Ship Detection in satellite imagery using Machine Learning and Transfer Learning Approaches

Daniel Ekenedilichukwu Ezeani

Student Number: 202211563

Supervisor

Dr Lara Blythe

Reviewers

Dr Lara Blythe

December, 2023

Table of Contents

List of Figures

List of Tables

Abstract

This study proposes a multi-layer Convolutional Neural Network (CNN) model for ship detection using the Planet satellite imagery dataset. The architecture consists of input, classification and fully connected layers and employs various optimizers, model architectures and hyper parameters to achieve a high performance. The baseline model results and metrics are compared with four pretrained models (MobilenetV2, InceptionV3, Resnet50, VGG16) to ascertain overall model performance. Data preprocessing which involved data balancing, augmentation, conversion and resizing was carried out. Results reveal that the Stochastic Gradient Descent (SGD) model trained with a learning rate of 0.001 emerged as the best model amongst other models with an accuracy of 99.5% and F1 score of 99%. This implies that the model was able to accurately predict 99.5% of the two classes correctly while taking note of false positive and false negative predictions. The loss function plot indicates that the model is able generalize well to unseen data. Findings shows that the study's model is competitive amongst other state of the art models for detecting ships on the waterways.

1.0 Introduction

Maritime surveillance plays an important role in ensuring activities on the waterways are monitored and regulated. It helps to improve safety and security, regulate commercial activities, and control illegal activities. About 90% of traded goods around the world are transported on the waterways (Organization for Economic Co-operation and Development, n.d). The United Nations Convention on the Law of the Sea (n.d) report noted that all countries in the world are affected by threats taking place on the sea such as piracy, armed robbery, terrorism, and human trafficking. These activities have given rise to the need to develop machine learning algorithms and artificially intelligent systems that can be used to track, manage, and control shipping activities.

1.1 Background

1.1.1 Machine Learning

International Business Machine (n.d) defines machine learning as a branch of artificial intelligence that utilizes huge amounts of data and computer algorithms to imitate or simulate the way humans learn. These systems enable computers to learn from data and make decisions without relying on human intervention. Machine learning is divided into 3 broad areas namely: supervised learning, this branch of ML utilizes labeled dataset to make predictions, unsupervised learning, utilizes unlabeled dataset to understand patterns from the data, and re-enforcement learning, which develops models that utilize trial and error for predictions.

Another highly influential sub-field in machine learning is Deep learning. It is a field that utilizes computer systems to learn multiple and successive layers from data to make predictions (Chollet, 2018). One popular technique in deep learning is the use of Convolutional Neural Networks (CNN). CNNs belong to a class of neural networks that utilize input, hidden and fully connected layers for recognition of objects within the environment (International Business Machine, n.d).

1.1.2 Transfer Learning

Transfer learning is a machine technique that utilizes knowledge from tasks that have been trained on large amounts of data on a new set of tasks (Torrey & Shavlik, 2010). They are known as pretrained models and form a foundation or a base for working on a new set of problems. Popular pretrained models include: VGG16, VGG19, ResNet, MobileNet and InceptionNet amongst others.

In previous studies, Apoorva et al. (2021) developed a Convolutional Neural Network model for ship detection using the Planet satellite imagery dataset. The dataset contains 2800 color images with two classes: ship and no-ship. The training process involved the use of cross validation where the dataset was split into different subsets and trained over 12 iterations (epochs). Results reveal that the model possessed an accuracy of 0.94 (94%). This result was compared with the scores of other classifiers such as XG-Boost XGB-(0.95); Random Forest-0.93 (93%) and K-Nearest Neighbour- 0.92 (92%). The study failed to properly outline the data preprocessing tasks carried out such as data augmentation and normalization before proceeding to model building.

Richa and Joshi (2019) utilized a CNN transfer learning technique known as the Resnet50 algorithm on the Planet satellite imagery dataset. Restnet50 is a 50-layer deep learning CNN architecture that introduces residual connections, allowing the model to learn effectively. To prepare the model for training, data augmentation was carried out using image data bunch. The model was trained over 2 cycles. The first involved freezing the weights of the pretrained model;

this led to an error rate of 1%, the second involved unfreezing the weights of the model. This resulted in an error rate of 0.5% and an accuracy of 99.5%.

Marfu'ah and Kurniawardhani (2020) performed a comparative analysis between Support Vector Machine (SVM) and CNN models for ship detection using Planet satellite imagery dataset. The dataset was split into 80% training and 20% validation respectively. Results reveal that the CNN model outperformed the SVM model with an accuracy of 98.25% compared to 94.4% for the SVM. However, the CNN model required more time to train.

Liu et al. (2021) adopted the use of an enhanced convolutional neural network (eYOLOv3) for accurately detecting ships on waterways. The model architecture uses the YOLOv3 framework with some changes to improve performance. A unique data augmentation technique was utilized to enlarge the dataset and boost model performance. The eYOLOv3-608 model outperformed other models in identifying five ship categories, achieving a mean average precision (mAP) of 87.74%.

Tanveer et al. (2019), utilized the Faster R-CNN VGG16 hybrid model for accurately detecting ships on Sentinel 1 SAR satellite images. The model was deployed to detect and count the number of ships in Shanghai port. Data was preprocessed using back-geocoding, averaging, binary conversion and image morphology. The results reveal that there was a slight difference between the actual labels and the predicted labels.

Shao et al. (2018) introduced the SeaShips dataset for detecting ships on large datasets. The dataset covers various types of ship and image qualities. Data was captured using 156 cameras placed at different locations in China between the year 2017 and 2018. Modifications to the images were performed by using three different image scales; images were obtained under different lighting conditions, viewpoints, and backgrounds. The model evaluated three detectors with Faster (ResNet101) achieving a (mAP) of 0.92, demonstrating superior performance above other models.

