Ctrl-Alt-SegmentAnything

Kristóf Géró, István Somodi, Sára Hugyecz

December 8, 2024

1 Abstract

Maritime transportation is a rapidly growing industry, posing significant environmental and security challenges. This project aimed to develop a deep learning model capable of segmenting ships in satellite images, contributing to maritime monitoring and global safety efforts. Using a dataset of 192,556 RGB satellite images with annotated segmentation masks, we implemented the DeepLabV3 model with a ResNet-50 backbone for precise ship detection.

The model was trained using weighted random sampling to address class imbalance, achieving a high validation Intersection over Union (IoU) of 88.85% after 5 epochs, with a training IoU of 76.20%, demonstrating strong performance in distinguishing ships from the background. Data augmentation techniques improved the model's robustness, while a graphical user interface (GUI) was developed for intuitive testing and visualization.

The results indicate that the model effectively identifies ships and their contours, offering a valuable tool for maritime monitoring systems. Future work includes exploring newer architectures and improving generalization capabilities to other datasets.

2 Introduction

Maritime transportation is one of the fastest-growing industries today, offering numerous economic and logistical opportunities while also posing significant risks. The increase in the number of ships raises the likelihood of environmental disasters, pirate attacks, illegal fishing, or drug trafficking. Satellite technologies play a key role in monitoring and preventing these activities. The goal of this project was to develop a deep learning model capable of efficiently identifying the positions of ships based on satellite images while accurately determining their contours. The resulting solution can be utilized in environmental monitoring systems and international security projects.

The problem of ship detection is not a new one, but the availability of more extensive data and advances in artificial intelligence have opened up new opportunities in this field. In the past, ship detection tasks were primarily carried out using manual annotation, which was a time-consuming and resource-intensive process. Modern machine learning techniques, particularly deep learning models such as convolutional neural networks, have revolutionized this area. Many models are well-suited to segmentation problems; however, the class imbalance, i.e., the disproportion between ship-containing and empty images in the dataset, continues to pose a challenge.

In the Airbus Ship Detection Challenge, the central problem was ship segmentation. The competition used a data annotation method known as Run-Length Encoding (RLE), which describes the position of ships in the images in a compressed format. Although technological advancements and existing models provided a foundation, the structure of the data and the class imbalance required new approaches.

3 System Design

For this project, we selected the DeepLabV3 model, which uses the ResNet-50 as its backbone architecture. This model is highly effective for segmentation tasks, particularly for high-resolution images, as it can efficiently handle a large field of view without losing the ability to detect fine details. DeepLabV3 employs atrous convolution, which maintains a larger effective receptive field by skipping pixels, allowing for detailed and rich results.

The loss function used was Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss), which is well-suited for binary segmentation problems. During training, the performance was evaluated using the

Intersection over Union (IoU) metric, complemented by the F1 score, which represents the harmonic mean of precision and recall.

Additionally, we explored alternative models such as UNet, YOLO, and the Segment Anything Model (SAM). UNet is effective in segmentation tasks but lacks advanced features like atrous convolution. YOLO excels in real-time object detection but lacks precision for detailed segmentation. SAM provides state-of-the-art segmentation capabilities but is computationally intensive and less adaptable to our pipeline. DeepLabV3 stood out due to its balance between detail preservation and efficiency, making it the ideal choice for this task.

4 Implementation

4.1 Data Preparation

The dataset consisted of 192,556 satellite images, each measuring 768x768 pixels and stored in RGB format. Of these, 42,556 images contained ships, while 150,000 images were empty (ship-free). The ratio of ship-containing to empty images was approximately 0.28.

We decoded the RLE masks into two-dimensional binary masks, where a pixel value of ”1” indicated the presence of a ship and ”0” represented the background. Additionally, each image was labeled with a binary ”has_ship” label, which indicated whether it contained a ship. This label was critical for balancing the dataset during training and was used extensively in the weighted random sampling process.

To standardize the dataset and reduce computational overhead, the images and masks were resized to 256x256 pixels and converted to PyTorch tensors.

