**Image Classification Project – COSC 6324**

Objective:

The objective of this project is create a multiclass classifier to efficiently classify among 10 classes (namely: 'airplane','automobile','bird','cat','deer','dog','frog','horse','ship','truck')

Dataset:

The dataset contains 9999 rows and 3072 columns. There are columns that are very much necessary for our prediction since each column contains pixel values for each image.. 9999 rows indicate that there are 9999 images in total. There is another dataset that we need as it contains the classes for each row of the image dataset.

Approach:

We see that there are no missing values which is a good thing, because if we had a single value missing it would not have been possible to fill it since these are pixel values. We went ahead and generated rgb images from the dataset and stored them in a list and used matplotlib library to visualise the images. We then converted into black and white for faster model training. We then split the dataset into training and testing (70% into the training set and 30% into the testing set). We named the training set x\_tain and y\_train and the testing test as x\_test and y\_test.

Model Creation:

A CNN model was created using the tensorflow library where:

* The number of neurons in the input layer are 32 × 32 = 1024
* There are 5 convolution and pooling layers.
* The number features are 512.
* The number of neurons in the classification input layer the size of the feature vector.
* The number of layers for the classification are 2.
* The number of neurons in the output are 10.
* The loss function is categorical\_crossentropy.
* The optimizer of choice is adam.
* The model is trained for 10 epochs (we tried training it for even more, but did not see a significant improvement in accuracy).
* The metric of choice is accuracy.

Model Evaluation:

We evaluated the classifier by the accuracy measurement on x\_test by dividing the total correct classification by total samples. We got the accuracy to be 41.33%

This is the graph for loss is:

