, scale variation is extreme objects such as distant ships, kayaks, or floating debris may appear minuscule even in high-resolution imagery.

Traditional detectors such as Faster R-CNN [14], SSD [15], and YOLOv3 [16] perform inadequately under these conditions due to their reliance on features extracted in fixed receptive fields and their tendency to lose fine-grained spatial detail through downsampling.

To address multiscale detection, Feature Pyramid Networks (FPN) [17] and PANet architectures have been widely adopted to combine low-level detail and high-level semantics. Such designs have been used successfully in maritime datasets like SeaDronesSee [18] and MODS (Maritime Object Detection System) [1]. However, these methods still suffer from background interference caused by sea clutter.

The MODS [1] dataset, introduced in the MaCVi'23 [19] challenge, is crucial real-world marine scenarios for object detection and segmentation. MaCVi'23 participants demonstrated innovative methods to detect small objects in challenging conditions by balancing accuracy with extraction speed. The prominent problems in this challenge were the difficulties created by variables such as light, waves, fog in the marine environment, data scarcity and lack of standardized benchmarks, model optimization for real-time applications, etc. The challenge results indicate a growing interest in transformer-based architectures, although CNNs continue to be the primary models used for these tasks. Among them, YOLOv7 [20], a simple but effective single-stage detector, remains a widely favored option. Achieving high accuracy is frequently linked to utilizing high-resolution inputs. This can be achieved through architectures that support large resolutions or by using targeted augmentation techniques such as clipping.

The Mastr1325 [21] and MODD [22] datasets, which are widely used in this field, further demonstrate the usefulness of semantic segmentation for the detection and avoidance of sea obstacles. Mastr1325 contains 1,325 pixel-wise annotated images for segmentation, while MODD consists of 12 videos (4,454 frames) captured by USVs under various conditions. These datasets support efficient horizon detection and obstacle segmentation, enabling real-time navigation with over 90% accuracy in various lighting and traffic scenarios. The results highlight the practical utility of segmentation techniques in improving safety and autonomy in USVs.

Another important work includes the WaSR (Water Segmentation and Fine-Tuning) [23] model, which uses semantic segmentation to identify water, obstacles, and sky in sea images. This model increases detection robustness by dividing the scene into meaningful regions and aiding in obstacle identification. One variant, eWaSR [24], adapts the model to embedded systems by reducing computational requirements while maintaining comparable accuracy. Another variant, WaSR-T [25], incorporates temporal texture information to improve performance in dynamic lighting and water conditions. These models highlight the potential of segmentation-based approaches in improving obstacle detection for USVs, especially in computationally constrained environments.

To further enhance discriminability, attention mechanisms like SENet [26] and SKNet [27] have been integrated into maritime detection pipelines. These modules help focus on salient regions by recalibrating feature maps, making them particularly valuable for identifying boats or obstacles in cluttered oceanic scenes.

In transformer-based designs, the Swin Transformer [7] introduces shifted windows and hierarchical attention, capturing long-range dependencies and improving robustness against deformations caused by waves and motion blur — a frequent issue in videos captured from USVs or surveillance platforms.

One of the largest bottlenecks in maritime detection is the scarcity of annotated data for small or rare maritime targets. Maritime datasets often exhibit extreme class imbalance, with a predominance of large vessels and few small floating targets. To

alleviate this, copy-paste augmentation and GAN-based synthesis (e.g., Perceptual GAN [28]) have been employed to simulate small targets under various sea conditions. Kisantal et al. [29] proposed scale-aware data augmentation strategies to increase the presence and visibility of small objects in training samples, which is especially effective in domains like aerial surveillance over ports or coastlines.

Anchor-based detectors typically require high overlap between predicted and ground truth boxes a condition rarely met for small, fast-moving maritime targets. Anchor-free methods such as FCOS [30] and CenterNet [31] provide improved localization by directly regressing from feature points, increasing sensitivity to small-scale objects.

In time-critical USV applications, real-time inference is crucial. Lightweight variants of YOLO (e.g., YOLOv5s [3], YOLOv8n [32]) and specialized detectors like FFEDet [33] and KDSMALL [34] offer an optimal trade-off between speed and accuracy. These models are often paired with knowledge distillation to transfer knowledge from large networks to smaller ones, enabling deployment on embedded systems like.

1.2.2 Ship classification

Having the ability to accurately distinguish different types of ships is critical to the success of situational awareness and decision support systems in military and civilian applications. In this context, large-scale datasets have been created that cover a variety of imaging conditions. MARVEL [2] is the largest and most comprehensive dataset ever created for ship classification tasks. Various studies have trained various models using this dataset and tried to solve the problem of classification of ships. For example, the creators of MARVEL used AlexNet [35] for ship classification and achieved 73.14% accuracy on 26 classes, which is an improvement over traditional methods such as Support Vector Machines [36].

Further advancements were achieved using transfer learning and optimized CNN models. Studies using EfficientNet [37] and MobileNetV2 [38] demonstrated significant improvements in accuracy, with EfficientNet B5 achieving a classification accuracy of 91.60%. Other research incorporated pre-trained models such as

ResNet-152 [39] and InceptionV3 [40] to optimize performance and reduce resource consumption, enabling faster training and deployment.

The integration of advanced CNN architectures has proven effective in overcoming the challenges of class imbalance and limited data availability. These models leverage weights pretrained in large scale datasets such as ImageNet, enabling efficient adaptation to maritime-specific tasks. With classification accuracies exceeding 90%, these methods mark substantial progress in the field.



Table 1.1 : Summary of maritime datasets with the number of classes, class names, and total number of images or frames.

Dataset Name	Challenge	No. of Classes	Class Names	No. of Frames
ABOShips [41]	Detection	11	boat, cargoship, cruiseship, ferry, militaryship, miscboat, miscellaneous, motorboat, passengershship, sailboat, sea-mark	9880
MoDS [1]	Detection Segmentation	4	ship, person, other, negative	8175
McShip [12]	Detection Classification	13	aircraft carrier, auxiliary ship, container ship, destroyer, fishing boat, landing ship, missile boat, passenger ship, sailboat, speed boat, submarines, support ship, tugboat	14709
LaRS [9]	Detection Semantic Segmentation Instance Segmentation	8	boat, swimmer, buoy, animal, paddle board, raw boat, float, other	4006
SeaShips [10]	Detection	6	bulk cargo carrier, container ship, fishing boat, general cargo ship, ore carrier, passenger ship	31455
SMD [11]	Detection Tracking	10	ferry, buoy, ship, speed boat, boat, kayak, sail boat, swimming person, flying bird/plane, other	1196 (on-board) 16,254 (on-shore)
MARVEL [2]	Verification Retrieval Recognition	26	(1) tankers, (2) carrier/floating (timber carrier, bulk carrier etc.), (3) supply vessels (supply vessel, tug etc.), (4) fishing vessels (trawler, fishing vessel etc.), and (5) dredgers (suction dredger, hopper dredger etc.)	1,607,190
VAIS [13]	Detection Classification	6	merchant, sailing, passenger, medium, tug, and small ships	2865 (1623 visible, 1242 IR, including 154 night IR)

1.3 Hypothesis

This study primarily aims to assess and enhance the performance of deep learning-based models in real-time obstacle detection and ship classification tasks for Unmanned Surface Vehicles (USVs). In this direction, the following contributions have been achieved.

- We show that the TPH-YOLOv5 [4] model improves detection performance, particularly in identifying small and dynamic maritime objects such as buoys, humans, and small vessels, compared to the standard YOLOv5 [3].
- We demonstrate that Vision Transformer (ViT) [5]-based models outperform traditional CNNs in terms of classification accuracy for large-scale ship classification. As far as we are aware, this is the first work employing ViTs for ship classification, and with this approach, we achieved superior performance compared to the existing state-of-the-art.
- We show the effects of both structured and unstructured pruning techniques on inference latency of transformer-based models. Our results shows that notable reductions in inference time can be achieved while avoiding excessive degradation of classification accuracy.
- We utilized the attention layers within these models to generate attention flow maps, a specialized type of heat map designed to visualize the areas focused by the transformers. This study proposes a general method for applying attention maps to different versions of the ViT model. This method provides valuable insights on the decision-making process of ViTs.

2. BACKGROUND

2.1 YOLOv5

YOLO is a family of computer vision models developed to perform real-time object detection within images and videos, simultaneously providing object classifications and corresponding bounding box coordinates [42]. Among these models, YOLOv5 [3] is particularly notable for its GPU-accelerated optimization, enabling high-speed object detection with high accuracy.

As shown in Figure 2.1, the YOLOv5 architecture includes three core components: the backbone, the neck, and the head—which collaboratively complete the object detection process. The image first passes through the input layer and is processed by the backbone to extract multi-scale feature maps. These features are then fused in the Neck, generating feature maps—P3 (80×80), P4 (40×40), and P5 (20×20)—each responsible for detecting small, medium, and large objects, respectively. The feature maps are forwarded to the Head, to generate multi-dimensional array (BBboxes) containing class labels, confidence scores, and box coordinates. Finally, thresholding and non-maximum suppression (NMS) are applied to filter results and output the final detections.

- The Backbone is a CNN that responsible for extracting low-level and mid-level image features.
- The Neck processes these features further, performing multiscale feature fusion and enhancement to generate high-level representations.
- The Head is the last stage responsible for assigning object classes and predicting bounding box coordinates using the predefined set of object categories.

Due to its efficiency and performance, YOLOv5 is extensively used in various object detection tasks. The model is available in four variants—YOLOv5s (small),

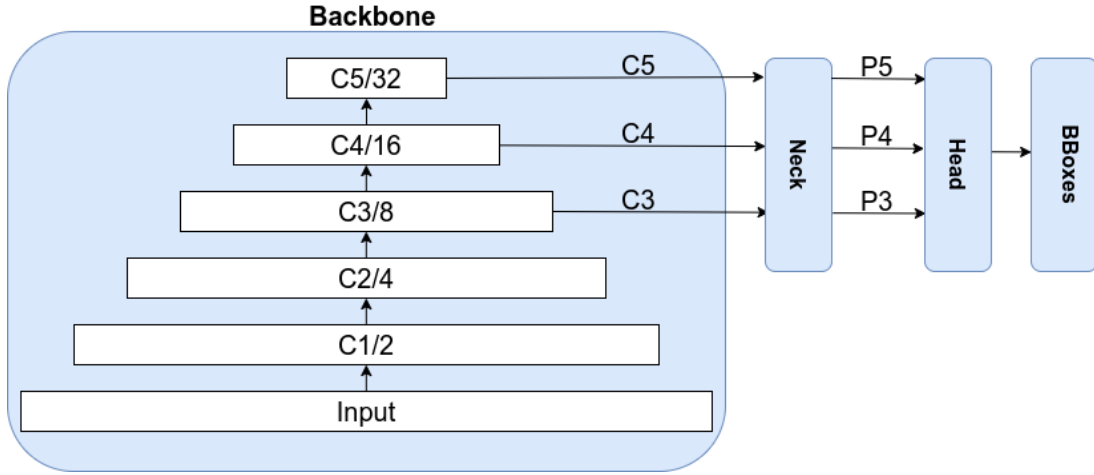


Figure 2.1 : The architecture and components of YOLOv5 [3] model.

YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra-large). Each configuration presents a compromise between accuracy and computational efficiency, with larger models achieving higher accuracy at the expense of increased training time and resource usage.

2.2 TPH-YOLOv5

To address the need for object detectors that can handle small-scale, densely clustered, and visually ambiguous targets, conditions typical in USV or UAV imagery, TPH-YOLOv5 [4] has been introduced as an improved variant of the YOLOv5 [3] architecture, specifically enhanced for detection tasks in drone-captured scenarios. The overall architecture of TPH-YOLOv5, as depicted in Figure 2.2, highlights its modular enhancements over standard YOLOv5. It has a CSPDarknet53 [43] backbone integrated with three transformer encoder blocks positioned at its end, a Neck that adopts a PANet-like [44] structure for enhanced feature aggregation and four Transformer Prediction Heads (TPHs) that utilize feature maps derived from the transformer encoder blocks within the Neck. Additionally, the quantity of each block is indicated by orange numerals displayed to the left of the respective blocks. By combining a CSPDarknet53 backbone with integrated transformer encoder blocks and a PANet-based neck, the model captures both low and high level features effectively.

The four transformer-based prediction heads, each specialized for different object scales, allow for more accurate detection across varied object sizes.

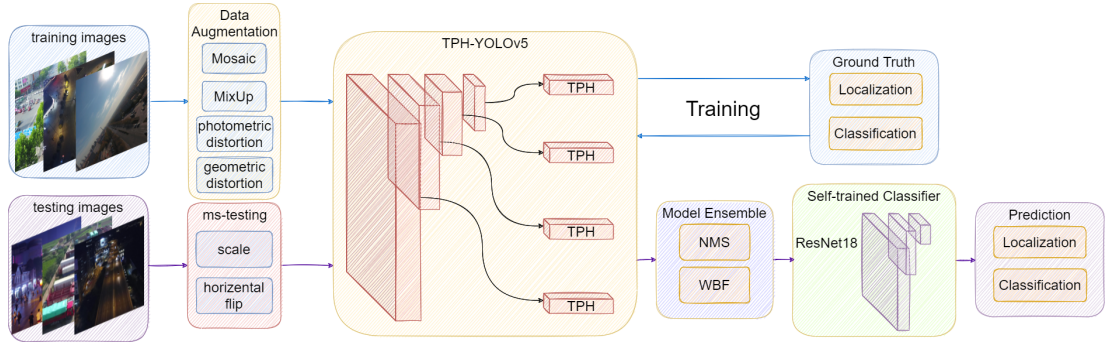


Figure 2.2 : The architecture of TPH-YOLOv5 model. Taken from [4].

The foundation of TPH-YOLOv5 lies in the original YOLOv5 framework, which employs CSPDarknet53 [43] as the feature extractor (backbone) and a PANet [44] structure to aggregate multi-scale features (neck). However, several significant upgrades have been integrated into the detector's head to enhance its ability to handle complex data.

1. **Expanded Detection Heads:** While YOLOv5 typically includes three output heads to detect large, medium and small objects, TPH-YOLOv5 introduces a fourth head dedicated to very small objects.
2. **Transformer-Based Heads:** Traditional convolution-based heads are replaced with Transformer encoder blocks that incorporate self-attention mechanisms. These blocks can capture long-range spatial dependencies and contextual information.
3. **CBAM:** An attention module is integrated into the network to enhance its focus on salient regions. Through the application of both channel and spatial attention, the model achieves better discrimination between targets and background clutter.

TPH-YOLOv5 improves on YOLOv5 by introducing several key enhancements. These modifications lead to substantial accuracy gains, making TPH-YOLOv5 more effective for the detection of small objects under challenging conditions.

2.3 ViT (Vision Transformers)

For many years, CNNs have been our go-to solution in the field of image classification. Architectures based on CNN principles have helped us effectively tackle various computer vision problems. Transformers, particularly Vision Transformers (ViTs) [5], have recently been introduced in deep learning, employing self-attention mechanisms to efficiently model global dependencies. The concept of Transformers was initially developed for Natural Language Processing (NLP) tasks. The landmark paper "Attention Is All You Need" [45] introduced groundbreaking solutions to NLP challenges, and Transformer-based architectures have since become the standard throughout numerous NLP applications. It was not long before researchers adapted this architecture to the visual domain. A pivotal work in this transition is the paper "An Image is Worth 16x16 Words" [5]. ViTs have demonstrated strong performance in various classification tasks, particularly after being trained on large-scale datasets, making them a competitive alternative to CNNs with respect to both accuracy and scalability.

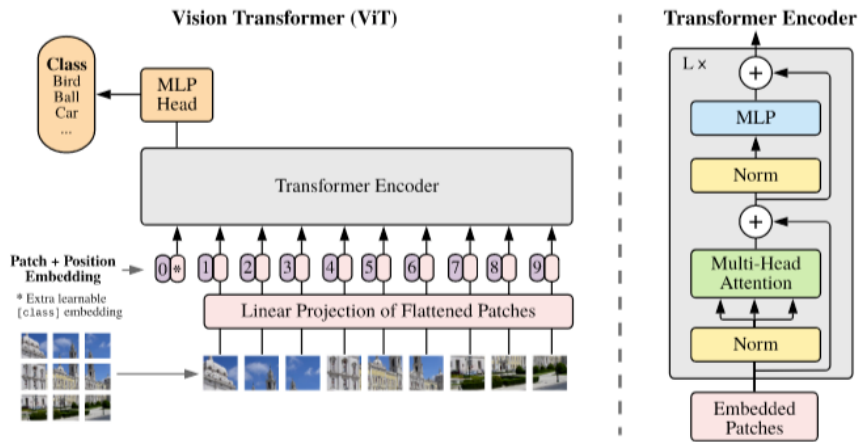


Figure 2.3 : Schematic representation of the Vision Transformer (ViT) pipeline.
Taken from [5].

Figure 2.3 shows the core idea of ViTs which involves dividing an image into fixed-size patches (typically 16×16), and then extracting embeddings from these patches, which are flattened and linearly projected into embedding vectors. A learnable classification token is attached to the sequence, and positional encodings are added to retain spatial

ordering. This sequence is processed through multiple layers of a Transformer encoder, which includes multi-head self-attention and feed-forward networks with residual connections and normalization. The final classification is obtained from the output corresponding to the class token via a multilayer perceptron (MLP) head. Before diving into the underlying mechanics, it is essential to introduce some foundational concepts to ensure a comprehensive understanding. The standard ViT architecture consists of the following key components:

1. **Patch Embeddings** Let the input image be $x \in \mathbb{R}^{H \times W \times C}$, where H , W , and C denote height, width and number of channels, respectively. The image is partitioned into non-overlapping patches of size $P \times P$, each of which is flattened and linearly projected into a fixed-dimensional vector. This reduces the computational cost of the self-attention mechanism, which scales quadratically with the number of input tokens.
2. **Positional Embeddings** Since transformers are invariant to the input order, positional embeddings are added to the patch embeddings to retain spatial information. These are learnable vectors with the same dimensionality as the patch embeddings and are optimized during training. They allow the model to distinguish between patches based on their position in the original image.
3. **Transformer Encoding Details** A stack of transformer encoder blocks forms the core of ViT. Each transformer block applies the following components in sequence:
 - **Multi-Head Self-Attention (MSA):** Captures relationships between all patches, enabling the learning of both local and global dependencies.
 - **MLP Block:** A two-layer feed-forward neural network with GELU [46] activation.
 - **Layer Normalization:** Applied before Multi Head Self Attention (MSA) and Multi Layer Perceptron (MLP) layers for training stability.
 - **Residual Connections:** Enable effective backpropagation and support deep network training.

4. **Classification Head** For image classification tasks, a classification head is added to the top of the architecture. It typically consists of a hidden layer followed by a linear output layer. During pre-training, this head enables the model to learn generic visual representations, while during fine-tuning, it adapts to specific downstream tasks. The lower layers of ViT tend to capture both local and global patterns, whereas the upper layers focus on more abstract and global features.

2.4 DeiT (Data-efficient Image Transformer)

DeiT [6] is a ViT model developed for visual recognition tasks. Transformer-based models are known to be effective in tasks such as image classification, but usually require large datasets and intensive computational resources. DeiT reduces these requirements and offers an efficient alternative that can be trained only with a more modestly sized dataset such as ImageNet.

As in ViT, the input image is divided into fixed-sized patches (e.g. 16x16 pixel blocks), each patch is converted into a set of vectors, and these vectors are sent to the transformer layers. However, DeiT's difference comes with two key innovations that increase efficiency:

1. **Efficient Training:** DeiT can be trained on a single machine in 2-3 days with only ImageNet data. Thus, the need for massive datasets and infrastructure is eliminated.
2. **Distillation Token:** DeiT uses a "teacher-student" learning technique during training. This method added to ViT is called a "distillation token". This token is included in the transformer layers along with the "class token" which performs class prediction and aims to learn the output of the teacher model. This strategy provides better generalization of the transformer model, especially when a CNN is used as a teacher.

DeiT competes with state-of-the-art CNN architectures such as EfficientNet in terms of performance; even the distillation version gives better results in many cases. In addition, the fact that it can be trained with much fewer resources makes it attractive for academic and practical applications.

2.5 Swin Transformer (Shifted Windows Transformer)

Swin Transformer [7] is a Vision Transformer architecture developed for use in visual classification tasks. Its design mutation was to solve the efficiency problems experienced by ViTs in high-resolution images. In addition, it has been made suitable for use in tasks such as object detection and semantic segmentation, apart from the purpose of image classification.

In this architecture, the input image is initially divided into fixed-size patches (e.g. 4×4 pixel blocks). These patches are represented as 'tokens' and self-attention operations are performed on these tokens. However, unlike classical Transformers, Swin implements self-attention in small, local windows, and these windows are shifted between layers.

Thanks to the local window-based self-attention design, i.e. since self-attention is only performed in small windows, the complexity is linearly proportional to the image size. In addition, thanks to sliding windowing, it provides connectivity between different windows and supports the model to learn the global context. Swin Transformers provide easy integration in intensive tasks because feature maps are generated at different resolutions. This sliding window approach allows the model to combine local contexts to create more robust representations.

2.6 Attention Rollout

The heatmap is a useful tool for visualizing the regions within an image that influence the most a model's predictions. By highlighting these areas, heatmaps give insight into how machine learning models process and interpret visual data. Typically applied in image classification and text analysis, they utilize a color scale where cooler tones, e.g. blue, indicate low attention, while warmer tones, e.g. red, indicate high-influence regions. Such visualizations improve model transparency, making AI models easier to understand, helping to identify biases or errors, and improving trust by showing how the model processes inputs.

