Ships in Satellite Imagery

Yash Tanaji Jadhav
1002074308
Arlington, Texas
ytj4308@mavs.uta.edu

***Abstract*—In this project I developed a deep-learning-based system using the Faster R-CNN model to classify ships in satellite imagery data. The dataset used for this project is a diverse dataset from Kaggle, Ships in Satellite Imagery. The proposed model aims for high accuracy and robust performance. Key deliverables for this project include a trained model, comprehensive evaluation metrics, contributing significantly to maritime surveillance and remote sensing.)**

***Keywords*—Faster R-CNN, deep-learning, accuracy, performance**

I. Introduction

In the field of computer vision and remote sensing, the automated classification and detection of objects in images taken by satellites has garnered significant interest because of their vast applications in maritime surveillance, environmental monitoring and security. In this domain a major challenge is the task of accurately detecting ships in the vast and dynamic maritime landscape captured in satellite images. The need to discern small, often indistinct objects within vast oceanic backgrounds, under varying lighting and weather conditions, accounts for the complexity of this task.

This project addresses this challenge by employing the Faster Region-based Convolutional Neural Network (Faster R-CNN) algorithm. Faster R-CNN is one best object detection technology due it's efficiency and ability to accurately detect object, it distinguishes itself from other models with an integrated region proposal network that share full-image convolutional features with the detection network, which enables nearly cost-free region proposals which makes the process faster and more scalable.

In order to accurately detect and classify different types of ships in satellite images, this implementation harnesses the power of Faster R-CNN's deep learning capabilities. Using this advanced algorithm, combined with a comprehensive dataset that includes the Kaggle Ships in Satellite Imagery dataset will help us achieve our goal of, creating a robust model which will be highly precise with a good recall score for detecting ships.

Through this initiative, we aim to contribute to the advancements in the field of satellite image analysis and provide valuable tools for maritime surveillance, aiding in better navigation safety, environmental protection, and maritime domain awareness.

II. Related work

The paper Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks introduces the Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, enabling nearly cost-free region proposals. State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations [1]. In this paper they have demonstrated the effectiveness of Faster R-CNN in achieving state-of-the-art object detection accuracy on various datasets while maintaining practical speed for real-time applications. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position [1]. The proposed method alternates between fine-tuning for region proposal and object detection tasks, leading to shared convolutional features and improved efficiency. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection [1]. The method proposed in this paper was able to significantly surpass the accuracy of previous approaches and also successfully reduced the computational burden, particularly in test-time proposal generation.

Another paper which I referred to was LMO-YOLO: A Ship Detection Model for Low-Resolution Optical Satellite Imagery. This paper presents an innovative approach to address the problem of high false detection rates in low-resolution optical satellite images for ship detection. The solution this paper proposes is based on three key strategies. First a multiple linear rescaling scheme is developed to quantize the original satellite images into 8-b images [2]. This helps in retaining more detailed information in the images, crucial for identifying ships in low-resolution imagery. Second dilated convolutions are included in a YOLO network to extract object features and object-background features [2] . This enhancement is used to extract not just the features of the objects (ships in this case) but also the features of the surrounding background, and by doing so the model gains a better understanding of the imagery, enabling it to better differentiate between ships and ship-like objects or noise in the background, such as scattered clouds. Finally, an adaptive weighting scheme is designed to balance the loss between easy-to-detect ships and hard-to-detect ships [2]. This is done to balance the detection loss between ships that are easily detected and those that are more challenging, due to factors like size, orientation, or environmental conditions. The proposed method was validated by level 1 product images captured by the wide-field-of-view sensor on the Gao Fen-1 satellite. The experimental results demonstrated that our method accurately detected ships from low-resolution images and outperformed state-of-the-art methods [2].

The paper "Precise Ship Location with CNN Filter Selection from Optical Aerial Images" addresses the challenge of efficiently detecting small maritime objects in aerial images by introducing a novel convolutional neural network (CNN)-based approach. The proposed method employs aerial images in the visible spectrum as inputs to train a categorical convolutional neural network for the classification of ships [3]. It is important for the network to learn the characteristics of ships, which will assist the model in classifying them accurately. We will proceed to fit the filter selection algorithm, once the Convolutional Neural Network has been trained. This process basically consisted of selecting the most relevant filters from this network for the location of a target class c [3]. By focusing on these specific filters, the method improves precision in locating the target class. In the inference stage...an input sample is forwarded through the trained model and if the prediction for any of the target classes is positive (in our case if a ship is detected) the feature maps of the network for that class are used to obtain its precise localization [3]. The method will utilize the feature maps from the selected filters to accurately pinpoint the ship in the image. Calculation of the gradients from the selected filters is involved in this step, which enables the model to precisely locate the location of ships. This method is particularly advantageous because it requires labeling only a few images with bounding boxes for training, which is significantly less labor-intensive than traditional methods.

