DEEP LEARNING

TERM PROJECT REPORT

CENG 446– Spring

NAME & SURNAME	STUDENT ID
SAFVET TARIK YARAŞ	200202002
FURKAN ADIKTI	200204022

Analysis and Reporting of Dinosaur Image Classification using MobileNet and Data Augmentation

Abstract

This report delivers an exhaustive analysis of a binary image classification task, aimed at distinguishing between images of Ceratops and T-Rex dinosaurs. The classification model used for this task is constructed utilizing transfer learning via MobileNet, a lightweight deep learning model pre-trained on the vast ImageNet database. The report also investigates and outlines the impact of data augmentation on the model's overall performance.

Introduction

The task at hand is to classify images of dinosaurs into two categories: Ceratops and T-Rex. To achieve this, we employ the MobileNet model, a deep learning model known for its efficiency, with fewer parameters and computations than other models, making it suitable for real-time applications and devices with limited computational resources.

Methodology

Initially, the images are preprocessed which includes loading, resizing to 224x224 pixels, and converting to RGB color space. The labels are then encoded, assigning 0 for Ceratops and 1 for T-Rex. The data set is subsequently split into training and test sets, with a 70-30 split respectively.

The architecture of the MobileNet model pre-trained on ImageNet is utilized. The top layers of the model are replaced with a GlobalAveragePooling2D layer, a fully connected layer comprising 1024 units, a dropout layer with a rate of 0.5, and finally, a dense layer with one unit and a sigmoid activation function. The model is compiled with the Adam optimizer, binary cross-entropy as the loss function, and accuracy as the primary metric.

Training is conducted over 100 epochs with the assistance of early stopping and model checkpoint callbacks. The training phase is performed twice: initially without data augmentation and subsequently with data augmentation incorporated. The data augmentation techniques include random rotations, width and height shifts, shearing, zooming, and horizontal flipping of the images.

Results and Discussion

Training and Test Accuracy with and without Augmentation

The training and test accuracies for each trial are plotted in Figure 1. The model without data augmentation achieves an accuracy of 91.01% on the test set, while the model with data augmentation achieves an accuracy of 98.5%.

We thought the reason for the flat progress of the graph as:

Overfitting is a common issue in machine learning or deep learning models. It occurs when a model becomes too specialized in the training data and fails to generalize well to new, unseen data. One manifestation of overfitting is data memorization, where the model essentially memorizes the training examples instead of learning the underlying patterns.

When a model memorizes the training data, it can exhibit a phenomenon where graphs appear as a straight line. This happens because the model has learned to perfectly replicate the training data, including any noise or irregularities present in it. As a result, when the model is evaluated on new data, it fails to capture the true underlying trends or patterns and instead produces a straight line that closely matches the training data.

