ESKISEHIR TECHNICAL UNIVERSITY **Ship Classification using Xception**



BIM447 Introduction to Deep Learning Project

**Ship Classification using Xception**

Duygu Halisyama1 and Furkan D¨onmez2

1Eski¸sehir Technical University, Computer Engineering Department, 2Eski¸sehir Technical University, Computer Engineering Department,

# ABSTRACT

This study investigates the application of the Xception model, a deep learning architecture, for ship type classification using a dataset comprising 6,252 ship images categorized into five classes [1]. The dataset is split into training (60%), validation (20%), and test (20%) sets. Augmented data and custom callback functions are utilized for model training. Evaluation on the validation set using the F1 score demonstrates the model’s effectiveness in ship type classification. The performance is further analyzed through the visualization of the confusion matrix. This research contributes to the field of computer vision by showcasing the utility of deep learning techniques in maritime vessel identification and classification tasks.

**Keywords:** Deep Learning, Xception, Ship Classification, Convolutional Neural Networks, Image

Classification

**Corresponding Author:** Cahit Perkgöz

# 1 Introduction

Maritime transportation plays a crucial role in global trade and commerce, facilitating the movement of goods and materials across the world’s oceans [2]. With the increasing reliance on maritime routes, the need for efficient and effective ship monitoring and classification systems has become paramount. Ship classification, the process of identifying and categorizing vessels based on their characteristics and functions, is essential for various maritime activities, including vessel tracking, maritime surveillance, and port management. Traditional methods of ship classification often rely on manual inspection or radar-based identification, which can be labor-intensive and prone to errors. These methods also struggle to handle large volumes of data and may inaccurately classify complex ship types. Manual inspection processes are time-consuming and costly, potentially leading to delays in decision-making and compromising maritime safety and security [3].

This study addresses this challenge by investigating the application of deep learning techniques for ship type classification. By leveraging the capabilities of deep neural networks, we aim to develop a robust and scalable solution that can accurately identify and categorize ships based on their visual characteristics. Such a system would have wide-ranging applications in maritime operations, including vessel traffic management, maritime domain awareness, and environmental monitoring.

Previous research in ship classification has predominantly focused on traditional machine learning algorithms and rule-based systems. Studies have explored the use of feature extraction methods and classification algorithms such as Support Vector Machines (SVM) and Random Forests for ship type classification based on radar data or Automatic Identification System (AIS) signals[4]. While these approaches have achieved reasonable accuracy in certain scenarios, they often rely on handcrafted features and may struggle to generalize to new or unseen ship types.

In recent years, there has been a growing interest in applying deep learning techniques to ship classification tasks. Convolutional Neural Networks (CNNs) have shown promising results in automatically learning discriminative features from ship images and achieving high classification accuracy[5][6]. Several studies have proposed CNN-based architectures for ship type classification, leveraging pre-trained models such as VGG, ResNet, and Inception[7][8]. These models have demonstrated superior performance compared to traditional methods, offering improved accuracy and robustness across diverse datasets.

The primary objective of this study is to investigate the effectiveness of deep learning techniques, specifically the Xception model, for ship type classification using image data. The research aims are outlined as follows:

1. To develop a deep learning-based ship classification model using the Xception architecture.
2. To evaluate the performance of the proposed model on a diverse dataset of ship images, encompassing multiple ship types and variations in environmental conditions.
3. To assess the robustness of the deep learning model against variations in image quality, lighting conditions, and background clutter.

The research hypotheses to be tested is the performance of the Xception model will be robust across diverse environmental conditions and image variations. This study aims to contribute to the ongoing efforts to enhance maritime domain awareness by providing an efficient and accurate tool for ship classification, showcasing the practical applications of advanced CNN architectures in real-world scenarios.

# 2 Methodology

The Xception model, short for ”Extreme Inception” is a deep convolutional neural network architecture that leverages depthwise separable convolutions to efficiently capture spatial and channelwise features. It was proposed by Francois Chollet, the creator of the Keras library[4][14].

## 2.1 Architecture Overview

The Xception model is structured around the idea of replacing the standard Inception modules with depthwise separable convolutions, thus simplifying the model and improving computational efficiency. Here’s a high-level overview of its architecture:

1. Entry Flow: This part consists of convolution and max-pooling layers that downsample the input image.
2. Middle Flow: It has multiple residual blocks with depthwise separable convolutions that capture intricate spatial features.
3. Exit Flow: This segment transitions the middle flow’s output to the final classification layers.