III. DATASET

Google Drive link for dataset: https://drive.google.com/drive/folders/1gxeVZND0BZE_KabRlPHkIDwwJeRtDtVk?usp=sharing

The dataset in focus is derived from Planet's satellite imagery, specifically over California's San Francisco Bay and San Pedro Bay regions. It comprises 4,000 RGB images, each measuring 80x80 pixels. These images are annotated to indicate the presence or absence of ships, categorizing them into "ship" and "no-ship" classes [5].

This dataset is available in a JSON file named shipsnet.json. The structure of this file includes several key elements:

- data: The image data.
- label: This field is binary, with a value of 1 indicating the presence of a ship ("ship" class) and 0 indicating the absence ("no-ship" class).
- scene_ids: These are unique identifiers corresponding to the specific PlanetScope visual scene from which each image was sourced. Users can leverage these scene ids to locate and retrieve the full scene via the Planet API.
- longitude_latitude: This contains the geographic coordinates of each image's central point, formatted as longitude and latitude values separated by an underscore.

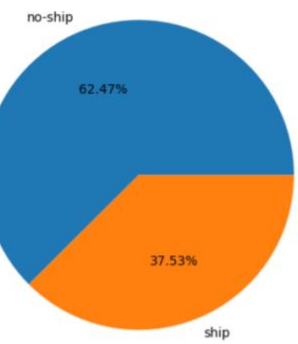

Fig 0.1: Pie-chat for distribution of images based on class.

Splitting of data:

Training – 70%
Validation – 20%
Testing – 10%

IV. IMPLEMENTATION

The implementation I have come up with consists of a Faster R-CNN model which consists of convolution layers, activation layer and max pooling layers. Parameters like number of filters, kernel size, strides to be taken can be saved, which allows us to make the model without having to repeatedly running same code many times. Once a predetermined number of convolutional layers are applied, the model introduces a flatten layer. This layer is crucial as it transforms the feature map output from the convolutional layers into a linear array, setting the stage for classification.

In the context of image detection using Faster R-CNN, the model diverges from this approach after the final convolutional layer. Instead of progressing to the flatten and dense layers, it utilizes the feature map directly from the last convolutional block, and feds it to a Region Proposal Network (RPN). The RPN is built on a basic CNN.

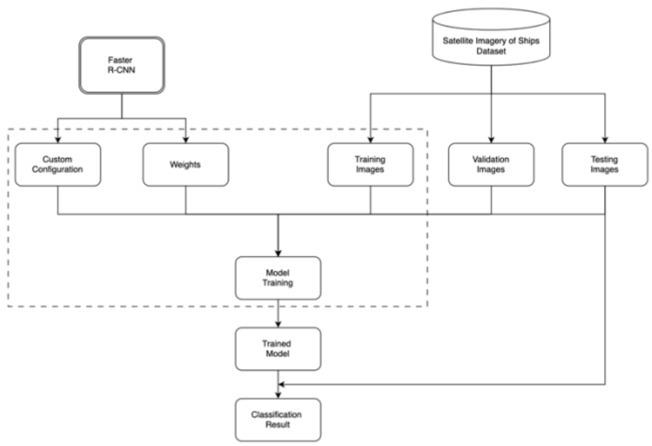

Fig 1.1: System architecture of implementation

Model Architecture:

Block 1:

An input layer is initialized using the Input Keras layer, this defines the number of neurons present in the input layer. ZeroPadding is applied to the input image, so that boundary features are not lost.

Block 2:

First Convolutioanl Layer, it starts with 16 filters and kernel size with (3,3) and strides (2,2). Padding is maintained same, so the image does not change spatially, until the next block in which MaxPooling occurs.

Block 3 – 4:

Similar structure in both with a convolutional layer followed by a MaxPooling and Dropout layers.

Output Block:

The feature map produced by the previous convolutional layers is converted into a single column using Flatten Layer and the classified using a Dense layer (output layer) with the number of classes present in the dataset, and sigmoid as activation function.

V. Results

Training the model involves a process spanning 50 epochs, each comprising of 16 samples per batch. During the training phase, the optimal weights determined for the model are saved in a file.
During the training phase the highest accuracy we obtain is of 98.295%.

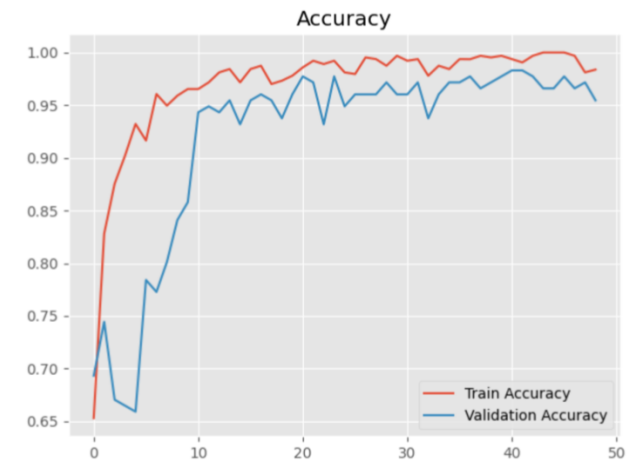

Fig 2.1: Accuracy for the training and validation datasets

In the accuracy plot we can observe a rapid increase in training accuracy within the initial epochs which eventually platues near the peak. The validation accuracy similarly increases and appears to closely follow the training accuracy, albeit with slight oscillations. Both accuracy curves begin to stabilize as the epochs increase, converging towards a high accuracy level, which suggests that the model is achieving a good fit to both the training and unseen validation data.