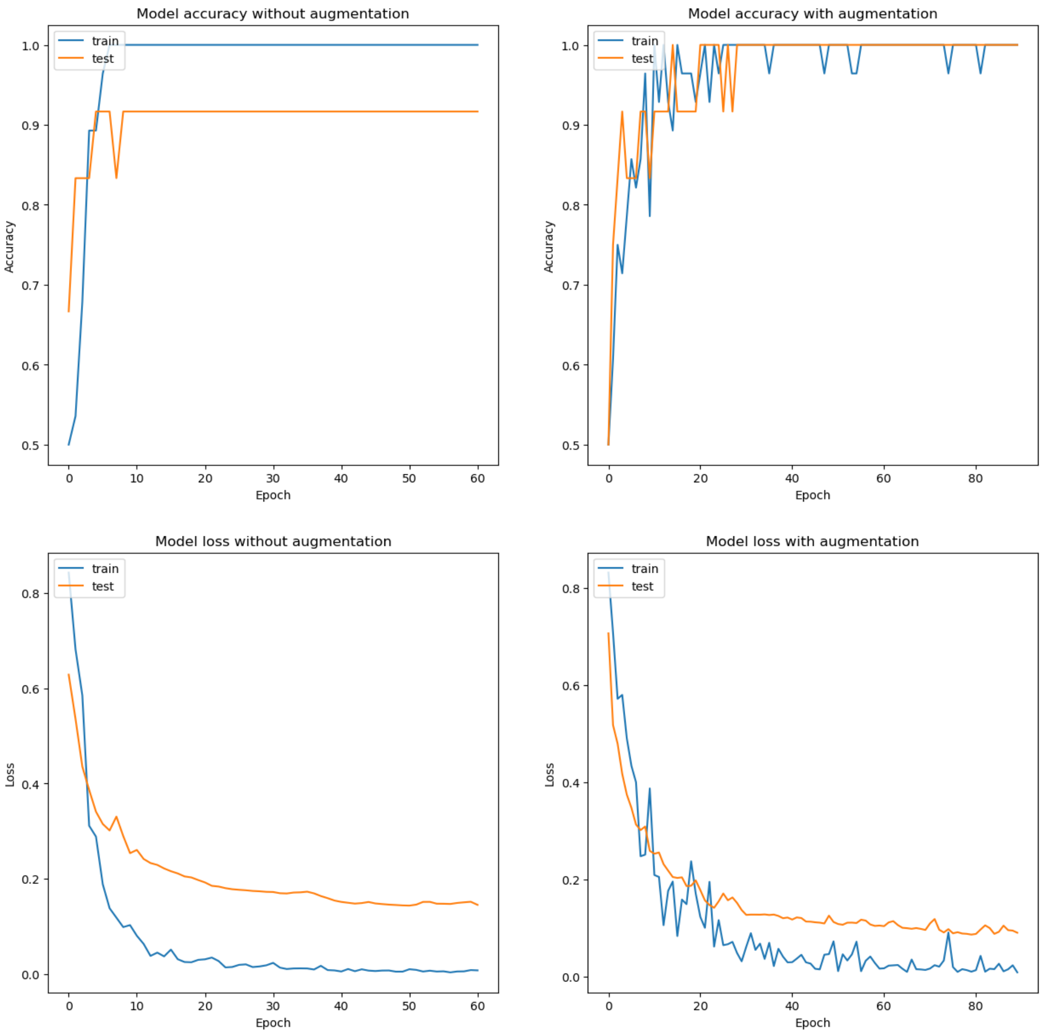
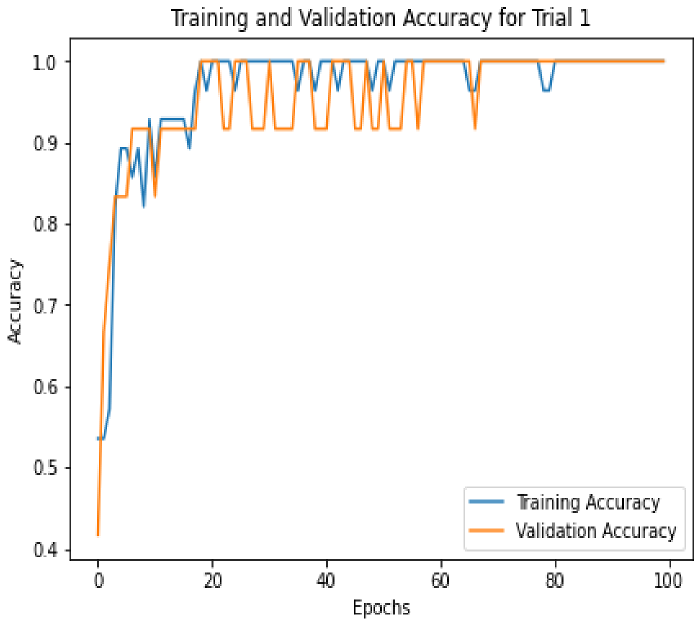
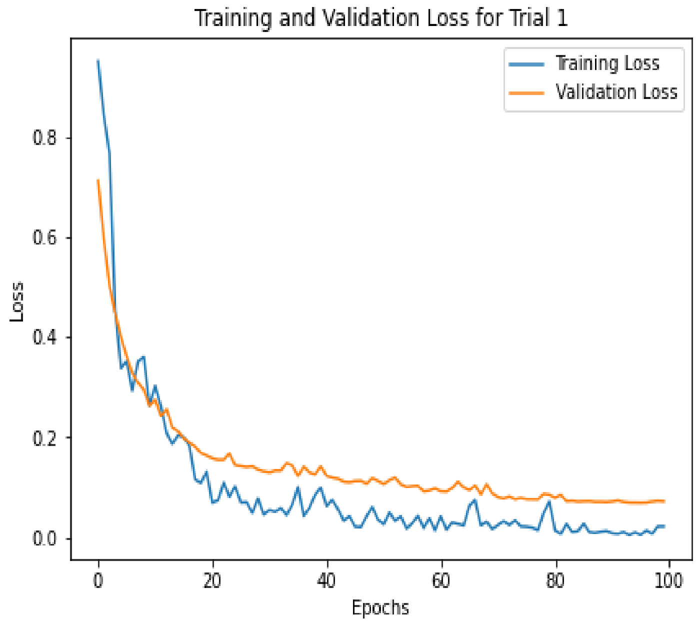