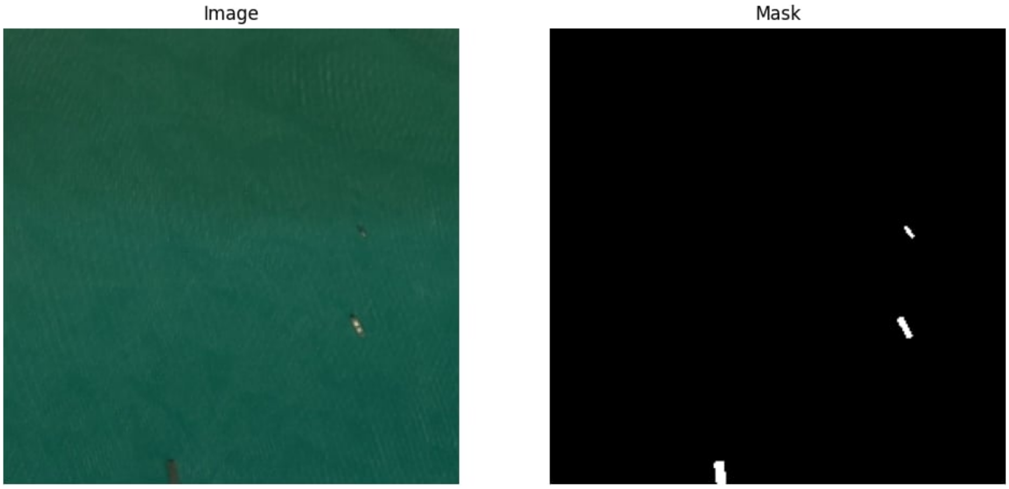

Figure 1: An example satellite image (left) and its corresponding mask (right) showing ship segmentation.

The image highlights how the model identifies ships against the background.

The dataset was split into training, validation, and test subsets. The training set was used to train the model, while the validation set was used for performance evaluation during training. The test set was reserved for the final evaluation of the model's performance. The initial smaller subset of 1,500 images was selected using the `dev_sample.py` script to enable rapid experimentation. This smaller dataset maintained the original proportion of ship-containing images (approximately 39%).

4.2 Training

During training, the class imbalance between ship-containing and empty images posed a significant challenge. To address this, we employed weighted random sampling, a technique that ensured balanced representation of both classes in each training batch. This approach enabled the model to learn effectively from both ship-containing and empty images.

After 5 epochs, the model achieved a validation IoU of 88.85% and a training IoU of 76.20%. The corresponding validation and training losses were 0.0008 and 0.0014, respectively, indicating effective convergence. These results demonstrated that the model performed well in both generalization and optimization.

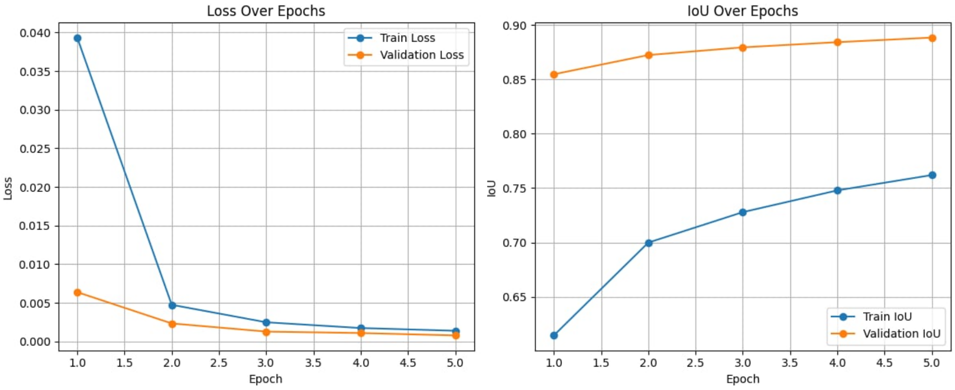

Figure 2: Training and validation performance across epochs. The left graph shows the loss reduction, while the right graph illustrates the improvement in IoU for both training and validation sets.

5 Evaluation

The choice of evaluation metrics played a crucial role in the project. The Intersection over Union (IoU) metric was selected to measure the overlap between the predicted and actual segmentation masks relative to their union. IoU is one of the most commonly used metrics in segmentation tasks because it is sensitive to cases where the model either over-predicts irrelevant areas as ships or under-predicts the actual ship regions. An IoU of 88.85% on the validation set demonstrated that the model accurately captured most of the ship contours, though occasional discrepancies between the predicted and actual masks were observed.

The F1 score was also a key metric in evaluating model performance. The F1 score, representing the harmonic mean of precision and recall, is particularly useful when there is an imbalance between classes. Given the lower proportion of ship-containing images, the F1 score helped assess how well the model identified ships while avoiding excessive false positives. High F1 scores indicated that the model successfully struck a balance between precision and recall.

