Automated Vessel Detection in Satellite Imagery
(Automatizált hajófelismerés műholdas képeken)

Fábián Csenge
KGM10V

Faluvégi Ármin
WYL4R7

Kubinyi Bálint
OMJWZ6

Abstract

This study is about a Kaggle competition called the Airbus Ship Detection Challenge, which focuses on important aspects of monitoring and ensuring safety at sea. The increasing number of ships raises concerns about accidents, piracy, illegal fishing, drug trafficking, and cargo movements. Big companies like Airbus contribute by offering thorough maritime monitoring services using satellite images. The challenge calls on Kaggle participants to develop a model for accurate ship detection in satellite images, even in challenging conditions like clouds or haze. The study explores some methods used in the past, such as convolutional neural networks and ensemble methods. It introduces the YOLOv8 model, emphasizing its advantages and components. The implementation details using Google Colab and the YOLOv8 model are discussed, covering preprocessing, training, evaluation, and testing. The study concludes by outlining future plans to refine the model.

Kivonat

Ez a tanulmány a Kaggle által szervezett Airbus Ship Detection Challenge versenyről számol be, mely a tengeri területek megfigyelésére és biztonságára összpontosít. Az egyre növekvő hajóforgalom számos aggodalmat vet fel a balesetek, kalózkodás, illegális halászat, kábítószer-kereskedelem és rakománymozgás tekintetében. Nagyvállalatok, köztük az Airbus, részt vesznek részletes tengerfigyelési szolgáltatások fejlesztésében, amelyek műholdképeket alkalmaznak. A verseny célja, hogy a Kaggle résztvevői olyan modellt fejlesszenek ki, amely pontosan képes detektálni a hajókat műholdképeken, még nehezen kezelhető körülmények között is, például felhők vagy homály esetén. A tanulmány áttekint néhány korábban alkalmazott módszert, beleértve a konvolúciós neurális hálózatokat és az ensemble módszereket is. Bemutatja a YOLOv8 modellt, kiemelve annak előnyeit és összetevőit. A megvalósítás részletesen ismerteti a Google Colab és a YOLOv8 modell felhasználásával végzett folyamatokat, beleértve az előfeldolgozást, a tanítást, a kiértékelést és a tesztelést. A tanulmány a modell finomítására vonatkozó jövőbeli tervek felvázolásával zárul.

1 Introduction

The Kaggle Airbus Ship Detection Challenge deals with important issues in monitoring the sea and ensuring safety. As more ships travel the seas, concerns arise about accidents harming the environment, piracy, illegal fishing, drug trafficking, and sneaky cargo movements. Big companies like Airbus are helping by providing thorough maritime monitoring services.

Airbus, a major player in the airplane industry, uses satellite images to cover large areas, conduct detailed analyses, monitor closely, and respond quickly to understand what's happening. They work with well-trained experts to support the maritime industry in learning more, preparing for potential

Submitted to the subject *Deep Learning in Practice with Python and LUA*

issues, sounding alarms, and improving efficiency at sea. The specific focus of this challenge is to develop a model that can accurately and rapidly detect ships in satellite images, even in challenging situations like the presence of clouds or haze.

In addition to promoting innovation in ship detection, the competition offers a special Algorithm Speed Prize, encouraging participants to optimize their programs for efficient processing on a substantial dataset of over 40,000 image chips. This reflects the real-world need for quick and accurate ship detection capabilities, aligning with Airbus's commitment to advancing automatic object extraction from satellite images.

2 Overview of the Research Area

In the last ten years, technology has improved how we watch over the seas, especially in automatically spotting things in satellite pictures. Even though we've made big steps forward, these improvements haven't really made a big difference in actual operations yet. The Airbus Ship Detection Challenge is a call to action for Kaggle participants to bring their skills and creative ideas to make automatic ship detection even more accurate and faster. The challenge isn't just about being really accurate; it's also about making the computer programs work faster. In addition, there's a special prize for the fastest program, called the Algorithm Speed Prize, as an extra motivation.