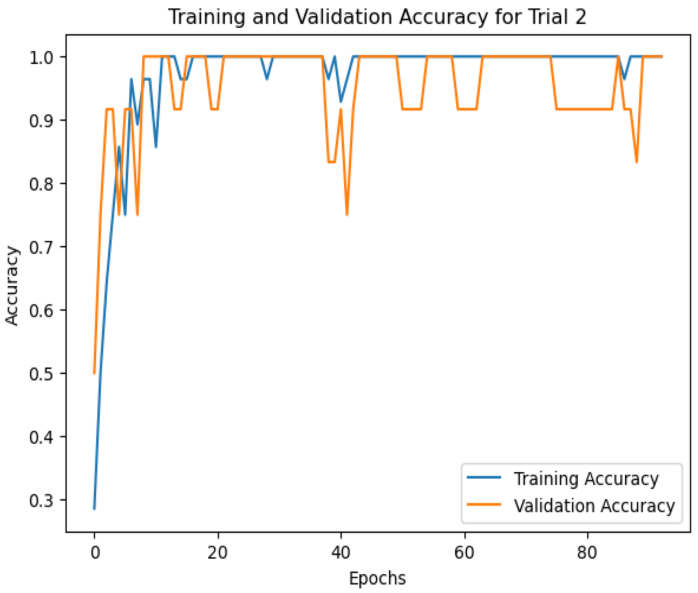
Figure 1.