Irfan and Karim (2022) utilized a Mobilenet pretrained Convolution Neural Network model for detecting ships using a multiclass (4?) ship dataset. The dataset consists of 9000 images, 80% of which was set aside for training and 20% for testing. A remote operated underwater vehicle (ROV) was used to capture images of the different ship categories. The model was evaluated on the ship dataset from different angles and positions, covering all four ship categories. Results reveal that the cargo cruise did slightly better than other ship categories with an accuracy of 39.92% when taken on a rectangular box angle of the ship, it performed reasonably well on cargo ship when captured from the top with an accuracy of 89.49%. Furthermore, the model did well on container ship when captured from the side with an accuracy of 77.57%, the model predicted cruise ship better than other classes when captured from the deck of the ship with an accuracy of 43.44%.

Most of these studies failed to address ethical issues like bias that could arise from imbalance in the dataset. Specifically, the works of Apoorva et al. (2021), Richa and Joshi (2019) and Marfu'ah and Kurniawardhani (2020) had this shortcoming. Also, a good number of these reviewed works failed to employ other conventional evaluation metrics like the precision, recall and F1-score. These metrics measures the robustness and significance of the models result in identifying false positive and negative predictions. These are some of the identified gaps that this study seeks to address.

This study seeks to evaluate if the development of a Customized Convolutional Neural Network model is robust in detecting ships in waterways using satellite images and seeks to achieve the following specific objectives:

1. To build a high-performance customized Convolutional Neural Network model that can accurately detect ships in waterways using satellite images.
2. To examine the impact of utilizing different model architectures, optimizers and hyperparameters on model performance.
3. To conduct a comparative analysis between the Custom CNN models with a few selected pretrained models.
4. To Examine model performance with respect to wider theory

2. Methodology

In this section, the primary objective is to outline the techniques and approaches employed to address the complexities of detecting ships using satellite images. The methodology will involve utilizing and harnessing the power of machine learning and deep learning approaches such as Convolution Neural networks and transfer learning to effectively detect ships using satellite images. The methodology will take the approach outlined in Figure 2.0.

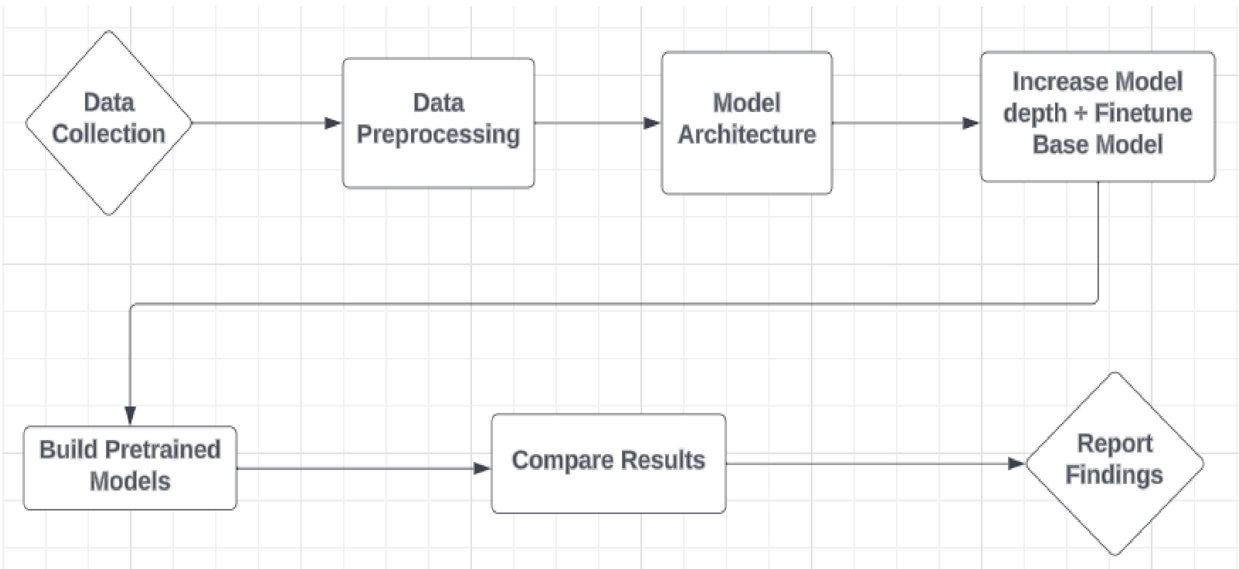

Figure 2.0: Methodology Workflow
Tool: (Lucidchart, https://lucid.app/lucidchart/)

2.1 Data Collection

The dataset comprises of images retrieved from Planet satellite images from the cities of San Francisco and San Pedro, in the State of California. It is available as an open-source dataset on Kaggle (Kaggle, n.d). It comprises 4,000 RGB (Red, Green, Blue) images divided into two classes. There are 3,000 images labeled as "0," representing the "No ship" class, and 1,000 images labeled as "1," representing the "Ship" class. The dataset is available in the form of a JSON file with 4 columns namely: data, labels, scene IDs, and locations. For the purposes of this study, only data and label columns will be used for analysis.

2.2 Data Preprocessing

This study will employ the flow in (Figure 2.2.1) for data preprocessing.