2.1 Previous Solutions

The challenge of detecting ships in satellite imagery has been a subject of extensive research. Many different approaches have been tried to handle the challenges that come with ships being different sizes, dealing with different weather conditions, and sometimes overlapping in the pictures. Researchers have explored various methods, taking inspiration from the vast field of computer vision and machine learning.

One interesting approach is using convolutional neural networks (CNNs) for image segmentation. These networks have been successful in understanding the detailed relationships between things in pictures, which makes them good at figuring out where ships are in satellite images. Another technique gaining attention is transfer learning, where models already trained on different tasks are adjusted to work for finding ships. This helps by using the knowledge gained from lots of different datasets to improve how well ships can be detected.

Moreover, some researchers have used semantic segmentation, a method that precisely outlines ship borders at the pixel level. This involves training models with labeled datasets to teach them to recognize and categorize each pixel in a picture, allowing for detailed segmentation of ships against complex backgrounds.

To make ship detection more reliable and generalizable, ensemble methods are being explored. These methods combine predictions from different models, bringing together their strengths to overcome weaknesses and boost overall performance.

Even though there has been significant progress, challenges remain, especially when ships are very close together, causing parts of them to overlap in the pictures. Recent research has focused on refining accuracy through improved post-processing techniques to address this issue.

3 System Design: Description of the Network Used

The YOLOv8 (You Only Look Once version 8) model is a cutting-edge technology in real-time object detection, known for being efficient, accurate, and versatile. YOLOv8 has evolved from its earlier versions, introducing architectural improvements and training strategies that make it well-suited for various applications, including the detection of ships in satellite imagery.

3.1 Key Components of YOLOv8

YOLOv8 uses CSPDarknet53 as its backbone architecture, which is a variant of the Darknet architecture with Cross Stage Partial (CSP) connections. CSPDarknet53 is the core feature extractor, capturing hierarchical features from input images. The CSP connection enhances information flow between different stages of the network, promoting efficient information exchange and mitigating information loss during processing. This architecture excels at capturing both low-level and high-level features, enabling the model to discern intricate patterns and contextual information crucial for ship segmentation in satellite imagery. The CSPDarknet53 backbone is vital for the success of our YOLOv8 model, providing a robust foundation for accurate ship detection.

Detection Head: The detection head of YOLOv8 predicts bounding boxes, confidence scores, and class probabilities, allowing for simultaneous detection of multiple objects in a single pass.

Anchor Boxes: To handle variations in object sizes, YOLOv8 utilizes anchor boxes. These predefined bounding boxes help to accurately locate objects of different scales.

Loss Function: During training, YOLOv8 employs a multi-component loss function, including classification loss, objectness loss, and bounding box regression loss. This comprehensive approach contributes to the model's ability to balance precision and recall.

3.2 YOLOv8 Training Strategy

The training process involves presenting the model with annotated datasets containing labeled ship instances. YOLOv8 learns to identify spatial features and patterns indicative of ship locations through the optimization of model parameters. The model is trained to output bounding boxes encompassing ships, along with associated confidence scores and class probabilities.

3.3 Advantages of YOLOv8

Speed: Known for real-time processing capabilities, YOLOv8 is suitable for applications requiring rapid object detection. The model's ability to process images in a single pass contributes to its exceptional speed.

Accuracy: YOLOv8's architecture considers contextual information within images, contributing to high accuracy in object detection. The model's capability to handle overlapping objects and varying scales enhances its performance in complex scenarios.

Versatility: YOLOv8 is versatile and adaptable to various object detection tasks. Its robust architecture and flexibility make it a popular choice for diverse applications, including ship detection in satellite imagery.

4 Implementation

For our project, we employed the advanced YOLOv8 model to tackle the complexities of detecting ships in satellite images. To carry out the implementation, we opted for Google Colab, a user-friendly cloud-based platform that facilitated collaborative coding, training, and validation of our YOLOv8 model. The platform's integration with GPU acceleration significantly sped up the training phase, enabling us to quickly iterate through model adjustments and hyperparameter tuning. The collaborative features of Colab also enhanced teamwork, allowing us to collectively develop, test, and refine our ship detection algorithm for the Airbus Ship Detection Challenge.