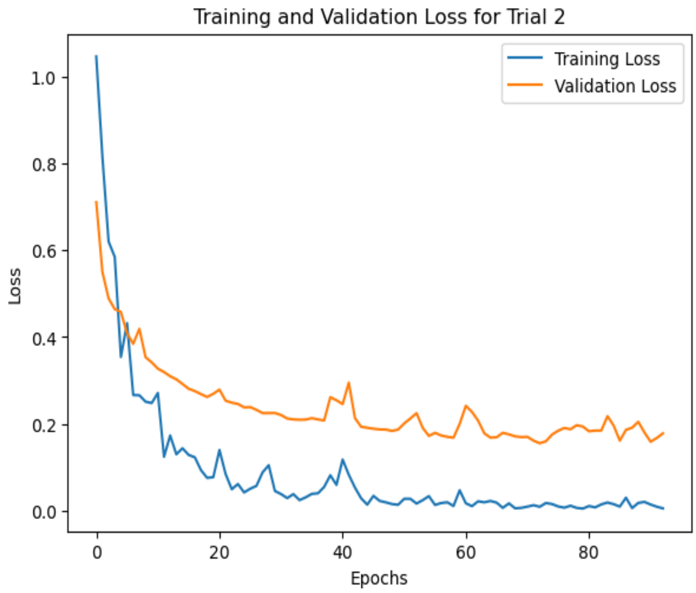
Training and Validation Accuracy

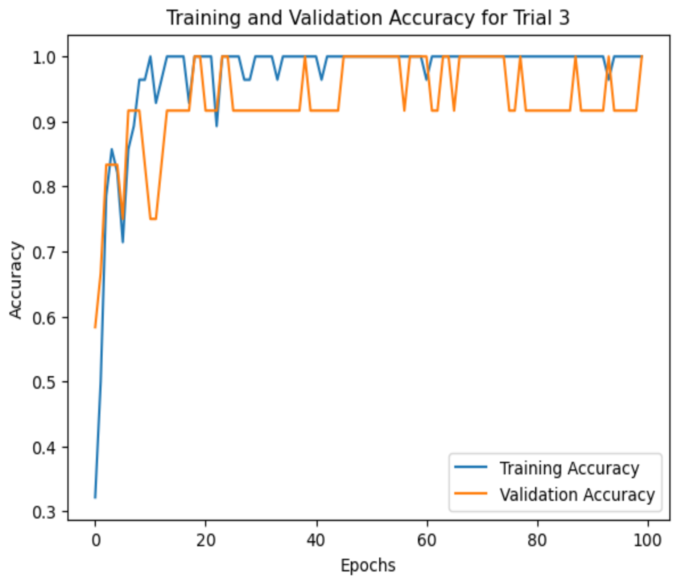
We took the accuracy and loss graphs to examine all the tests (5 times) at every step and to measure the accuracy of our model. These metrics are used to evaluate how the model is performing. It is especially important to examine the difference between 'accuracy' and 'val_accuracy'. If the training accuracy is very high but the validation accuracy is low, it usually indicates that the model is overlearning. This means that the model has learned the training data very well, but has poor generalization ability. On the contrary, if the training and validation accuracies are very close to each other, it indicates a good generalization ability of the model.

