**Section I: Dataset**

I used the CIFAR 10 dataset has 10 classes and 60000 images, with 6000 image per class. Each image is 32x32 and is of RGB type as opposed to a black and white image. The CIFAR10 dataset was split into train and test sets along a 5:1 ratio, therefore the training set has 50000 images.

The images (x train and test data) were normalized using min-max scaling.

(max(x) – min(x))

The labels were transformed into categorical data using the keras.utils.to\_categorical command.

**Section II: Select a Network**

I selected the ResNet50 model.

**Section III: Compare the trained/fine-tuned pretrained DCNN with the DCNN from assignment 2**

My pre-trained DCNN model which I retrained on the CIFAR10 dataset achieved a training accuracy of 88.1% while the testing accuracy was 75.1%. The overfitting was less severe but was also accompanied by a reduction in accuracy. My DCNN from assignment 2 was only trained on the CIFAR10 dataset and achieved a testing accuracy of 80.1% while the training accuracy was 95.9% - this suggests that there was strong overfitting. I tried to combat this by the following means; increasing the regularisation parameter value in the Dropout Layers, by decreasing the number of free parameters and by implementing early stopping. I also tried to use data augmentation for the images as well but I did not find very helpful results from this process when I used ImageDataGenerator class from tensorflow.keras.preprocessing.image, this is shown in the trial data from assignment 2.

**Section IV:**