4.1 Preprocessing

In preparing our dataset for training the YOLOv8 model on ship detection in satellite images, the preprocessing stage played a crucial role. We began by accessing a shared Google Drive folder that contained all the necessary data, and through Google Colab, we mounted Google Drive, providing direct access to the shared folder with our training data. This collaborative environment greatly simplified our work, promoting efficient collaboration and resource utilization.

To ensure we had the essential tools for our machine learning tasks, we installed the 'scikit-learn' and 'ultralytics' libraries. These libraries are crucial for working with YOLOv8 models.

The next step involved data preparation, where we executed the file extraction process, making the dataset readily available in the Google Drive directory. Preprocessing steps were then implemented to convert Run-Length Encoded (RLE) mask data into a format compatible with YOLOv8 segmentation annotations. This included decoding RLE masks, converting binary masks to the YOLOv8 segmentation format, and creating a new DataFrame with the necessary annotations.

For dataset creation and organization, we utilized a function designed to create a YOLOv8-compatible dataset structure. This resulted in the generation of three datasets ('train', 'valid', and 'test'), each with its own image and label directories. Additionally, we created a YAML configuration file, 'dataset.yaml,' specifying the dataset structure, including paths to train, validation, and test images.

4.2 Training

During the training phase of our YOLOv8 model, a crucial step is taken to empower the model with the capability to precisely identify and categorize ships in satellite imagery. Utilizing the carefully curated dataset from earlier stages, we make use of the 'ultralytics' library to streamline the training process.

The training starts by loading a pre-trained YOLOv8 model through the 'YOLO' class from the 'ultralytics' library. This pre-trained model, named 'yolov8n-seg.pt,' serves as a robust starting point, capturing generalized features that can be fine-tuned for our specific ship segmentation task.

Following this, the loaded model undergoes training using the 'train' method. Key parameters, such as the path to the dataset configuration file, the number of training epochs, and the desired image size, are specified. This step enables the model to adapt to the intricacies of our ship detection dataset, enhancing its capacity to accurately recognize and outline ships.

The training results, encapsulated in the 'results' variable, provide valuable insights into the model's performance. Metrics like loss values, performance metrics, and other statistics are monitored, allowing for an evaluation of the model's progress throughout the training process.

4.3 Evaluation

In the evaluation phase, we measured how effectively our trained YOLOv8 model performs in ship segmentation. This evaluation is essential to understand how well the model generalizes to unseen scenarios and to gain insights into its strengths and areas for improvement.

After the completion of YOLOv8 training, crucial evaluation data and visualizations are generated and stored in the 'runs' folder. These include essential metrics, such as the confusion matrix (Figure 1). This provides a detailed breakdown of the performance of a classification model by summarizing the number of correct and incorrect predictions for each class. It consists of four key metrics:

True Positive (TP): The number of instances correctly predicted as the positive class.
True Negative (TN): The number of instances correctly predicted as the negative class.
False Positive (FP): The number of instances incorrectly predicted as the positive class (false alarms).
False Negative (FN): The number of instances incorrectly predicted as the negative class (missed detections).

The normalized confusion matrix (Figure 1) is a variation of the confusion matrix that presents the same metrics (True Positive, True Negative, False Positive, False Negative) but as proportions or percentages, making it easier to interpret in the context of the distribution of actual class instances. Each entry in the matrix is divided by the sum of the corresponding row, transforming the counts into proportions.

In our specific case, the confusion matrix indicates that there are 93 true positive (TP) instances, meaning the model correctly identified and classified 93 instances of the positive class. However, there are 33 false positive (FP) instances, indicating that the model incorrectly predicted the positive class for 33 instances that actually belong to the negative class. Additionally, there are 226 false negative (FN) instances, signifying that the model failed to identify 226 instances that truly belong to the positive class. This high count of false negatives suggests a limitation in the model's ability to identify negative cases correctly. We acknowledge the need for further refinement and adjustments in our training strategy to address this discrepancy and enhance the model's overall performance.