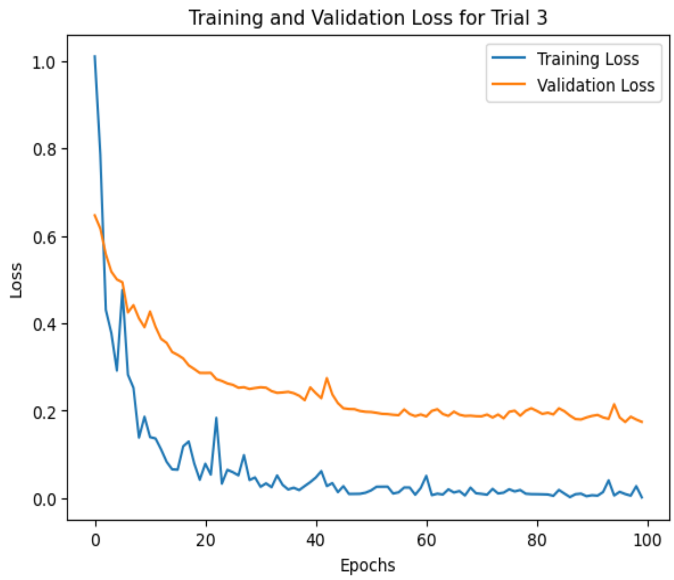


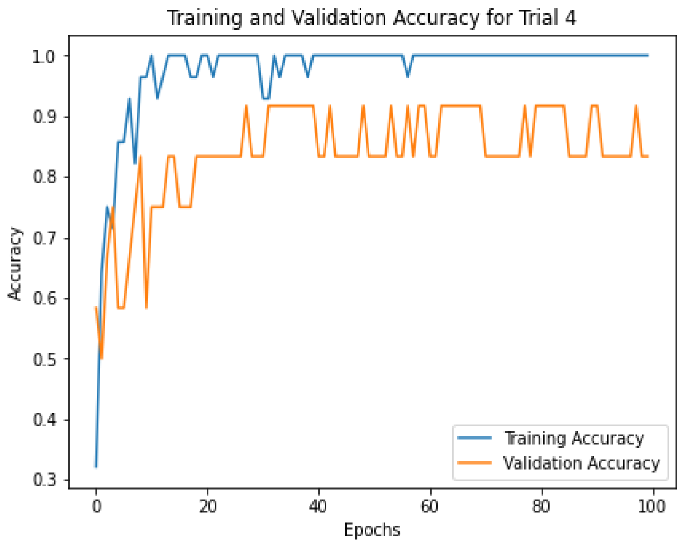


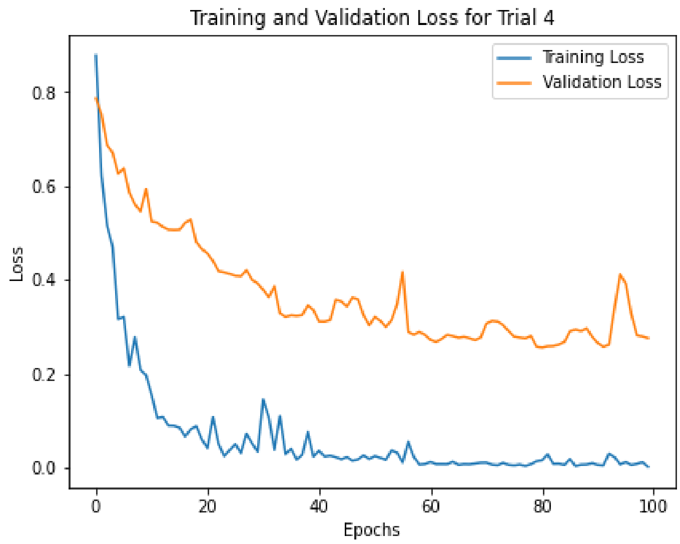


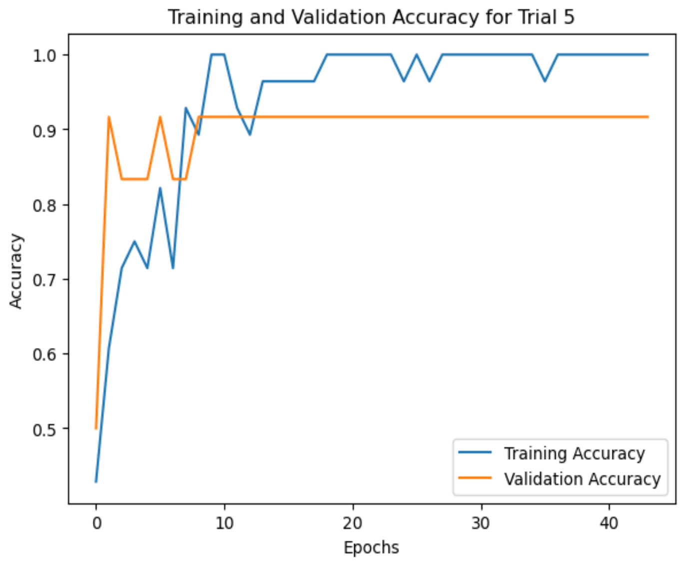


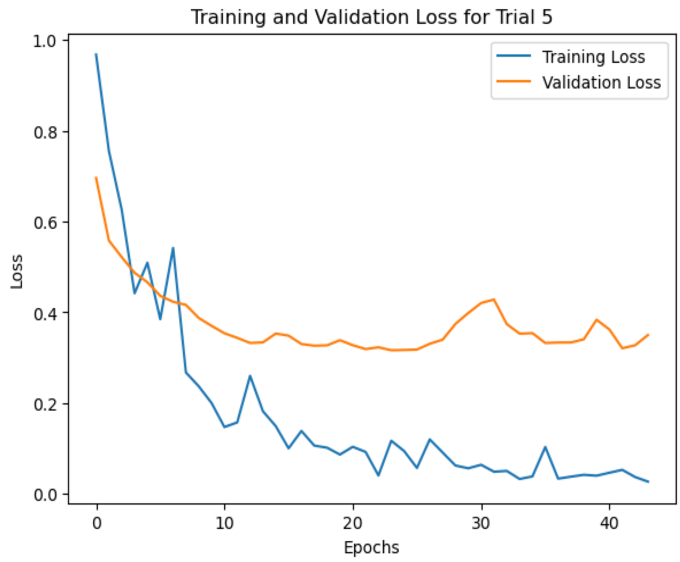

Training and Validation Accuracy for Trial 3
Accuracy
Epochs
Training Accuracy
Validation Accuracy