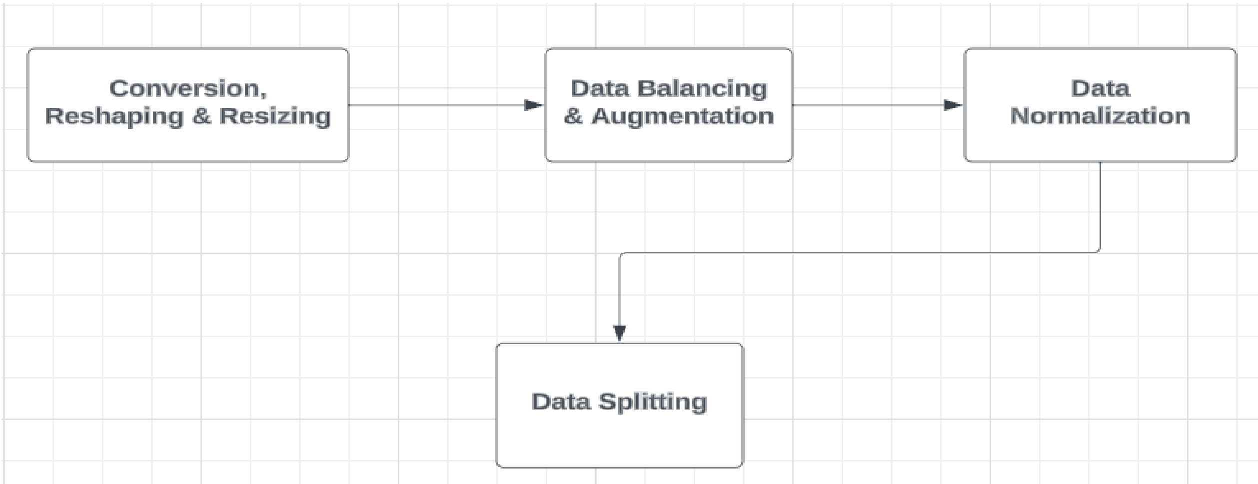

Figure 2.2.1: Data Preprocessing Workflow
Tool: (Lucidchart, https://lucid.app/lucidchart/)

The first step of this process involves flattening and converting the data into numpy arrays, reshaping and transposing the image to the original input image of (4000, 80, 80, 3). Where 4000 represents the number of images, (80, 80, 3) represents the original input dimension of height, width and 3 colour channels (Red, Green, Blue). A further resizing was carried out from dimension (80, 80, 3) to (224, 224, 3) to enable a balanced and fair comparison between the Custom CNN model and the pretrained models the pretrained models selected are trained with the predefined ImageNet dataset designed to take images with input dimensions 224 x 224 x 3 (Xie, et. al, 2020).

The second step involved applying data augmentation and data balancing. For an imbalanced dataset, machine learning models tend to be biased and favour the majority class but fail to recognize important features from the underrepresented class (Satiawan, et.al, 2021). As a corrective measure, the Oversampling technique has been applied because Undersampling technique has been observed to greatly reduce the performance of the model (Hasanin & Khoshgoftaar, 2018). The barplot below (Figure 2.2.2a) shows the class distribution of the original dataset.

a. Unbalanced dataset b. Balanced Dataset

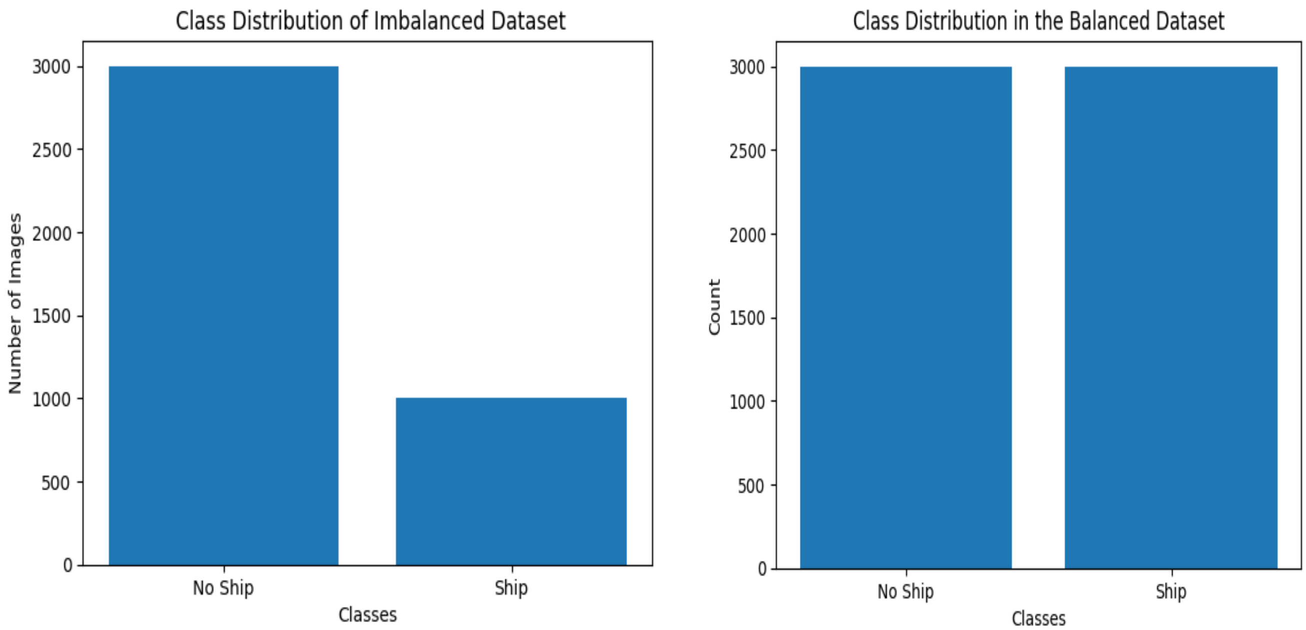

Figure 2.2.2: Class Distribution of Imbalanced Dataset and Balanced Dataset

Data augmentation and balancing was achieved by defining a function for ImageDataGenerator (Keras Documentation, n.d). It involves applying transformations to the image by flipping, shearing and zooming. The second part takes the original images belonging to the lesser class "Ship Class" and the desired number of new images to be generated, it loops through the original images of the class and appends it to a new list. Finally, the list of 2000 new images are added to the 'Ship' class. This will make up a total of 3000 images per class. See distribution of the new dataset in figure 2.2.2b.

