**Text

Description automatically generated**

A picture containing graphical user interface

Description automatically generated

**CIS6005 COMPUTATIONAL INTELLIGENCE**

<https://github.com/ebubesamuel/CIS6005_COMP_INT>

**23rdJanuary, 2024**

STUDENT NUMBER: ST20283947

NAME: SAMUEL CHIDIEBUBE CHIEMELA

Project Overview.......................................................3

Introduction........................................................................3

Dataset ...............................................................................3

Dataset Metadata......................................................3

Data Splitting.........................................................................3

Rationale for Validation Set.......................................4

Design of Neural Network.........................................4

Activation Function....................................................4

Loss Function............................................................5

Models Evaluation and Result...................................5

Model Structure and Architecture......................5

First Model........................................................5

Second Model....................................................6

Model Accuracy...................................................6

Comparison of Models............................................7

Accuracy and Loss..............................................7

Graphs.................................................................8

Classification Report...................................................9

Confuison Matrix........................................................10

Dataset Creation Experience.....................................10

Conclusion..................................................................10

Knowledge gained......................................................11

Table of Figures

Fig1: Splitting into Validation set................................................4

Fig 2: Building first model............................................................5

Fig 3: Building second model.......................................................6

Fig 4: First Model History.............................................................7

Fig 5: Second Model History........................................................7

Fig 6: Model Comparison.............................................................8

Fig 7: First Model Training vs Validation Accuracy......................8

Fig 8: Second Model Training vs Validation Accuracy.................8

Fig 9: First Model Training vs Validation Loss..............................9

Fig 10 : Second Model Training vs Validation Accuracy..............9

Fig 11: Classification Report........................................................10

Fig 12: Confusion Matrix..............................................................10

Project Overview

The project for this assignment is based on an image classification which takes in an input image and assigns a class label based on that input image. This project is developed on Convolutional Neural Networks(CNNs). These neural networks are crucial to this task as they help in identifying patterns in image input.

This Convolutional Neural Network is trained on the CIFAR-10 dataset and can be used in recognizing the following input images. E.g, an airplane, a dog, a frog, a horse, a ship, a truck, a car, a bird, a cat, a deer,

Introduction

This project model is of the classification type. The classification type involves building a machine learning model or program to classify data into predefined classes or categories – this case being the correlating class name to the given input image.

This project being a Multi-Class Classification project as we try to categorize given image input into 10 classes, this means that each data sample is assigned to a single label, i.e “ship”, which is assigned an image of a ship.

Dataset

For this project, the CIFAR-10 dataset from keras.datasets library is used. This dataset consists of 60,000 images of 32x32 pixels in the following classes with each having 6,000 dedicated images: an airplane, a dog, a frog, a horse, a ship, a truck, a car, a bird, a cat, a deer.

Dataset Metadata

The metadata for the CIFAR-10 datasets are:

* Number of images: 60,000
* Number of classes: 10
* Class distribution: 6,000 images (5,000 for training and 1,000 for testing/validation).
* Image size: 32x32 pixels
* Image format: colour(rgb)
* task: Object recognition

Data Splitting

After loading the CIFAR-10 dataset from the keras dataset library, the next step is to split it into a training set and a validation set. We do that by first splitting the original dataset into training set (80% of the data) and a validation set (20% of the data).

Out of the 60,000 images in the dataset, Training accounts for 40,000 images, testing accounts for 10,000 images and validation accounts for the further 10,000 images. Thereby making the validation 20% of training and testing data.

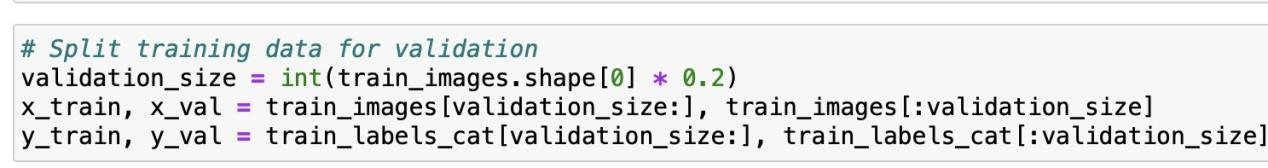


Figure 1: Splitting into Validation set

Rationale for Validation Set

There are different reasons to have a validation set when working on a classification project. Some reasons include:

1. **Model Evaluation during training**: The primary purpose of a validation set is to evaluate the performance of your machine learning or deep learning model during the training process.
2. **Avoiding Overfitting**: Overfitting occurs when a machine learning model learns to perform exceptionally well on the training data but fails to generalize to new, unseen data.

