Traffic Signal Classification is a key algorithm for the development of the autonomous driving field. It eliminates the need of the driver to drive according to the need of the road rules without the need of the attention to be paid. Also, with the help of traffic signal classification the number of accidents that are happening due to the mistakes caused the human beings without noticing the road signs. Best example can be we can avoid driving at a speed more than the road limit, driving in one-ways can be avoided. It helps to check the driving behavior of humans making sure that they strictly follow rules laid by the local government bodies. According to the surveys carried across the globe it was known that more than 1 million people died yearly.

Deep Learning techniques were developed and used to detect and classify information available in the image. Here we employ the deep learning algorithm to detect the traffic sign in the image and classify it to the respective class.

**Road Map of the Project:**

* Loading the dataset
* Summarizing the dataset
* Visualizing the dataset
* Data Preprocessing
* Design and training the model
* Testing the model
* Making predictions on the new images from internet

**Dataset Information and Visualization:**

The dataset is built with actual German Traffic Signs. The size of the images used here are with Height-32, Width-32 and with number of channels-3 (RGB channels).

Total count of images in training set are 34799,

Testing set count is 12630 and

Validation size is 4410.

Total number of unique classes in the dataset is 43

Histograms added in the chart shows the distribution of various classes of images in respective dataset.

Chart, histogram

Description automatically generated

Clearly here it can be noticed that the images are not evenly distributed. More images to be added to the dataset to make it more uniformly distributed. Spikes here shows that the images of that particular class are more in the respective dataset.

**Data Preprocessing:**

The images that are captured and stored in the dataset are in RGB format and here we convert them into grayscale in the first step. Advantage of turning the RGB image to grayscale is it helps to speed up the computation leading to optimize the process for faster reaction of the autonomous car to the sudden sign board appearing on the road.

After converting the images to grayscale, we normalize them. Normalization can be done by dividing each image with value of mean and then dividing it by standard deviation. This results in all image pixel values in range [0,1]. Having all in range between 0 and 1 helps to avoid too much difference between values. For example, white takes value 255 and black takes value 0. This can bring the effects of non-uniformity during training. Normalization helps to achieve consistency in image pixel values.

**Model Architecture:**

The model is built with 3 convolution layers along with Maxpooling and dropout. Convolutional layer-1 is built with filters of 32 and activation function ‘relu’ is used. Kernel size of 3\*3 is used for convolution. Max Pooling layer with stride size of (2,2). Following that dropout layer is employed with 0.25% of dropout. In the end softmax function with classes of 43 is given for classification.

The model results are shown in the graphs shown below. We achieved a model accuracy of 99.88% on training and 99.87% for validation set.

Chart

Description automatically generated