Final Project Report: Airbus Ship Detection Challenge

Sherali Ozodov
Instructor: Jason Pacheco
CSC 480 - Principles of Machine Learning

May 1, 2025

Contents

1 Introduction

This report presents the results of my final project for CSC 480: applying deep learning models to solve the Airbus Ship Detection Challenge on Kaggle. The goal of the challenge is to detect ships in satellite images by predicting binary segmentation masks. The output masks are evaluated using the F2 Score calculated over multiple Intersection over Union (IoU) thresholds. The challenge has significant real-world applications, including maritime surveillance, piracy detection, and environmental monitoring.

2 Background and Dataset

The dataset consists of over 190,000 high-resolution satellite images (768x768 pixels) with associated run-length encoded (RLE) segmentation masks. Some images contain multiple ships, while others contain none, making class imbalance a notable issue.

The ground truth is provided in RLE format, which efficiently encodes binary masks. Images were decoded, resized to 384x384 for memory efficiency, and serialized into TFRecord format using a separate script (`airbus-ship-detection-tfrecords.ipynb`). This conversion allowed for faster I/O during model training and validation in the main notebook (`airbus-ship-detection-challenge.ipynb`).

3 Problem Formulation and Approach

This task is formulated as a binary semantic segmentation problem: classifying each pixel as either ship or background. The models were trained using TensorFlow with custom loss functions, data augmentations, and mixed precision for performance.

3.1 Data Pipeline

- **TFRecords:** Preprocessed and stored data in TFRecord format with GZIP compression.
- **Augmentations:** Applied Albumentations transforms including flips, fog, brightness contrast, and random cropping.

3.2 Model Architectures

Three deep learning architectures were implemented:

- **ResUNet:** Introduced residual blocks for stable training.
- **U-Net++:** Utilized nested skip connections for feature fusion.
- **Attention U-Net:** Integrated attention gates to improve focus on relevant regions.

All models used the Adam optimizer, a composite Focal Tversky + Dice loss, and validation metrics including Accuracy, IoU, and Dice Coefficient.

4 Novel Contributions

1. **TFRecord Pipeline:** Created TFRecords for efficient data loading, enabling faster training compared to loading raw images.
2. **Custom Ensemble:** Combined predictions using a weighted average of ResUNet (50%), Attention U-Net (30%), and U-Net++ (20%).
3. **Custom Loss:** Used a novel combo loss (Focal Tversky + Dice) to address class imbalance and improve segmentation sharpness.

5 Ablation Study

To validate the effectiveness of each component, I performed the following comparisons:

- Using cross-entropy instead of combo loss resulted in lower Dice and F2 scores.
- Disabling data augmentations caused overfitting and poor generalization on the validation set.
- Individual models underperformed compared to the ensemble. The ensemble resulted in smoother and more accurate masks.

6 Evaluation and Results

6.1 Validation Metrics

- ResUNet: Dice ~ 0.64, IoU ~ 0.50
- U-Net++: Dice ~ 0.66, IoU ~ 0.52
- Attention U-Net: Dice ~ 0.63, IoU ~ 0.49

6.2 Leaderboard Scores (Kaggle)

Model	**Public LB**	**Private LB**
ResUNet	0.517	**0.755**
U-Net++	0.477	0.715
Attention U-Net	0.505	0.728
Ensemble (Weighted)	0.51299	**0.75679**

The ensemble strategy outperformed all individual models on the private leaderboard.