In the context of the CIFAR-10 dataset, using a validation set is crucial for optimizing and evaluating the performance of your classification models. It is used to select the best model among the two models by comparing their performance on the validation set. The best model can then be used to make predictions on the unseen test set.

Design of Neural Network

Stated above in the project overview, I stated the reason why I made use of a convolutional neural network to train the images. CNNs have the capacity to recognize specific patterns in images thus making them a valid choice to proceed with when tackling image classification projects.

In order to optimize the performance of a neural network, there asre some aspects to consider when creating the neural network, such as activation functions and loss functions. Below are the activation and loss functions used in designing my neural network:

Activation Functions

1. **Rectified Linear Unit (ReLU) activation function:**

ReLU is an activation function commonly used in Convolutional Neural Networks (CNNs) and other deep learning architectures. It is a piecewise linear function that introduces non-linearity to the model's decision-making process. I returns 0 for all negative input values and returns the same value for positive input values.

1. **Softmax activation function:**

This is a commonly used activation function in Convolutional Neural Networks (CNNs) and other neural network architectures, especially in the output layer of multi-class classification problems. It is used to convert the raw model outputs, often referred to as logits or scores, into a probability distribution over multiple classes. It is a generalization of the logistic function and it maps the input to a probability distribution over the output classes.

Loss Function

I employed the **categorical cross-entropy loss function** when creating this neural network. The reason to adopt this loss function is because we are dealing with CIFAR-10 dataset which is a multi-class classification problem with 10 classes. The primary purpose of a loss function is to guide the model's optimization process during training by providing a measure of how well or poorly the model is performing. Key aspects of loss functions include measuring the errors between the predicted values and the true target values. They also define the training objective for the model.

Models Evaluation and Results

Model Architecture

First Model

This convolutional nueral network (CNN) is designed for image classification. The network has a the following layers:

* One convolutional layer(Conv2D) with 32 filters, followed by a max pooling layer (MaxPooling2D) to reduce dimensionality and fish out the most important features.
* A flatten layer to reshape the feature maps into a 1-dimesnional array.
* A dense layer with 10 neurons and a softmax activation function to produce the probability scores for each class.

The first model makes use of the Adam optimizer and categorical **cross entropy** as loss function and **accuracy** as the evaluation metric.

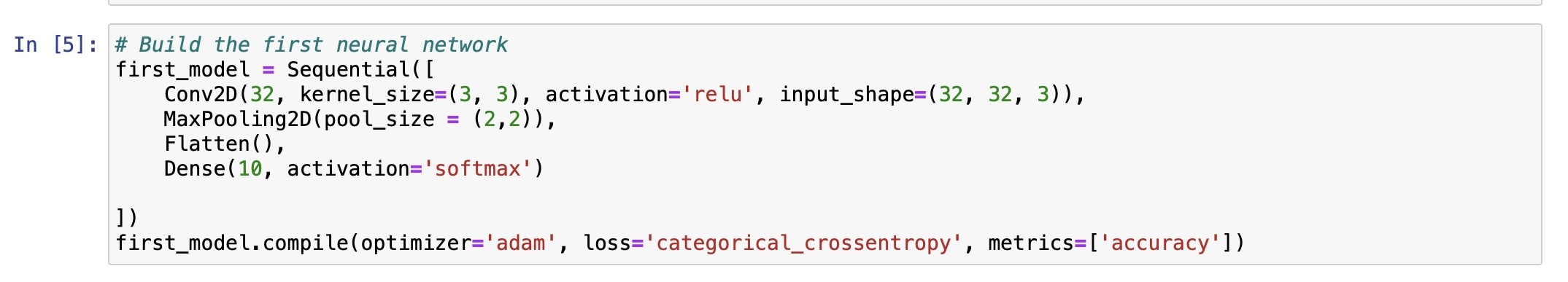


Fig 2: Building first model

Second Model

The structure of this neural network is a convolutional neural network (CNN) designed for image classification as well. The network has the following layers:

* Two convolutional layers (Conv2D) with 32 and 64 filters respectively, each Conv2D layer is followed by a max pooling layer (MaxPooling2D) to reduce dimesnionality and fish out the most important features.
* A flatten layer to reshape the feature maps into a 1-dimesnional array.
* A dense layer with 64 neurons with a ReLU activation function.
* A dense layer with 10 neurons and a softmax activation function to produce probability scores for each class

The second model makes use of the Adam (**Adam()**) optimizer as well, **categorical cross-entropy** loss function and **accuracy** as evaluation metric.

The architecture of the second model is more complex as it contains mod´re layers, more complex dense layers and filters. This is so that it can be able to extract more complex features from the images in the CIFAR-10 dataset.

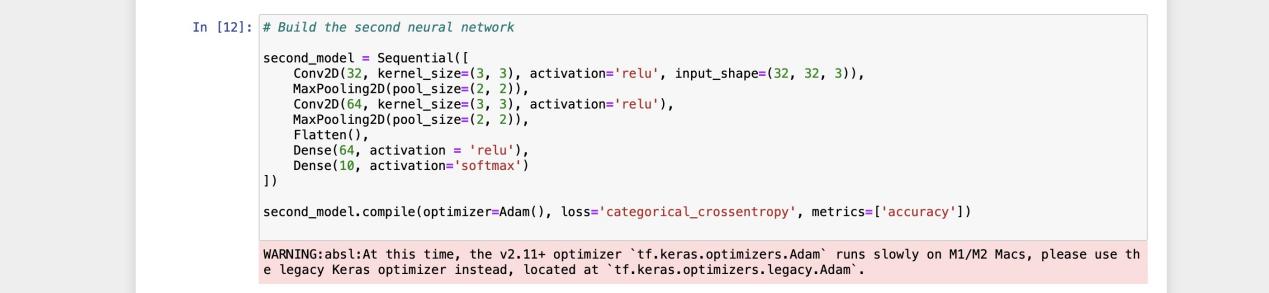


Fig 3: Building second model

Model Accuracy

Using 10 epochs, we can see that the accuracy for the first model improves after each epoch. With the first epochs accuracy being about **46%** and by the tenth the accuracy being **72%.**