Attention rollout is a method used to understand and visualize how information is transferred between layers in transformer models. Its main purpose is to provide a

tracking method that reflects the effect of input tokens on the model more accurately, instead of raw attention values that become uninterpretable, especially in deep layers.

$$\text{Let } A^{(l)} = \text{attention matrix for layer } l, \quad (2.1)$$

$$\hat{A}^{(l)} = \max(A^{(l)}) \quad (\text{Fused Attention}) \quad (2.2)$$

$$A^{(l)} = \frac{\hat{A}^{(l)} + I}{\sum(\hat{A}^{(l)} + I)} \quad (\text{Normalized}) \quad (2.3)$$

The implementation involved adjusting attention matrices to filter out less informative attention scores based on a predefined discard threshold. Among the three available options for attention head aggregation—maximum, minimum, and mean—we chose for a maximum fusion strategy (equation 2.2) to emphasize the most significant regions highlighted by any single attention head. This was further enhanced by discarding lower attention values, which improved the clarity of attention maps by emphasizing the most activated features. Additionally, residual connections were unified by adding an identity matrix to each layer’s attention matrix, as shown in equation 2.3.

$$\tilde{A}^{(l)} = \begin{cases} A^{(l)} \cdot \tilde{A}^{(l-1)} & \text{if } l > 0 \\ A^{(0)} & \text{if } l = 0 \end{cases} \quad (2.4)$$

To accumulate attention across layers, the final attention map was derived by multiplying the refined attention matrices, as shown in equation 2.4, followed by normalization to ensure a balanced attention distribution across the image. If $A^{(l)}$ denotes the attention matrix in the l th layer, the rollout matrix up to the l th layer is defined as $\tilde{A}^{(l)}$.

2.7 Pruning

Pruning is a widely used model compression technique in deep learning that reduces the size and computational complexity of neural networks by eliminating redundant

parameters. This process helps optimize storage and computational efficiency while preserving model accuracy. By removing low-impact weights and neurons, typically those with minimal magnitude or redundant activations, pruning results in a more lightweight model with improved generalization, making it well suited for deployment on resource-constrained devices, such as embedded systems.

Pruning Strategies in Neural Networks

There exist various pruning techniques, each focusing on different structural components of a neural network. The most commonly used methods include unstructured pruning, structured pruning, and dynamic pruning.

- **Unstructured Pruning:** This method removes individual connections within the network. A widely used approach is magnitude-based pruning, where weights with the smallest magnitudes are eliminated, under the assumption that their impact on model performance is minimal.
- **Structured Pruning:** Unlike weight pruning, structured pruning removes entire groups of parameters, such as convolutional filters, neurons or layers to make model more compatible. This includes filter pruning, which eliminates less impactful convolutional filters in CNNs, and neuron pruning, which discards redundant neurons in fully connected layers. Layer pruning is a more aggressive approach that removes entire layers if they contribute negligibly to performance.
- **Dynamic Pruning:** This adaptive pruning technique takes place during training, allowing the network to iteratively adjust and identify important connections while learning. This results in a more optimized structure without requiring post-training modifications.



3. METHODS

This section focuses on our deep learning framework aimed at solving object detection and classification problems, particularly for USVs in maritime settings. It offers a detailed walkthrough of the modifications applied to the datasets, the preprocessing steps for images, and the transfer learning strategies used to optimize model outcomes. It also discusses the rationale for choosing TPH-YOLOv5 [4] for object detection, and highlights the strengths of this specialized YOLO-based [42] model in providing a balance between accuracy, speed, and expertise in detecting small objects. The chapter then focuses on the methodology adopted for the classification problem. It discusses the use of image transformer-based models, and highlights their ability to process images and capture global context through self-attention mechanisms. It also provides information on the steps taken to compare transformer-based models with traditional CNNs in terms of accuracy, efficiency, and suitability for real-time applications.

3.1 Datasets

This study uses two datasets, MODS [1] and MARVEL [2], to address the challenges of obstacle detection and classification of ships, respectively, in maritime environments. These datasets are specifically designed to reflect real-world conditions by incorporating various environmental factors like changing weather, lighting variations, and complex scenes that involve multiple types of objects. They are comprehensive benchmarks for evaluating computer vision models for maritime surveillance systems.

3.1.1 MODS dataset

The Maritime Obstacle Detection System (MODS) [1] dataset is tailored for Unmanned Surface Vehicles (USVs) and focuses on real-world scenarios involving obstacle detection and segmentation. Approximately 81,000 stereo images, paired



Figure 3.1 : Example images from the MODS dataset representing various environmental conditions. (a) a harbor scene with partially distorted visibility due to sun reflection, (b) a wavy sea surface close to the shore, (c) a foamy water trail created by a moving boat, (d) a small boat moving on a calm water surface in the open sea, (e) sailboats positioned in different directions, (f) a harbor scene with buoys and the sun directly reflected on the camera near sunset.

with data from an on-board Inertial Measurement Unit (IMU), are included in this dataset. The annotations include dynamic obstacles such as ships, swimmers, and buoys, as well as static obstacles like shorelines and piers, marked by water-edge boundaries. Image acquisition took place under diverse environmental settings, such as different weather patterns, varying levels of sunlight, and complex maritime features such as sea foam and reflections on water as in Figure 3.1. This diversity ensures that MODS-trained models can handle a wide range of real-world scenarios, including small and challenging-to-detect obstacles. MODS supports both object detection and segmentation tasks, making it an essential dataset for training models aimed at improving the safety and navigational capabilities of USVs.

Key Characteristics of the MODS Dataset:

- **Stereo imagery:** Facilitates depth perception and obstacle segmentation.
- **Annotated classes:** Covers dynamic (e.g., vessels, buoys) and static (e.g., piers, shorelines) obstacles.

- **Evaluation protocol:** Designed to measure performance in real world USV navigation scenarios, emphasizing detection accuracy which is critical for collision avoidance.

The rich annotations of the MODS dataset enable the development and evaluation of both object detection and segmentation algorithms, making it a benchmark in maritime obstacle detection research.

3.1.2 MARVEL dataset

The MARVEL [2] dataset is a large-scale resource created for fine-grained ship classification, containing approximately 2 million images of maritime vessels. It categorizes ships into five main types—cargo, military, carriers, cruise ships, and tankers—which are further divided into 26 specific subtypes. The images were collected from public sources and are annotated with metadata such as vessel type, year of build, and International Maritime Organization (IMO) number. The vast size and variety of MARVEL position it as an ideal resource for developing deep learning models based on both CNNs and ViTs [5]. The dataset includes extensive variations of viewing angles, lighting conditions, and vessel appearances, enabling the models to generalize effectively in real-world applications. MARVEL is widely regarded as a critical resource for advancing the accuracy of ship classification models and improving maritime safety, surveillance, and vessel identification systems.

Key attributes of the MARVEL dataset:

- **Class diversity:** 26 superclasses representing a wide variety of vessel types, including cargo ships, military vessels, and passenger ships.
- **Imaging conditions:** Includes visual data recorded in different lighting and weather environments, ensuring generalization across different maritime environments.
- **Applications:** Supports tasks such as vessel verification, retrieval, and recognition in addition to classification.

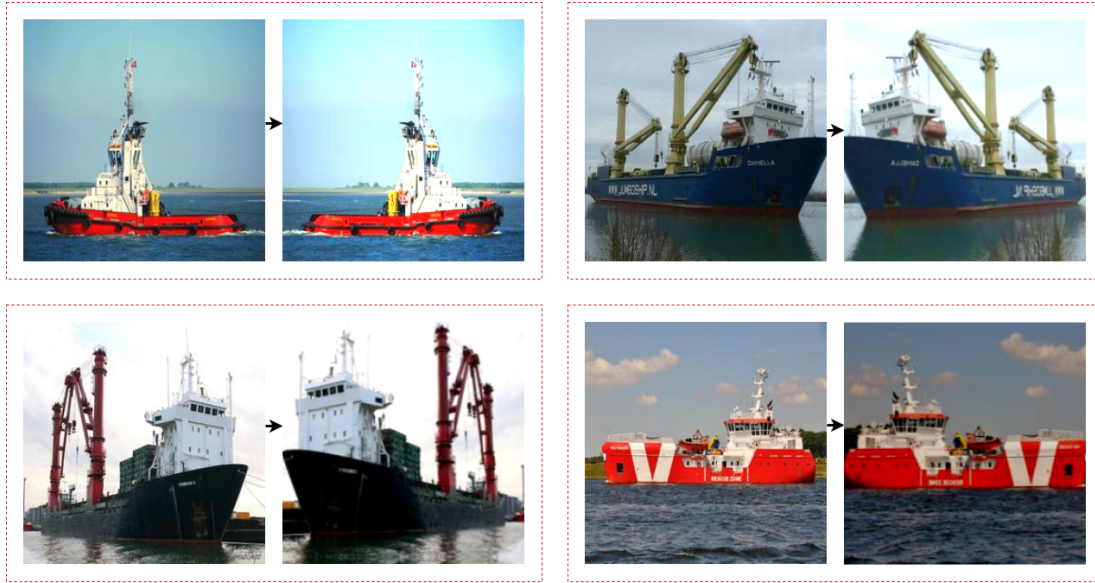


Figure 3.2 : Examples of raw and preprocessed images from the MARVEL dataset.

Original images as collected from the dataset and the corresponding images after preprocessing, including operations such as horizontal flipping, translation, brightness and contrast adjustments, and scaling.

3.2 Data Preprocessing and Adaptation

Datasets underwent necessary preprocessing steps to prepare them for the specific models and tasks in this study:

3.2.1 Adaption of MODS

Since the original annotations of MODS were not in a YOLO-compatible format, they were converted accordingly to support the YOLOv5-based models used for object detection. The dataset was then separated into three subsets: 80% for training, 10% for validation, and 10% for testing, resulting in total approximately 9000 images for experimentation. This ensured that the model could be thoroughly trained and validated across diverse obstacle types and conditions.

3.2.2 Adaption of MARVEL

To enable a fair comparison with the prior state-of-the-art approach by Salem et al. [47] conducted on the MARVEL dataset, we aimed to repeat a similar experimental setup. Specifically, we had a dataset of 3600 training images, 800 validation images, and 800 test images per class to maintain consistency. However, some classes were found

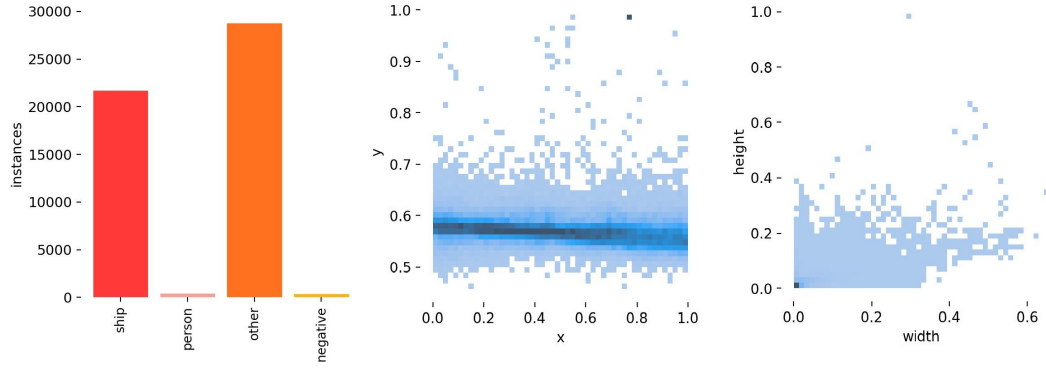


Figure 3.3 : Post-training analysis from YOLOv5 for MODS. (a) The bar chart shows the distribution of object classes in the dataset. (b) The scatter plot represents object positions in the image space. (c) The scatter plot visualizes the width-height distribution of detected objects.

to have an insufficient number of samples. To resolve this, various data augmentation operations such as horizontal flipping, translation, brightness and contrast adjustments, and scaling were applied until the desired sample size was achieved. This approach was crucial for achieving reliable performance, particularly for less common vessel classes. Figure 3.2 shows some example images before and after applying preprocessing steps. For classes with an excessive number of samples, a random selection was made to match the intended number of samples.

