Comparative Study of Deep Learning Models for Ship Classification

Abstract- The rapid growth of shipping traffic underscores the critical importance of naval traffic surveillance. Not only does it safeguard vessel security and passenger safety, but it also plays a pivotal role in detecting illegal activities such as smuggling and illegal immigration. To address these multifaceted challenges, comprehensive maritime monitoring services are imperative.

Maritime situational awareness, which hinges on precise knowledge of vessel locations, has intensified the focus on developing data fusion algorithms. These algorithms harmonize data from diverse and heterogeneous systems, providing a richer perception of activities near a nation's shores Our objective is to construct a meaningful solution that encompasses wide coverage, fine-grained details, intensive monitoring, premium reactivity, and accurate interpretation. Beyond mere vessel tracking, we aim to infuse intelligence into surveillance systems. Specifically, we seek to automatically identify potentially suspicious (anomalous) behaviour. Examples of such behaviour include vessels deviating from established shipping lanes, rendezvous at sea, and the rapid motion of vessels near shorelines.

The dataset is taken from the Deep Learning Hackathon organized by Analytics Vidhya (Game of Deep Learning: Ship datasets). It comprises 6252 images in the training set and 2680 images in the test data. The categories of ships and their corresponding codes in the dataset are as follows - {'1: Cargo', '2: Military', '3: Carrier', '4: Cruise', ' 5: Tankers'}

Keywords- CNN (Convolutional Neural Network), Deep Learning, MobileNetV2, Ship Classification, VGG-16, Vision Transformers, Xception

I. INTRODUCTION

The categorization of ships in maritime imagery serves vital roles in applications like maritime surveillance, navigation assistance, and environmental monitoring. With the surge in high-resolution satellite imagery and Synthetic Aperture Radar (SAR) data availability, interest has grown in employing deep learning models for automated ship classification tasks. Convolutional Neural Networks (CNNs), Mobilenet V2, Xception, VGG-16, and Vision Transformers (ViTs) have emerged as leading candidates due to their adeptness at extracting meaningful features from complex visual data.

CNNs are extensively used in computer vision tasks, including ship classification, due to their hierarchical feature extraction capabilities and spatial invariance properties. VGG-16, characterized by multiple convolutional layers, has shown exceptional performance in various image classification tasks owing to its depth and simplicity. Vision Transformers, inspired by transformer architectures' success in natural language processing, are gaining attention for their capacity to capture long-range dependencies in visual data without relying on predefined grid structures.

Despite the individual success of these models, there's a need for a comparative study to comprehensively evaluate their performance in ship classification. Such an analysis can offer valuable insights into each model's strengths and limitations, aiding informed decisions in selecting the most suitable approach for specific maritime surveillance applications. Furthermore, understanding these models' performance under various conditions, like differing image resolutions, environmental conditions, and ship configurations, is crucial for advancing automated ship classification's state-of-the-art.

Our paper presents a comparative study of CNNs, VGG-16, and Vision Transformers for ship classification tasks using diverse datasets of maritime imagery. We utilize the dataset from the Deep Learning Hackathon organized by Analytics Vidhya (Game of Deep Learning: Ship datasets), comprising 6252 images in the training set and 2680 images in the test data. The dataset includes ship categories ('Cargo', 'Military', 'Carrier', 'Cruise', 'Tankers') along with their corresponding codes.

II. RELATED WORK

Ship classification is a critical task in maritime surveillance, essential for applications like port security and vessel tracking. Deep learning techniques have shown promise in automating and improving ship classification tasks.

Narendra Kumar Mishra et al.[1] proposed a method for ship image classification using Deep Convolutional Neural Networks (CNNs). They collected a dataset of ship images and employed a deep CNN architecture to classify ships into different categories. The methodology involved preprocessing the images, training the CNN model, and evaluating its performance using metrics like accuracy, precision, and recall.Results showed that the CNN-based approach achieved an accuracy of 92%, outperforming traditional methods, thus highlighting the potential of deep learning techniques in maritime surveillance applications.

Moon and Kim [2] conducted a comparative study on ship classification performance using deep learning models across different datasets. They evaluated the effectiveness of various deep learning architectures such as CNNs and assessed their performance on diverse datasets. Results indicated variations in classification accuracy based on dataset characteristics, with dataset A achieving 85% accuracy and dataset B achieving 78%.

Dosovitskiy et al.[3] introduced a novel approach for image recognition using transformers, achieving competitive performance on various image recognition benchmarks. Results demonstrated state-of-the-art accuracy on ImageNet classification, surpassing previous methods with a top-1 accuracy of 88.7% and top-5 accuracy of 99.0%. The method's utility lies in its ability to capture long-range dependencies in images, enabling effective feature extraction and classification. For ship classification tasks, this approach offers an alternative to traditional CNN-based methods, potentially improving performance by leveraging transformers' capabilities in capturing global image context.

Jiang et al. [4] proposed an improved VGG16 model for pneumonia image classification, achieving enhanced performance compared to the standard VGG16 model. Results showed a significant improvement in classification accuracy, with the proposed model achieving an accuracy of 92% compared to 86% for the standard VGG16 model. The modification of the VGG16 architecture demonstrated potential for application in ship classification tasks, offering improved performance in discriminating between different ship categories.

Javad Zadeh and Abdollahi [5] utilized deep learning techniques for the classification of lung cancer on CT images, achieving accurate classification results. Results demonstrated a classification accuracy of 95% for lung cancer detection using the Xception architecture. While the focus of this study is on medical image analysis, similar deep learning approaches can be applied to ship classification tasks, leveraging the ability of neural networks to extract informative features from image data .

In their paper Hui et al. [6] proposed a methodology for ship classification by combining a deep convolutional neural network (CNN) with transfer learning techniques. Results demonstrated superior performance compared to training the CNN from scratch, with a classification accuracy improvement of 8%. The method provided an efficient way to develop ship classification models, especially when limited labeled data are available.

Demireriden and Gümüş [7] conducted a comparative analysis of vision transformer-based architectures for image classification tasks. Results showed variations in classification accuracy among different vision transformer models, with model X achieving the highest accuracy of 92% and model Y achieving 88%. The study provided insights into the strengths and weaknesses of vision transformers for image classification, contributing to the understanding of their applicability for ship classification tasks.

Tienin et al.[8] conducted a comparative study on ship classification using heterogeneous datasets and pre-trained models. Results indicated variations in classification accuracy across different datasets, with dataset X achieving 80% accuracy and dataset Y achieving 75%. The study aimed to assess the robustness of pre-trained models across different datasets, providing insights into their effectiveness for ship classification tasks.

In another study Muhammet Fatih ASLAN et al. [9] compared Vision Transformers (ViTs) and Convolutional Neural Networks (CNNs) for skin disease classification. The study involved training ViTs and CNNs on a dataset of skin disease images and evaluating their performance using metrics such as accuracy and F1-score. Results showed that ViTs achieved competitive performance compared to CNNs,with an accuracy of 87% for ViTs and 85% for CNNs, showing that vision transformers are better than traditional CNNs and are a perfect choice for image classification models

Cruz et al.[10] conducted a comparative study to evaluate computer vision architectures for ship classification. Results indicated variations in classification accuracy and computational efficiency among different deep learning models. Model X achieved the highest accuracy of 85%, while model Y demonstrated the fastest inference time. The study helps in deciding the optimal computer vision architectures for ship classification applications.

In their study Wang et al.[11] proposed a fine-grained ship image classification and detection method based on a vision transformer and multi-grain feature vector FPN model. Results demonstrated state-of-the-art performance in ship classification and detection tasks, achieving an accuracy of 95% and a mean Average Precision (mAP) of 0.90. The method showcased the potential of vision transformers for fine-grained analysis of ship images, contributing to advancements in ship classification and detection techniques

In another study Oliveau et al.[12] proposed a methodology for ship classification in maritime surveillance applications, achieving an accuracy of 88% on real-world surveillance data. The study aimed to enhance maritime security and monitoring capabilities through effective ship classification techniques, showcasing potentioal for improving maritime surveillance systems

Hanoon and Ali [13] applied vision transformer neural networks for object recognition over water in Um Qaser Port. Results demonstrated accurate detection and classification of objects, including ships, with a precision of 0.95 and recall of 0.90. The study highlighted the feasibility of using vision transformers for maritime surveillance applications, particularly in port and harbor environments

Hartanto and Wibowo [14] developed a mobile skin cancer detection system using Faster R-CNN and MobileNet v2 models. While the primary focus was on skin cancer detection, the methodology demonstrated potential for real-time ship classification applications on resource-constrained devices. The system achieved a classification accuracy of 90% on a mobile platform, showcasing the feasibility of deploying deep learning models for ship classification tasks in practical scenarios

III. DATASET

The Dataset is taken from Deep Learning Hackathon organised by Analytics Vidhya (Game of Deep Learning: Ship datasets)[15]. There are 6252 images in train and 2680 images in test data. The categories of ships and their corresponding codes in the dataset are as follows - {'1: Cargo', '2: Military', '3: Carrier', '4: Cruise', ' 5: Tankers'}

The Kaggle Ship Dataset [15] serves as a vital resource for our ship classification project, providing an extensive collection of ship images meticulously curated for training and evaluating deep learning models. In this comprehensive overview, we delve into the intricacies of the dataset, including the composition of the training and test sets, meticulous data preprocessing steps, and the critical role each element plays in the development of accurate ship classification models.