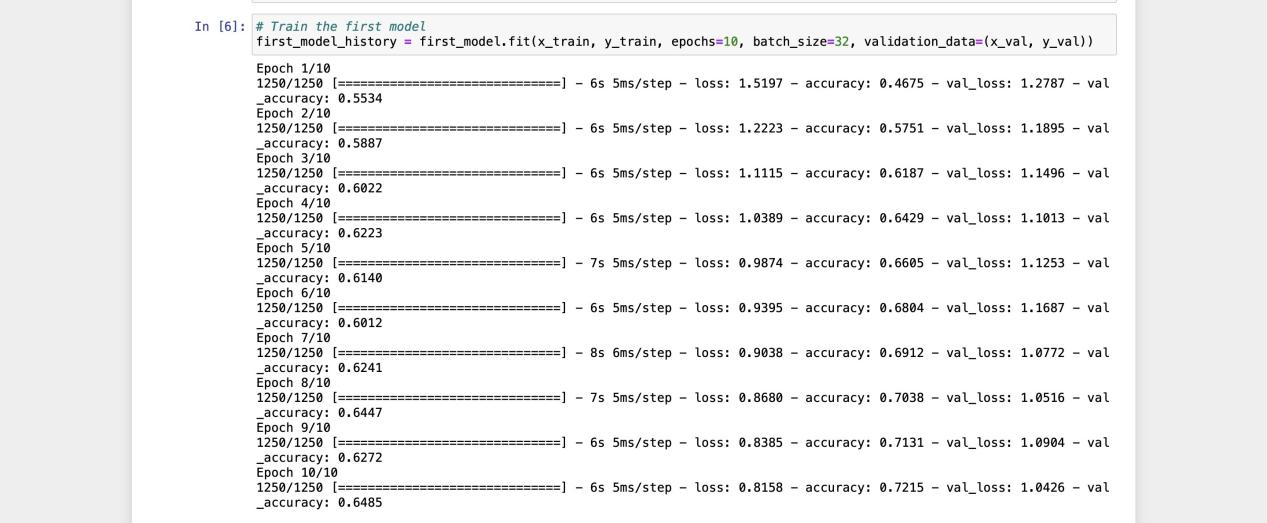
****

Fig 4: First Model History

Using 10 epochs as well on the second model, we can see that the accuracy for the second model improves after each epoch. With the first epochs accuracy being about **42%** and by the tenth the accuracy being **75%.**

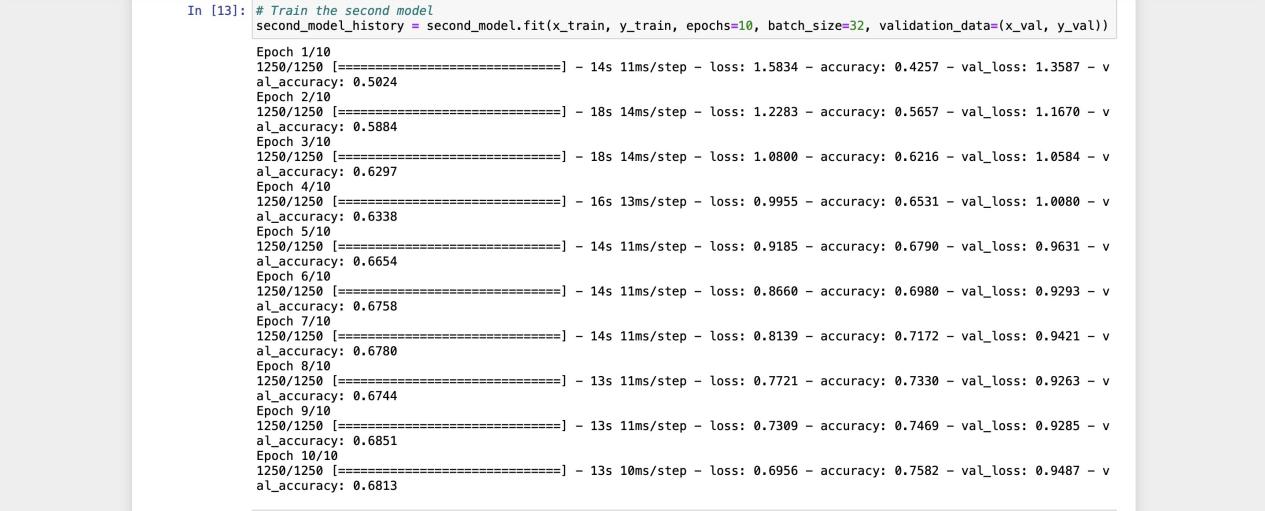
****

Fig 5: Second Model History

This shows that the second model has a better performance and accuracy level and that it can make decent predictions on patterns and images.

Comparison of Models

Loss and Accuracy

It is ideal to understand that accuracy is a measure of how well the model is able to classify correctly the samples in the dataset. Accuracy is a very good method to use when evaluating the model but should not be the only metric used.