## 2.2 Mathematical Functions and Layers

### 1. Standard Convolution

A standard convolution operation in a neural network can be represented as:



where:

* *X* is the input feature map.
* *W* is the weight of the filter.
* *bk* is the bias term.
* *Z* is the output feature map.

### 2. Depthwise Separable Convolution

In the Xception model, the depthwise separable convolution replaces the standard convolution. This operation is split into two parts:

* Depthwise Convolution: Applies a single filter per input channel.



* Pointwise Convolution: Uses a 1x1 convolution to combine the outputs of the depthwise convolution across channels.



Here, Z is the output of the depthwise convolution, and W1,1 represents the 1x1 convolution kernel.

#### 2.2.1 Entry Flow

The entry flow in the Xception model consists of several convolutional layers followed by max-pooling layers. The primary goal here is to downsample the image while capturing essential features.

* Initial Convolution: 2D convolution with 32 filters.



* Max-Pooling: Reduces the spatial dimensions (height and width) while retaining the most critical information.



## 2.3 Middle Flow

The middle flow consists of multiple blocks of depthwise separable convolutions. Each block has:

1. Depthwise Convolution
2. Batch Normalization
3. Pointwise Convolution
4. Residual Connection

The residual connection adds the input of each block to its output, which helps in training deeper networks by mitigating the vanishing gradient problem[10].



## 2.4 Exit Flow

The exit flow transitions the feature maps from the middle flow to the final classification layers. It involves depthwise separable convolutions followed by global average pooling and fully connected layers for classification.

* Global Average Pooling: Reduces each feature map to a single value by averaging.

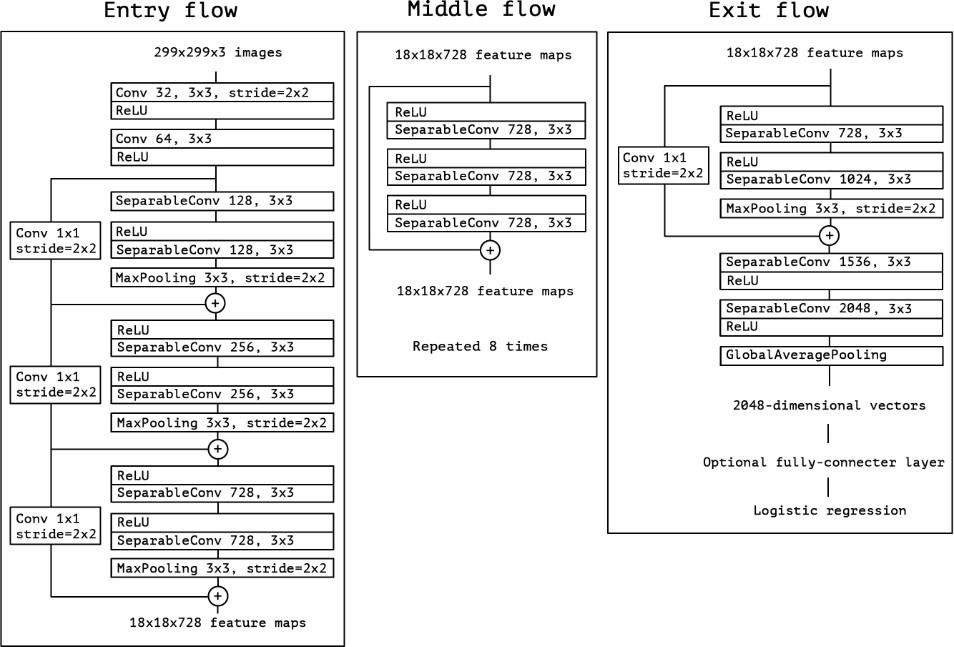


* Fully Connected Layer: Maps the pooled features to the final output classes.



## 2.5 Visualization

Below is a simplified diagram of the Xception model’s architecture:



*Fig1: Xception model architecture [4].*

## 2.6 Advantages of Xception

1. Efficiency: Depthwise separable convolutions significantly reduce computational cost compared to standard convolutions.
2. Performance: Despite its simplicity, Xception often outperforms more complex models on various image classification tasks.

