Image Classification of Everyday Objects using Convolutional Neural Networks

Team Members:

- Naveen Seelam : 002794970
- Rahul Kumar : 002779804
- Vaishnavi Mada : 002775010
- Dipak Bhattarai : 002776875
- An Nguyen : 900935293

Introduction

In the rapidly evolving realm of image detection, precision in classifying images into predefined categories is paramount. This report delves into Convolutional Neural Networks (CNNs) to tackle the challenge of image classification for everyday objects, with a focus on discerning between airplanes, ships, and cars. Leveraging the deep learning capabilities of CNNs, our objective is to develop a robust model capable of accurately capturing the subtle nuances that distinguish these categories, thereby achieving high levels of accuracy and reliability in our predictions.

As digital imagery proliferates across diverse domains, the demand for automated systems capable of comprehending and categorizing visual data has surged. Convolutional Neural Networks have emerged as a potent solution, showcasing remarkable efficacy in tasks spanning image recognition to object detection. **Our project endeavors to harness the potential of CNNs to address the intricacies inherent in classifying everyday objects. By training a model that not only distinguishes between airplanes, ships, and cars** but also captures nuanced differences, we aim to contribute to advancements in image classification, paving the way for practical applications in transportation, surveillance, and autonomous systems. Through this exploration, we seek to unlock new avenues for innovation and progress in the burgeoning field of computer vision.

Dataset Overview

The dataset consists of **3,000 training images and 582 test images**, meticulously balanced across three distinct classes: airplanes, ships, and cars. Each image is a vibrant color photograph sized at **1200x1200** pixels, ensuring ample detail and clarity for comprehensive analysis.

The dataset was sourced from Kaggle, a renowned platform for datasets and data science competitions. You can access the dataset via the following URL:
https://www.kaggle.com/datasets/abtabm/multiclassimagedatasetairplanecar

```
Train class counts: {'airplanes': 1000, 'cars': 1000, 'ship': 1000}
Train image shape: (1200, 1200)
Test class counts: {'airplanes': 189, 'cars': 193, 'ships': 200}
Test image shape: (1200, 1200)
```

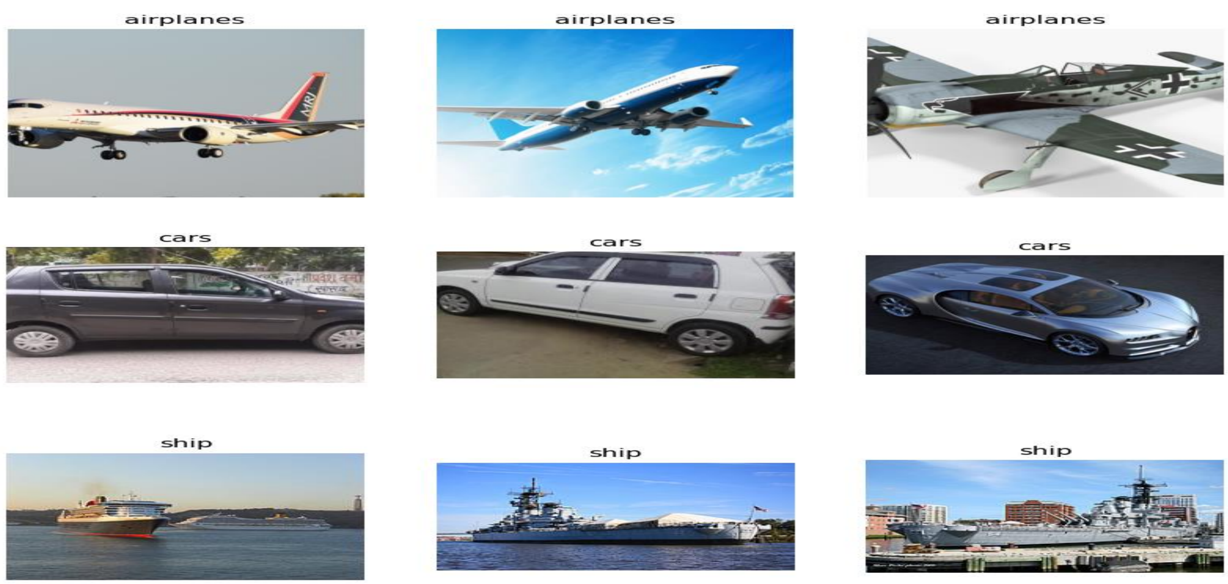
Data Cleaning and Feature Extraction

Our initial inspection confirmed the absence of corrupted images within the dataset. To standardize the input for our CNN, all images were **resized to 224x224** pixels, and pixel values were **normalized to a [0, 1]** range to facilitate model convergence. We employed data augmentation techniques such as rotation, zoom, and horizontal flipping to enhance the dataset's diversity and improve the model's generalization capabilities.