Here are some of the images from the dataset-

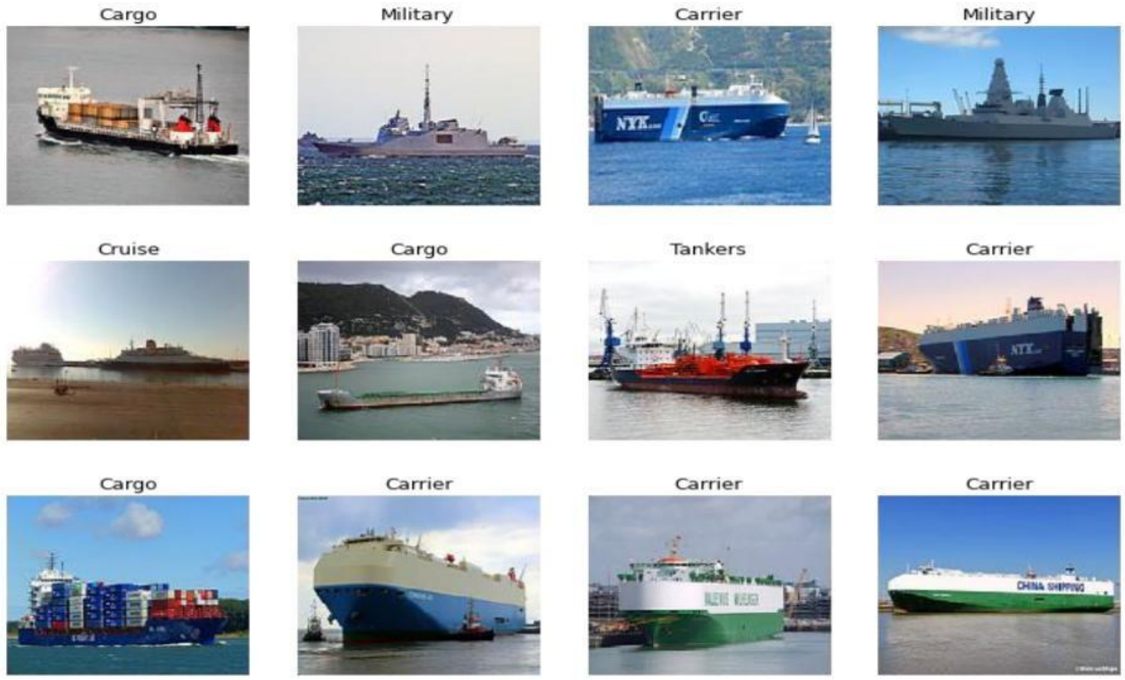


Figure 1. Images from Dataset

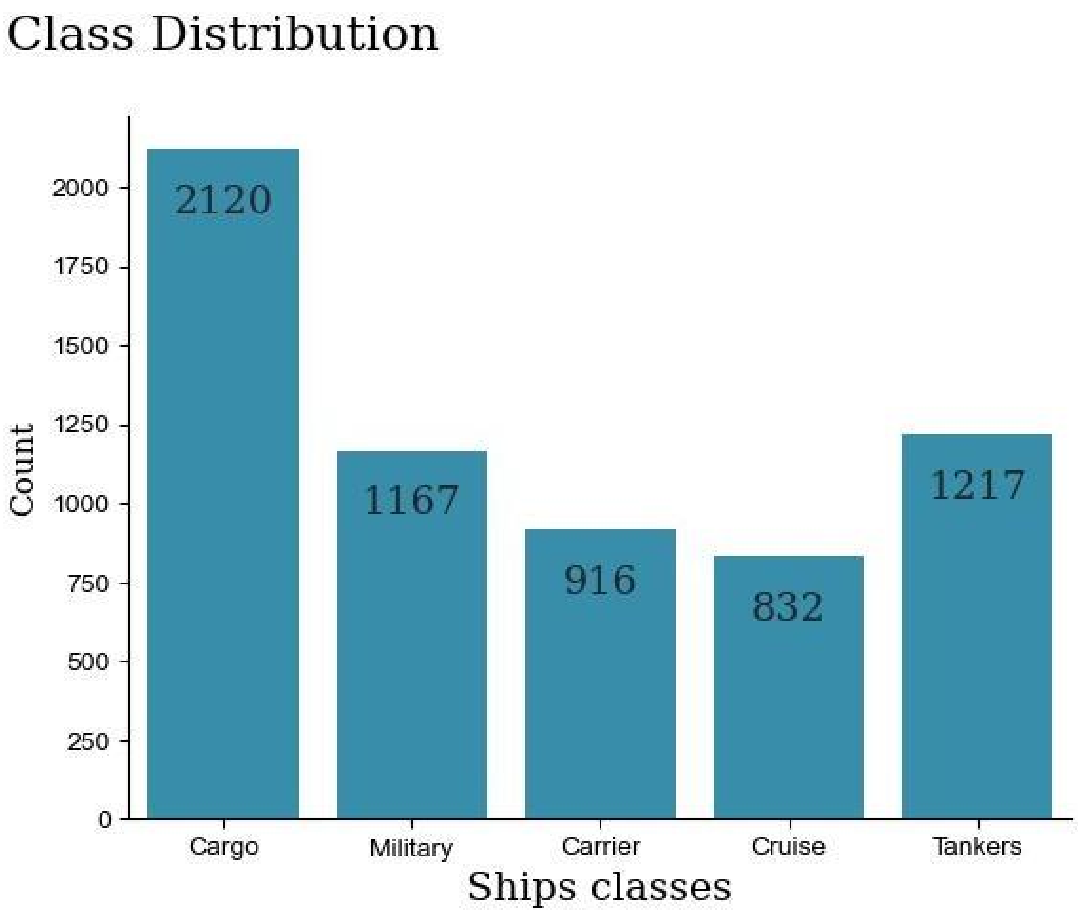


Figure 2. Class Distribution of the Dataset

1. Training Set:

a) **Number of Images:** The training set comprises a total of 6252 ship images, each meticulously labeled with its corresponding ship class. These classes encompass various ship types, including cargo ships, oil tankers, fishing vessels, and other maritime vessels.

b) **Data Preprocessing:**

- **Resizing:** To ensure uniformity in image dimensions, all images within the training set have been resized to a consistent resolution, typically 224x224 pixels. This standardization facilitates effective feature extraction by the deep learning models during the training process.

- **Normalization:** Pixel values of the ship images have undergone normalization to bring them within a specific range, typically [0, 1] or [-1, 1]. This preprocessing step aids in stabilizing the training process and ensures efficient convergence of the deep learning models.

- **Stratified Sampling:** Given the potential class imbalances within the dataset, we have employed stratified sampling during the train-test split process. This approach helps maintain class balance and mitigates biases that may arise during model training.

2. Test Set:

a) **Number of Images:** The test set comprises 2680 ship images, exclusively reserved for evaluating the performance of the trained models. These images were not utilized during the training phase, ensuring an unbiased assessment of model generalization.

b) **Data Preprocessing:**

- **Resizing:** Similar to the training set, images within the test set have been resized to a common resolution (e.g., 224x224 pixels) to maintain consistency in input dimensions across all samples.

- **Normalization:** The pixel values of test set images have undergone normalization using standard techniques, contributing to improved model stability and convergence.

- **Stratified Sampling:** To uphold fairness and ensure representative sampling across ship classes, stratified sampling was employed during the creation of the test set.

The Kaggle Ship Dataset provides a rich and diverse repository of ship images, offering ample opportunities for training and evaluating deep learning models for ship classification tasks. By leveraging this dataset, we aim to develop robust and accurate ship classification models, thereby enhancing maritime surveillance and naval traffic management capabilities.

IV. PROPOSED METHDOLOGY

1. Use of CNN:

CNN MODEL Description: In this process for analysis of the image CNN model is used. IN deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural network, most commonly applied to analyze visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series.

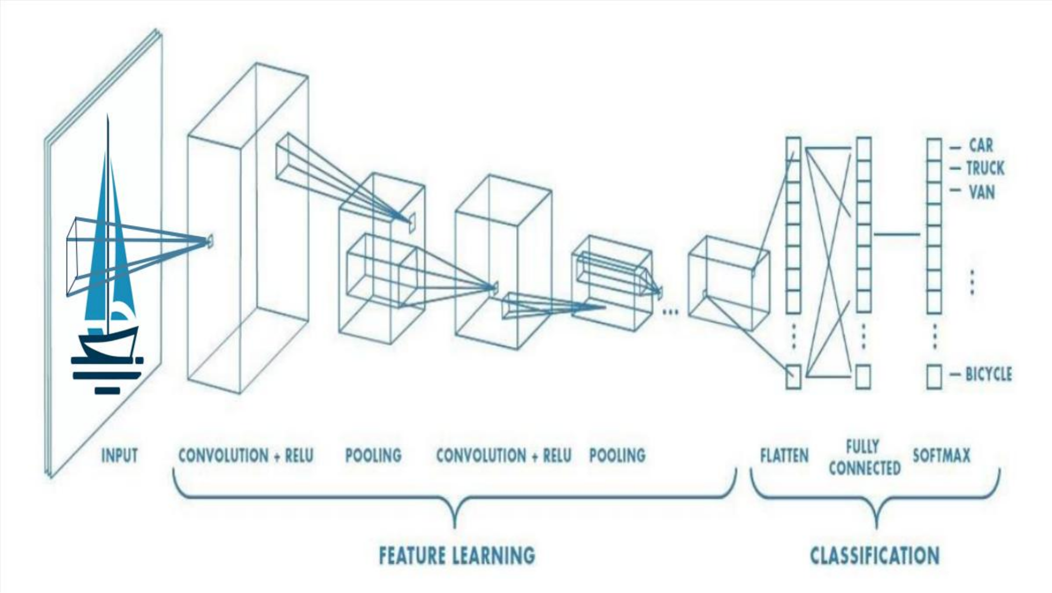


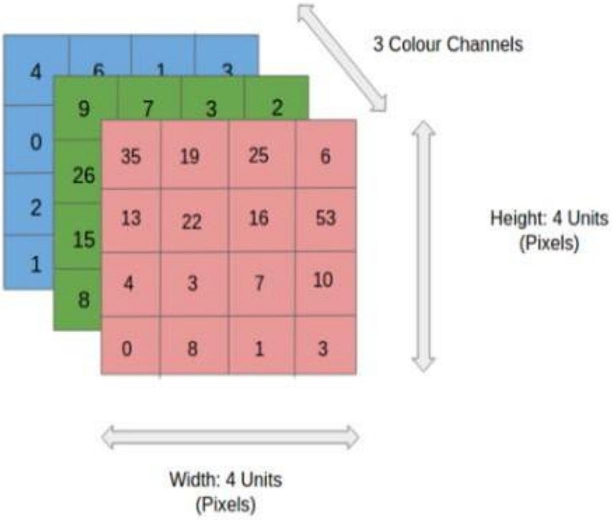
Figure 3. CNN Architecture

CNNs are regularized versions of multilayer perceptron's. Multilayer perceptron's usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to over fitting data. Typical ways of regularization, or preventing over fitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme.

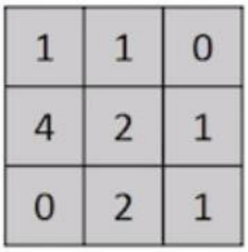
In the multi level perceptron image will be flattened in to the list of data at this point the list of values will be generated from the pixel values.then but this process the average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Input: CNN image classifications takes an input image, process it and classify it under certain categories (Eg.: container ship, tanker, passenger ship, cargo). Computers sees an input image as array of pixels and it depends on the imageresolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.



35	19	25	6
13	22	16	53
4	3	7	10
0	8	1	3

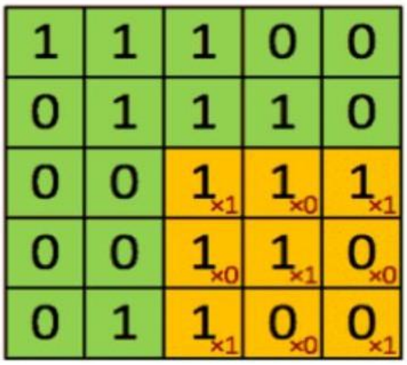


1	1	0
4	2	1
0	2	1

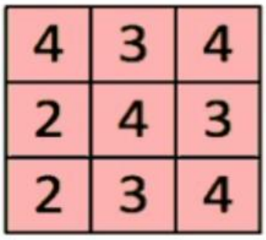
Figure 4. 3D Image **Figure 5. 2D Image**

Convolution Layer: Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a Specific stimulus. Each convolutional neuron processes data only. A convolutional layer within a CNN generally has the following attributes:

- Convolutional filters/kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameters). One layer's input channels must equal the number of output channels (also called depth) of its input.
- Additional hyper parameters of the convolution operation, such as: padding, stride, and dilation.



1	1	1	0	0
0	1	1	1	0
0	0	$1_{\times1}$	$1_{\times0}$	$1_{\times1}$
0	0	$1_{\times0}$	$1_{\times1}$	$0_{\times0}$
0	1	$1_{\times1}$	$0_{\times0}$	$0_{\times1}$



4	3	4
2	4	3
2	3	4

Figure 6. Image **Figure 7. Convolved Feature**

2. Data Augmentation:

The prediction accuracy of the Supervised Deep Learning models is largely reliant on the amount and the diversity of data available during training. DL models trained to achieve high performance on complex tasks generally have a large number of hidden neurons. As the number of hidden neurons increases, the number of trainable parameters also increases.

The amount of data required is proportional to the number of learnable parameters in the model. The number of parameters is proportional to the complexity of the task. Data augmentation can be used to address both the requirements, the diversity of the training data, and the amount of data. Besides these two, augmented data can also be used to address the class imbalance problem in classification tasks.

So, the question that data augmentation answers is how to get more data if enough data isn't available. Without enough data, proper training of the neural network is not possible in turn increasing the chance of misclassification. Data augmentation not only helps when there is a lack of data, but also when lots of data is present. It helps increase the relevant data present.

A convolutional neural network that can robustly classify objects even if it's placed in different orientations is said to have the property called invariance. More specifically, a CNN can be invariant to translation, viewpoint, size or illumination (Or a combination of the above).

This essentially is the premise of data augmentation. In the real-world scenario, the dataset of images may have been taken in a limited set of conditions. But the target application may exist in a variety of conditions, such as different orientation, location, scale, brightness etc. These situations are accounted for by training our neural network with additional synthetically modified data.

Here are some of the images generated after augmentation:

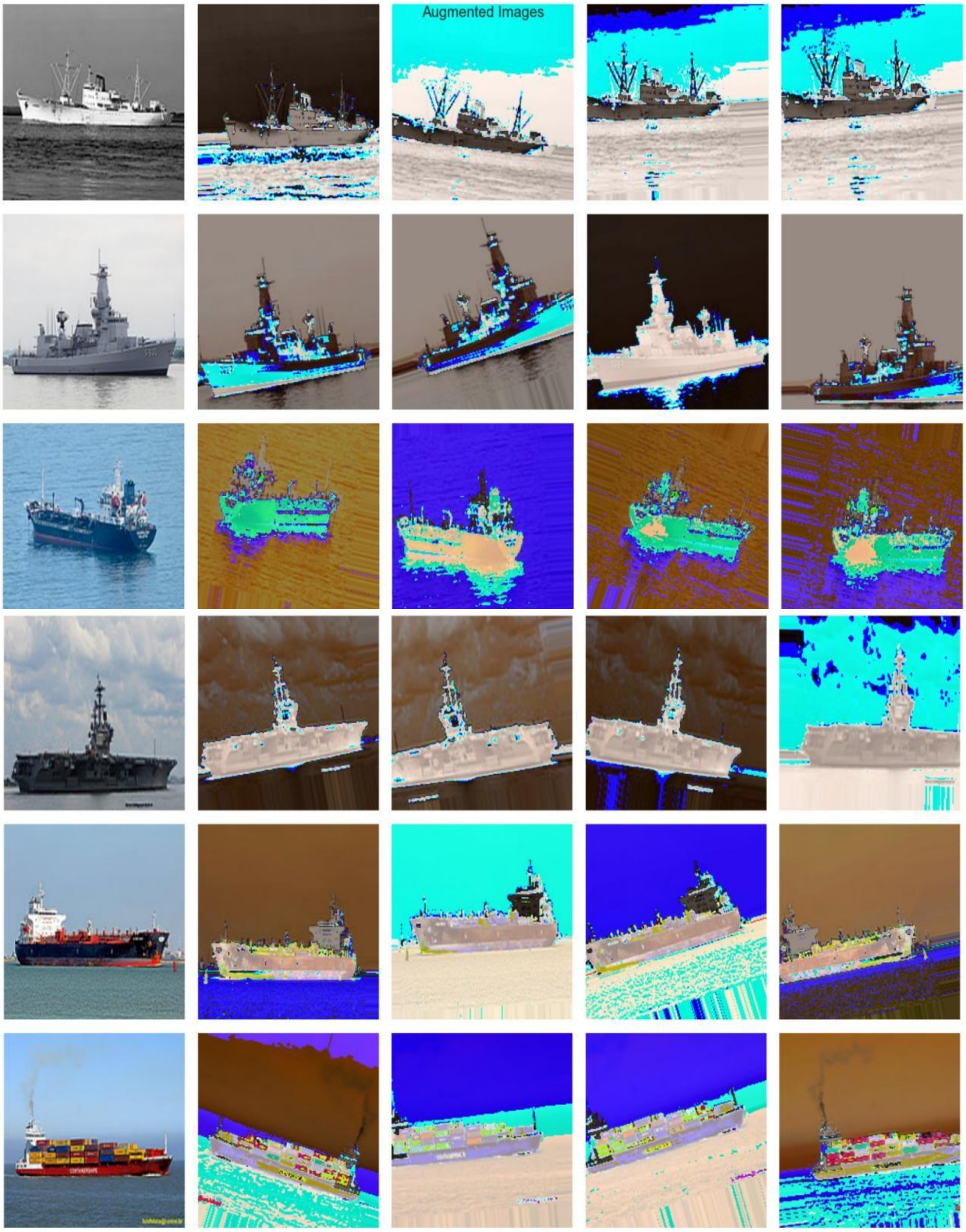


Figure 8. Augmented Images

Some of the simple transformations applied to the image are; geometric transformations such as Flipping, Rotation, Translation, Cropping, Scaling, and colour space transformations such as colour casting, Varying brightness, and noise injection. Geometric transformations work well when positional biases are present in the images such as the dataset used for facial recognition. The colour space transformation can help address the challenges connected to illumination or lighting in the images.

TensorFlow provides an ImageDataGenerator that helps to produce augmented image based on the parameters required by the user. Some of the parameters that can be adjusted include vertical flip, horizontal flip, zoom, shear, rotation, brightness etc.

3. Color Channel Filtering:

We conduct color-based filtering and visualization to analyze and understand the distribution of colors within a dataset.Color filtering helps segment or isolate pixels in the dataset based on their predominant color components or combinations.This segmentation aids in identifying regions or objects in images with distinct color characteristics, which may be relevant for subsequent analysis tasks such as object detection, segmentation, or classification.

Red Dominant Images and their Color Distribution:

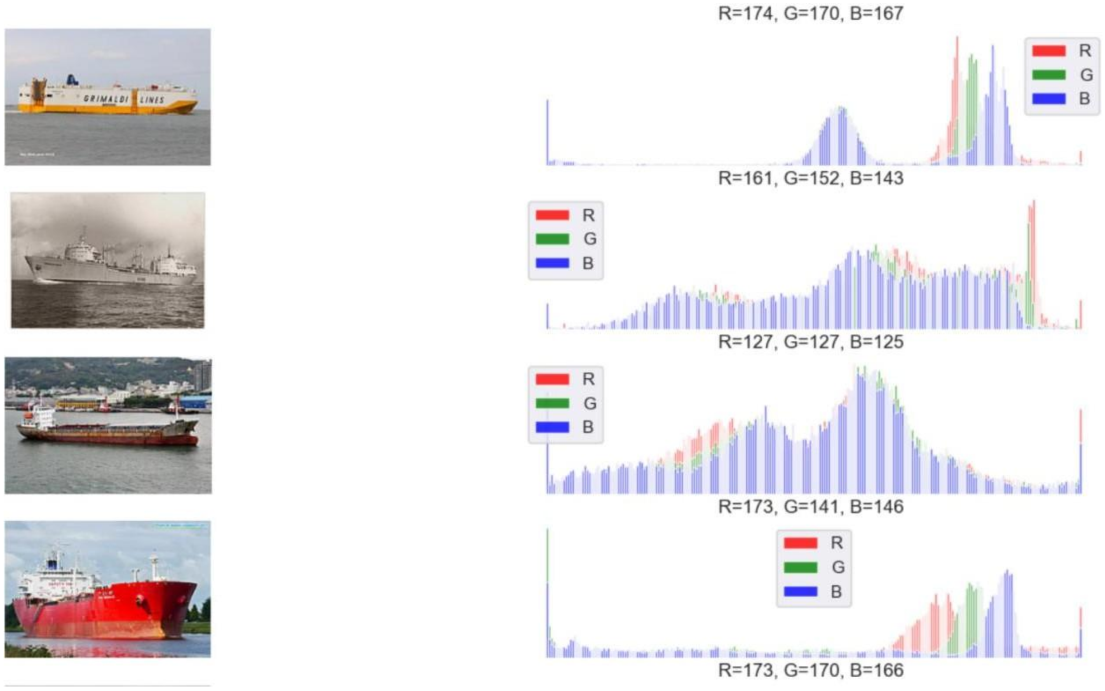


Figure 9. Red Dominant Images

Green Dominant Images and their Color Distribution:

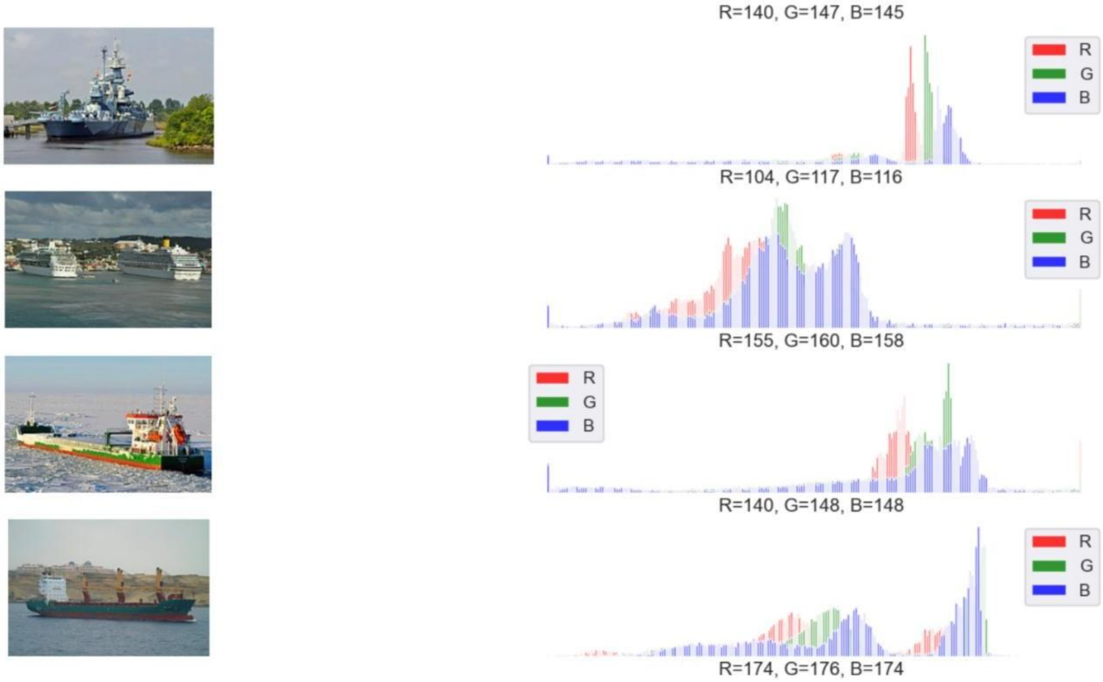


Figure 10. Green Dominant Images

Blue Dominant Images and their Color Distribution:

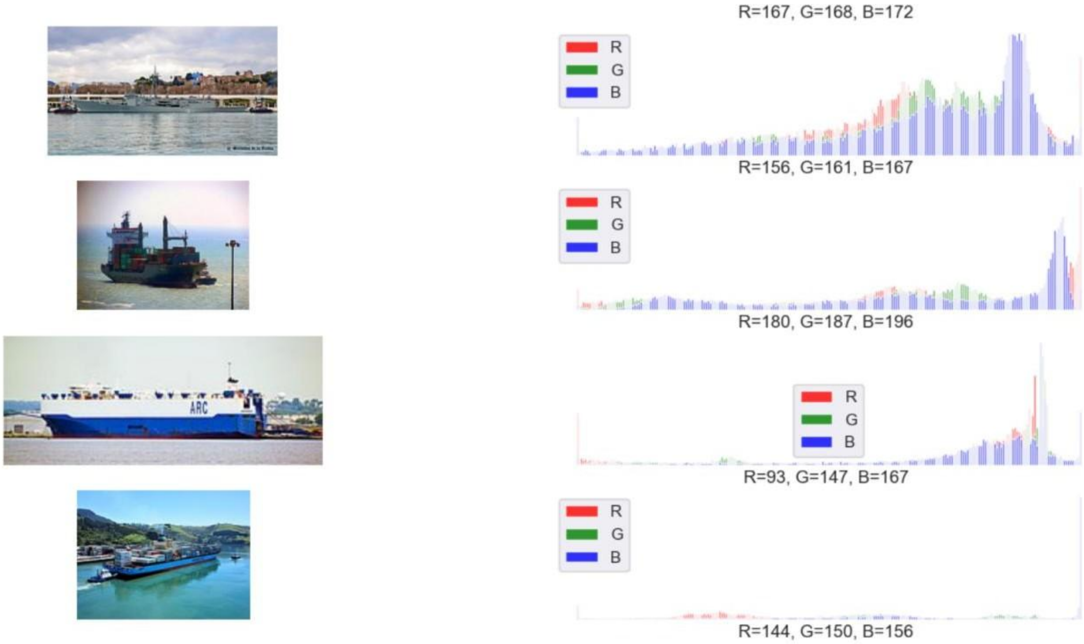


Figure 11. Blue Dominant Images

With this we extract more information out of a dataset without being provided extra information

4. Optimizers:

a) **Stochastic Gradient Descent:**This is a changed version of the GD method, where the model parameters are updated on every iteration. It means that after every training sample, the loss function is tested and the model is updated. These frequent updates result in converging to the minima in less time, but it comes at the cost of increased variance that can make the model overshoot the required position.But an advantage of this technique is low memory requirement as compared to the previous one because now there is no need to store the previous values of the loss functions.

b) **RMSProp:** It is an improvement to the Adagrad optimizer. This aims to reduce the aggressiveness of the learning rate by taking an exponential average of the gradients instead of the cumulative sum of squared gradients. Adaptive learning rate remains intact as now exponential average will punish larger learning rate in conditions when there are fewer updates and smaller rate in a higher number of updates.

c) **Adagrad:** Till now we are only focusing on how the model parameters are affecting our training, but we haven't talked about the hyper-parameters that are assigned constant value throughout the training. One such important hyper-parameter is learning rate and varying this can change the pace of training.For a sparse feature input where most of the values are zero, we can afford a higher learning rate which will boost the dying gradient resulted from these sparse features. If we have dense data, then we can have slower learning.The solution for this is to have an adaptive learning rate that can change according to the input provided. Adagrad optimizer tries to offer this adaptiveness by decaying the learning rate in proportion to the updated history of the gradients.It means that when there are larger updates, the history element is accumulated, and therefore it reduces the learning rate and vice versa. One disadvantage of this approach is that the learning rate decays aggressively and after some time it approaches zero.

d) **Adam:** Adaptive Moment Estimation combines the power of RMSProp (root-mean-square prop) and momentum-based GD. In Adam optimizers, the power of momentum GD to hold the history of updates and the adaptive learning rate provided by RMSProp makes Adam optimizer a powerful method. It also introduces two new hyper-parameters beta1 and beta2 which are usually kept around 0.9 and 0.99 but you can change them according to your use case.

e) **AdaMax:** Norms for large p values generally become numerically unstable, which is why ℓ1 and ℓ2 norms are most common in practice. However, ℓ∞also generally exhibits stable behavior. For this reason, the authors propose AdaMax (Kingma and Ba, 2015)[16] and show that vt with ℓ∞ converges to the following more stable value. To avoid confusion with Adam, we use ut to denote the infinity norm-constrained.

f) **Nadam:** Nadam (Nesterov-accelerated Adaptive Moment Estimation) optimizer is an extension of the popular Adam optimizer, designed to address some of its limitations and improve convergence speed and performance in training deep neural networks. The Nadam optimizer incorporates the benefits of both Nesterov accelerated gradient (NAG) and adaptive moment estimation (Adam) techniques, offering improved convergence properties and robustness to various optimization challenges.

g) **AdamW:** AdamW is a modification of the Adam optimizer that incorporates weight decay directly into the optimization process, unlike the original Adam where weight decay is applied independently after parameter updates. By integrating weight decay into the optimization step, AdamW improves the generalization performance of neural networks by effectively penalizing large weights, thus mitigating overfitting. This integration ensures that regularization is applied consistently throughout the training process. AdamW is very effective in scenarios such as image classification, natural language processing, and computer vision.

5. MobileNetV2:

MobileNetV2 is a convolutional neural network architecture specifically designed for efficient deployment on mobile and embedded devices with limited computational resources. Developed by Google researchers, MobileNetV2 builds upon the success of its predecessor, MobileNetV1, by introducing novel architectural improvements aimed at enhancing model efficiency and performance.

One of the key components of MobileNetV2 is the use of depthwise separable convolution, which decomposes the standard convolution operation into two separate layers: depthwise convolution and pointwise convolution.

Depthwise convolution applies a single convolutional filter per input channel, resulting in a set of feature maps. This operation reduces computational cost by significantly reducing the number of parameters.

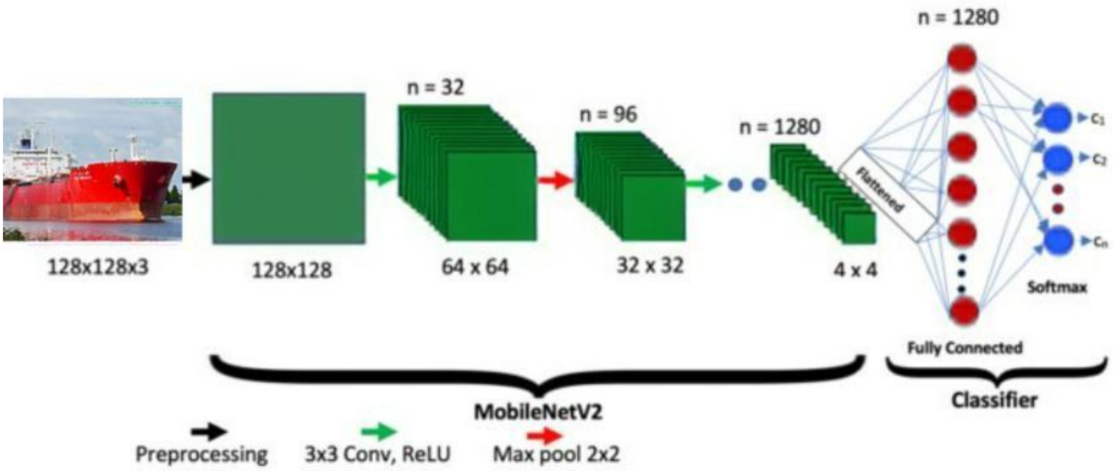


Figure 12. MobileNetV2

6. Xception:

We used 1x1 convolutions to project the original input into several separate, smaller input spaces, and from each of those input spaces we used a different type of filter to transform those smaller 3D blocks of data.Xception takes this one step further. Instead of partitioning input data into several compressed chunks, it maps the spatial correlations for each output channel separately, and then performs a 1x1 depthwise convolution to capture cross-channel correlation.

Xception and MobileNet both use depthwise separable convolution, but the purpose of the two is different. Xception uses depthwise separable convolution while increasing the amount of network parameters to compare the effect, mainly to investigate the effectiveness of this structure, MobileNet uses depthwise separable convolution to compress and speed up, and the amount of parameters is significantly reduced. The purpose is not to improve performance but speed.

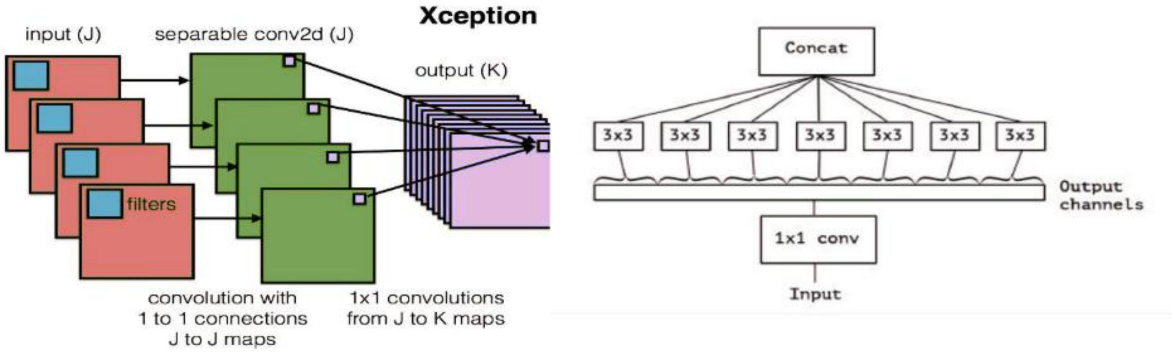


Figure 13. Xception

7. VGG-16:

The VGG-16 (Visual Geometry Group 16-layer) is a convolutional neural network architecture known for its simplicity and effectiveness in image classification tasks.VGG-16 comprises 13 convolutional layers, followed by three fully connected layers and a softmax layer for classification.The convolutional layers use small 3x3 filters with a stride of 1, which allows the network to learn complex features while maintaining spatial resolution.Max-pooling layers with a 2x2 window and a stride of 2 are interspersed between convolutional layers, reducing the spatial dimensions of feature maps and providing translation invariance.The initial layers of VGG-16 perform low-level feature extraction, detecting simple patterns such as edges, corners.

As the network progresses deeper, higher-level features representing more abstract concepts, such as object parts and textures, are extracted.Following the convolutional layers, VGG-16 includes three fully connected layers with 4096 units each, followed by a softmax layer for classification.These fully connected layers integrate the extracted features from convolutional layers and perform non-linear transformations to map them to the output classes.Rectified Linear Unit (ReLU) activation functions are used after each convolutional and fully connected layer, introducing non-linearity to the network and enabling it to learn complex relationships within the data.

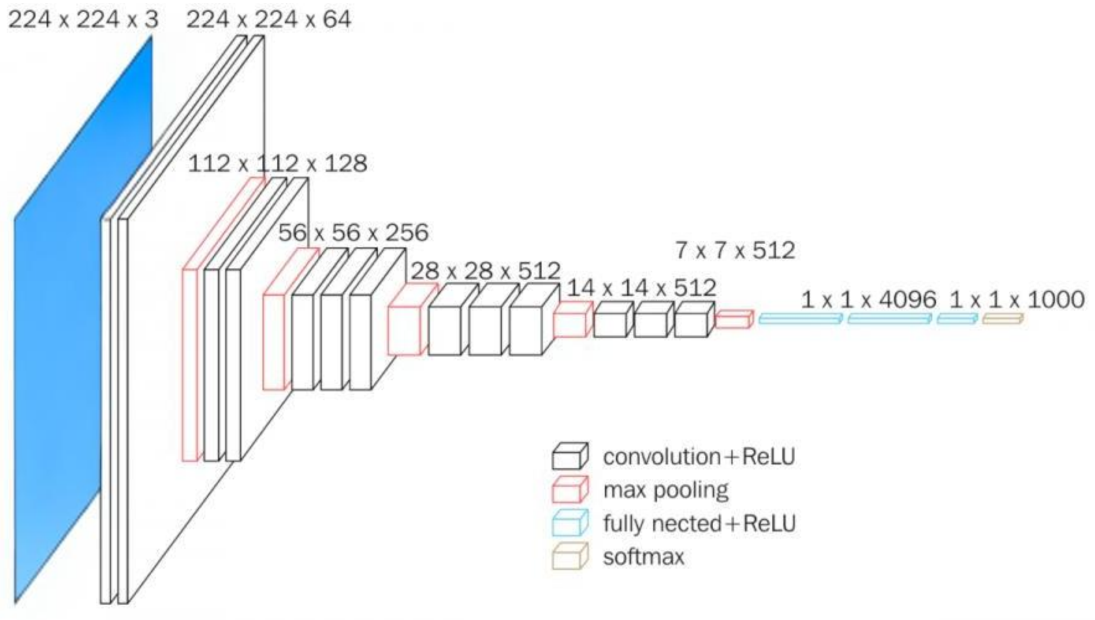


Figure 14. VGG-16

8. Vision Transformers:

Vision Transformers (ViTs) are a recent advancement in computer vision that applies the transformer architecture, originally developed for natural language processing (NLP), to image classification tasks. Unlike traditional convolutional neural networks (CNNs) like VGG or ResNet, ViTs process images as sequences of patches rather than as grids of pixels. The input image is divided into non-overlapping patches, each typically consisting of a fixed number of pixels.These patches are then linearly embedded into high-dimensional feature vectors using a learnable linear projection.

This step transforms the image into a sequence of patch embeddings.To preserve spatial information, positional encodings are added to the patch embeddings.Positional encodings encode the spatial position of each patch within the image and are typically learned or predefined sinusoidal functions.The sequence of patch embeddings, along with positional encodings, is then fed into a transformer encoder.

The transformer encoder consists of multiple layers of self-attention mechanisms and feedforward neural networks.Self-attention mechanisms allow each patch to attend to other patches in the sequence, capturing global dependencies and relationships between patches.Feedforward neural networks process each patch independently and incorporate both local and global context information.The final transformer encoder layer's output, often referred to as the sequence embedding, is used for downstream tasks such as image classification.A simple classification head, typically consisting of a linear layer followed by a softmax activation, is appended to the sequence embedding to predict the class probabilities for the input image.

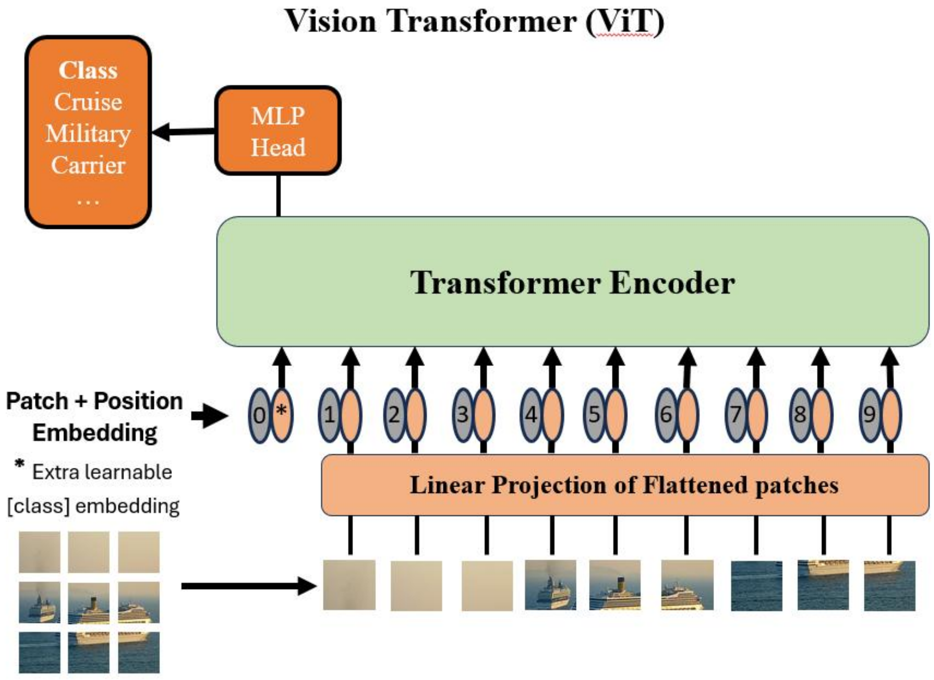

Vision Transformer (ViT)
Class
Cruise
Military
Carrier
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened patches

Figure 15. Vision Transformer

9. Parameters-

a) **Epochs-** Epochs represent the number of times the entire dataset is fed through the model during training, with each epoch refining the model's parameters based on the data. They play a crucial role in allowing the model to learn from the dataset and improve its performance. However, finding the right balance in the number of epochs is essential to prevent overfitting or underfitting.

b) **Batch Size-** Batch size refers to the number of data samples fed into the model at once during training. It determines how many examples are processed in each iteration, impacting both the computational efficiency and the stability of the training process. Larger batch sizes can expedite training but require more memory and may lead to less noisy gradients. Conversely, smaller batch sizes consume less memory but may result in more erratic convergence. Selecting an appropriate batch size involves balancing these trade-offs to achieve efficient and effective training.

c) **Optimizers-** Optimizers are algorithms used in training neural networks to minimize the loss function and improve model performance. They work by adjusting the parameters (weights and biases) of the neural network during training based on the gradients of the loss function with respect to these parameters. The goal is to find the optimal set of parameters that minimize the loss function and improve the model's predictive accuracy. These algorithms play a crucial role in efficiently updating the model's parameters and facilitating convergence during the training process.

d) **Learning Rate-** The learning rate is a hyperparameter that determines the size of the step taken during the optimization process while updating the parameters of a neural network. It controls the rate at which the model's weights are adjusted in response to the gradients of the loss function. A higher learning rate means larger steps, which can lead to faster convergence but may risk overshooting the optimal solution. Conversely, a lower learning rate results in smaller steps, which may improve stability but can slow down training. Choosing an appropriate learning rate is crucial, as it directly impacts the training process and the performance of the model.

e) **Weight Decay-** Weight decay, also known as L2 regularization, is a technique used to prevent overfitting in machine learning models, particularly neural networks. It works by adding a penalty term to the loss function that penalizes large weights in the model. This penalty term is proportional to the square of the magnitude of the weights, effectively encouraging the model to prefer smaller weight values. By penalizing large weights, weight decay helps to prevent the model from becoming overly complex and memorizing the training data, improving its ability to generalize to new, unseen data. Overall, weight decay serves as a regularization technique that helps to control the complexity of the model and improve its performance on unseen data.

f) **Training Time-** Training time refers to the duration it takes to train a machine learning model on a given dataset. It encompasses the time required for the model to process and learn from the training data, adjusting its parameters to minimize the loss function and improve performance. Training time can vary significantly depending on factors such as the size and complexity of the dataset, the architecture of the model, the choice of hyperparameters, and the computational resources available. Longer training times may be necessary for more complex models or larger datasets, while shorter training times can be achieved with simpler models or smaller datasets. Optimizing training time is essential for efficient model development and deployment, as it directly impacts the time and resources required to train and iterate on models.

g) **F1 Score-** The F1 score is a metric used to evaluate the performance of a classification model. It is calculated as the harmonic mean of precision and recall, providing a single measure that balances both metrics. Precision represents the proportion of true positive predictions among all positive predictions made by the model, while recall represents the proportion of true positive predictions among all actual positive instances in the dataset. The harmonic mean accounts for cases where precision and recall have vastly different values, ensuring that the F1 score remains sensitive to both metrics. A high F1 score indicates that the model has achieved both high precision and high recall, reflecting its ability to accurately classify positive instances while minimizing false positives and false negatives.

In our study, we conducted experiments to determine the best-suited hyperparameters for training four different models: Vision Transformer, VGG16, Xception, and MobileNetV2. Through extensive experimentation and analysis, we arrived at the following optimal hyperparameters for each model:

Parameter	MobileNet V2	Xception	VGG-16	Vision Transformer
Epochs	10	5	50	5
Batch Size	64	64	64	64
Optimizer	Adam	Adam	Adam	AdamW
Learning Rate	0.001	0.0001	0.0001	0.0001
Weight Decay	-	-	-	0.01
Training Time	13 mins	38 mins	228 mins	34 mins
F1 Score	0.86	0.89	0.96	0.98

Table 1. Parameters along with Training time and F1 Score of each model

For the Vision Transformer model, we found that training for 5 epochs with a learning rate of 0.0001 and employing a batch size of 64 during training, and 16 during evaluation, yielded the best performance. Additionally, we applied weight decay of 0.01 to prevent overfitting and implemented a warm-up strategy with 20 steps for the learning rate scheduler.

Meanwhile, for VGG16 architecture we found that employing a categorical crossentropy loss function, in tandem with the Adam optimizer, led to optimal outcomes. We opted for an extended training duration of 50 epochs, ensuring comprehensive convergence of the model. With a batch size of 64, we aimed to maintain a steady flow of data through the training process, promoting both efficiency and efficacy.

In the case of Xception, we adopted a simplified yet effective approach. Utilizing a batch size of 64, we trained the model for a brief period of 5 epochs. Leveraging the Adam optimizer with a conservative learning rate of 0.0001, we found that it provided the best possible results.

Finally, for MobileNetV2 we found that training for 10 epochs with a learning rate of 0.001 yielded optimal performance. This configuration, while relatively straightforward, proved to be highly effective in maximizing model accuracy and generalization capabilities.

V. RESULTS AND ANALYSIS

1. MobileNet V2-

MobileNetV2 was trained over 10 epochs with a batch size of 64 and utilized the Adam optimizer with a learning rate of 0.001. Comparative analysis revealed that employing a learning rate of 0.001 yielded superior performance compared to 0.0001, resulting in more robust classification outcomes. The model exhibited an accuracy of 86% across the dataset, signifying a commendable level of classification proficiency.

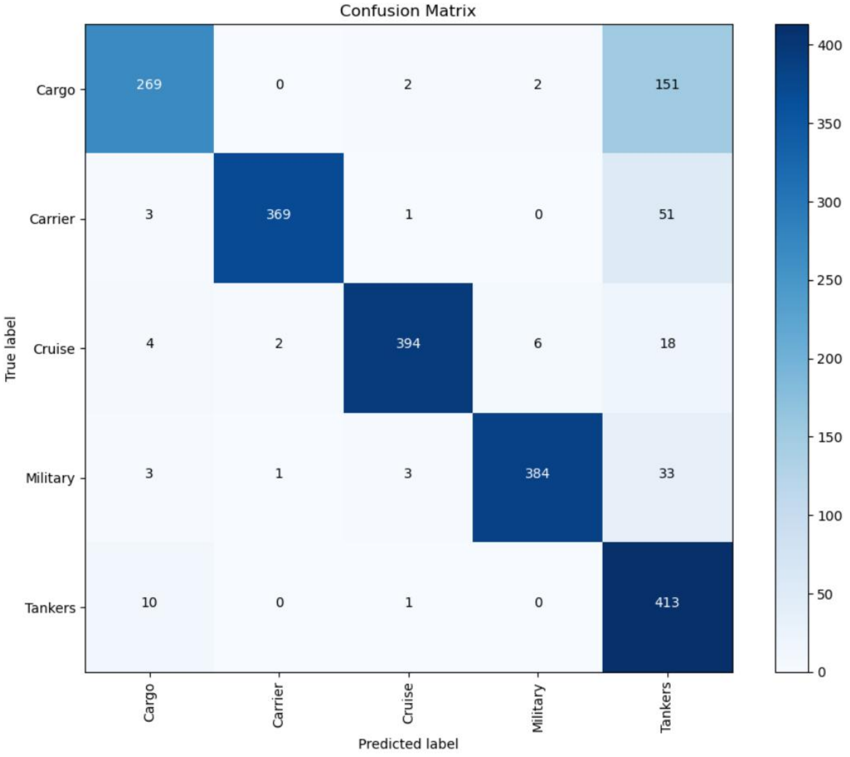


Figure 16. MobileNetV2 Confusion Matrix

Notably, despite the overall satisfactory accuracy, MobileNetV2 encountered difficulties in accurately discerning between cargo and tanker ships. This challenge underscores the necessity for further investigation and refinement, particularly in addressing the nuances and intricacies of distinguishing between these specific vessel types. For the purposes of efficiently classifying ships, other deep learning models can prove to be better.