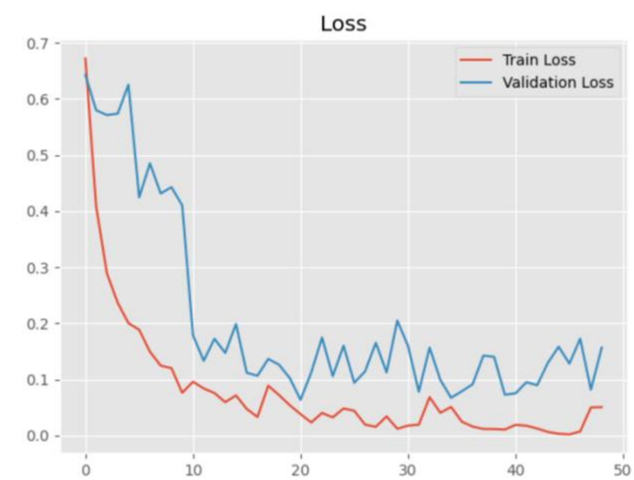

Fig 2.2: Loss for the training and validation datasets

In the loss plot, we observe that the training loss starts relatively high but decreases sharply and consistently as the epochs progress, indicating that the model is effectively learning from the training data. The validation loss also decreases, but with some fluctuations, suggesting the model is generalizing to new data with varying degrees of success across epochs.

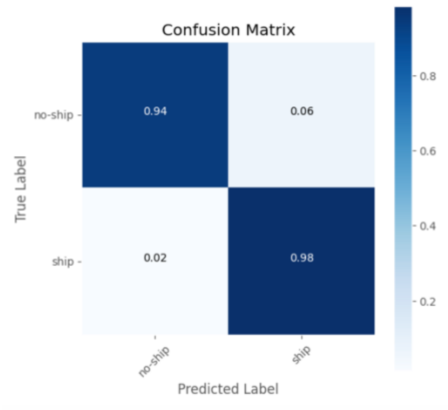

Fig 2.3: Confusion Matrix for Validation dataset

Precision:

- No-ship = 0.94
- Ship = 0.98

Recall:

- No-ship = 0.98
- Ship = 0.94

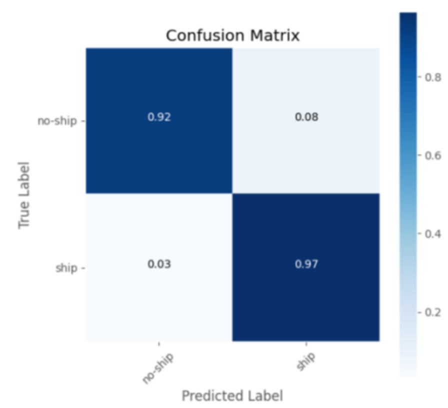

Fig 2.4: Confusion Matrix for Testing Dataset

Precision:

- No-ship = 0.92
- Ship = 0.97

Recall:

- No-ship = 0.9684
- Ship = 0.9238

Testing our model on random images from the dataset:

Fig 2.5: Model testing results

VI. Conclusion

To conclude, in this project I have effectively implemented a deep learning model utilizing Faster R-CNN for ship detection in satellite imagery, achieving notable accuracy and efficiency. The optimized filter selection strategy and minimal labelling requirement underscore the model's potential for scalable maritime surveillance applications. The training outcomes point towards a significant step forward in satellite-based monitoring systems.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Proc. 28th Int. Conf. Neural Inf. Process. Syst. - Vol. 1, 2015, pp. 91–99.I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[2] Q. Xu, Y. Li and Z. Shi, "LMO-YOLO: A Ship Detection Model for Low-Resolution Optical Satellite Imagery," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 4117-4131, 2022, doi: 10.1109/JSTARS.2022.3176141.

[3] S. Alashhab, A. -J. Gallego, A. Pertusa and P. Gil, "Precise Ship Location With CNN Filter Selection From Optical Aerial Images," in IEEE Access, vol. 7, pp. 96567-96582, 2019, doi: 10.1109/ACCESS.2019.2929080.

[4] https://www.kaggle.com/code/harshranjan01/ship-detection-using-faster-r-cnn-part1

[5] https://www.kaggle.com/datasets/rhammell/ships-in-satellite-imagery/data