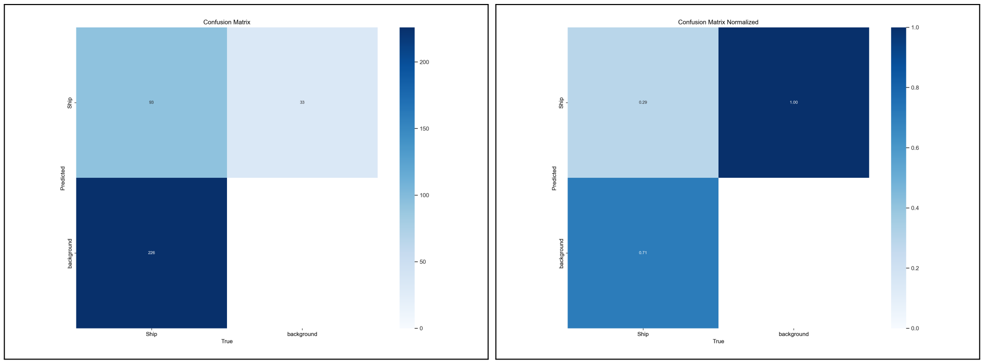

Figure 1: Confusion Matrix and Confusion Matrix Normalized

The 'results.png' (Figure 2) provides an insight into the model's performance on the validation set.

The **box loss** metric indicates how well the model is performing in terms of accurately predicting bounding boxes around ships. A lower box loss suggests that the model is effectively learning to locate and delineate ship boundaries.

The **segmentation loss** metric reflects the model's performance in segmenting ships from the background in satellite images. A lower segmentation loss indicates that the model is successfully capturing the spatial features of ships, contributing to accurate segmentation.

The **classification (cls) loss** measures how well the model is categorizing detected objects, in this case, ships. A lower classification loss implies that the model is becoming adept at assigning correct labels to identified ships.

The **dense focal (dfl) loss** is another important metric related to the focal loss, a specialized loss function. It addresses the issue of class imbalance by assigning different weights to different instances. A lower dense focal loss signifies that the model is effectively handling the challenges associated with imbalanced ship detection classes.

The **precision** is the accuracy of positive predictions, calculated as TP / (TP + FP).

The **recall** is the ability of the model to identify all relevant instances of the positive class, calculated as TP / (TP + FN).

The **mean Average Precision at 50% IoU (mAP50-95)** metric calculates the average precision of a model at a specific IoU threshold of 50%. IoU is a measure of how well the predicted bounding box overlaps with the ground truth bounding box.

The **mean Average Precision across IoU thresholds from 50% to 95% (mAP50-95)** metric extends the evaluation to a range of IoU thresholds, specifically from 50% to 95%. It calculates the average precision across this spectrum, providing a more comprehensive assessment of a model's performance at different levels of bounding box overlap.

