**Text

Description automatically generated**

A picture containing graphical user interface

Description automatically generated

**CIS6005 COMPUTATIONAL INTELLIGENCE**

[**LINK TO REPOSITORY**](https://github.com/LucilleNU/CIS6005_S1BUL_22)

**24th January, 2023**

STUDENT NUMBER: ST20205448

NAME: LUCILLE UMBEYE NAWA

Table of Contents

[Brief Project Overview 3](#_Toc125474923)

[Introduction 3](#_Toc125474924)

[Dataset 3](#_Toc125474925)

[Data Splitting 3](#_Toc125474926)

[Training Set 3](#_Toc125474927)

[Validation Set 4](#_Toc125474928)

[Metadata 4](#_Toc125474929)

[Neural Network design 4](#_Toc125474930)

[Activation functions 5](#_Toc125474931)

[Loss Function 5](#_Toc125474932)

[Models Evaluation and results 5](#_Toc125474933)

[Model Structure/ Architecture 5](#_Toc125474934)

[Model1 5](#_Toc125474935)

[Model2 6](#_Toc125474936)

[ModelAccuracy 7](#_Toc125474937)

[Comparison of Models 7](#_Toc125474938)

[Loss and Accuracy 7](#_Toc125474939)

[Graphs 7](#_Toc125474940)

[Classification Report and Confusion Matrix 8](#_Toc125474941)

[Conclusion 9](#_Toc125474942)

[knowledge Gained 9](#_Toc125474943)

Table of Figures

[Figure 1: Training Data 3](#_Toc125474886)

[Figure 2: Validation Data 4](#_Toc125474887)

[Figure 3: Building the first model 6](#_Toc125474888)

[Figure 4: Building the second model 7](#_Toc125474889)

[Figure 5: Model Comparison 8](#_Toc125474890)

[Figure 6: Model1 Training Loss Vs Validation Loss. Figure 7: Model2 Training Loss Vs Validation Loss 8](#_Toc125474891)

[Figure 8: Model1 Training Accuracy Vs Validation Accuracy Figure 9: Model2 Training Accuracy Vs Validation Accuracy 9](#_Toc125474892)

[Figure 10: Classification Report Figure 11: Confusion Matrix 9](#_Toc125474893)

# sBrief Project Overview

This project is an image classification project that assigns a class label based on the input image. The project is built with a Convolutional Neural Network (CNN). CNNs are great for image classification because they can detect patterns in any given input image. The CNN is trained on the cifer10 dataset and can then recognize the following input images e.g A plane, a car, a bird, a cat, a deer, a dog, a frog, a horse, a ship, and a truck.

# Introduction

The project model is of the classification type, which is a supervised learning algorithm that predicts the class of given values, in this case the corresponding class name to the given input image. This project employs Multi-Class Classification, which means that each data sample is assigned to a single label, such as "Dog," which is assigned to an image of a Dog.

# Dataset

I used the cifar10 Dataset from the Keras dataset library for this project; it contains 60,00 coloured images of 32x32 pixels in the ten classes mentioned above, with 6,000 images dedicated to each.

## Data Splitting

### Training Set

CIFAR-10 dataset is imported from the keras.datasets library and then split into a training set, validation set, and testing set using the train\_test\_split() function from the sklearn.model\_selection library.

The original dataset is split into a training set and a testing set first, using the train\_test\_split() function:

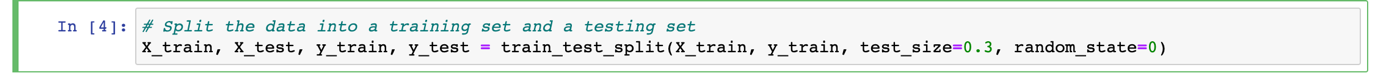


Figure 1: Training Data

This splits the original dataset into a training set (70% of the data) and a testing set (30% of the data).

### Validation Set

#### Creation

To create the validation set, the training set is further split into a new training set and a validation set using the train\_test\_split() function again:

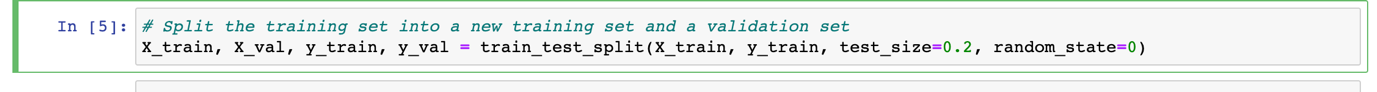


Figure 2: Validation Data

#### % Of data used

This splits the training set into a new training set (80% of the training data) and a validation set (20% of the training data).

#### Reason for having a validation set

The validation set is used to evaluate the performance of the models during training, by comparing the accuracy on the validation set to the accuracy on the training set. This allows to detect overfitting of the model on the training set and adjust the training accordingly, by modifying the architecture, number of layers, and hyperparameters of the models.

Additionally, the validation set is also used to select the best model among the two models, by comparing their performance on the validation set. The best model is then used to make predictions on the unseen test set.

It's important to note that the validation set is created from the training set, and it is not used during the training process but only to evaluate the model's performance and adjust its parameters.

## Metadata

The metadata of the CIFAR-10 dataset are:

* Number of images: 60,000
* Image size: 32x32 pixels
* Number of classes: 10
* Number of images per class: 6,000
* Image format: colour (RGB)
* Collection: Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton
* Task: Object Recognition
* Distribution: 70% training, 20% validation, 10% testing

# Neural Network design

As stated in the brief overview, a CNN was used to train the images. One of the main reasons I chose to use a CNN was its high accuracy; CNNs have the ability to recognize specific patterns in images, making them the best choice for image classification. CNNs are fully connected neural networks that are very effective at reducing parameter size without sacrificing model quality.