By employing these datasets with appropriate preprocessing and balanced splits, this study ensures the development and evaluation of high-performing models for both maritime obstacle detection and ship classification tasks.

3.3 Training and fine-tuning

In machine learning, a model initially trained for one task can be reused for a different task through various transfer learning strategies. A pre-trained model can either be used for feature extraction or refined through fine-tuning to better adapt to the new problem. Fine-tuning enables the model to leverage the weights learned from a previous task, facilitating faster and more effective adaptation to similar problems. However, training deep neural networks typically demands large datasets, and when the available dataset is small, the model may experience overfitting or memorization issues. Given these factors, choosing the appropriate model and dataset becomes

critical for real-time applications like maritime navigation, where fast and dependable models are essential.

3.3.1 Object detection

One primary objective of this work is to build a reliable object detection framework for identifying ships, humans, and other obstacles in maritime environments. Given the dynamic and unpredictable nature, rapid and accurate detection is essential for the safe operation of USVs.

In this study, object detection was chosen over segmentation due to its efficiency in real-time applications, particularly for USVs operating in dynamic maritime environments. While segmentation provides pixel-level classification, making it useful for tasks requiring detailed shape and boundary recognition, it is computationally expensive and often unsuitable for time-sensitive navigation systems. USVs require rapid decision-making to avoid obstacles and ensure safe navigation, making detection a more practical approach as object detection models provide faster inference speeds and lower computational costs. Additionally, segmentation models often struggle with water reflections, waves, and lighting variations, which can introduce noise and reduce accuracy.

When analyzing object sizes in the MODS dataset, we observe that most objects are relatively small, making detection challenging. This YOLOv5 post-training analysis on Figure 3.3 highlights key dataset characteristics. The bar chart shows class imbalance, with ships and other objects dominating, while persons and negative samples are sparse. The scatter plots reveal that objects are clustering in a narrow horizontal band. The width-height distribution confirms that most objects occupy a minor portion of the image, emphasizing the challenge of small object detection. Additionally, the clustering of points near the (0,0) coordinate, visible as a dense dark blue region, further indicates a concentration of very small objects in the dataset. The Figure 3.4 illustrates the annotations present in the dataset, providing insight into the distribution of small objects across the images. To address this issue, our primary focus has been on small object detection. This is where TPH-YOLOv5 comes into



Figure 3.4 : Sample images from the MODS dataset, highlighting annotations for small-sized objects.

play which is a modified version based on the standard YOLOv5, aimed at improving the detection of small objects.

TPH-YOLOv5 strengthens the ability to identify small objects, a crucial factor in maritime navigation where precise recognition of buoys, swimmers, and similar obstacles is vital. By integrating Transformer Prediction Heads (TPH), this model leverages attention mechanisms to enhance feature extraction, improving detection performance in cluttered and complex environments.

This study presents a comparative analysis between YOLOv5 and TPH-YOLOv5 as the core detection models. We conducted experiments on both to evaluate whether TPH integration effectively enhances the detection of small objects.

Both models were initialized with pre-trained ImageNet weights to leverage transfer learning, facilitating faster convergence and improved detection performance. The training process involved 130 iterations of fine-tuning using the MODS dataset. The dataset was preprocessed to align with the YOLO framework, ensuring compatibility and efficient training.

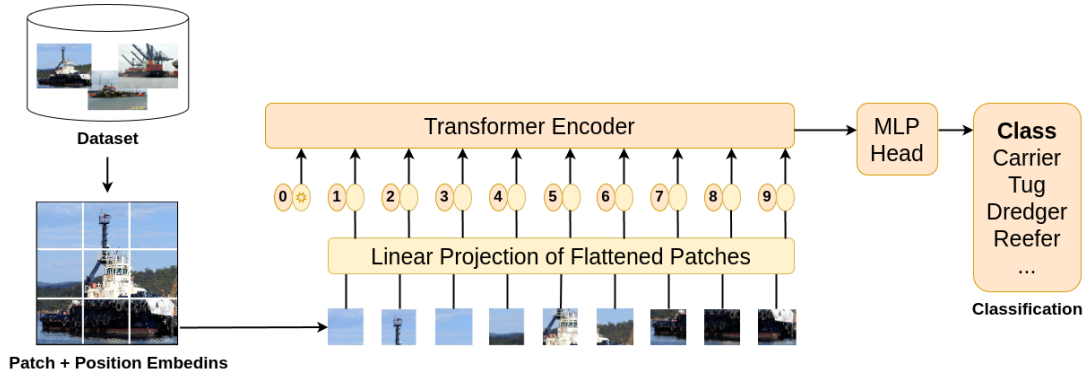


Figure 3.5 : Graphical representation of training process of the vision transformers on MARVEL dataset.

3.3.2 Ship classification

In this research, we examine the effectiveness of Vision Transformers-based models for ship classification by experimenting with ViT [5], DeiT [6], Swin Transformers [7], and ConvNext-v2 [8] on the MARVEL dataset. To establish a fair comparison and assess whether transformer-based models provide solid improvements, we also evaluated CNN-based models, specifically ResNet50 and ResNet101. Given the extensive size of the MARVEL dataset, previous studies have relied on random subsets for evaluation, making direct model comparisons challenging. To maintain consistency, we adopted a similar approach, using a controlled subset while applying data preprocessing and augmentation techniques to mitigate class imbalance. Additionally, convolutional models were fine-tuned to ensure a balanced comparison with transformer-based models.

Figure 3.5 illustrates the training pipeline of Vision Transformers. Through our experiments, we systematically compared CNN-based and transformer-based architectures, analyzing their accuracy and computational efficiency. The results provide an understanding of the strengths and weaknesses of each method in ship classification tasks within dynamic maritime environments.

3.4 Attention Rollout Extraction

In this study, heatmaps are generated from the attention mechanisms of transformers. This approach helps determine whether models are correctly focusing on key

characteristics, such as a ship’s shape, while discarding irrelevant background elements like water reflections. Unlike prior work [48], which primarily tailored heatmaps for standard ViT architectures, our study adapts this methodology to be applied across diverse transformer-based models, including DeiT and Swin Transformers.

To gain deeper insights into the focus areas of vision transformer models, we utilized attention layers to produce rollout maps, a specialized form of heatmap. These maps highlight the portions of an image that receive the most attention during classification. Through the examination of these visualizations, we are able to more clearly interpret the model’s decision-making behavior and its emphasis on various parts of the image.

For models DeiT, which utilize a classification token (CLS token), we extracted this token from all attention layers before merging them, as described in equation 3.1. Here, the variable k represents the number of CLS tokens, and this operation ensures that we select all elements from the last two dimensions starting from the index k onwards, effectively skipping or ignoring the first k entries in both dimensions.

$$\tilde{A}_i = A_i[\dots, k :, k :] \quad (3.1)$$

In the Swin Transformers, we addressed varying window sizes by averaging across the batch dimension, as indicated in equation 3.2. This step involves computing the mean attention across all samples in the batch (dimension 0), effectively reducing window size while maintaining interpretability.

$$\tilde{A}_i = \text{mean}(A_i, \text{dim} = 0, \text{keepdim} = \text{True}) \quad (3.2)$$

This refined approach not only builds on the findings from [48] but also extends them. By making attention visualization more adaptable across architectures, this method ensures better transparency and understanding of how ViTs and their derivatives operate in different vision tasks.

3.5 Pruning

Our analysis revealed that Vision Transformers (ViTs) struggle for real-time use because of their limited frame rates (FPS). To address this, we explored different pruning techniques to boost their efficiency. Among the models evaluated, DeiT achieved the highest accuracy, making it the primary focus of our pruning experiments. We applied multiple post-training pruning strategies to assess their impact on the performance and accuracy of ship classification systems.

In this work, we examined both local and global pruning strategies, employing unstructured pruning and L2 structured pruning techniques. Each pruning strategy was tested at two different sparsity levels (20% and 30%), and the corresponding pruning ratios were accordingly adjusted and applied. Local pruning targeted specific model components, including the classification layer, the initial attention layer, and all attention layers, whereas global pruning was applied throughout the model to obtain a greater reduction in complexity. L2 structured pruning focused on removing entire structural elements such as neurons or filters based on their L2 norm.

Although DeiT demonstrated the highest accuracy compared to the other tested models, its low frame rate made it inappropriate for real-time deployment. To mitigate this issue, pruning techniques were employed to reduce computational demands while preserving classification performance and improving inference speed.

4. EXPERIMENTAL RESULTS

In this section, the performance values obtained as a result of the experiments carried out with our experimental setups and models are explained. Detailed information on the data sets and experimental setups is given in the following sections.

4.1 Evaluation Metrics

For the evaluation of our models' performance in object detection and classification, we utilize standard evaluation metrics. These metrics provide insights into the accuracy, precision, and efficiency of our models, helping to determine their suitability for real-world applications.

4.1.1 F1 score

The F1 score is used as a metric to deliver a combined measure of precision and recall, which is crucial when handling imbalanced datasets where one class dominates over another. It is calculated as follows:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.1)$$

Precision reflects the model's ability to correctly identify true positives among all positive predictions:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (4.2)$$

Recall measures how many actual positive cases were correctly identified:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (4.3)$$

The F1 Score varies between 0 and 1, with 1 representing perfect precision and recall, and 0 indicating a complete failure to make correct predictions.

4.1.2 mAP (Mean Average Precision)

The mAP is a widely used metric in object detection, measuring how well the model identifies and localizes objects. It considers both precision and recall at various confidence levels and Intersection over Union (IoU) thresholds.

IoU measures the degree of overlap between the predicted and actual bounding boxes. It is determined by the following formula:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (4.4)$$

Where the intersection is the area where the predicted and actual bounding boxes overlap, and the union is defined as the total region spanned by both bounding boxes.

An IoU score of 1 indicates perfect overlap, while 0 means no overlap. Different IoU thresholds are used to determine whether a prediction is considered a TP (True Positive) or FP (False Positive).

- **mAP@0.5:** Uses a threshold of 0.5, meaning that a predicted box must overlap at least 50% with the ground truth box to be counted as a correct detection.
- **mAP@[0.5:0.95]:** Calculates the mean AP evaluated at IoU thresholds between 0.5 and 0.95, with increments of 0.05, providing a broader view of model effectiveness.

The **mAP** calculation involves the following steps:

1. **Precision-Recall Curve:** A curve is generated by plotting precision against recall at different confidence thresholds.
2. **Average Precision (AP):** The area under the precision-recall curve is computed for each class. AP for a single class is computed as:

$$AP = \int_0^1 P(R) dR \quad (4.5)$$

3. **Mean Average Precision (mAP):** The AP scores of all classes are averaged.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4.6)$$

where AP_i is the AP score of the i -th class and N is the number of object classes. A higher mAP value indicates better model performance in detecting and localizing objects with high accuracy while minimizing false positives.

4.1.3 Accuracy

Accuracy is a basic yet important metric for evaluating classification models. It measures the proportion of correct predictions relative to the total number of samples. It can be formulated as:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \quad (4.7)$$

Accuracy is a useful metric when classes are balanced, but in cases of class imbalance, additional metrics, like the F1 score, provide a more detailed performance analysis.