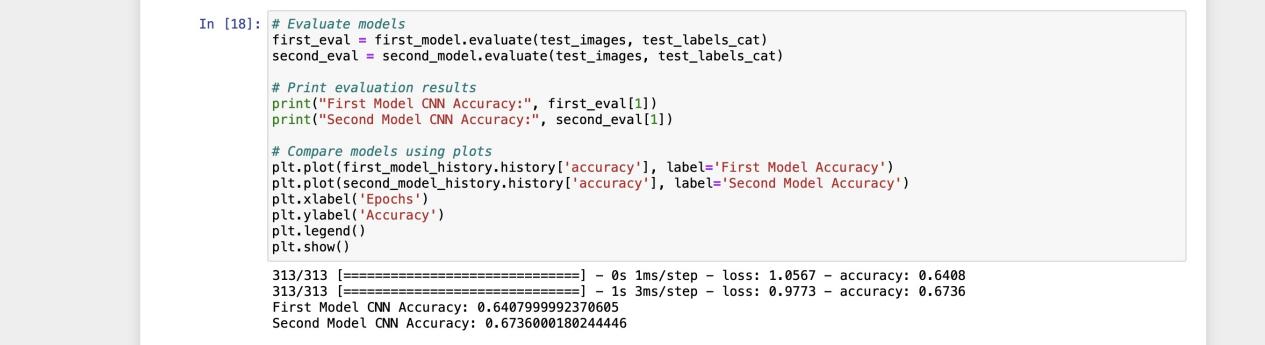


Fig 6: Model Comparison

The accuracy of the two models is evaluated and compared on the validation set by using the evaluate() method. This method returns the loss and accuracy of the model on the given dataset.

Graphs

A training vs validation accuracy graph is a visualization used to monitor