The next step under preprocessing was to normalize the dataset. This is achieved by dividing the training, test and validation set to ensure the pixel values range between zero and one.

The final preprocessing step involves splitting the dataset into training, testing and validation sets. A split of 80%, 10%, 10% is applied for training, testing and validation sets respectively. This split ratio is a good and popular technique which has been used by the works of Torén (2020), Kim, D.H. and MacKinnon (2018) and (Pandey et al., 2021).

2.3 Model Architecture

This study has adopted a multilayer Convolutional Neural Network model for classification and detection of ships using satellite images. CNN has been selected for this image classification task because it is a technique widely used to develop deep learning algorithms that can learn spatial information from the image data (Girdhar et al., 2022).

The architecture is broadly divided into 3 layers namely the input layer, feature learning or hidden layer and the classification or output layer. See below a simple architecture design.

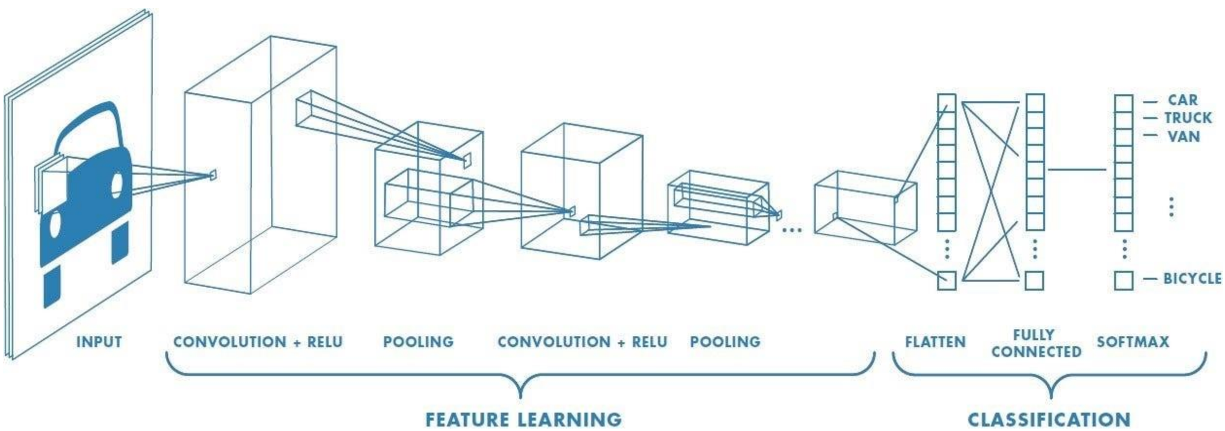

Figure 2.3.1- (Saha S., 2018): Generic model architecture for a CNN model

2.3.1 Input Layer: The model takes the image as pixel of values with the dimension (h x w x c), where h = height, w = width, c = colour channels. The original input of the dataset comes with a shape of (80, 80, 3).

2.3.2 Feature Learning: This layer is also called the feature extraction layer. It comprises of the convolution, activation function (Relu) and pooling layers. For this study, a filter value of 64 has been used; and kernel size of (4, 4); these hyperparameters enable the model to learn different features as it moves around the image from top left to bottom right. This process is known as the convolution operation. Table 2.3.1 (below) reflects this.

Mathematically, it is expressed as the equation 1 below

Z = W $*$ X + W_0-- Equation 1

Where z = activation map, W = weights, X = inputs, W_0 = bias

1	0	0	1
2	1	0	0
0	0	1	0
-1	2	1	1

1	0	0	1
2		2	0
0	0	1	0
-1	2	1	1

1	0	0	1
2	1	1	0
0	0	1	0
-1	2	1	1

1	0	0	1
1	0	2	0
0	0	1	0
-1	2	1	1

Table 2.3.2: Convolution operation in a CNN Model. The highlighted portion in yellow shows how the filter moves from the top of the image to the bottom right to learn features from the image.

A ReLu activation function is used. This makes the model non-linear and computationally efficient, turning all negative values to zero. The pooling layer makes the input data smaller by keeping only the most important information. MaxPooling has been adopted with a pool size of (5, 5).

2.3.3 Classification layer: This is also known as the output layer. This layer is responsible for transforming the input to output. It consists of a dense layer of 128 neurons and ReLU activation function that ensures the model is efficient. A final dense layer is utilized for converting the predicted classes ('Ship', 'No ship') into probabilities by employing the Softmax activation function.

2.3.4: Regularization Methods: To manage overfitting, two measures have been employed. The first is to introduce dropout, 0.5 has been selected. This ensures that 50% of the inputs are randomly set to zero to prevent the model from heavily relying on certain units (i.e neurons) of the model. The second measure adopted is the introduction of early stopping, this measure prevents the model from training further when no improvements are observed after several iterations. For this study, a value of 10 has been selected.