4.1.4 FPS (Frames Per Second)

FPS (Frames Per Second) is a key metric for evaluating the speed and real-time efficiency of object detection models. It represents how many frames (or images) the model can process per second. A higher FPS indicates faster inference speed, making the model more suitable for real-time applications.

FPS is calculated as:

$$FPS = \frac{\text{Total Processed Frames}}{\text{Total Processing Time (seconds)}} \quad (4.8)$$

For real-time applications, an FPS of 30 or higher is generally considered optimal, while values below 10 FPS may indicate slow inference unsuitable for real-time deployment.

These evaluation metrics provide a comprehensive understanding of model performance in terms of accuracy, precision, recall, and speed, helping to select the best model for the intended application.

4.2 Experimental Setup

In this study, a suitable infrastructure was created for model training and evaluation by applying preprocessing steps to the datasets. First, the original annotations of the MODS dataset were transformed to be compatible with YOLOv5-based object detection models. The dataset was then divided into 80% training, 10% validation and 10% test. In total, there are approximately 9,000 images were prepared for the experiments.

For the MARVEL dataset, an experimental setup similar to the study by Salem et al. [47] was targeted. It was aimed to create 3,600 training, 800 validation and 800 test images for each class. However, since there were not enough examples in some classes, the number of examples was increased by using data augmentation methods, such as horizontal flip, shift, brightness and contrast adjustment, and scaling (see Figure 3.2). For classes with a large number of examples, the number of examples were reduced by random sampling.

The detailed distribution showing the number of samples per class before and after data augmentation is presented in Table 4.1. Reviewing the table shows that some classes (e.g., *Bulk Carrier*, *Container Ship*, *Fishing Vessel*) have already met their initial sample targets, and thus, additional data augmentation is not required. However, there are some classes where the amount of original data is quite insufficient such as *Combat Vessel* (1730 training samples), *Icebreaker* (1482 training samples) and *Replenishment Vessel* (1351 training samples). In these classes, the number of samples was increased with data augmentation, reaching 3,600 training, 800 test and 800 validation images for each. Thus, the sample imbalance between classes was eliminated and the model was aimed to exhibit strong performance in minority classes.

In this way, by using balanced and appropriately preprocessed datasets, high-performance models have been developed and evaluated in obstacle detection and ship classification tasks in maritime environments.

Table 4.1 : MARVEL class distribution before (B) and after (A) applying augmentation to balance the number of samples per classes.

Class Name	Train/B	Train/A	Test/B	Test/A	Val/B	Val/A
Bulk Carrier	3600	3600	800	800	800	800
Combat Vessel	1730	3600	370	800	372	800
Container Ship	3600	3600	800	800	800	800
Dredger	3600	3600	786	800	788	800
Fire Fighting Vessel	2181	3600	467	800	468	800
Fishing Vessel	3600	3600	800	800	800	800
General Cargo	3600	3600	800	800	800	800
Heavy Load Carrier	2631	3600	563	800	565	800
Icebreaker	1482	3600	317	800	319	800
Livestock Carrier	2209	3600	473	800	474	800
Passengers Ship	3600	3600	800	800	800	800
Patrol Vessel	2119	3600	454	800	455	800
Platform	1944	3600	416	800	418	800
Reefer	3600	3600	800	800	800	800
Replenishment Vessel	1351	3600	289	800	290	800
Ro-ro Cargo	3600	3600	800	800	800	800
Ro-ro-passenger Ship	3600	3600	800	800	800	800
Sailing Vessel	3564	3600	763	800	765	800
Standby Safety Vessel	1757	3600	376	800	378	800
Supply Vessel	3600	3600	800	800	800	800
Tanker	3600	3600	800	800	800	800
Training Ship	1663	3600	356	800	358	800
Tug	3600	3600	800	800	800	800
Vehicles Carrier	3600	3600	800	800	800	800
Wood Chips Carrier	2225	3600	476	800	478	800
Yacht	3600	3600	800	800	800	800

4.3 Object Detection Results

This section proposes a comparable analysis of obstacle detection on MODS dataset using YOLOv5 and TPH-YOLOv5. The basic hyperparameters used for training the YOLOv5 models are summarized in Table 4.2. Optimization parameters such as initial learning rate (`lr0`), momentum coefficient (`momentum`) and weight decay coefficient (`weight_decay`) were determined during the training process. In order to provide a

more balanced progress at the beginning of training, the `warmup_epochs` value was selected as 2. This hyperparameter selection was chosen to improve both the training performance and the overall validation performance of the models.

Table 4.2 : Selected key hyperparameters for YOLOv5 training.

Parameter	Value	Description
<code>lr0</code>	0.0032	Initial learning rate
<code>momentum</code>	0.843	SGD momentum
<code>weight_decay</code>	0.00036	Weight decay coefficient
<code>warmup_epochs</code>	2.0	Warm-up epochs
<code>mosaic</code>	1.0	Mosaic augmentation probability
<code>mixup</code>	0.2	MixUp augmentation probability
<code>hsv_h/s/v</code>	0.4/0.3/0.5	HSV color augmentation factors
<code>scale</code>	0.4	Image scaling augmentation

Table 4.3 offers a comparison of three YOLOv5-based object detection models: YOLOv5-s, YOLOv5-l, and TPH-YOLOv5, using performance metrics including precision, recall, F1 score, and mean average precision (mAP), measured at two IoU thresholds. The results show that TPH-YOLOv5 achieves the highest F1 score (0.860) and mAP@0.5 (0.863). Compared to YOLOv5-l, which is a larger model in terms of parameters, TPH-YOLOv5 better performs on detection problem while maintaining similar precision (0.894 vs. 0.897) and improving recall (0.830 vs. 0.812). In contrast, YOLOv5-s, being the smallest and lightest model, obtains the lowest scores on all metrics. It achieves low recall and mAP@0.5:0.95 which indicates that it may struggle with more complex detection scenarios.

Table 4.3 : Detailed comparison of performance metrics for various YOLOv5 models on the MODS [1] dataset.

	Precision	Recall	F1	mAP@.5	mAP@.5:.95
YOLOv5-s [3]	0.845	0.745	0.791	0.793	0.403
YOLOv5-l [3]	0.897	0.812	0.852	0.844	0.487
TPH-YOLOv5 [4]	0.894	0.830	0.860	0.863	0.493

Table 4.4 explores the performance per class for three YOLOv5-based models evaluated on the MODS dataset. This detailed analysis shows how each model performs detecting specific object categories, namely, ships, people, and other objects commonly found in maritime environments. The noticeable improvement in scores

for the person and similar small-object classes suggests that the detection of smaller targets has become more effective. Looking at the results, it can be seen that all models achieve high precision and recall scores for the class of the ship, with YOLOv5-l slightly outperforming others in both recall (0.938) and mAP@.5 (0.95). However, for person class, which generally represents smaller and more difficult-to-detect objects, TPH-YOLOv5-s demonstrates a significant advantage with the highest mAP@.5 score of 0.852. Furthermore, for the other object class, TPH-YOLOv5-s again leads with a mAP@.5 of 0.666, surpassing both standard YOLOv5 variants. This consistency across smaller and less distinct object classes suggests that TPH-YOLOv5-s not only maintains competitive performance on larger objects like ships, but also is better at detecting smaller targets.

Table 4.4 : Class-specific performance metrics for YOLOv5 models on the MODS dataset.

	Pr / ship Re / ship mAP@.5 / ship	Pr / person Re / person mAP@.5 / person	Pr / other Re / other mAP@.5 / other
YOLOv5-s [3]	0.924	0.714	0.812
	0.923	0.625	0.484
	0.947	0.687	0.607
YOLOv5-l [3]	0.919	0.916	0.799
	0.938	0.818	0.545
	0.95	0.824	0.648
TPH-YOLOv5-s [4]	0.915	0.895	0.794
	0.920	0.818	0.609
	0.941	0.852	0.666

Figure 4.1 shows a side-by-side visual comparison of the detection results of three variants of the model in the MODS dataset, covering a variety of maritime conditions. Each row corresponds to a different model and illustrates its ability to detect and classify objects under different environmental scenarios. Among the models, TPH-YOLOv5 clearly performs better in detecting smaller objects. This capability is particularly valuable in complex maritime environments where tasks such as navigation and surveillance require high detection precision.

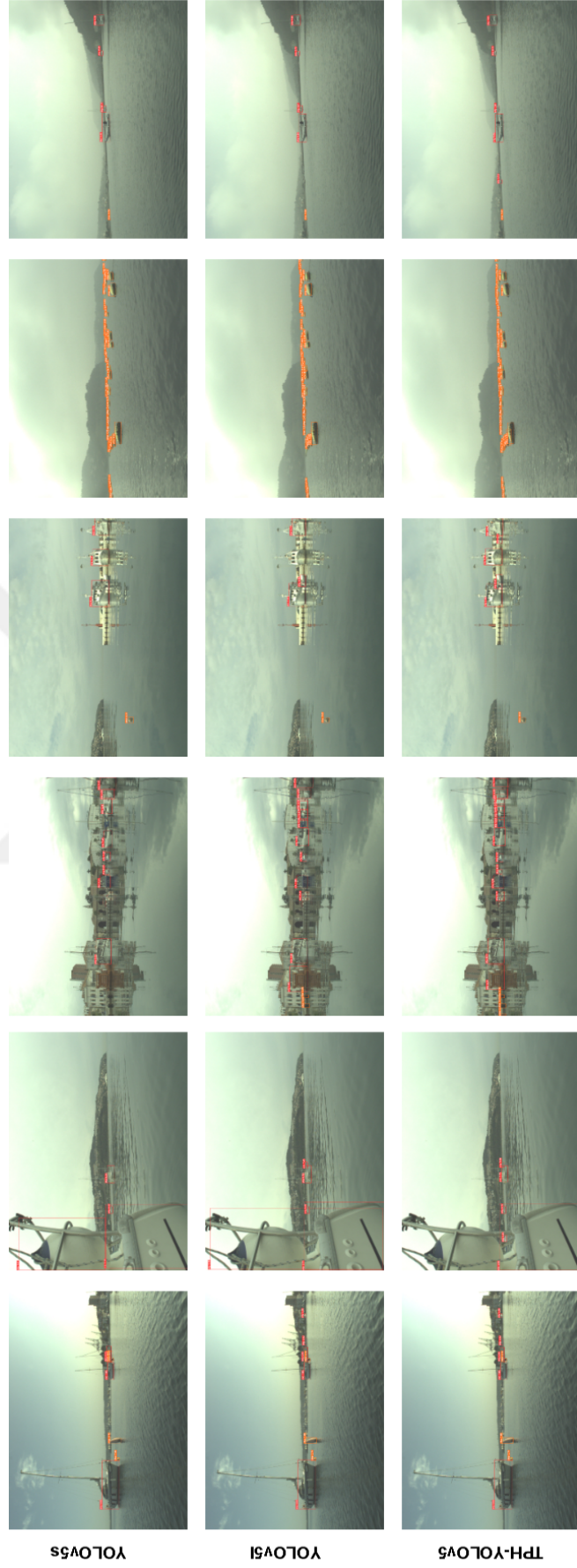


Figure 4.1 : Comparison of object detection performance across three YOLO model variants (YOLOv5s , YOLOv5l, TPH-YOLOv5) on the MODS dataset, illustrating detection outcomes in various maritime scenes.