and analyze the performance of a machine learning model during training. It displays how the accuracy of the model on both the training dataset and the validation dataset changes as the training progresses over multiple epochs (training iterations).

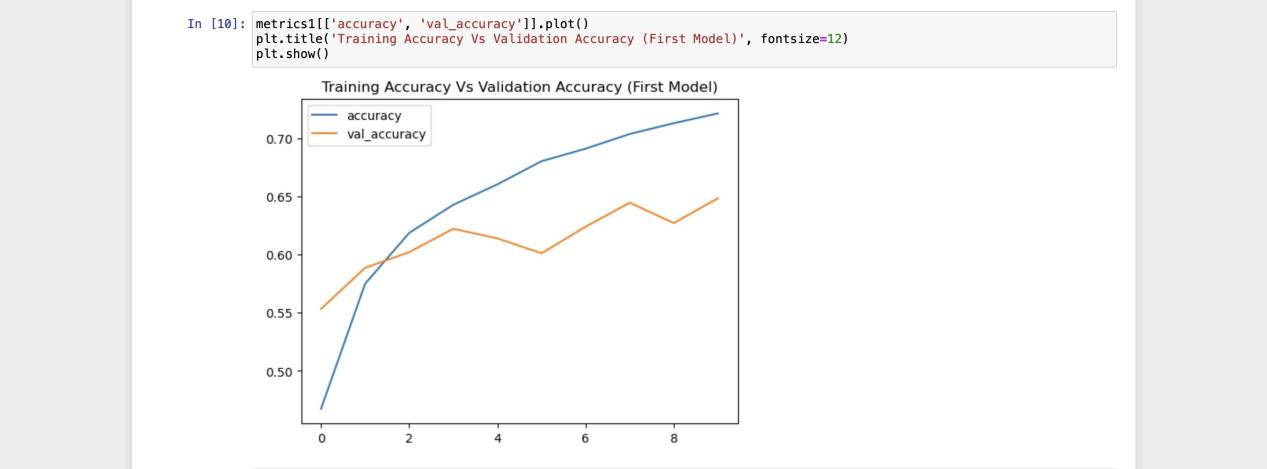
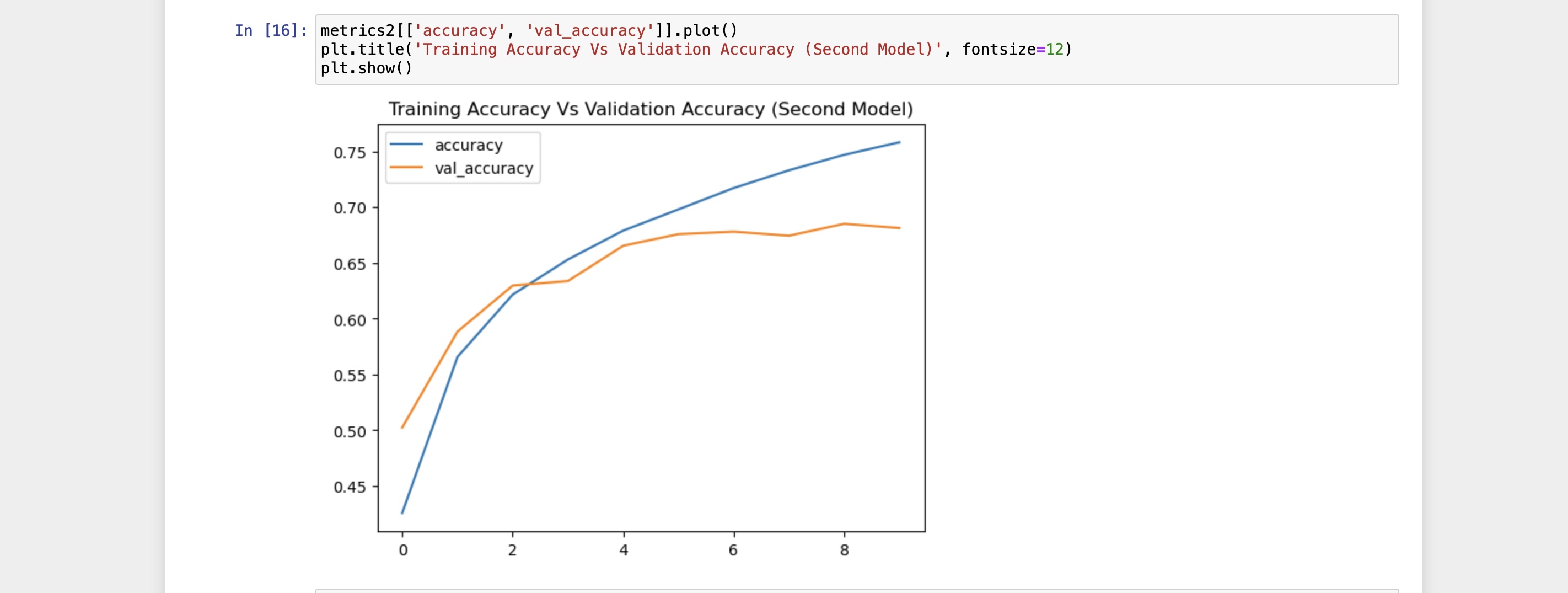