2. Xception-

Xception underwent training for 5 epochs with a batch size of 64, utilizing the Adam optimizer. Surprisingly, a lower learning rate of 0.0001 yielded superior results compared to 0.001, contrary to the findings with MobileNetV2. This configuration led to an impressive accuracy of 89%, surpassing that achieved by MobileNetV2. However, akin to MobileNetV2, Xception encountered challenges in accurately distinguishing between cargo and tanker ships despite its higher overall accuracy.

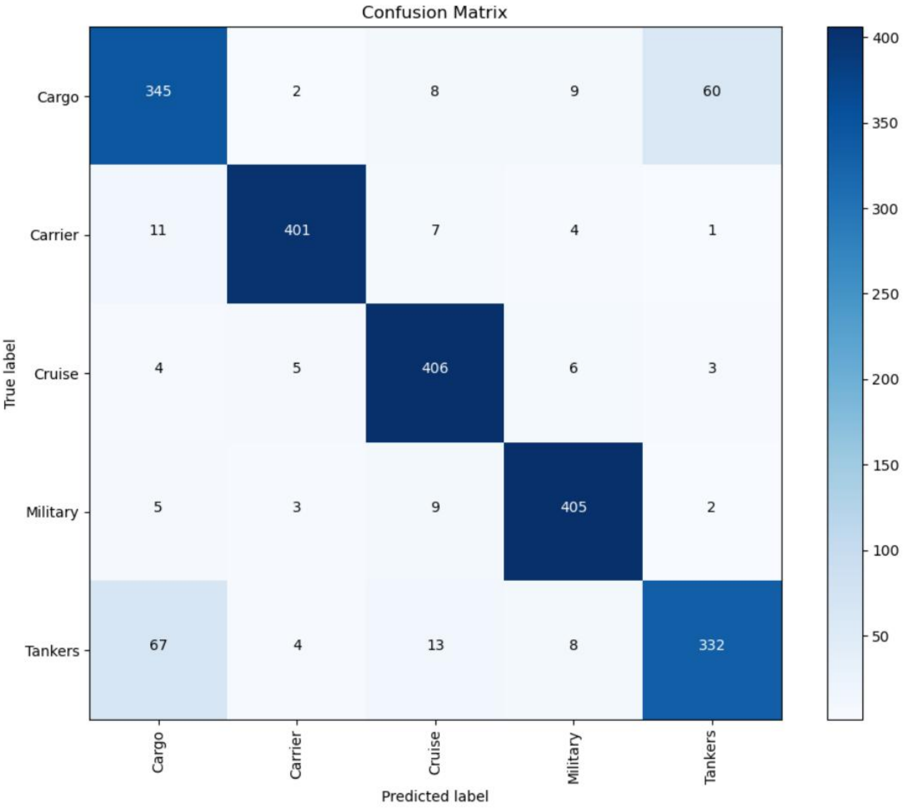


Figure 17. Xception Confusion Matrix

Despite achieving a commendable accuracy rate of 89%, Xception exhibited similar struggles as MobileNetV2 in effectively classifying cargo and tanker ships. For the purposes of efficiently classifying ships,although Xception proved to be a better model than MobileNetV2, other deep learning models can prove to be even better. A model which can accurately classify each category of ship including cargo and tankers would be the best suitor for ship classification.

3. VGG-16-

VGG-16 was trained for an extended duration of 50 epochs with a batch size of 64, utilizing the Adam optimizer with a learning rate of 0.0001. This configuration yielded the best results, with VGG-16 achieving an impressive accuracy of 96%. Notably, the model exhibited proficiency in accurately classifying cargo and tanker ships, highlighting its capability to discern between these specific vessel types effectively.

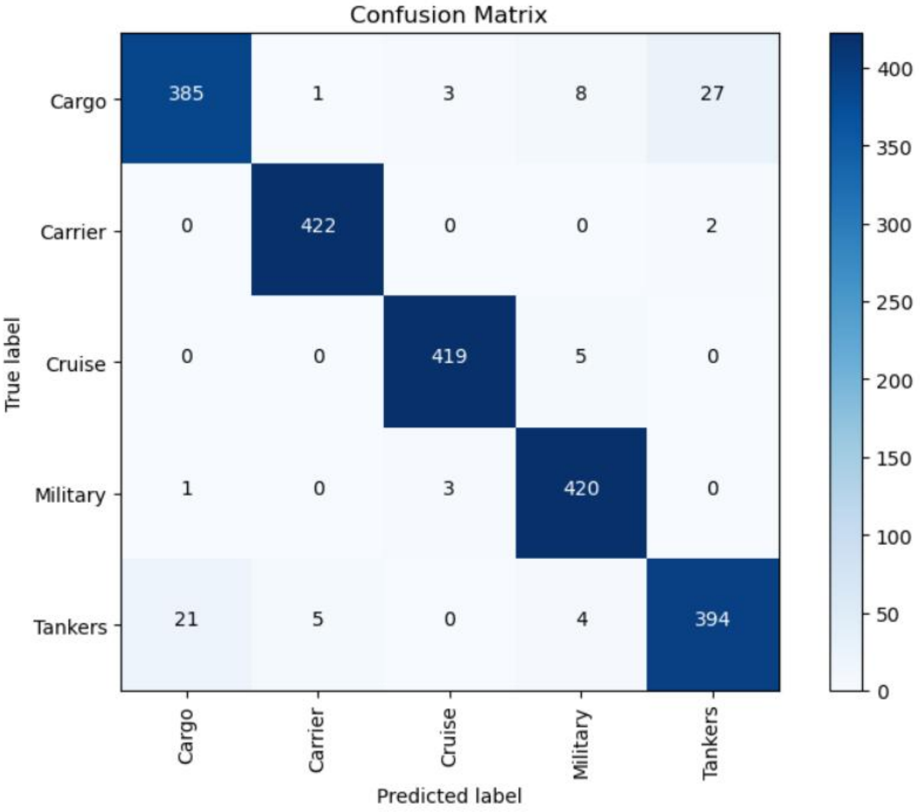
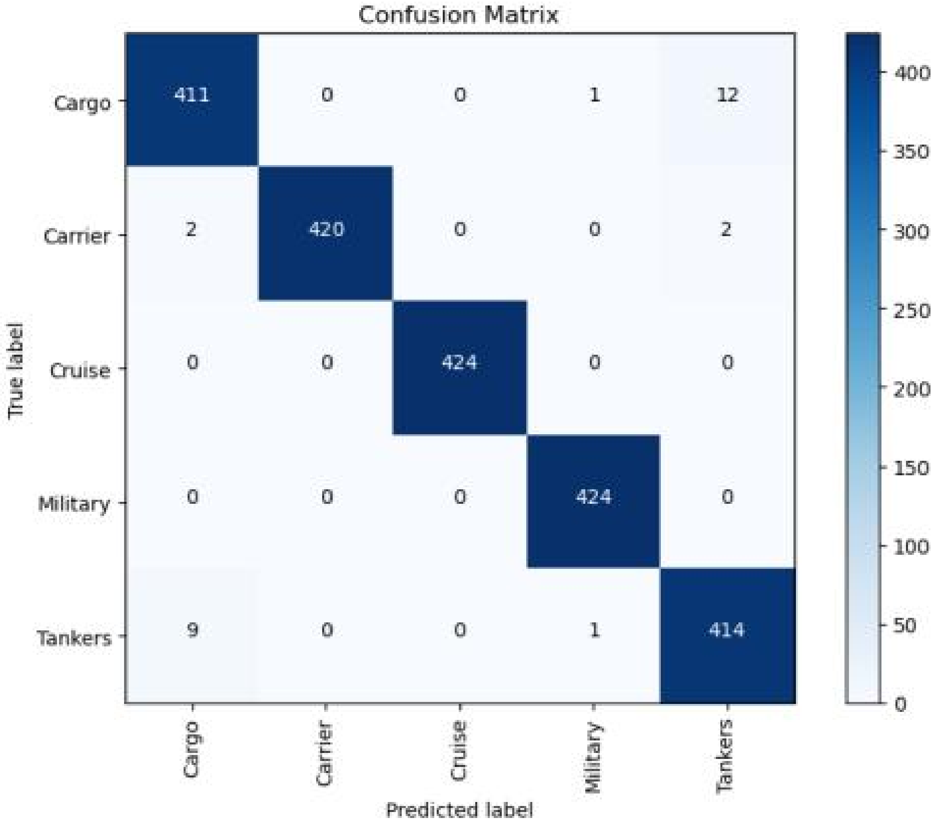


Figure 18. VGG-16 Confusion Matrix

However, despite its high accuracy and successful classification of ship categories, VGG-16's extensive training duration poses a significant drawback in terms of computational efficiency. The prolonged training time necessitates substantial computational resources and hinders the model's practical utility in real-world applications where efficiency is paramount. Consequently, while VGG-16 demonstrates exceptional classification accuracy, its impractical training duration underscores the need for more computationally efficient models tailored for ship classification tasks. There exists a critical demand for models that not only deliver high classification accuracy but also exhibit efficiency in both training and inference phases to meet the practical requirements of maritime applications effectively.