Task Definition

At the heart of this analysis lies a multi-class image classification task, wherein our objective is to precisely categorize images into one of three classes: airplane, ship, or car. The accuracy of these predictions holds immense significance across a multitude of applications, spanning from automated surveillance systems and content categorization platforms to educational tools and beyond. By effectively distinguishing between these categories, we can unlock enhanced capabilities in object recognition, enabling advancements in diverse fields and fostering innovation in artificial intelligence-driven solutions.

Displaying sample images in each class



Checking for corrupt images

During our meticulous examination of the dataset, we conducted thorough checks to identify any corrupted images. Delightfully, **our efforts yielded a result of 0 corrupted images** found. This assurance underscores the integrity and reliability of the dataset, instilling confidence in the quality of our analysis and the subsequent model development process.

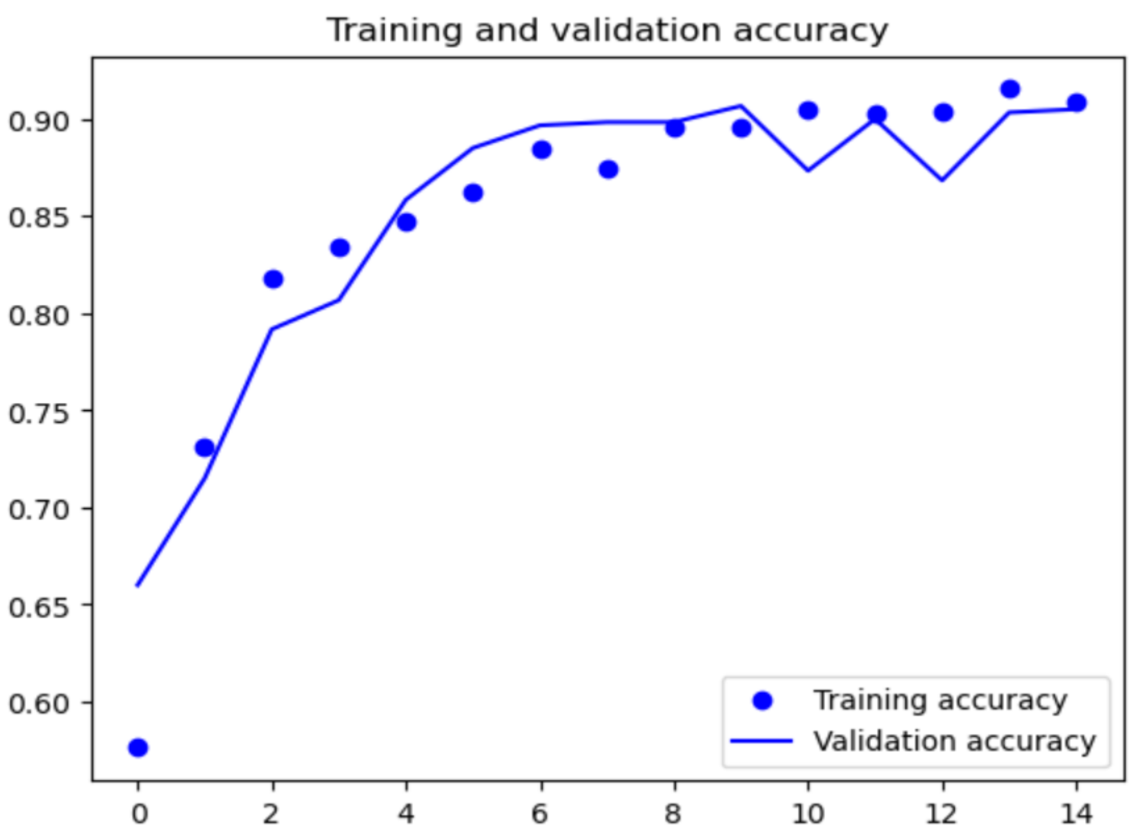
```
Found 0 corrupted images.
```

Convolutional Neural Network Analysis

Our CNN architecture comprises **three convolutional layers featuring ReLU activation and max-pooling, succeeded by a flattening operation and two dense layers.** To produce probabilities for the three classes, the final layer utilizes SoftMax activation. We opted for this architecture due to its adept balance between model complexity and computational efficiency. Training the model entailed employing the Adam optimizer alongside a categorical cross-entropy loss function across **30 epochs**, with early stopping implemented to alleviate overfitting concerns.