4.4 Ship Classification Results

This section proposes a comparable analysis of ship classification on MARVEL dataset using Vision Transformer [5] based models (eg. DeiT [6], Swin Transformer [7], and ConvNext-v2 [8]). The effects of model training, hyperparameter optimization, performance comparison between CNN-based and transformer-based architectures, visualization of attention rollouts, and pruning techniques on model efficiency are studied in detail. Each experiment is designed to evaluate the effectiveness, generalizability, and suitability of the models for real-world maritime applications.

Training process

Hyperparameter optimization of ViTs

For Vision Transformer models, hyperparameter optimization is particularly essential due to their sensitivity to parameter selection and their unique architecture. In contrast to CNNs, ViTs handle images by dividing them into sequences of patches and rely on optimal choices for parameters like learning rate, patch size and embedding dimensions. Improper selection of hyperparameters can lead to slow convergence, suboptimal feature extraction, or even training failure. Additionally, transformers require careful tuning of regularization methods like weight decay and dropout in order to mitigate overfitting, particularly when dealing with smaller datasets.

In this study, we focus primarily on optimizing the learning rate for all of our transformer-based models, as it is one of the most crucial hyperparameters that influence training stability and convergence speed. Several techniques can be employed to fine-tune the learning rate effectively; among them, we preferred the manual search due to its simplicity and direct control over parameter selection.

In manual search, hyperparameter values are chosen manually based on experience and intuition rather than automated optimization techniques. To ease the process and reduce computational cost, we observed model performance at different learning rates without completing the full training process. Instead of running the models through all epochs, we conducted preliminary training for a few epochs to assess trends in loss

behavior and convergence. This approach allowed us to efficiently identify the best learning rates while avoiding unnecessary computation and long training times.

The comparative graphs in Figure 4.2 illustrate the impact of different LRs. Each row corresponds to a specific model, with the left column depicting training loss trends and the right column showing the evaluation accuracy over the course of training. The goal of this analysis was to identify the most effective learning rate that ensures stable convergence and high accuracy.

For the ViT model, the training loss curves indicate that a lower learning rate (e.g., $1e-4$) leads to more stable and gradual convergence, while a lower learning rate ($5e-5$) also performs well. However, the lowest learning rate ($1e-5$) struggles, as seen by its higher loss throughout training. The evaluation accuracy graph confirms this, where ViT with $1e-4$ learning rate achieves better generalization.

For DeiT, a similar trend is observed. The higher learning rate ($5e-4$) shows unstable loss reduction, suggesting difficulties in convergence, while lower learning rates ($1e-5$, $1e-4$) result in more controlled and stable training. The evaluation accuracy further highlights that $1e-4$ provides better accuracy.

Both Swin-T and Swin-B models exhibit smoother loss curves across different learning rates. However, Swin-T trained with $5e-4$ appears to struggle with convergence initially, while Swin-T with $1e-4$ and $5e-5$ maintain a better trajectory. Swin-B, being a larger model, shows stable training behavior across all learning rates, although $5e-5$ outperforms in terms of final accuracy.

For the ConvNext v2 model, the training loss curves indicate that the lowest learning rate ($1e-5$) leads to unstable convergence, while $1e-4$ and $5e-5$ demonstrate a more stable decrease in loss. The evaluation accuracy confirms this observation, with $1e-4$ achieving the best generalization.

Table 4.5 presents the final hyperparameters used for training different vision transformer-based models for ship classification. The learning rate (LR) was set to $1e-4$ for most models, except for SwinT-Tiny, which utilized a lower learning rate of $5e-5$ to facilitate stable convergence. All models employ a linear learning rate scheduler, ensuring a gradual adjustment of the learning rate over time. Warmup ratios

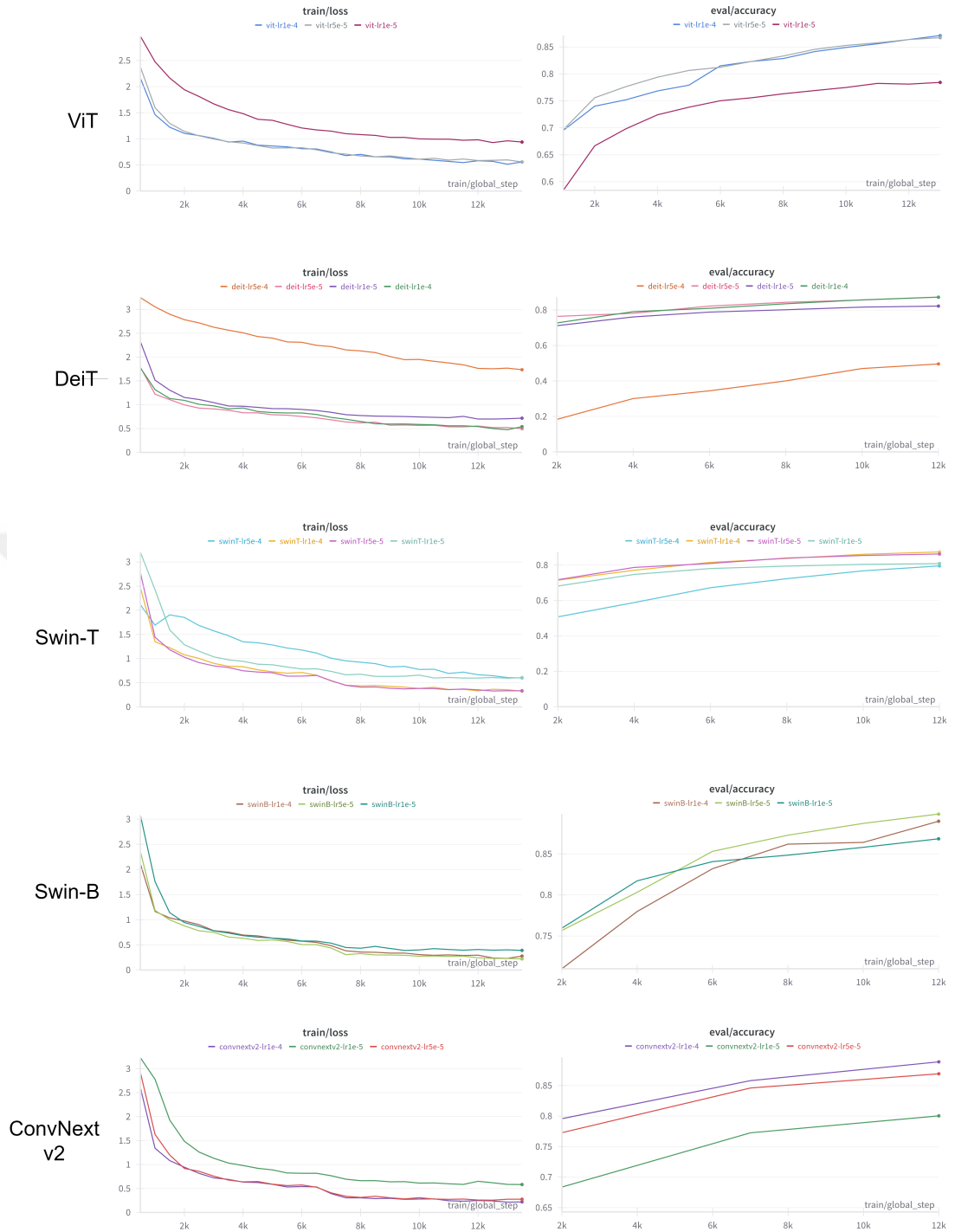


Figure 4.2 : Comparative analysis of learning rate optimization for different transformer-based and convolutional models. Each row represents the performance of a specific model—ViT, DeiT, Swin-T, Swin-B, and ConvNext v2—trained with different learning rates. The left column shows the training loss curves, while the right column presents the evaluation accuracy over training steps.

Table 4.5 : Hyperparameters used for training different models.

Model	LR	Scheduler LR	Warmup Ratio	Weight Decay	Epoch	Train/Eval Batch Size
ViT	1e-4	Linear	0	0.01	15	10 / 4
DeiT	1e-4	Linear	0	0.01	15	10 / 4
SwinT-Base	1e-4	Linear	0.1	0.01	15	10 / 4
SwinT-Tiny	5e-5	Linear	0.1	0.01	15	10 / 4
ConvNext-T	1e-4	Linear	0.1	0.01	15	10 / 4

are set to 0 for ViT and DeiT, whereas Swin Transformers and ConvNext-T apply a warmup ratio of 0.1 to stabilize initial training. A weight decay of 0.01 is consistently applied across all models to prevent overfitting. Training is conducted for 15 epochs, with a batch size of 10 for training and 4 for evaluation, ensuring a balance between computational efficiency and model performance. These hyperparameter settings were optimized through manual search to enhance model generalization.

Training

The training process of the models, as illustrated in the Figure 4.3, reveals several key insights regarding accuracy improvements and loss convergence. The training loss plot (first figure) shows a sharp initial drop for all models, as expected in deep learning optimization. As training progresses, the loss stabilizes, with some minor fluctuations. The lower loss for DeiT and Swin indicates their superior convergence, while ViT and ConvNext-T exhibit slightly higher values. The evaluation accuracy plot (second figure) highlights the progression of classification performance over training steps. All models show a steady increase in accuracy, with DeiT and Swin outperforming the others slightly towards the end of training, demonstrating that a more conservative learning rate can still yield strong results.

Evaluation results

This study comprises evaluation of vision transformer-based models for the task of ship classification. Beyond just analyzing the effectiveness of transformer architectures, a key objective was to determine whether these models have a significant advantage over CNNs when applied to this specific problem. To make a fair comparison, two widely