By utilizing depthwise separable convolutions, Xception separates spatial and channel-wise feature learning, leading to a more efficient and effective model architecture.

## 2.7 Conclusion

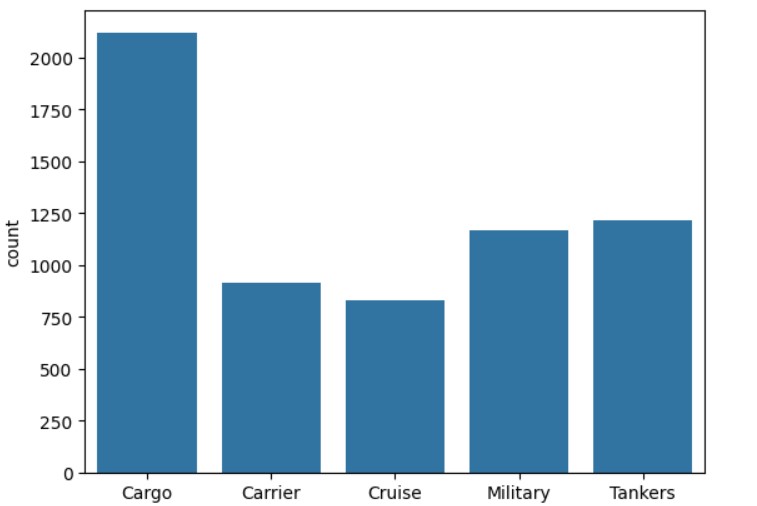
The Xception model’s innovative use of depthwise separable convolutions streamlines the learning process while maintaining high performance, making it a valuable tool in the realm of convolutional neural networks.

# 3 Results

This section details the methodology employed in developing the ship classification model using the Xception architecture, focusing on data preparation, model architecture, training procedures, and evaluation metrics.

## 3.1 Data Preparation

The dataset comprises 6252 images of ships, categorized into five classes: Cargo, Military, Carrier, Cruise, and Tankers [1]. The dataset was split into training (60%), validation (20%), and test (20%) sets.



*Fig2: The graph illustrates the proportion of each ship type in the dataset.*



*Fig3:* *Illustrative images for each type of ship.*

Image data were loaded and preprocessed using the following steps:

1. Loading and Preprocessing Images:
   * Images were loaded from the specified directory using the load\_img function from TensorFlow Keras, resized to 224x224 pixels [13][14].
   * Each image was converted to an array and normalized by scaling pixel values to the range [0,

1].

1. One-Hot Encoding of Labels: The ship categories were converted to numerical labels and then one-hot encoded using OneHotEncoder from scikit-learn to facilitate categorical cross-entropy loss computation [12].

1. Data Augmentation: Data augmentation was applied to the training set using ImageDataGenerator from TensorFlow Keras [14]. Augmentation techniques included rotation, horizontal flipping, and width and height shifts to increase the diversity of the training data and improve the model’s robustness[13].

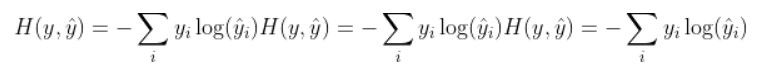


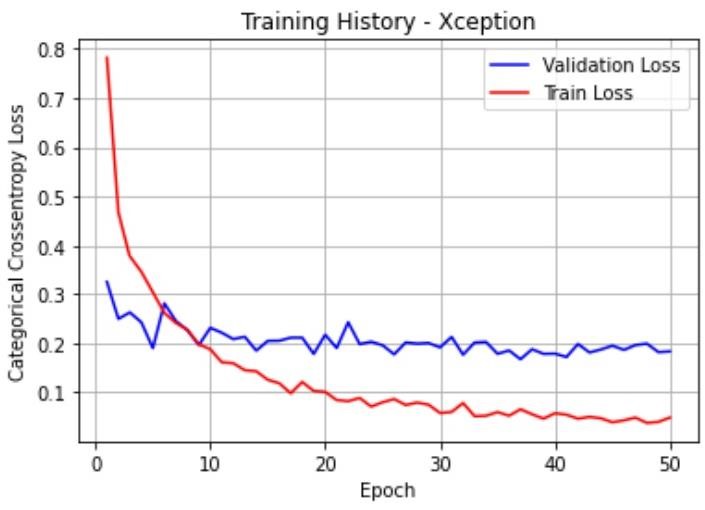
*Fig4: Examples of augmented images: rotation, flipping, and shifting.*