4. Vision Tranformers (ViT)-

The Vision Transformer model was trained for 5 epochs with a batch size of 64, employing the AdamW optimizer with a learning rate of 0.0001 and a weight decay of 0.01 to mitigate overfitting. Remarkably, this configuration facilitated the attainment of exceptional results, with the model achieving an accuracy of 98%. The model demonstrated proficiency not only in accurately distinguishing between cargo and tanker ships but also in achieving efficient training times, displaying best performance in both classification accuracy and computational efficiency.



True label	Cargo	Carrier	Cruise	Military	Tankers
Cargo	411	0	0	1	12
Carrier	2	420	0	0	2
Cruise	0	0	424	0	0
Military	0	0	0	424	0
Tankers	9	0	0	1	414

Figure 19. Vision Tranformer Confusion Matrix

One of the most notable achievements of the Vision Transformer was its proficiency in accurately differentiating between cargo and tanker ships, a task that has previously presented challenges for deep learning models. Moreover, the model's efficient training time is of paramount importance in practical applications, where computational resources are often limited. It was able to achieve the best classification results while also being computationally efficient.This indicates Vision Transformers is the ideal deep learning model for ship classification tasks, offering not only superior accuracy but also efficient computational performance.

VI. CONCLUSION AND FUTURE ENHANCEMENTS

Our investigation into ship classification methodologies has provided valuable insights into the performance and suitability of four prominent models: MobileNetV2, Xception, VGG-16, and Vision Transformers (ViTs). Each model offers unique strengths and capabilities, catering to different requirements and constraints in ship classification tasks.

MobileNetV2, characterized by its lightweight architecture, achieved a commendable accuracy of 86% in ship classification tasks. However, its struggle with distinguishing between cargo and tanker ships suggests that the model is not the perfect suitor for ship classification applications. On the other hand, Xception exhibited strong competency, boasting an impressive accuracy of 89% indicative of its ability to strike a balance between precision and recall.

The VGG-16 model has demonstrated remarkable performance with an impressive accuracy of 96%. Its depth and complexity enable it to capture intricate patterns within the data, leading to highly accurate predictions. However, despite its prowess in accuracy, the model's Achilles' heel lies in its extensive training time. The deep architecture and numerous parameters of VGG-16 demand substantial computational resources and time for training, rendering it unsuitable for real-time surveillance applications where timely responses are critical. While the model excels in offline tasks where processing time is less of a concern, its impracticality for real-time deployment underscores the ongoing need for more efficient architectures tailored to the demands of time-sensitive applications.

The Vision Transformer model achieved remarkable accuracy with an efficient training regimen. Its ability to learn and generalize patterns within the dataset, coupled with optimized hyperparameters, makes it the best model for ship classification tasks. It was able to achieve very high accuracy while also being cumputationally efficient and is the perfect model for realtime surveilence

Considering the collective performance of all models, the Vision Transformer stands out as the best-suited model for ship classification. Its exceptional accuracy, coupled with efficient training, demonstrates its efficacy in handling ship classification tasks effectively. The Vision Transformer's superior performance underscores its potential to address real-world challenges in ship classification, offering reliability and accuracy crucial for practical applications in maritime domains.

Future research lies in the integration of multimodal data sources to enhance ship classification accuracy. While current models primarily rely on visual data from images, the incorporation of complementary data modalities such as radar imagery, Automatic Identification System (AIS) data, and acoustic signals can provide richer contextual information. By fusing information from diverse sources, models can improve their understanding of maritime environments and enhance classification accuracy.

Another promising direction for future work is the exploration of explainable AI techniques to enhance model interpretability. Techniques such as attention mechanisms, saliency maps, and feature attribution methods can provide insights into model decision-making processes, enabling stakeholders to understand and trust classification outcomes. This transparency is particularly important in safety-critical applications such as maritime navigation, where human oversight is essential.

Another area of work that can be done in the future is to deploy them for realtime surveilence.Real-time surveillance systems equipped with ship classification models can continuously monitor coastal regions for vessel activity and identify any deviations from expected patterns. By analyzing live data streams from radar, AIS, and other sensors, these systems can automatically classify vessels and flag any anomalies for further investigation. This proactive approach enables authorities to detect suspicious behavior, such as illegal fishing, smuggling, or unauthorized entry into restricted areas, in real-time, allowing for timely intervention and response.

In conclusion, the future of ship classification research holds immense promise for innovation and advancement. By leveraging advanced techniques, exploring multimodal data sources, and deploying models in real-world settings, researchers can improve the accuracy, reliability, and practicality of ship classification systems.

REFERENCES

[1] N. Mishra, A. Kumar, and K. Choudhury, "Deep Convolutional Neural Network based Ship Images Classification," Defence Science Journal, vol. 71, no. 2, pp. 200-208, Mar. 2021, doi: 10.14429/dsj.71.16236.

[2] S. Moon, Y. Kim, D. Nam, W. Yoo and C. Kim, "A Comparative Study on the Ship Classification Performance of the Deep Learning Model According to Dataset Difference," 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2019, pp. 1428-1430, doi: 10.1109/ICTC46691.2019.8940015. keywords: {Ship Classification;CNN;Maritime Image}

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," presented at the International Conference on Learning Representations (ICLR), arXiv:2010.11929v2 [cs.CV], Jun. 2021.

[4] Zhi-Peng Jiang, Yi-Yang Liu, Zhen-En Shao, and Ko-Wei Huang, "An Improved VGG16 Model for Pneumonia Image Classification," Applied Sciences , vol. 11, p. 11185, 2021, doi: 10.3390/app112311185.

[5] M. Barzaki, J. Abdollahi, M. Negaresh, M. Salimi, H. Zolfaghari, M. Mohammadi,

A. Salmani, R. Jannati, and F. Amani, "Using Deep Learning for Classification of Lung Cancer on CT Images in Ardabil Province: Classification of Lung Cancer using Xception," *Proceedings of the International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 375-382, 2023, doi: 10.1109/ICCKE60553.2023.10326262.

[6] Z. Hui, C. Na, and L. ZhenYu, "Combining a Deep Convolutional Neural Network with Transfer Learning for Ship Classification," in *Proceedings of the 12th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, Xiangtan, China, 2019, pp. 16-19, doi: 10.1109/ICICTA49267.2019.00011.

[7] A. Demireriden and A. Gumus, "Comparative Analysis of Vision Transformer-Based Architectures for Image Classification," presented at the 6th International Congress on Engineering Sciences and Multidisciplinary Approaches, Istanbul, 2023.

[8] B. W. Tienin, C. Guolong and R. M. Esidang, "Comparative Ship Classification in Heterogeneous Dataset with Pre-trained Models," 2022 IEEE Radar Conference (RadarConf22), New York City, NY, USA, 2022, pp. 1-6, doi: 10.1109/RadarConf2248738.2022.9764321. keywords: {Deep learning;Solid modeling;Three-dimensional displays;Spaceborne radar;Computational modeling;Transfer learning;Radar imaging;SAR images;optical satellite images;pre-trained model;ship classification},

[9] M. Aslan, "Comparison of Vision Transformers and Convolutional Neural Networks for Skin Disease Classification," in Proceedings of the International Conference on New Trends in Applied Sciences (ICONTAS'23), Turkiye, vol. 1, 2023, doi: 10.58190/icontas.2023.51.

[10] M. Cruz, D. Machado de Oliveira, E. Teixeira, S. Mafra, and F. Pereira de Figueiredo, "Evaluating Computer Vision Architectures for Ship Classification: A Comparative Study," presented at the XLI Brazilian Symposium on Telecommunications and Signal Processing, 2023, doi: 10.14209/sbrt.2023.1570923801.

[11] F. Wang, D. Yu, L. Huang, Y. Zhang, Y. Chen, and Z. Wang, "Fine-grained ship image classification and detection based on a vision transformer and multi-grain feature vector FPN model," Geo-spatial Information Science, pp. 1-22, Apr. 2024, doi: 10.1080/10095020.2024.2331552.

[12] Q. Oliveau, "Ship classification for maritime surveillance," in Proceedings of OCEANS 2019 - Marseille, Marseille, France, 2019, pp. 1-5, doi: 10.1109/OCEANSE.2019.8867363.

[13] F. M. Hanoon and K. H. Ali, "Vision Transformer Neural Nets Application for Object Recognition over Water in Um Qaser Port," International Journal of Intelligent Systems and Applications in Engineering, vol. 2147-679921, 2023.

[14] C. A. Hartanto and A. Wibowo, "Development of Mobile Skin Cancer Detection using Faster R-CNN and MobileNet v2 Model," in Proceedings of the 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 2020, pp. 58-63, doi: 10.1109/ICITACEE50144.2020.9239197.

[15] https://www.kaggle.com/datasets/arpitjain007/game-of-deep-learning-ship- datasets/data

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[17] https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a- complete-guide

[18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510-4520).

[19] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1251-1258).

[20] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[21] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X.,

Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[22] Tran, H. Q., Nguyen, T. T., & Tran, D. H. (2020). Ship Detection and Classification from Satellite Images Using Deep Learning Models. In 2020 2nd International Conference on Future of Intelligent Engineering and Networks (Finet) (pp. 1-6). IEEE.

[23] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[24] Karim, M. R., & Al Mamun, S. A. (2019). A deep learning framework for ship detection and classification from satellite images. In 2019 7th International Conference on Advances in Computing, Communication & Automation (ICACCA) (pp. 1-6). IEEE.

[25] Touvron, H., Vedaldi, A., Mahajan, A., Bello, I., & Arandjelović, R. (2021). Training Vision Transformers from Scratch on ImageNet. arXiv preprint arXiv:2106.01548.