2.3.5 Performance Evaluation: The following results in table 2.3.5 below will be used to evaluate model performance: Considering that the dataset has been balanced, accuracy has been selected as a metric to evaluate overall model performance. Also, F1-score strike the balance between precision and recall. It has also been selected to account for false positive and negative predictions. See table 2.3.5 below

Table 2.3.5: Performance Evaluation Metrics

Metric	Formula	Description
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	This metric is used to evaluate the overall performance of the model. It is the ratio of the total number of correct predictions to the total number of predictions. TP = True positive predictions, TN = True negative predictions, FP = False positive, FN = False negative predictions
Precision	$\frac{TP}{(TP+FP)}$	This metric measures the model’s correct positive predictions with respect to the total number of positive predictions. TP = True positive predictions, FP = False positive predictions
Recall	$\frac{TP}{(TP+FN)}$	This metric measures the model's ability to find all the correct positive predictions. It is the ratio of correct positive predictions to the total number of actual positive predictions. TP = True positive predictions, FP = actual positive predictions
F1 score	$2\left(\frac{\Pr e\,cision\ \times Recall}{\Pr e\,cision\ +\ Recall}\right)$	This is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives. [1]

[1] https://www.statology.org/sklearn-classification-report/.

2.4 Computer Specification

The experiments were carried out using Dell Core i9-12900 CPU, with processing speed of 2.40 GHz, 32 Gigabyte RAM, and an NVIDIA GeForce RTX 3090 GPU.

3.0 Presentation of Results:

Table 3.0.1 and 3.0.2 below shows the results of the best performing customized convolutional neural network models and pretrained models. The top performing models are bracketed with asterisk.

Table 3.0.1: Results of top custom CNN model architectures and pretrained models

Model	Weight layers	Optimizer	Learning Rate	Val accuracy	Val loss	Training Time
Custom CNN Models						
Balanced_Block3 Model	5	Adam	0.001	0.99	0.0541	15 mins
Balanced_Aug_model	5	Adam	0.001	0.99	0.0439	15 mins
Bal_block4_model	6	Adam	0.001	0.99	0.0439	20 mins
block5_model	7	Adam	0.001	0.99	0.0397	78mins
balanced_sgd_model2 (*)	5	SGD	0.001	0.98	0.0671	15 mins
balanced_model_rmsprop	5	RMSProp	0.001	0.49	0.6941	8 mins
Batch64_model	5	Adam	0.001	0.98	0.0502	14 mins
model3_sgd	5	SGD	0.0001	0.95	0.1422	20 mins
Pretrained Models						
MobilenetV2 (*)	3	Adam	0.001	0.99	0.0479	13 mins
Resnet50 Model	50	Adam	0.001	0.88	0.3241	40 mins
VGG16 Model	16	Adam	0.001	0.94	0.1732	77 mins
InceptionV3 Model	48	Adam	0.001	0.98	0.0778	23 mins

Table: 3.0.2: Performance Metrics of various top custom CNN and pretrained models

Model	Accuracy	Precision	Recall	F1-score	Test loss
Custom CNN Models					
Balanced_Block3 Model	0.98	0.98	0.98	0.98	0.0653
Balanced_Aug_model	0.98	0.98	0.99	0.98	0.0529
Bal_block4_model	0.98	0.98	0.98	0.98	0.0679
block5_model	0.99	0.99	0.99	0.99	0.0391
balanced_sgd_model2_model (*)	0.995	1.00	0.99	0.99	0.0244
balanced_model_rmsprop	0.48	0.24	0.50	0.33	0.6945
Batch64_model	0.98	0.98	0.98	0.98	0.0536
model3_sgd	0.95	0.95	0.95	0.95	0.1221
Lr4_adam_model	0.98	0.98	0.98	0.98	0.0538
Pretrained Models					
MobilenetV2 (*)	0.99	0.99	0.99	0.99	0.0282
Resnet50 Model	0.89	0.89	0.88	0.88	0.3206
VGG16 Model	0.95	0.95	0.95	0.95	0.1353
InceptionV3 Model	0.98	0.98	0.98	0.98	0.0459
Ensemble (Custom CNN, Mobilenet v2)	0.99.5	-	-	-	-