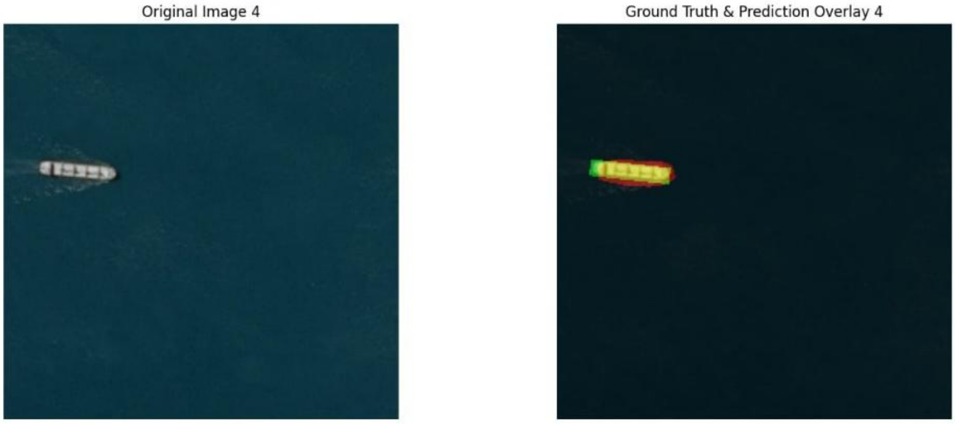

Figure 3: Example from the dataset: The original image (left) and its corresponding ground truth and predicted overlay (right). This demonstrates the model's accuracy in detecting and segmenting ships.

6 User Experience

As part of the project, we developed a graphical user interface (GUI) to facilitate easy testing of the model and visualization of results. The GUI was built using Python and implemented with the Flask framework for the backend. It was accessible in two ways: through a Docker environment or directly via Python.

The GUI allowed users to upload custom images, which were processed by the model, and displayed visualizations comparing the original image, the model-generated mask, and the ground truth mask.

Additionally, it included a feature to randomly select test or training data and display metrics such as IoU, precision, recall, and F1 score.

7 Summary

The project successfully demonstrated the potential of deep learning in maritime monitoring. By combining robust data preparation, the DeepLabV3 model, and techniques like weighted random sampling, we created a system capable of accurately segmenting ships in satellite images. Future work will focus on exploring newer architectures and further optimizing the model for diverse datasets.

8 AI Usage

We used artificial intelligence tools in several areas to assist with this project. Specifically:

- **Code Documentation:** AI was leveraged to generate detailed, clear comments in the Python code, ensuring its understandability for future developers or team members.
- **Translation:** AI tools facilitated translations between Hungarian and English, which was essential for creating bilingual documentation.

References

[1] F. van Beers, *Using Intersection over Union Loss to Improve Binary Image Segmentation*, University of Groningen Thesis, 2018. `https://fse.studenttheses.ub.rug.nl/18139/1/AI_BA_2018_FlorisvanBeers.pdf`

[2] M. Rahimzadeh, S. Attar, and F. Puech, *Deep Neural Networks with Intersection over Union Loss for Binary Image Segmentation*, Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2019, pp. 135–141. `https://www.scitepress.org/Papers/2019/73475/73475.pdf`

[3] A. Akagić, et al., *Early Stage Flame Segmentation with DeepLabv3+ and Weighted Cross-Entropy*, ResearchGate, 2023. `https://www.researchgate.net/publication/372292148_Early_Stage_Flame_Segmentation_with_DeepLabv3_and_Weighted_Cross-Entropy`

[4] M. Kar, et al., *Cross-Entropy Loss Functions: Theoretical Analysis and Applications*, arXiv preprint arXiv:2304.07288, 2023. `https://arxiv.org/abs/2304.07288`

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*, arXiv preprint arXiv:1606.00915, 2016. `https://arxiv.org/abs/1606.00915`

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, *Focal Loss for Dense Object Detection*, arXiv preprint arXiv:1708.02002, 2017. `https://arxiv.org/abs/1708.02002`

[7] Z. Yuan, S. Wu, S. Zhang, J. Liu, X. Bai, and J. Li, *Dice Loss for Data-imbalanced NLP Tasks*, arXiv preprint arXiv:1911.02855, 2019. `https://arxiv.org/abs/1911.02855`

[8] N. Becker, *Understanding Binary Cross-Entropy / Log Loss: A Visual Explanation*, Towards Data Science, 2018. `https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a`

[9] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, *Class-Balanced Loss Based on Effective Number of Samples*, arXiv preprint arXiv:1901.05555, 2019. `https://arxiv.org/abs/1901.05555`

[10] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, arXiv preprint arXiv:1506.01497, 2015. `https://arxiv.org/abs/1506.01497`

[11] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, *A Comprehensive Review on Ensemble Deep Learning: Opportunities and Challenges*, arXiv preprint arXiv:2001.06554, 2020. `https://arxiv.org/abs/2001.06554`