When designing a neural network, there are several aspects to consider in order to optimize its performance for a specific task some of which have been explained below:

# Activation functions

1. **Rectified Linear Unit (ReLU) activation function:** ReLU is used in the convolutional layers of the neural networks. It is a non-linear activation function that returns 0 for all negative input values and returns the same value for positive input values. It is used to introduce non-linearity to the model and allow the model to learn more complex features from the input data.
2. **Softmax activation function:** Softmax is used in the output layer to produce probability scores for each class and select the class with the highest probability as the final prediction. It is a generalization of the logistic function that maps the input to a probability distribution over the output classes.

# Loss Function

The loss function used is the **categorical cross-entropy loss function**. This is because the CIFAR-10 dataset is a multi-class classification problem with 10 classes, and the output of the model is a probability distribution over these classes. The categorical cross-entropy loss function measures the dissimilarity between the predicted probability distribution and the true probability distribution and is used to guide the training of the model.

# Models Evaluation and results

## Model Structure/ Architecture

### Model1

The structure of this neural network is a convolutional neural network (CNN) designed for image classification tasks. The network has a total of 5 layers:

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

It uses the Adam optimizer and categorical cross entropy as loss function and accuracy as evaluation metric.

Text

Description automatically generated with medium confidence

Figure 3: Building the first model

### Model2

The structure of this neural network is a convolutional neural network (CNN) designed for image classification tasks. The network has a total of 8 layers:

* Three convolutional layers (Conv2D) with 32, 64 and 64 filters respectively, each followed by a max pooling layer (MaxPooling2D) to reduce the dimensionality of the feature maps and extract the most important features.
* A flatten layer to reshape the feature maps into a 1-dimensional array.
* A dense layer with 512 neurons and relu activation function
* A dense layer with 10 neurons and a softmax activation function to produce probability scores for each class.

It uses the Adam optimizer, categorical crossentropy as loss function and accuracy as evaluation metric. This architecture is a bit more complex than the previous one, it has more layers, more filters and more complex dense layer which is used to classify the images. It is designed to extract more complex features from the images.

Graphical user interface, text, application

Description automatically generated

Figure 4: Building the second model

## ModelAccuracy

The accuracy and loss of model1, Using 15 epochs, we can see that the accuracy improves slightly after each epoch. Within the first epoch we can see that the accuracy was about **27.81%** and by the fifteenth epoch the accuracy jumped up to **69.90%.**

The accuracy and loss of model2, Using 15 epochs, we can see that the accuracy improves slightly after each epoch. Within the first epoch we can see that the accuracy was about **31.24%** and by the fifteenth epoch the accuracy jumped up to **74.61%.** This shows that it has a better performance and accuracy level and is a good percentage for the model to make decent prediction on images.

## Comparison of Models

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

Text

Description automatically generated

Figure 5: Model Comparison

It is important to note that accuracy is a measure of how well the model is able to correctly classify the samples in the dataset. It can be a good metric to use when evaluating the model, but it should not be the only metric used.

### Graphs

The training loss vs validation loss graph is a visual representation of the performance of a machine learning model during the training process. It plots the training loss and validation loss over time, typically against the number of training iterations or epochs.

#### Chart, line chart Description automatically generatedChart, line chart Description automatically generated

Figure 6: Model1 Training Loss Vs Validation Loss. Figure 7: Model2 Training Loss Vs Validation Loss

Training accuracy vs validation accuracy graph is a visual representation of the performance of a machine learning model during the training process. The goal is to have both training accuracy and validation accuracy increasing and converge to a similarly high value which can be seen below. This indicates that the model is generalizing well to new unseen data

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Figure 8: Model1 Training Accuracy Vs Validation Accuracy Figure 9: Model2 Training Accuracy Vs Validation Accuracy

# Classification Report and Confusion Matrix

A classification report and a confusion matrix are two commonly used evaluation metrics for assessing the performance of a classification model. A classification report shows the precision, recall, f1-score, and support for each class in a classification task. A confusion matrix is a table that is often used to describe the performance of a classification algorithm, each row and column represents the instances in a predicted and actual class respectively.

Both of these metrics are useful for understanding how well a model is performing and for identifying areas where the model may be struggling, they can help to identify the misclassification and give an idea of the bias and variance of the model.

Keep in mind that this metrics have been implemented and predictions made on the validation set. However, this could also be implemented to the test set.

Table

Description automatically generatedA picture containing text

Description automatically generated

Figure 10: Classification Report Figure 11: Confusion Matrix

# Conclusion

Finally, this application was created with the TensorFlow library, and it predicts the ten image types mentioned in the project overview. One of the models (model2) has an accuracy of **74.61%,** which means it can detect most images without issues. In general, image classification is extremely useful because it opens endless possibilities; for example, it could be used to detect objects around self-driving cars.

## knowledge Gained

Creating neural networks involves several key concepts and techniques such as data pre-processing, model architecture, training, evaluation, regularization, transfer learning, hyperparameter tuning, optimization and data augmentation. It also requires experience with programming and the ability to use deep learning frameworks like TensorFlow, Keras, PyTorch, etc. which I have been able to learn and gain experience in with this assignment.