Fig 7: First Model Training vs Validation Accuracy

 Fig 8: Second Model Training vs Validation Accuracy

A training vs validation loss graph, also known as a "loss curve" or "learning curve," is a visualization used to monitor and analyze the performance of a machine learning model during training. It displays how the model's loss (error) on both the training dataset and the validation dataset changes as the training progresses over multiple epochs (training iterations).

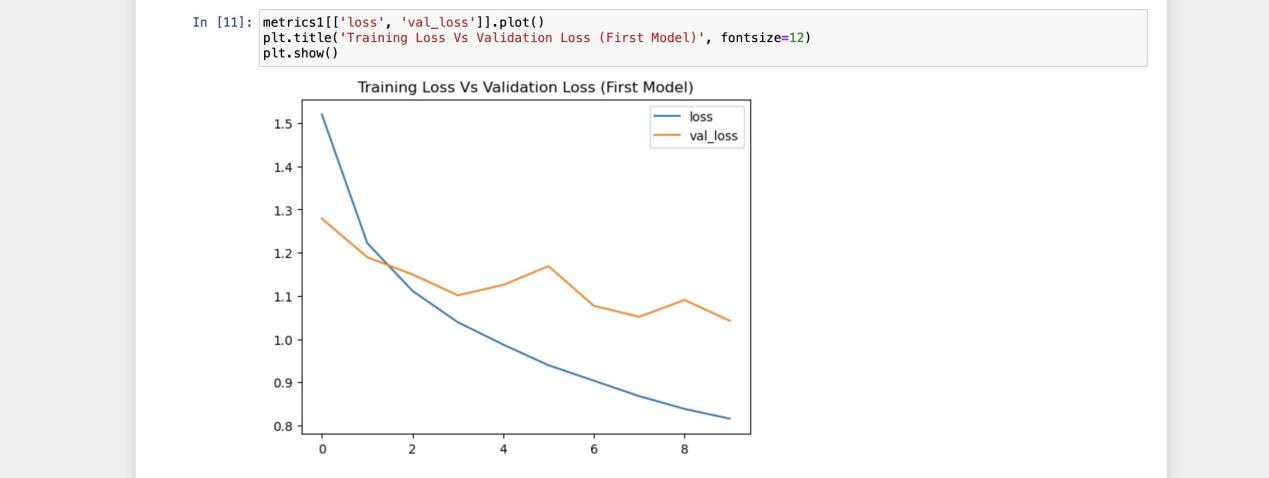


Fig 9: First Model Training vs Validation Loss

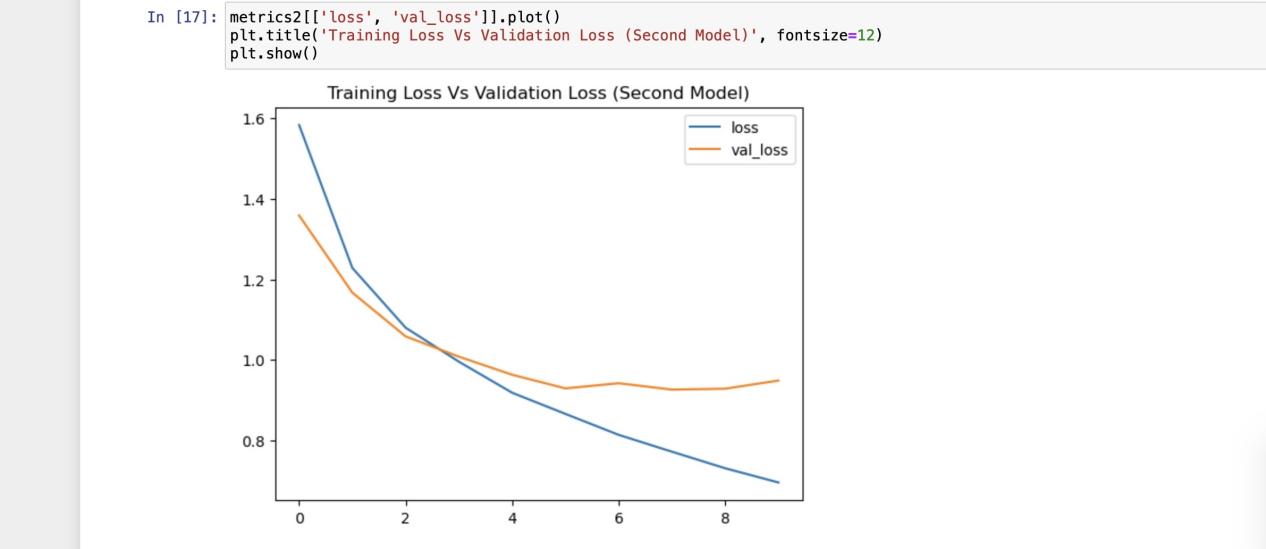


Fig 10 : Second Model Training vs Validation Accuracy

Classification Report

A classification report is a summary of the performance of a classification model that is often used in machine learning and data analysis. It provides a detailed breakdown of various evaluation metrics for each class in a classification problem. A classification report usually includes the following: precision, recall, f1-score, support, accuracy, macro average and weighted average for each classification task.