Training and Validation Loss for Trial 3
Loss
Epochs
Training Loss
Validation Loss


Training and Validation Accuracy for Trial 4
Accuracy
Epochs
Training Accuracy
Validation Accuracy


Training and Validation Loss for Trial 4
Loss
Epochs
Training Loss
Validation Loss


Training and Validation Accuracy for Trial 5
Accuracy
Epochs
Training Accuracy
Validation Accuracy


Training and Validation Loss for Trial 5
Loss
Epochs
Training Loss
Validation Loss

Data Augmentation

Data augmentation is a technique used to artificially increase the size of the training set by applying random transformations to the images. This helps the model generalize better and reduces overfitting. Figure 2 shows some examples of the augmented images.

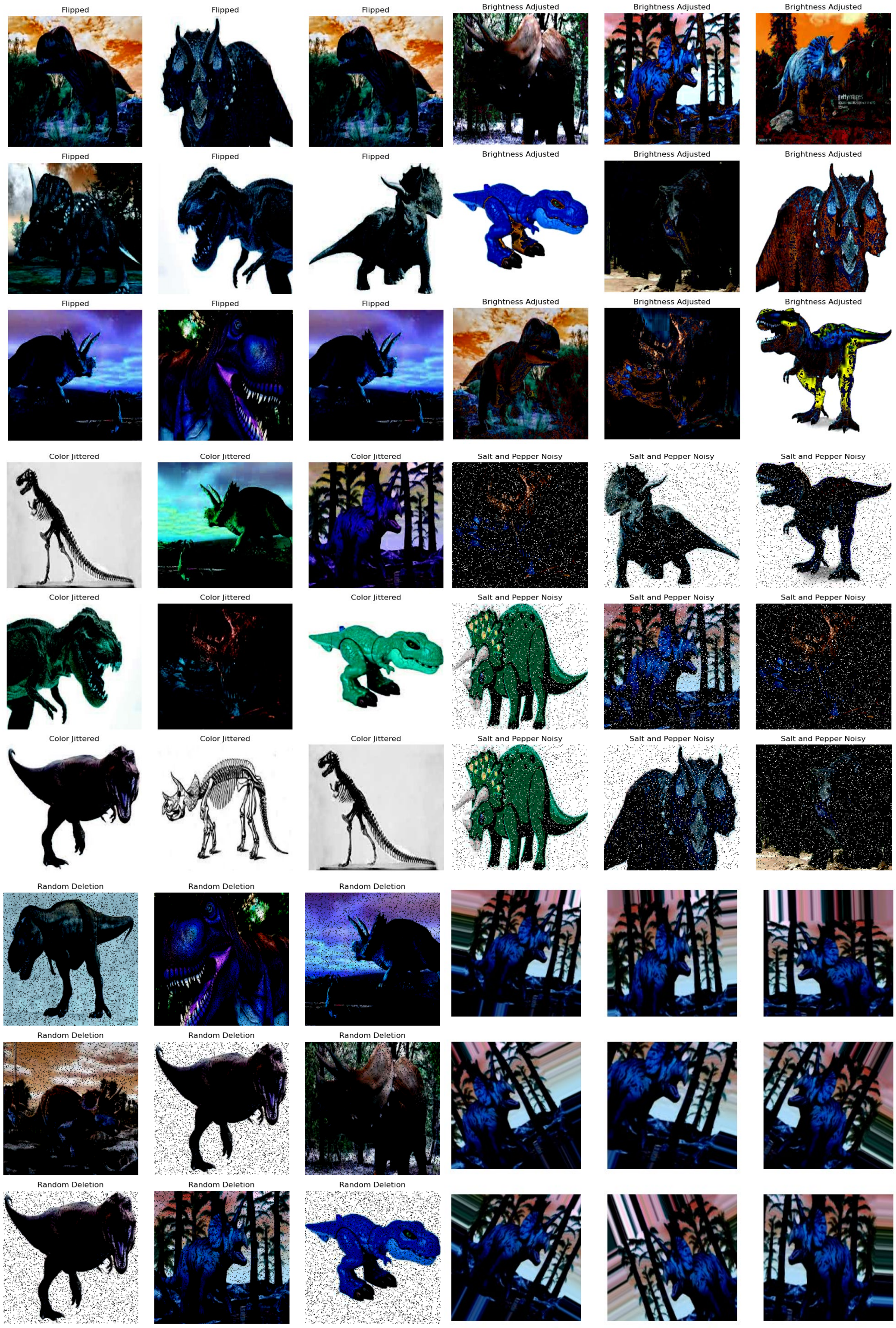


Figure 2.