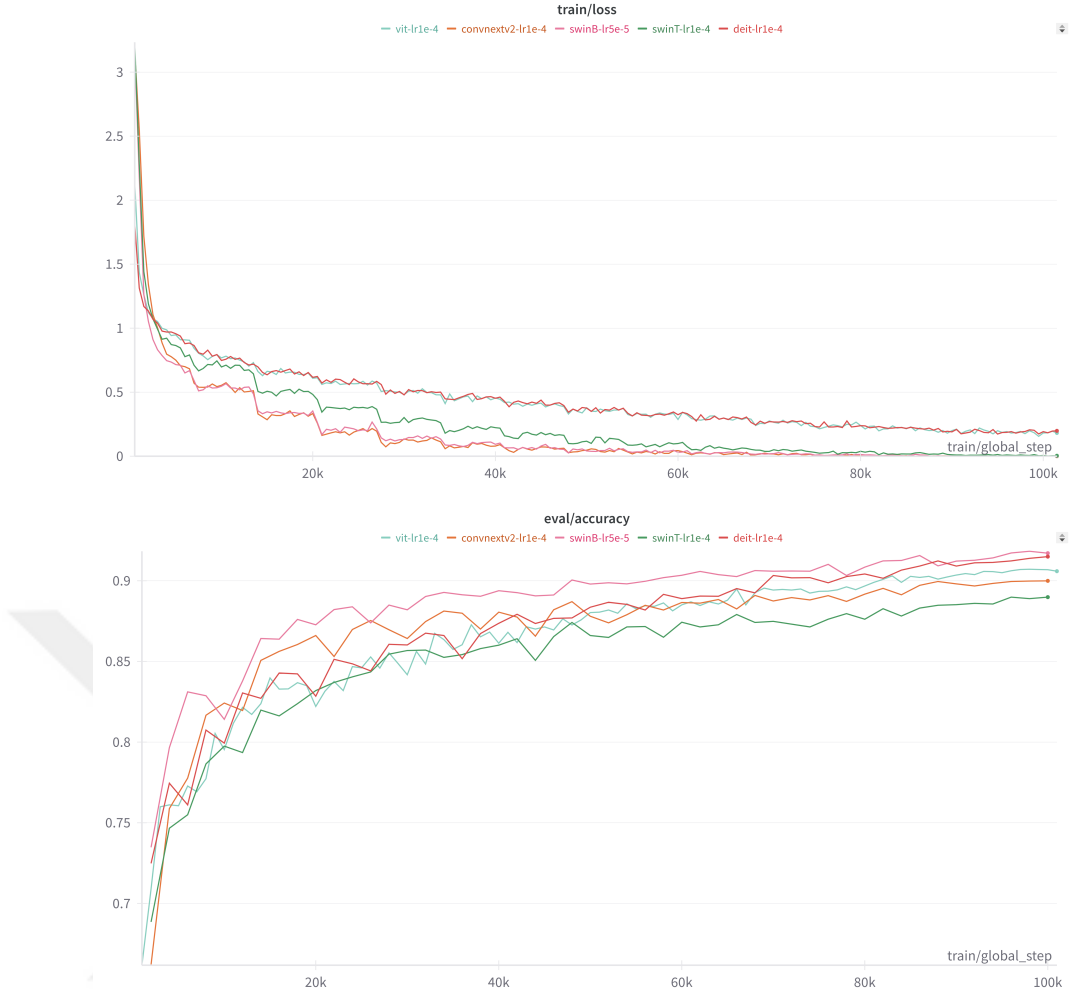


Figure 4.3 : Training loss and evaluation accuracy curves for different models. The training loss graph illustrates the reduction in loss over training steps, while the evaluation accuracy graph shows the accuracy progression throughout the training process.

used CNN architectures, ResNet50 and ResNet101, were selected as base networks. Both models were fine-tuned with the same experimental setups and on the same subset of the MARVEL dataset used for training the transformer-based models. The study aimed to provide a clear understanding of the strengths and limitations of both model families in terms of classification accuracy, computational cost, and suitability for real-world maritime applications.

Table 4.6 summarizes the performance of different neural network architectures evaluated on the MARVEL dataset. Unlike previous studies such as [2], [49], and [50], which primarily reported validation set performance, our study evaluates models on the test set for a more accurate assessment. Our approach, utilizing the DeiT model,

Table 4.6 : Performance Comparison of Various Neural Network Architectures on the MARVEL [2] Dataset.

Study	#class	method	acc.type	acc.(%)
Erhan et al. [2]	26	AlexNet [35]	Val	73.14
Erhan et al. [2]	26	SVM [36]	Val	53.89
Leclerc et al. [49]	26	Inception-v3 [40]	Val	78.73
Leclerc et al. [49]	26	ResNet [39]	Val	75.84
Salem et al. [50]	5	EfficientNetB2 [37]	Test	97.04
Salem et al. [47]	26	EfficientNet-B5 [37]	Val	91.60
This study	26	ResNet50 [39]	Test	91.76
This study	26	ResNet101 [39]	Test	89.87
This study	26	ViT-B [5]	Test	82.68
This study	26	DeiT-B [6]	Test	92.87
This study	26	Swin-T [7]	Test	87.72
This study	26	Swin-B [7]	Test	90.44
This study	26	ConvNext-v2 [8]	Test	90.07

Table 4.7 : Comparison of Image Classification Models' Computation Loads

Model	FPS
ResNet50	47.05
ResNet101	44.65
ViT-B [5]	0.95
DeiT-B [6]	1.31
Swin-B [7]	0.51
Swin-T [7]	1.14
ConvNext-v2 [8]	0.13

achieved a test accuracy of 92.87%, exceeding the previous best accuracy of 91.60%, reported in [50] using EfficientNet-B5. Since both studies conducted classification across 26 distinct ship classes, this improvement demonstrates the advantages of transformer-based architectures like DeiT over conventional CNN-based models, particularly in handling complex image classification tasks within large-scale maritime datasets.

Table 4.7 compares the computational efficiency and the speed of the inference, both crucial factors for the classification of ships in real time in maritime operations. ResNet-based CNN models (ResNet50, ResNet101) demonstrate a strong balance between accuracy and inference speed, achieving relatively high Frames Per Second

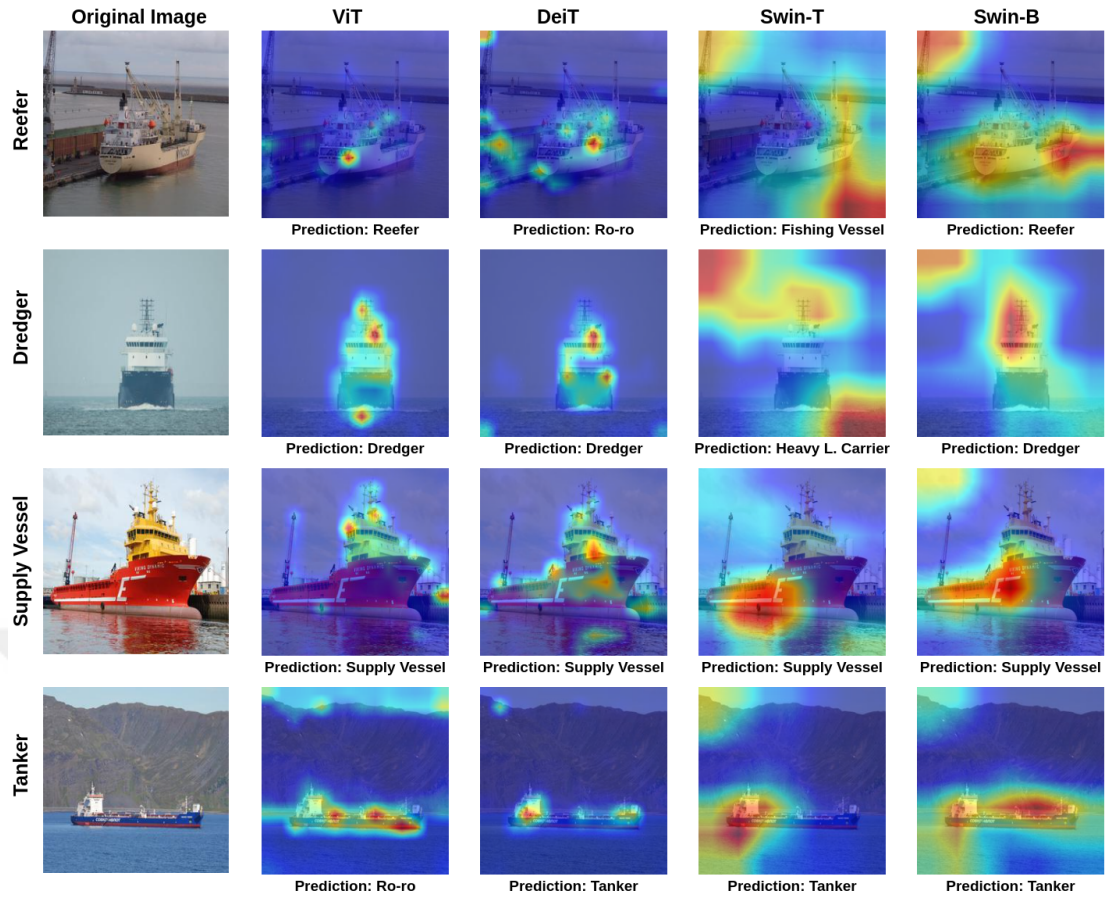


Figure 4.4 : Attention rollout maps for true and false predicted samples on vision transformer models.

(FPS) values. This makes ResNet architectures a compelling choice for practical real-time deployment, where both performance and speed are critical.

In contrast, transformer-based models, despite achieving superior classification accuracy, exhibit significantly lower FPS due to higher computational complexity. DeiT-B emerges as the best-performing model in terms of accuracy, reaching 92.87%, but its high Floating Point Operations per Second (FLOPs) and slower inference speed make it less suitable for real-time applications. ConvNext-v2, while computationally demanding, also achieves competitive accuracy at 90.07%. Ultimately, CNN models remain more practical for real-time maritime tasks due to their faster inference times, whereas transformer-based models like DeiT are better suited for applications where accuracy is the primary concern, rather than speed.

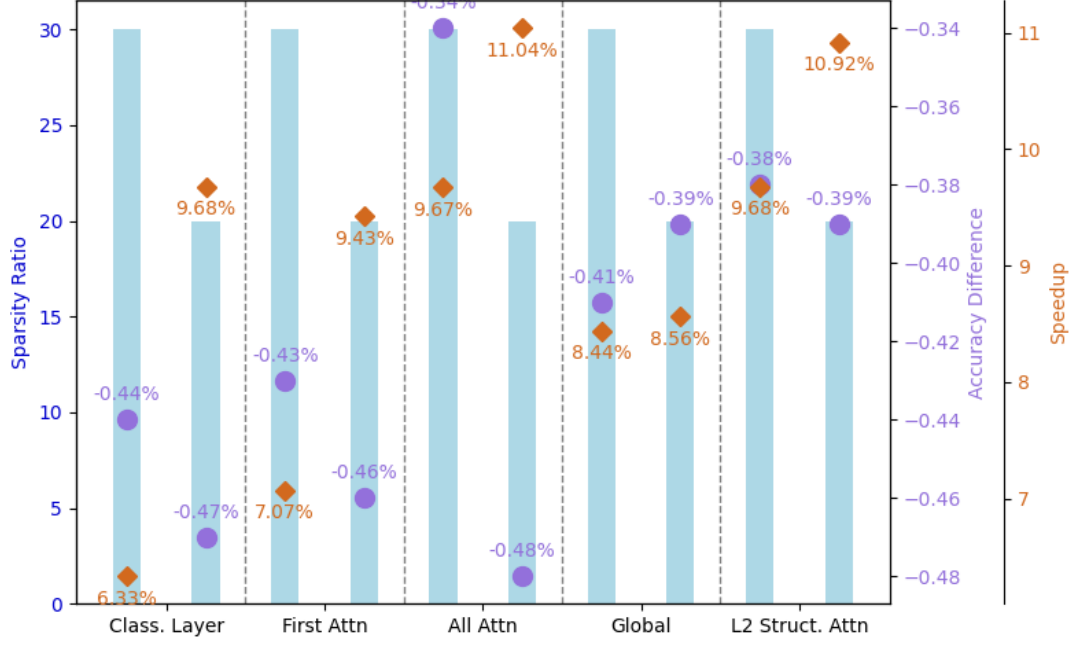


Figure 4.5 : Sparsity vs Accuracy and Inference time change when applying pruning to our fine-tuned DeiT model.

Attention rollout results

The other task in this work is extracting attention regions of vision transformer-based models. This step was important to understand whether our models focus on the proper areas of the input image during the decision-making process.