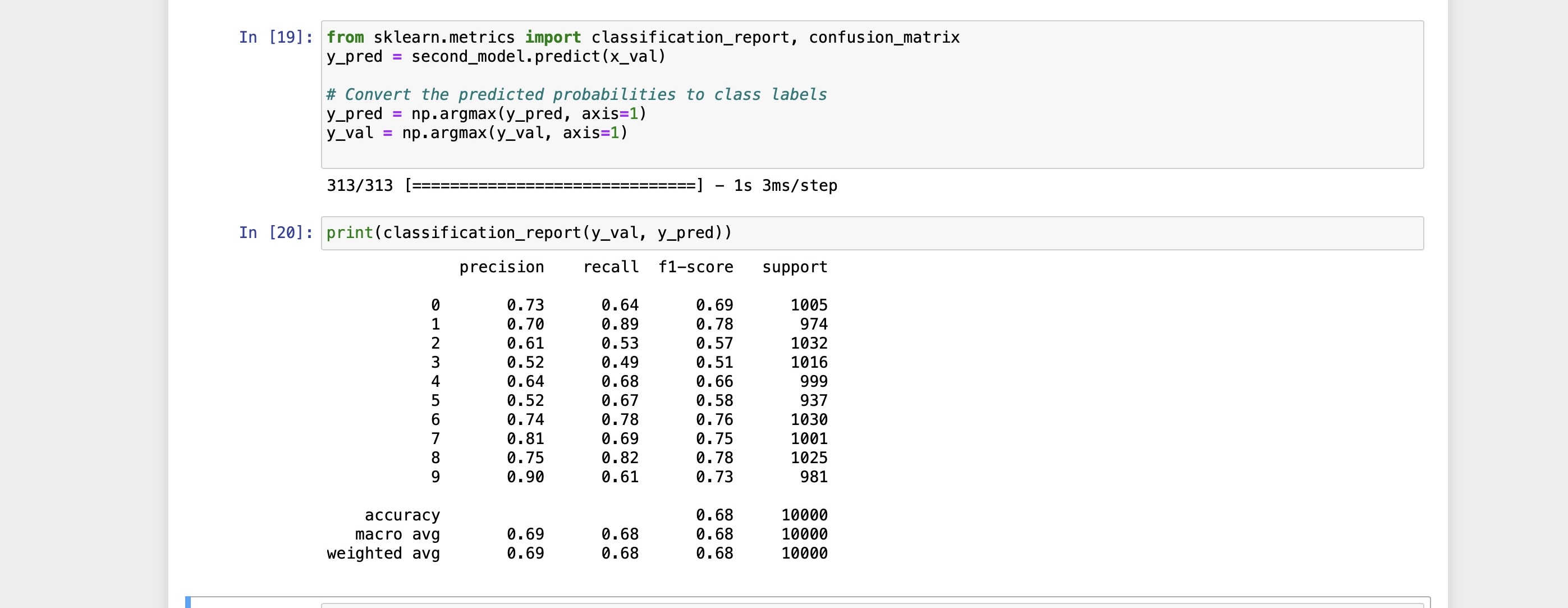


Fig 11: Classification Report

Confusion matrix

A confusion matrix is a table or matrix that is commonly used to evaluate the performance of a classification model, more specifically in machine learning and data analysis. It provides a detailed summary of the model's predictions and the actual class labels for a given dataset.

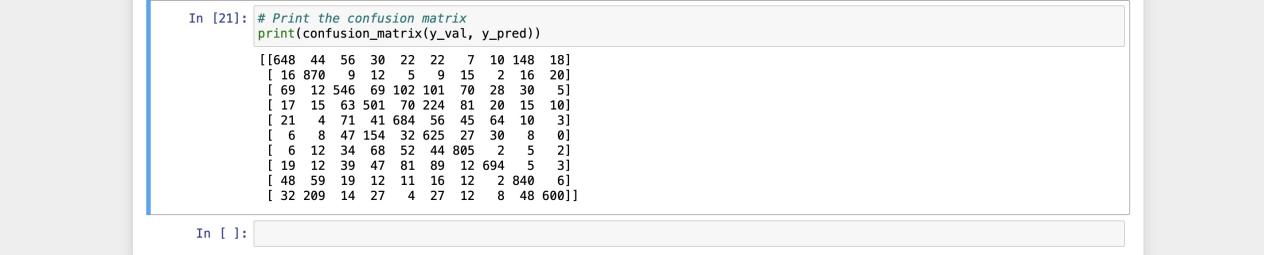


Fig 12: Confusion Matrix

Conclusion

In conclusion, this application was developed using the TensorFlow library, enabling it to accurately predict the ten image categories outlined in the project overview. Notably, one of the models achieved an impressive accuracy rate of **75%**, demonstrating its capability to effectively identify most images. The broader significance of image classification is evident, offering a wide range of potential applications, such as object detection for autonomous vehicles, emotion analysis, food quality control, medical diagnosis and beyond.

Knowledge Gained

Developing neural networks encompasses several fundamental principles and methodologies, including tasks like data preprocessing, designing model architectures, conducting training and evaluation, implementing regularization techniques, exploring transfer learning, fine-tuning hyperparameters, optimizing models, and applying data augmentation. It also necessitates proficiency in programming and the utilization of deep learning frameworks such as TensorFlow, Keras, PyTorch, and others. Through this assignment, I have acquired the knowledge and practical experience required for these aspects of neural network development.