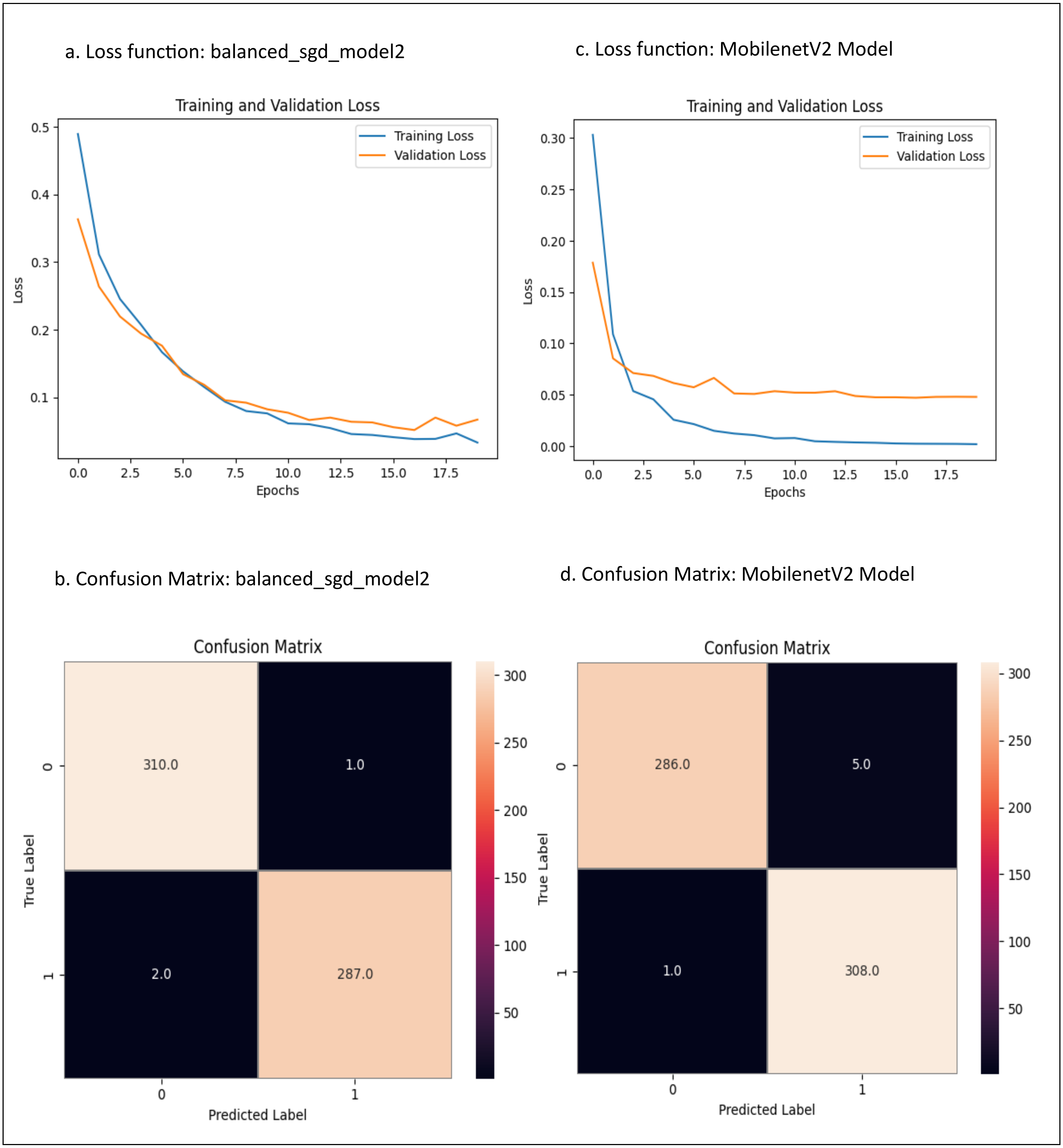

Figure 3.0.1: Classification Report and Confusion Matrix of Top 2 Custom Models. Figure 3.0.1a and 3.0.1b model is trained with SGD Optimizer and default learning rate of 0.001. Figure 3.0.1c and Figure 3.0.1d is trained with SGD optimizer and learning rate of 0.0001.

4.0 Discussion

Table 3.0.1 and table 3.0.2 above shows all the results of the top performing custom CNN models and a few selected pretrained models. All the models performed well except the model trained with RMSprop optimizer which possessed a poor accuracy of 0.48 (48%).

Among all the other custom models, the model (with asterisk) trained with 5 weight layers, SGD optimizer and default learning rate of 0.001 outperformed all other models. With an accuracy of 0.995 (99.5%) and F1-score of (0.99) 99%, the model was to classify 99.5% of the classes correctly while considering the false positive and false negative predictions. The model was able to predict 310 instances of the "No ship class" (0) correctly, misclassifying only 2 instances as belonging to the ship class (1). Furthermore, 287 instances of the "Ship class" was predicted as correct instances while a single instance was misclassified as belonging to the "No ship" class. The model was able to generalize well with very minimal overfitting compared to other models, see figure 3.0.1a above. Evaluating the model's performance on unseen data (test set), the model possessed the lowest test loss of 0.0244 amongst other models. The test loss measures the difference between the predicted labels and the actual labels on the test set, the lower the test loss value the better the performance of the model. Furthermore, the model was able to strike a balance between achieving high accuracy with lesser training time. Obtaining a high accurate and fast model is essential for any maritime surveillance system (Alghazo et al., 2021).

On the other hand, the Mobilenet v2 pretrained model outperformed other pretrained models with an accuracy and F1 score of 0.99 (99%), misclassifying 5 instances of "No ship" class and 1 instance of the "Ship" class. However, there exists a wide margin between the training loss and validation loss which is a clear sign of overfitting. This implies that the model may not perform optimally on unseen data.

The ensemble method which is a hybrid of the custom CNN and Mobilenet v2 models. It produced a similar and high accuracy of 99.5%. Constructing ensemble models is a technique used to boost the accuracies of poor performing models (Ngo et al., 2021). As such, there exist no justification for adopting this approach as the non-hybrid models in figure 3.0.1 were observed to be performing very well.

The custom CNN (balanced_sgd_model2) highlighted with red asterisk has proven to be the best model among other custom model architectures and pretrained models. It has been adopted as the baseline model.

Table 4.0.1 below shows the how well this study's baseline CNN model performs in relation to other existing works. The works of Richa and Joshi (2019) and Apoorva et.al (2021) were all trained using the Planet satellite imagery dataset. The former, trained on the Resnet50 pretrained model with unfrozen layers produced an accuracy of 99.5%. However, loss function plots were not utilized to reflect how well the model is able to generalize to unseen data. On the other hand, the baseline model displayed very competitive accuracy of 99.5% and F1 score of 99%. This reflects the overall performance of the model and its ability to detect false positive and false negative predictions. The loss function plot displayed high convergence between the training loss and validation loss, thereby minimizing overfitting on the validation set. See figure 3.0.1a above.