## 3.2 Model Architecture

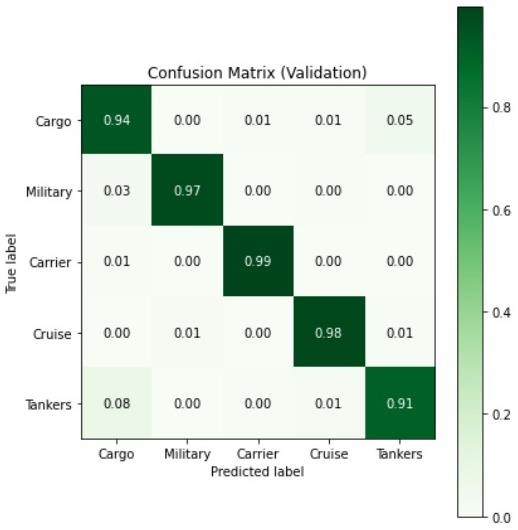
The Xception model, a deep Convolutional Neural Network (CNN) architecture pre-trained on the ImageNet dataset, was used as the base model. Key steps in the model setup included:

1. Loading the Pre-Trained Xception Model: The base Xception model was imported without its top (fully connected) layers using Xception from TensorFlow Keras applications, configured to accept input images of size 224x224x3.
2. Adding Custom Layers:
   * A Global Average Pooling layer was added to the output of the Xception model to reduce the spatial dimensions.
   * A Dense layer with a softmax activation function was appended to output predictions for the five ship classes.
3. Model Compilation:
   * The model was compiled using the Adam optimizer with a learning rate of 0.0001.
   * The loss function was set to categorical cross-entropy, and the accuracy metric was used for evaluation.

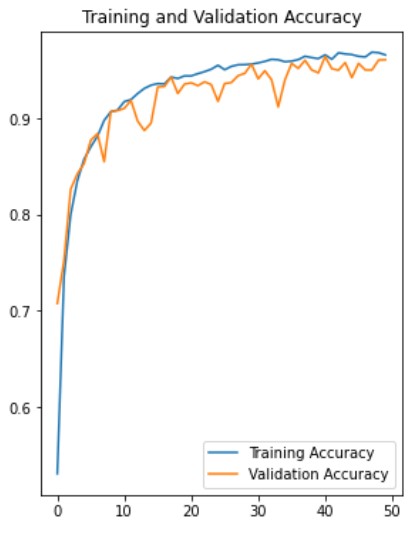




*Fig5: The plot illustrates the train-validation loss curve.*



*Fig6: The validation set's confusion matrix summarizes the model's classification performance.*



*Fig7: Accuracy plot for training and validation sets.*

## 3.3 Training Procedures

The model was trained using the following procedures:

1. Train-Validation Split: The training set was further split into training and validation subsets using stratified sampling to maintain consistent class distributions.
2. Callbacks and Checkpoints:
   * Custom callback functions were implemented to monitor the training process and save the best model based on validation loss.
   * The ModelCheckpoint callback was used to save the model with the lowest validation loss.
   * A LearningRateScheduler callback was used to adjust the learning rate dynamically during training.
   * The printf1 custom callback was used to print the F1 score at the end of each epoch for both training and validation sets.
3. Model Training:
   * The model was trained for 50 epochs with a batch size of 8 using the augmented data from the ImageDataGenerator.
   * The training history, including the loss and accuracy metrics, was recorded for analysis.