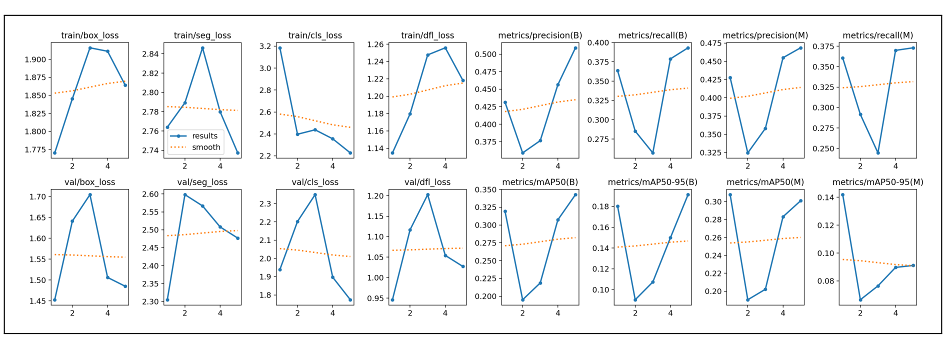

Figure 2: Confusion Matrix and Confusion Matrix Normalized

Our diagrams show quite good results for the critical metrics. Unfortunately, due to constraints in the availability of resources, we were only able to execute a limited number of training epochs. Despite this limitation, there is a promising aspect to highlight — our model's learning capabilities. Even within this constrained timeframe, it's evident that our model exhibits a propensity for learning, consistently enhancing its performance and yielding progressively better results after each training iteration. This observation encourages optimism about the model's potential given more extensive training opportunities.

The batch-specific visualizations (Figure 3) showcase the model's predictions on specific sets of training or validation data, allowing for a detailed examination of its output.

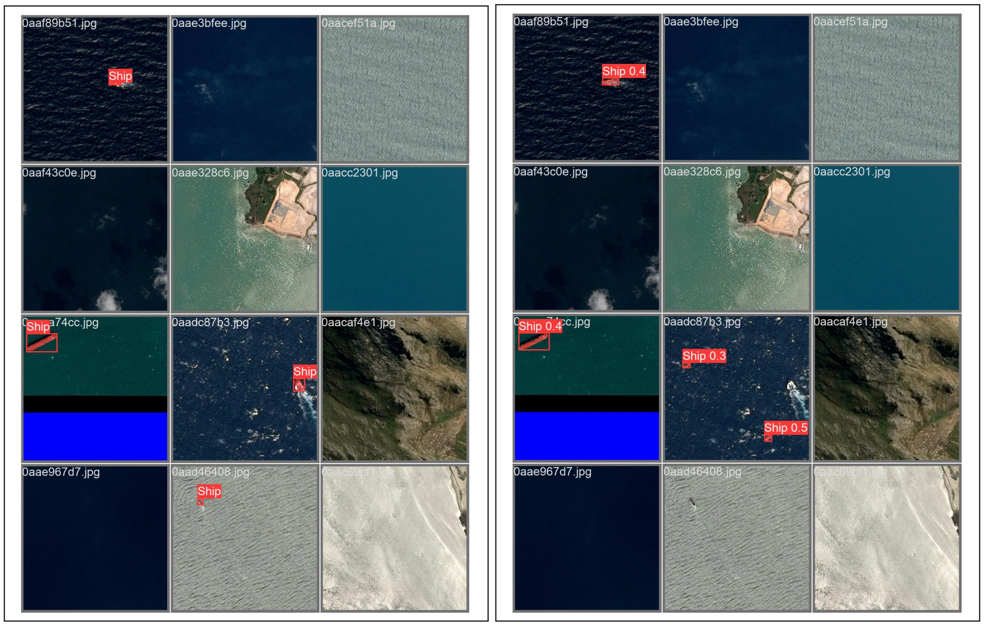

Figure 3: Confusion Matrix and Confusion Matrix Normalized

Within the depicted batch, it's notable that the model successfully recognized three ships accurately. However, it also encountered challenges, as one ship went unnoticed, and interestingly, a wave was misclassified as a ship.

4.4 Testing

In the testing phase, we analysed how well our model performs when making predictions on new and unseen data. This phase serves as a practical evaluation, offering insights into the model's ability to generalize beyond the training dataset.

We are pleased to report that our model demonstrated success in accurately detecting and delineating ships within the test images. This positive outcome signifies the robustness and generalization capabilities of our trained YOLOv8 model, showcasing its effectiveness beyond the training dataset.

5 Summary

The study begins with an overview of the Kaggle Airbus Ship Detection Challenge, emphasizing its significance in addressing maritime safety concerns. It introduces the role of big companies like Airbus in providing comprehensive maritime monitoring services using satellite imagery.

The subsequent section delves into the past decade, highlighting technological developments in maritime surveillance. The paper navigates through challenges in ship detection and surveys various methodologies, including convolutional neural networks, transfer learning, semantic segmentation, and ensemble methods.