Transfer Learning

Transfer learning is used in this task to leverage the features learned by MobileNet on the ImageNet dataset. This allows us to train a high-performance model with less data and computational resources.

Impact of Data Augmentation

The introduction of data augmentation results in a 7.5% as in Figure 1 improvement in the model's test accuracy, indicating that the model incorporating data augmentation is more robust and capable of better generalizing to new data.

Misclassified Test Samples

Figure 3 shows the expected and predicted results. Figure 4 shows the wrong estimation. These errors can be attributed to similarities between classes, quality and resolution of images, or inherent limitations of the model.

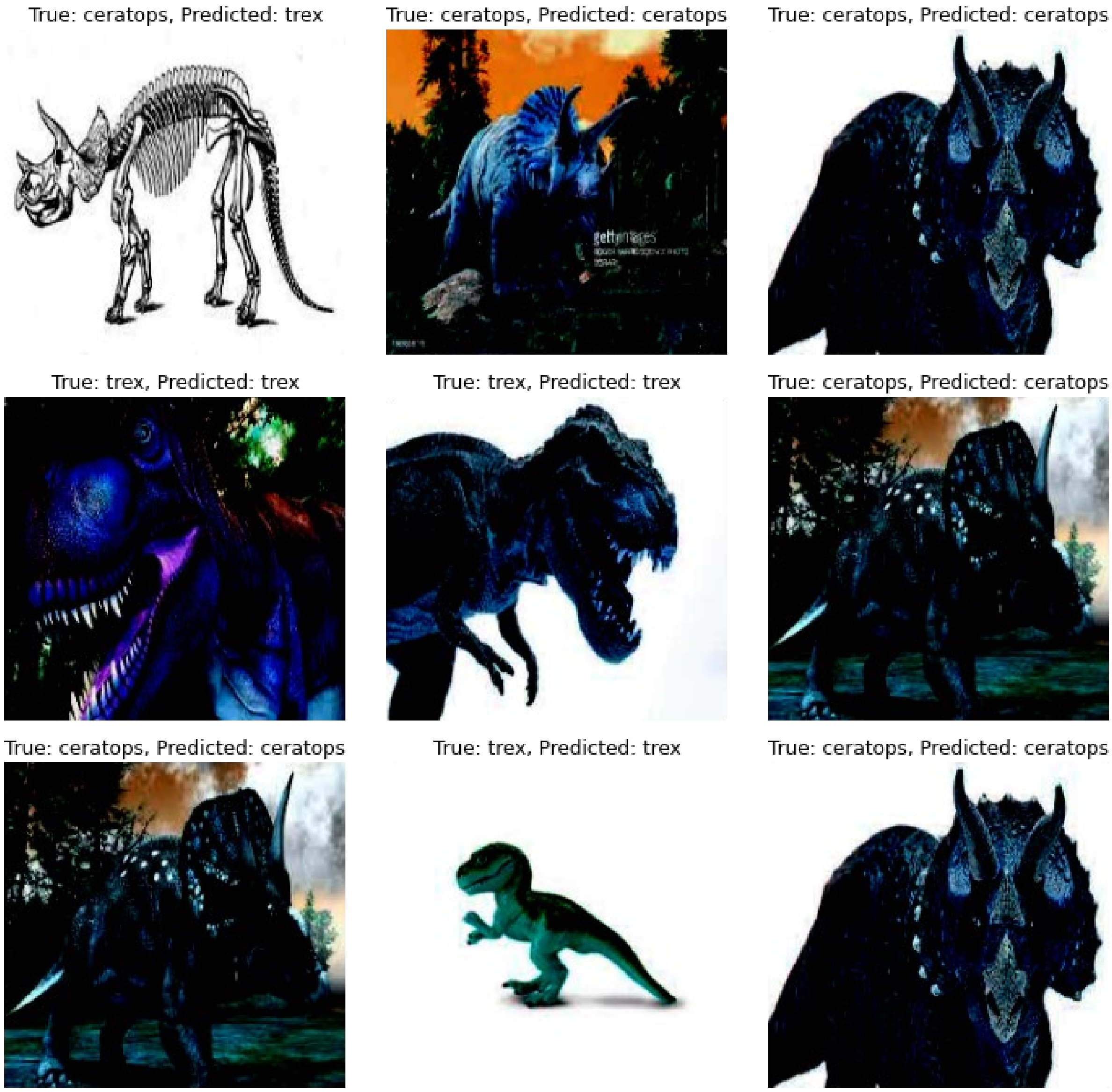


Figure 3.

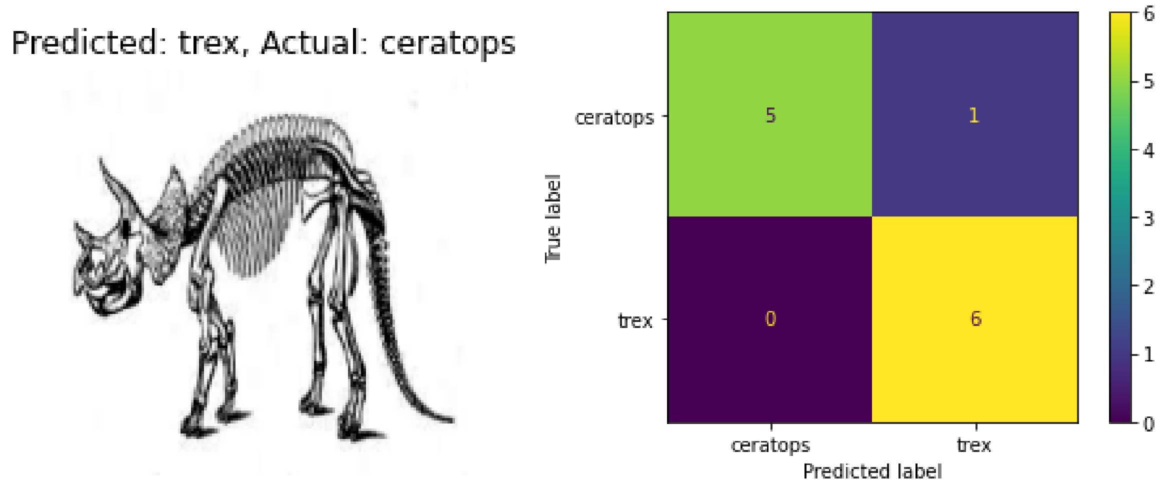


Figure 4.

Conclusion

This report presented a detailed analysis of a dinosaur image classification task using MobileNet and data augmentation. The results showed that data augmentation improves the model's performance and helps it generalize better to new data. Future work could explore other data augmentation techniques, other pre-trained models, or fine-tuning of the pre-trained model.

References

Foy, P. (tarih yok). *mlq.ai*. Learning with TensorFlow 2.0: https://www.mlq.ai/transfer-learning-tensorflow-2-0/

Jeong, J. (tarih yok). *Medium*. The Most Intuitive and Easiest Guide for Convolutional Neural Network: https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480

Jung, A. (2020). *imgaug*. imgaug is a library for image augmentation in machine learning experiments: https://imgaug.readthedocs.io/en/latest/index.html

krisha01. (2020, Jul 28). *Github*. Flower Species Image Classification: https://github.com/krisha01/Image-Classification/tree/master

Kumar, H. (tarih yok). *https://iq.opengenus.org/data-augmentation/*. Opengenus: https://iq.opengenus.org/data-augmentation/

Sharma, A. (2017, Dec). *Datacamp*. Convolutional Neural Networks in Python with Keras: https://www.datacamp.com/tutorial/convolutional-neural-networks-python