Table 4.0.1: Comparative analysis of the baseline model with existing works

Model	Accuracy Score (%)	Precision	Recall	F1-score
Richa and Joshi (2019)	99.5%	-	-	-
Apoorva et. al (2021) (CNN)	(0.9357) 94%	0.7975(80%)	0.9692 (97%)	0.8763 (88%)
Apoorva et. al (2021) (XGB Classifier)	(0.946) 95%	-	-	-
Baseline model (balanced_sgd_model2)	0.995 (99%)	1.00	0.99	0.99

Apoorva et.al (2021) utilized a Convolutional Neural Network model. The results were compared to 8 standard classification models with XGB Classifier emerging as the best with an accuracy of 0.95(95%). Their model was able to generalize well as regards to overfitting. See below figure 4.0

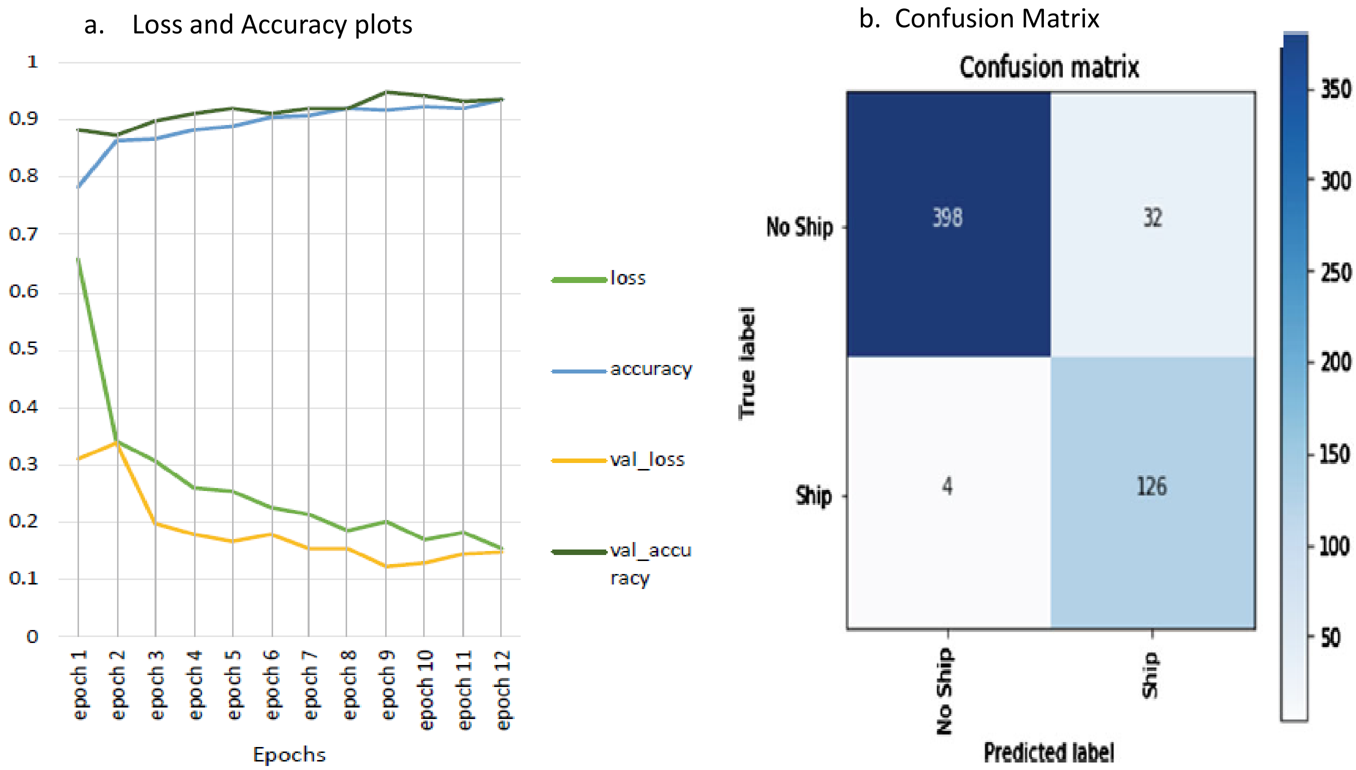

Figure 4.0.1- (Apoorva et. al, 2021): Loss and Accuracy plot and Confusion Matrix

Comparing the performance of the above model in figure 4.0.1 with this study's baseline model, the latter CNN model outperformed their model with an accuracy of 99.5% and F1 score of 99%. From the confusion matrix in figure 4.0.1 above, their CNN model misclassified a total of 36 instances from a total of 560 instances of both classes while a total of 3 instances were misclassified from a total of 600 samples of the test set of the baseline model. See figure 3.0.1b above. The implication of this finding is that the study's baseline model will be able to detect ship accurately and swiftly in moving waters with very minimal deviations from the true or actual values. The result obtained could be attributed to the unique data preprocessing strategy adopted which involved upward resizing of the image to account for balanced comparison between custom models and pretrained models, data balancing to eliminate bias from the data and data splitting strategy to ensure sufficient samples are allotted to training, validation and testing.

While the current findings demonstrate robustness, other alternative machine learning and transfer learning models, varied split ratios and techniques (i.e., cross-validation) can be applied to ensure a comprehensive assessment of model performance.

5.0 Conclusion

In summary, the Convolutional Neural Network model has proven to be one of the best techniques in automatically identifying ships using satellite images. Its ability to learn complex patterns and features from spatial data makes it a top choice for maritime surveillance. The study's baseline CNN model trained with SDG optimizer emerged as the best among other model architectures trained with different optimizers and transfer learning techniques. When deployed live, the study's baseline model would be able to autonomously detect ships very quickly and accurately, leading to enhanced operational efficiency, improved search and rescue missions, environmental monitoring and advancement of law enforcement and border control activities.

5.0.1 Recommendation

The most notable recommendation proposed by this study is that model builders should adopt a solid and unbiased data preprocessing strategy by ensuring ethical issues (i.e., bias) are treated before proceeding to model building as the quality and outcome of results is largely dependent on this.