Figure 4.4 visualizes some input images correctly or incorrectly predicted. The rows correspond to specific vessel types, with the true class labels shown on the far left of each row. Each column depicts the output of a different model: the first pictures the input image and the remaining four columns picture the output of ViT, DeiT, Swin-T, and Swin-B models, respectively. The maps in each cell show the image regions that each model focuses on during classification and highlight the most important areas in the model’s decision-making process. Below each output given in the attention map, the prediction result of the corresponding model for that output is indicated in the row.

Inaccuracies in the predictions commonly occurred when the models focused on the background of images or the water surface. However, when we examine the focus area on images where the models correctly predict the ship type, we observe that the input generally points to the correct place. This visualization provides insight into how the

models' attention mechanisms function during prediction and helps explain the reasons behind incorrect predictions.

Pruning results

Since our best-performing model was DeiT, we observed the effects of pruning on this model. Figure 4.5 displays the impact of different pruning techniques applied to the DeiT model, focusing on various components of the model. Each pruning approach is compared in terms of the sparsity ratio (blue bars), inference speed-up (orange diamonds), and accuracy change (purple dots). Each method has been observed with two sparsity ratios 30% and 20%. The findings show only minor accuracy drops, ranging from 0.34% to 0.48%, while the inference speed improves significantly, with gains between 6.33% and 11.04%. Among the strategies tested, L2 structured pruning with 20% sparsity provides the best trade-off, delivering a 10.92% speedup with just a 0.39% reduction in accuracy. However, even with these optimizations, the speed of the DeiT model remains below that of typical CNN architectures in terms of FPS.



5. CONCLUSIONS

The use of Unmanned Surface Vehicles (USVs) in the maritime sector has created a major transformation in both civil and military operations. Especially in terms of safe navigation and environmental monitoring, it is of great importance that USVs perceive their surroundings accurately and detect critical objects in a timely manner. However, the dynamic nature of the marine environment, variable weather conditions, water surface reflections and small object density make this detection process difficult. In order to overcome these difficulties, deep learning-based object detection and classification methods are being developed. However, the ability of these models to generalize in maritime environments, the suitability of their inference speeds for the application, and their ability to focus on the right regions are still among the issues that need to be investigated.

In this thesis, object detection and ship classification problems are addressed in order to enable USVs to navigate safely and autonomously in marine environments. The study presents a two-stage approach. In the first phase, the aim is to detect small and dynamic objects in real time using the MODS dataset with YOLOv5 and its derivative models. In this phase, the focus is on the accurate detection of small-sized and moving obstacles on the water. In addition to standard YOLOv5 models, the TPH-YOLOv5 model, which was developed to increase the performance of small object detection, has been examined. Thanks to components such as Transformer Prediction Heads (TPH) and Convolutional Block Attention Modules (CBAM), TPH-YOLOv5 has achieved success in detecting small objects compared to classical YOLOv5 models. In the experiments, the TPH-YOLOv5 model has shown better performance in metrics such as precision, recall and mean Average Precision (mAP).

In the second stage, an in-depth analysis was performed on the ship classification problem using the MARVEL dataset. In this stage, the focus was on the Transformer-based models such as Vision Transformer (ViT) and its different variants,

DeiT, Swin Transformer and ConvNext-v2. As a result of the comparisons, it was observed that Transformer-based models provided a significant superiority over classical CNN architectures (e.g. ResNet50 and ResNet101), especially in terms of classification accuracy. While the DeiT model achieved the highest success with 92.87% accuracy rate, the Swin Transformer and ConvNext-v2 models also demonstrated the effectiveness of the Transformer architecture in maritime classification tasks.

In this study, the developed attention rollout method has been successfully applied to make the decision-making processes of Vision Transformer (ViT) based models more understandable. Unlike previous studies, the proposed method has been adapted not only to standard ViT architectures but also to variants such as DeiT and Swin Transformers. The image regions that the models focus on have been visualized through heat maps, and the attention differences between correct and incorrect predictions have been analyzed. It has been observed that the models focus on target object features in correct classifications and are distracted by background elements in incorrect classifications.

In addition, model pruning techniques were used to increase the computational efficiency of the classification models. Pruning experiments performed on the DeiT model provided speed gains of up to 11%, while causing only a small loss of 0.39% in accuracy. This shows that transformer-based models can be made more suitable for real-time applications with optimization methods, but CNNs are still the best choice when speed is more important than precision.

As a result, this thesis contributes to operation of USVs in maritime missions by presenting models that provide both high accuracy and processing efficiency. In future studies, it is aimed to combine detection and classification tasks under a single system, further simplify the models and test them under different environmental conditions.

REFERENCES

- [1] B. Bovcon, J. Muhovič, D. Vranac, D. Mozetič, J. Perš, and M. Kristan, “MODS—A USV-Oriented Object Detection and Obstacle Segmentation Benchmark,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13403–13418, 2022.
- [2] E. Gundogdu, B. Solmaz, V. Yücesoy, and A. Koç, “MARVEL: A Large-Scale Image Dataset for Maritime Vessels,” in *Computer Vision – ACCV 2016*, pp. 165–180, 2017.
- [3] G. Jocher, “ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements,” oct 2020. <https://github.com/ultralytics/yolov5>.
- [4] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, “TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios,” in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2778–2788, 2021.
- [5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in *9th International Conference on Learning Representations, ICLR 2021*, 2021.
- [6] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation through attention,” 18–24 Jul 2021.
- [7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” *CoRR*, vol. abs/2103.14030, 2021.
- [8] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, “ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16133–16142, 2023.
- [9] L. Žust, J. Perš, and M. Kristan, “LaRS: A Diverse Panoptic Maritime Obstacle Detection Dataset and Benchmark,” 2023.
- [10] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, “SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection,” *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

- [11] J. Jaekel, S. Moosbauer, D. König, and M. Teutsch, “A Benchmark for Deep Learning Based Object Detection in Maritime Environments,” 06 2019.
- [12] Y. Zheng and S. Zhang, “Mcships: A Large-Scale Ship Dataset For Detection And Fine-Grained Categorization In The Wild,” in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2020.
- [13] M. M. Zhang, J. Choi, K. Daniilidis, M. T. Wolf, and C. Kanan, “VAIS: A dataset for recognizing maritime imagery in the visible and infrared spectrums,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 10–16, 2015.
- [14] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *CoRR*, vol. abs/1506.01497, 2015.
- [15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, “SSD: single shot multibox detector,” *CoRR*, vol. abs/1512.02325, 2015.
- [16] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018.
- [17] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” *CoRR*, vol. abs/1612.03144, 2016.
- [18] L. A. Varga, B. Kiefer, M. Messmer, and A. Zell, “Seadronessee: A maritime benchmark for detecting humans in open water,” *CoRR*, vol. abs/2105.01922, 2021.
- [19] B. Kiefer *et al.*, “1st Workshop on Maritime Computer Vision (MaCVi) 2023: Challenge Results,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 265–302, January 2023.
- [20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” 2022.
- [21] Bovcon, Borja and Muhovič, Jon and Perš, Janez and Kristan, Matej, “The MaSTr1325 dataset for training deep USV obstacle detection models,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3431–3438, 2019.
- [22] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan, “Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation,” *Robotics and Autonomous Systems*, vol. 104, p. 1–13, June 2018.
- [23] B. Bovcon and M. Kristan, “WaSR—A Water Segmentation and Refinement Maritime Obstacle Detection Network,” *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 12661–12674, 2022.

- [24] M. Teršek, L. Žust, and M. Kristan, “eWaSR – An Embedded-Compute-Ready Maritime Obstacle Detection Network,” *Sensors*, vol. 23, no. 12, p. 5386, 2023.
- [25] L. Žust and M. Kristan, “Temporal Context for Robust Maritime Obstacle Detection,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6340–6346, 2022.
- [26] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” *CoRR*, vol. abs/1709.01507, 2017.
- [27] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” *CoRR*, vol. abs/1903.06586, 2019.
- [28] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, “Perceptual generative adversarial networks for small object detection,” *CoRR*, vol. abs/1706.05274, 2017.
- [29] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, “Augmentation for small object detection,” *CoRR*, vol. abs/1902.07296, 2019.
- [30] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: fully convolutional one-stage object detection,” *CoRR*, vol. abs/1904.01355, 2019.
- [31] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “Centernet: Keypoint triplets for object detection,” *CoRR*, vol. abs/1904.08189, 2019.
- [32] R. Varghese and S. M., “Yolov8: A novel object detection algorithm with enhanced performance and robustness,” in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pp. 1–6, 2024.
- [33] F. Zhao, J. Zhang, and G. Zhang, “Ffedet: Fine-grained feature enhancement for small object detection,” *Remote Sensing*, vol. 16, no. 11, 2024.
- [34] W. Zhou, X. Wang, Y. Fan, Y. Yang, Y. Wen, Y. Li, Y. Xu, Z. Lin, L. Chen, S. Yao, L. Zequn, and J. Wang, “Kdsmall: A lightweight small object detection algorithm based on knowledge distillation,” *Computer Communications*, vol. 219, 04 2024.
- [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [36] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support Vector Machines,” *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.
- [37] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, 09–15 Jun 2019.

- [38] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation,” *CoRR*, vol. abs/1801.04381, 2018.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR ’16, pp. 770–778, June 2016.
- [40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2015.
- [41] B. Iancu, V. Soloviev, L. Zelioli, and J. Lilius, “ABOShips – An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations,” 2021.
- [42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
- [43] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN,” 2019.
- [44] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path Aggregation Network for Instance Segmentation,” *CoRR*, vol. abs/1803.01534, 2018.
- [45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.
- [46] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus),” 2023.
- [47] M. H. Salem, Y. Li, and Z. Liu, “Transfer Learning on EfficientNet for Maritime Visible Image Classification,” in *2022 7th International Conference on Signal and Image Processing (ICSIP)*, pp. 514–520, 2022.
- [48] S. Abnar and W. Zuidema, ““Quantifying Attention Flow in Transformers”,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, July 2020.
- [49] M. Leclerc, R. Tharmarasa, M. C. Florea, A.-C. Boury-Brisset, T. Kirubarajan, and N. Duclos-Hindié, “Ship Classification Using Deep Learning Techniques for Maritime Target Tracking,” in *2018 21st International Conference on Information Fusion (FUSION)*, pp. 737–744, 2018.
- [50] M. H. Salem, Y. Li, Z. Liu, and A. M. AbdelTawab, “A Transfer Learning and Optimized CNN Based Maritime Vessel Classification System,” *Applied Sciences*, vol. 13, no. 3, 2023.

CURRICULUM VITAE

Name SURNAME: Sevda SAYAN

EDUCATION:

- **B.Sc.:** 2020, Hacettepe University, Engineering Faculty, Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2019 - Present Senior Software Engineer, ASELSAN A.Ş.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Sayan, S.** and Ekenel, H. K. (2025). Obstacle Detection and Ship Recognition System for Unmanned Surface Vehicles. In Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP, ISBN 978-989-758-728-3, ISSN 2184-4321, pages 113-122. DOI: 10.5220/0013152000003912

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Ömer Faruk Yaşar, Sevda Sayan, Meltem Tokgöz, Fatmanur Turhan, Furkan Çağlar Gülmez, Buğra Fırat Çıtak, Yahya Koçak, Samet Hicsonmez, Pınar Duygulu. Recognition of Daily Activities in Kitchen. 28st Signal Processing and Communications Applications Conference (SIU) Poster Presentation October 5-7, 2020, Turkey.