In the third segment, we introduce the YOLOv8 model, known for its efficiency in real-time object detection. We provide a detailed insight into the backbone architecture and key elements of CSP-Darknet53, such as the detector head, anchor boxes and the loss function. The chapter further details the training strategy and highlights the advantages of YOLOv8, including its speed, accuracy and versatility.

In the fourth section, we shifted from theory to hands-on implementation, demonstrating the practical application of the YOLOv8 model. We introduced the use of Google Colab as our platform for developing and walked through the essential steps, covering pre-processing, model training, as well as evaluation and testing.

5.1 Future Plans

Looking ahead, the study outlines plans to enhance the YOLOv8 model's performance, exploring different strategies for improvement. We plan to delve into methods that optimize how the model works, and adjust its settings for better performance. This should make the model stronger and more accurate in spotting ships. We're also keeping an eye on the newest ideas and improvements in computer vision and machine learning. Our goal is to keep improving our model for ship detection in satellite images and stay on top of the latest methods in the field.

References

[1] D. Hordiiuk, I. Oliinyk, V. Hnatushenko and K. Maksymov, "Semantic Segmentation for Ships Detection from Satellite Imagery," *2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO)*, Kyiv, Ukraine, 2019, pp. 454-457, doi: 10.1109/ELNANO.2019.8783822.

[2] L. Ting, Z. Baijun, Z. Yongsheng and Y. Shun, "Ship Detection Algorithm based on Improved YOLO V5," 2021 *6th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Dalian, China, 2021, pp. 483-487, doi: 10.1109/CACRE52464.2021.9501331.

[3] L. Shen, B. Lang and Z. Song, "DS-YOLOv8-Based Object Detection Method for Remote Sensing Images," in *IEEE Access*, vol. 11, pp. 125122-125137, 2023, doi: 10.1109/ACCESS.2023.3330844.

[4] Fan W, Zhou F, Bai X, Tao M, Tian T. "Ship Detection Using Deep Convolutional Neural Networks for PolSAR Images" in *Remote Sensing*, 2019, 11(23):2862. https://doi.org/10.3390/rs11232862

[5] X. Chen, X. Wu, D. K. Prasad, B. Wu, O. Postolache and Y. Yang, "Pixel-Wise Ship Identification From Maritime Images via a Semantic Segmentation Model," in *IEEE Sensors Journal*, vol. 22, no. 18, pp. 18180-18191, 15 Sept.15, 2022, doi: 10.1109/JSEN.2022.3195959.

[6] Zhang T, Zhang X. "High-Speed Ship Detection in SAR Images Based on a Grid Convolutional Neural Network" in *Remote Sensing*, 2019; 11(10):1206. https://doi.org/10.3390/rs11101206

[7] Yuanyuan Wang, Chao Wang & Hong Zhang (2018) "Combining a single shot multibox detector with transfer learning for ship detection using sentinel-1 SAR images" in *Remote Sensing Letters*, 9:8, 780-788, DOI: 10.1080/2150704X.2018.1475770

[8] Y. Li, Z. Ding, C. Zhang, Y. Wang and J. Chen, "SAR Ship Detection Based on Resnet and Transfer Learning," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, 2019, pp. 1188-1191, doi: 10.1109/IGARSS.2019.8900290.

[9] Zhijun Chen, Depeng Chen, Yishi Zhang, Xiaozhao Cheng, Mingyang Zhang, Chaozhong Wu, "Deep learning for autonomous ship-oriented small ship detection" in *Safety Science*, Volume 130, 2020, 104812, ISSN 0925-7535, https://doi.org/10.1016/j.ssci.2020.104812.

[10] Chang Y-L, Anagaw A, Chang L, Wang YC, Hsiao C-Y, Lee W-H. "Ship Detection Based on YOLOv2 for SAR Imagery" in *Remote Sensing*, 2019, 11(7):786. https://doi.org/10.3390/rs11070786