5.0.2 Future Works

The performance of any machine learning model is highly dependent on the quantity and quality of the data fed into the model. In the future, researchers could benefit from applying advanced data generation techniques such as Generative Adversarial Networks and Variational Auto Encoders to train their models on new set of images.

References

Alghazo, A., Bashar, G., Latif and M. Zikria (2021). Maritime Ship Detection using Convolutional Neural Networks from Satellite Images. *10th IEEE International Conference on Communication Systems and Network Technologies (CSNT).* Bhopal, India. pp. 432-437, doi: 10.1109/CSNT51715.2021.9509628.

Apoorva, A., Mishra, G.K., Sahoo, R.R., Bhoi, S.K., and Mallick, C. (2021) 'Deep Learning-Based Ship Detection in Remote Sensing Imagery Using TensorFlow', in S. Patnaik, X.S. Yang, and I. Sethi (eds.). *Advances in Machine Learning and Computational Intelligence (Algorithms for Intelligent Systems),* Springer, Singapore. DOI: 10.1007/978-981-15-5243-4_14.

Chollet, F. (2018) *Deep Learning with Python*. New York: Manning Publications.

Girdhar, P., Johri, P., & Virmani, D. (2022). Deep Learning in Image Classification: Its Evolution, Methods, Challenges and Architectures. *In Advances in Distributed Computing and Machine Learning.* (Vol. 427, pp 381–392).

Hasanin, T., Khoshgoftaar, T.M. (2018). The Effects of Random Undersampling with Simulated Class Imbalance for Big Data. In Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA. pp. 70–79.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. arXiv:1905.02244v5

IBM. (n.d.). *Convolutional Neural Networks*. Available online: https://www.ibm.com/topics/convolutional-neural-networks

Irfan, F.A., Karim, M.M. (2022). Detection of Underwater Ships Categorically Using Convolutional Neural Network and Power Efficient Remotely Operated Vehicle. In Proceedings of MARTEC 2 The International Conference on Marine Technology, 21-22 December 2022, BUET, Dhaka.

Kaggle. (n.d.). *Ships in Satellite Imagery.* Available online: https://www.kaggle.com/datasets/rhammell/ships-in-satellite-imagery

Keras Documentation. (n.d) Keras.io. Available online: https://keras.io/ja/preprocessing/image/ [Accessed 23 Nov. 2023].

Kim, D.H. and MacKinnon, T. (2018) Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. *Clinical Radiology,* 73(5), 439–445.

Liu, R. W., Yuan, W., Chen, X., & Lu, Y. (2021). An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system. Ocean Engineering, 235, 109435.

Lucidchart. (2023). Lucidchart - Online Diagramming Tool. https://lucid.app/lucidchart/

Ngo, N.-T., Pham, A.-D., Truong, T.T.H., Truong, N.-S., Huynh, N.-T. & Pham, T.M. (2021) An Ensemble Machine Learning Model for Enhancing the Prediction Accuracy of Energy Consumption in Buildings. *Arabian Journal for Science and Engineering*, 47(4), 4105–4117.

OECD. (n. d). Ocean shipping and shipbuilding. Available online: https://www.oecd.org/ocean/topics/ocean-shipping/

Pandey, R.K., Kumar, A. and Mandal, A. (2021). A robust deep structured prediction model for petroleum reservoir characterization using pressure transient test data. *Petroleum Research.* 7, pp 204 - 219

Richa, Richa Joshi. (2019). Smart Ship Detection using Transfer Learning with ResNet. *International Research Journal of Engineering and Technology (IRJET),* Volume 06, pp. 1870.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.

Saha, S. (2018) A Comprehensive Guide to Convolutional Neural Networks — *the ELI5 way.* Towards Data Science. Available online: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

Setiawan, B.D., Serdült, U. and Kryssanov, V. (2021) A Machine Learning Framework for Balancing Training Sets of Sensor Sequential Data Streams. *Sensors*, 21(20), 6892.

Shao, Z., Wu, W., Wang, Z., Du, W., & Li, C. (2018). SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. IEEE Transactions on Multimedia, 20(10), 2593.

Ship Detection. (n.d.) kaggle.com. Available online: https://www.kaggle.com/code/randritasarkar/ship-detection

Tanveer, H., Balz, T., & Mohamdi, B. (2019). Using convolutional neural network (CNN) approach for ship detection in Sentinel-1 SAR imagery. Asia-Pacific Conference on Synthetic Aperture Radar (APSAR). IEEE, Xiamen.

Torén, R. (2020). Comparing CNN methods for de- tection and tracking of ships in satellite images. M.Sc. Thesis. Linköping University. Available online: https://www.diva-portal.org/smash/get/diva2:1448055/FULLTEXT01.pdf [Accessed 1 Jan. 2024].

Torrey, L. and Shavlik, J. (2010) *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*.

UN Convention on the Law of the Sea. (2023). Available online: https://www.un.org/depts/los/convention_agreements/convention_overview_convention.htm

Wu, Y., Sun, Y., Zhang, S., Liu, X., Zhou, K., & Hou, J. (2022). A Size-Grading Method of Antler Mushrooms Using YOLOv5 and PSPNet. *MDPI*, pp 6, Basel.

Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A. and Le, Q. (2020) Adversarial Examples Improve Image Recognition. Available online: https://openaccess.thecvf.com/content_CVPR_2020/papers/Xie_Adversarial_Examples_Improve_Image_Recognition_CVPR_2020_paper.pdf

Zach (2022) *How to Interpret the Classification Report in sklearn (With Example).* Statology. Available online: https://www.statology.org/sklearn-classification-report/.