6.3 Final Submission

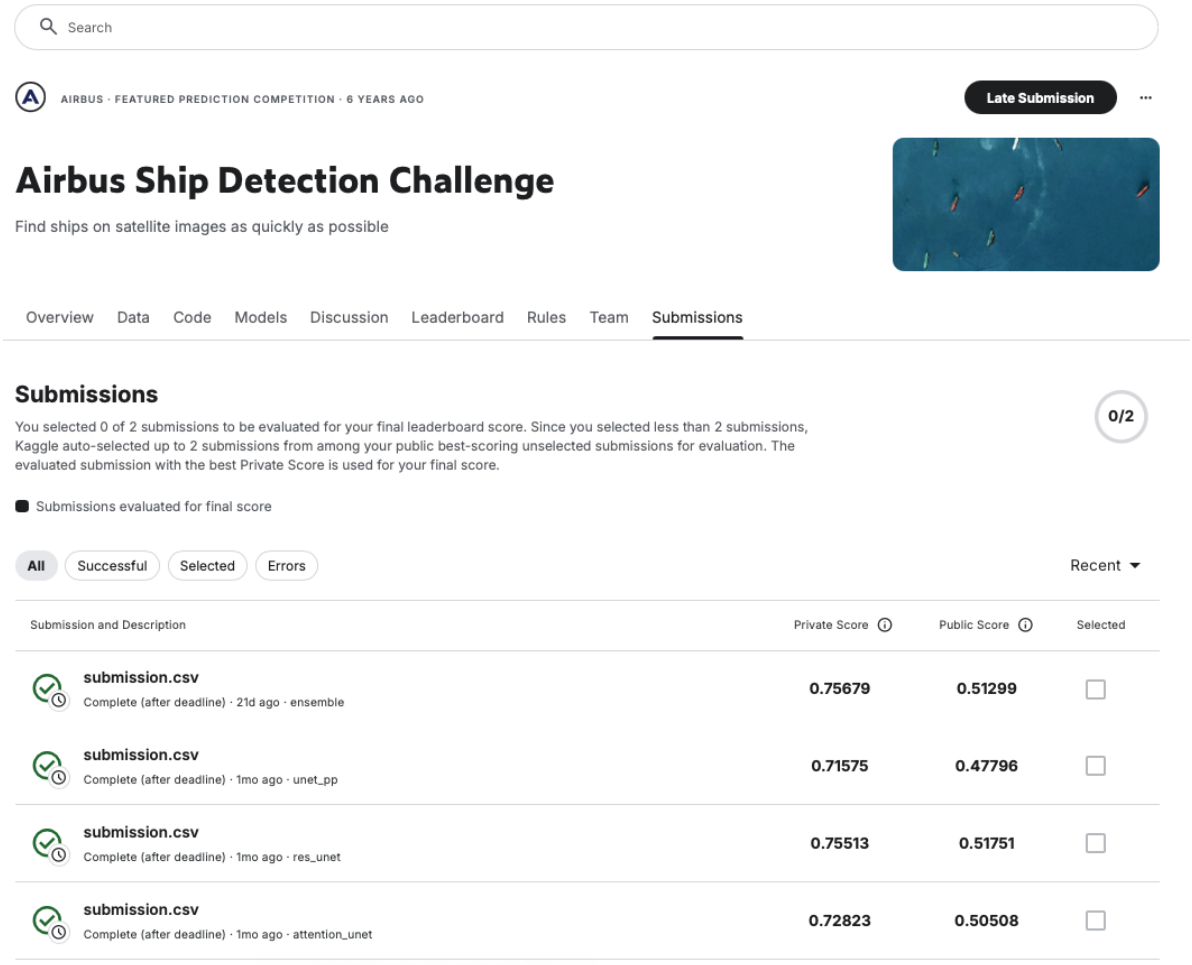

Figure 1: Kaggle submission results showing public and private scores for all models. The ensemble model achieved the highest private score of 0.75679, outperforming individual models.

7 Future Work

- Experiment with different image resolutions and input sizes to explore the impact on model performance.
- Try training with different data augmentation strategies to further improve generalization.
- Perform error analysis to better understand where models fail and adjust preprocessing or model parameters accordingly.
- Explore training with additional regularization techniques to reduce overfitting.

8 Project Repository

All source code, TFRecord scripts, notebook files, and proposal document are available at:
https://github.com/sheraliozodov77/airbus-ship-detection-cnn

9 Conclusion

This project demonstrated the successful application of deep learning to satellite image segmentation using TFRecords, novel loss functions, and ensemble strategies. The best private leaderboard score achieved was **0.75679**, highlighting the effectiveness of combining models with complementary strengths.

10 References

inversion, Jeff Faudi, and Martin. Airbus Ship Detection Challenge. 2018. Kaggle.
https://www.kaggle.com/competitions/airbus-ship-detection