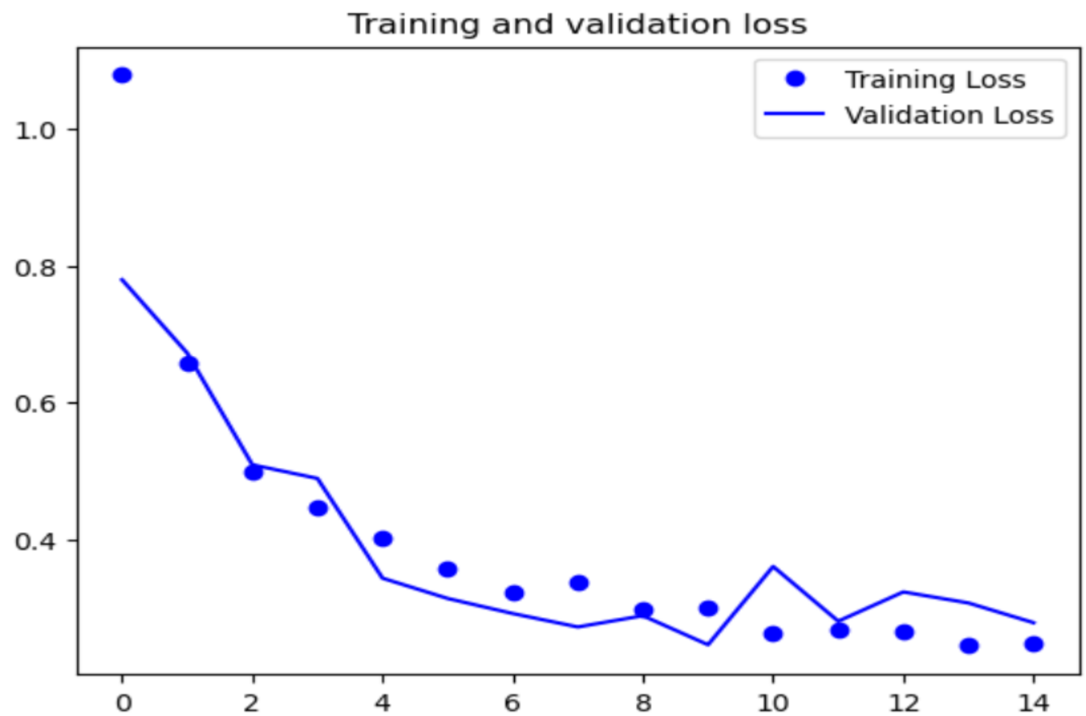
Plot for training and validation accuracy

The x-axis represents the time steps (epochs), and the y-axis represents the accuracy.The training and validation accuracy graph shows an increase in training accuracy over time, suggesting that the model is learning from the data. The validation accuracy also increases but shows some variability, again potentially indicating overfitting as the model learns to adapt too closely to the training data at the cost of its generalization to the validation data. Overall, the model performs well, achieving high accuracy on both training and validation sets.



Plot for training and validation loss

The x-axis represents the training epochs (iterations), and the y-axis represents the loss value. The training and validation loss graph shows a typical pattern where the training loss decreases sharply initially and then levels off, while the validation loss decreases and then exhibits some fluctuations. This could indicate that the model is starting to overfit the training data after the initial epochs, as evidenced by the divergence that appears later in training.



Results and Discussion

Evaluating the model on the test dataset yielded a **test accuracy of 90.17%** and **a loss of 0.293**, showcasing a high level of performance in classifying the images. The progression of accuracy and loss throughout the epochs revealed swift improvements early in the training, which then stabilized, highlighting the CNN's ability to effectively extract and learn meaningful features from the image data, thus distinguishing accurately between the three classes.

```
Found 582 images belonging to 3 classes.
29/29 [==============================] - 14s 453ms/step - loss: 0.2932 - accuracy: 0.9017
Test Loss: 0.29320842027664185
Test Accuracy: 0.9017241597175598
```

Conclusions and Suggestions

The CNN model devised in this analysis proficiently tackles the image classification task, with the results indicating a robust performance. Incorporating a more diverse and voluminous dataset could further challenge and enhance the model's generalization capability for future endeavors. Additionally, experimenting with deeper architectures and more sophisticated regularization techniques may provide deeper insights into the model's performance and potential areas for improvement.