## 3.4 Evaluation Metrics

The model’s performance was evaluated using the F1 score and confusion matrix:

1. F1 Score: The F1 score, a measure of a model’s accuracy considering both precision and recall, was computed for both the validation and test sets[15].

|  |  |
| --- | --- |
| Best Model Weighted F1 Score (Validation) | 0.9533 |
| Weighted F1 Score (Test) | 0.9536 |

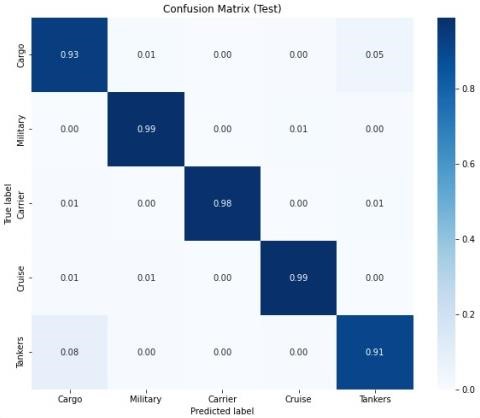
*Table1:* *Performance metrics (weighted F1 scores) for the best model on the validation and test sets.*

1. Confusion Matrix: Confusion matrices for the validation and test sets were plotted to visualize the classification performance across different ship categories [14].

## 3.5 Model Evaluation

The best model was selected based on the validation loss, and its performance was evaluated on the test set:

1. Loading the Best Model: The best model, saved during training, was loaded and used to predict the test set labels.
2. Performance Metrics: The F1 score and confusion matrix for the test set were computed and compared with the validation results to assess the model’s generalization ability.



*Fig8: The confusion matrix illustrates the classification performance of the model on the test set.*

# 4 Discussion

The Xception model significantly enhances convolutional neural networks (CNNs) by employing depthwise separable convolutions, which improve computational efficiency and performance [9]. This approach allows for effective feature extraction with fewer parameters, making Xception both powerful and resource efficient.

Compared to traditional models like VGG and Inception v3, Xception achieves superior accuracy while reducing complexity [8]. However, it requires substantial memory due to numerous intermediate feature maps and relies on careful hyperparameter tuning, which can be time-consuming.

Future research could focus on optimizing Xception for low-power devices and integrating attention mechanisms to further enhance performance [11]. Additionally, exploring its applications in other domains, such as medical imaging and autonomous driving, could validate its versatility and efficacy [17].

Overall, Xception represents a significant step forward in CNN architecture, balancing efficiency high performance, and setting a new standard for future deep learning models.

# 5 Conclusion

The Xception model represents a significant advancement in the field of convolutional neural networks (CNNs) by introducing depthwise separable convolutions. This architectural innovation allows for a more efficient and effective extraction of spatial and channel-wise features, resulting in a model that is both computationally less expensive and highly accurate [9].

By simplifying the traditional convolution process, Xception achieves superior performance compared to earlier models such as VGG, ResNet, and Inception v3, particularly in large-scale image classification tasks like those found in the ImageNet dataset[7][8][]. The model's success underscores the importance of architectural efficiency and modular design in deep learning, offering a compelling balance between complexity and performance.

Despite its advantages, the Xception model does have limitations, including increased memory consumption and a dependency on careful hyperparameter tuning. These challenges highlight areas for future research, such as integrating attention mechanisms, optimizing for low-power devices, and extending the model's application to diverse fields like medical imaging and autonomous driving [11][17].

In conclusion, the Xception model's use of depthwise separable convolutions marks a pivotal development in CNN architecture, providing a foundation for future innovations in the design of efficient, high-performance neural networks [9].

# References

1. Analytics Vidhya. (n.d.). *Game of Deep Learning: Computer Vision Hackathon*. Retrieved May 18, 2024. Retrieved fro[m https://datahack.analyticsvidhya.com/contest/game-of-deep-learning/](https://datahack.analyticsvidhya.com/contest/game-of-deep-learning/)
2. EconoFact. (2022). The Role of Shipping in World Trade. Retrieved from https://econofact.org/the-role-of-shippingin-world-trade
3. United Nations News. (2021). UN Report: The Shipping Industry and Global Trade. Retrieved from https://news.un.org/en/story/2021/11/1106082
4. Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
5. Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.
6. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
7. Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.
8. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
9. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
10. Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
12. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
13. TensorFlow Keras Documentation. (n.d.). Retrieved May 18, 2024, from https://www.tensorflow.org/api\_docs/python/tf/keras
14. Keras Documentation. (n.d.). Retrieved May 18, 2024, from https://keras.io/api/
15. Chollet, F. (2015). Keras. Retrieved from https://keras.io
16. Lever, J., Krzywinski, M., & Altman, N. (2016). Model selection and overfitting. Nature Methods, 13(9), 703-704.
17. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., & van Ginneken, B. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.