**Milestone Report:**

Started off with importing all the necessary libraries with using TensowFlow in the Backend. Loaded the csv files on to the training and testing dataframes. Performed the initial quality checks like checking the shape of the dataframes, checking for NaN values and any irregularities in the dataframe. Set the input features in the dataframe to a variable X and the output labels t a variable Y. The best way to split a dataset is by splitting the dataset to training and testing datasets. We have a module in sklearn model selection called train test split. The train test split will split the dataframe to training set inputs(features) and training set outputs and testing set inputs and testing set outputs. The training inputs and outputs will be used for training the model. The testing inputs will then be used to make predictions and the predicted outputs and the actual test set outputs will then be compared to check for accuracy of the algorithm. The training sets will be stored as X\_train, y\_train and testing sets will be stored as X\_test, y\_test.

Plotted the images of the different labels in the dataset. Reshaped the data to samples, channels, width and height. Then normalized the images to be in the range of 0-1 by dividing the entire training and testing datasets by 255. Then converted the labels to one hot encoded outputs. The plan is to create multiple models, a base model and a larger model. The base model will be a simple neural network with a convolutional layer of 32 inputs, then pooling the pixels data, then adding a dense layer of 128 nodes and then the output layer. This neural network is run over 10 epochs with a batch size of 200 and it gives an accuracy of 91%. It gives a precision, accuracy and f score of 0.91. The testing loss is about 0.25

The model is testing against the testing set. The testing set predictions are checked to see which predictions did the model get right and which predictions the model got incorrect. This is done by plotting the correctly and incorrectly predicted images.

The second larger neural network is a slight extension of the previous model. It has a convolutional layer with 30 features, a pooling layer, then again a convolutional layer of 15 features and pooling layer. Two fully connected dense layers of 128 and 50 neurons each and then the output layer. This model gives an accuracy of 0.89% and precision, recall and f score of 0.89 and testing loss of 0.29.

The plan is to further create another model with multiple convolutional layers and run it on around 50 epochs.