**RECOGNISIGN**

**Team ID - RS-206619**

**TASK**

In this competition, the main goal is to classify traffic signs. The participants must analyse possible techniques to classify the traffic signs (since no dataset is made available) and develop a computer vision model that gives a reasonable accuracy during prediction.

**PROBLEM STATEMENT**

This competition expects a Team to come with a model capable of detecting and classifying traffic signs. Some types of signs are mentioned at the end of the Problem statement. The Team has to build its own dataset for training the models. There is no restriction on using any kind of data for training the model. There are no restrictions on the dataset size and contents used for training, choice of programming languages, usage of inbuilt packages, and the type of framework used for attempting the task. The model should be robust enough so that it is capable of performing real-time classification.

**PRACTICAL APPLICATION**

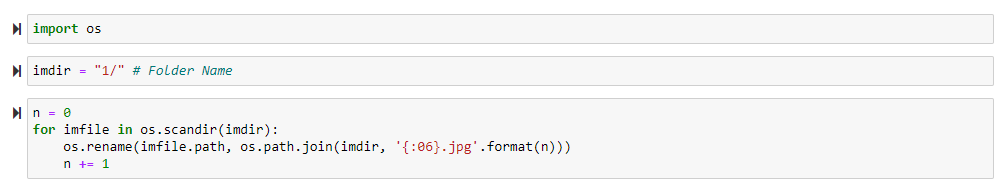
Traffic Safety is an overlooked topic and technology should be used to improve it. In developing countries like ours, road accidents are a cause of large percentage of deaths. A traffic sign detector will really prove useful in increasing traffic safety.

**OUR APPROACH**

* The first task was to create a dataset
* The second one was to train it and increase its accuracy

**CREATING DATASET**

* We decided to go with multiclass classification as it is a more practical solution than a binary classifier
* We decided 6 categories of signs which our model will be able to recognise and identify
  + 1. Men at work
    2. No Parking
    3. One Way
    4. Speed Breaker Ahead
    5. Speed Limit Sign
    6. Stop
* These categories were assigned Class IDs from 0 to 5
* To start with, we considered these 6 categories as these signs are the most common ones in India and we further plan to add more categories by increasing the dataset of images
* We used a google extension – ‘Image Downloader’ to scrape images from web and downloaded the images of the above 6 categories
* After that we placed the images in a folder with folder name corresponding to Class ID
* We then renamed all the images in an increasing numeric order, starting from 0 to n-1 using this code -



* We also created a “labels.csv” file to index the images to their respective class for further reference
* In this way we created our dataset of labelled images comprising of 6 categories

**APPLYING ML AND TRAINING THE MODEL**

We have used Convolutional neural networks as they are a part of deep learning and extensively used in image recognition. These convolutional neural networks consist of several layers.

We preferred it over YOLO because now-a-days highways are 3-4 lanes so the image of the traffic sign on the road will appear as a small object as compared to the entire image so there’s a risk of it being missed by YOLO.

**Feature Learning Phase Layers**

* A Conv2D layer, used for feature extraction with the help of filters. Number of filters are generally in power of 2 like 32, 64 or 128. We have used ReLU as our activation function. ReLU function is defined as maximum (0, x).
* The max pooling layer, which is used reduce the dimensions of the image. This is done to reduce the computation power required for processing the image.
* The dropout layer is used to prevent overfitting and to reduce the complexity of the model. In this layer some neurons are removed randomly.

The flatten layer which converts the 2-D data into a long 1-D vector of features for a fully connected layer that can be fed into the neural network.

The last layer is the dense layer which is used as an output layer. It has 6 nodes, same as the number of classes. It uses SoftMax activation function. SoftMax function gives the probability value (between 0 and 1) so that the model can predict which class has the highest probability and accordingly predicts and output corresponding to the Class ID

**WORKING OF MODEL**

* We first import all the necessary libraries
* We then convert all the images to a same size and store them in a NumPy array for faster processing
* We then split the dataset in train and test data
* We then build a sequential model using keras and put
  + 1. Conv2D Layer for feature extraction
    2. Max Pooling Layer for dimension reduction
    3. Dropout layer to reduce complexity of model
    4. Flatten layer to convert from 2D to 1D
    5. SoftMax layer to give the output
* We will compile the model and apply it using fit function. The batch size will be 32. Then we will plot the graphs for accuracy and loss.