**TRAFFIC SIGN RECOGNITION**

**A PROJECT REPORT**

***Submitted by***

Varsha Nair H

Samiksha S

Rajanandhini H

***in partial fulfillment for the award of the degree of***

**Bachelor of Engineering**

**IN**

Computer Science and Engineering

**Thiagarajar College of Engineering**

**ANNA UNIVERSITY : CHENNAI 600**

APRIL 2021

**ANNA UNIVERSITY : CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report **TRAFFIC SIGN RECOGNITION** is the bonafide work of  Varsha Nair H , Samiksha S , Rajanandhini H who carried out the project work under my supervision.

**Abstract**

Self-driving cars have the potential to revolutionize urban mobility by providing sustainable, safe, convenient and congestion free transportability. This vehicle autonomy as an application has several challenges like infallibly recognizing traffic lights, signs, unclear lane markings, pedestrians, etc. which can be overcome by using technological development in the fields of Deep Learning, Computer Vision and CNN algorithm.CNN- Convolutional Neural Networks have increased the computer vision tasks because of its enhanced recognition rate without any human supervision and faster execution. The experimental results have shown higher accuracy. Nowadays there is a high probability that the driver may miss some of the traffic signs on the road due to overcrowding of vehicles or distraction or inattentiveness. A traffic sign recognition system can be implemented in vehicles with an aim of recognizing all emerging traffic signs.

**Chapter 1 - Introduction**

Traffic signs are a very essential part of roads which provide details regarding the state of road, restrictions,warnings and also help for navigation such as pedestrian crossing, railroad crossing.

Failing to notice these traffic signs may directly or indirectly lead to accidents.Traffic sign recognition (TSR) is an important application in advanced driver assistance systems (ADAS), which helps drivers with vital information.

Traffic Sign Recognition (TSR) system is an important part or module of Intelligent Transport System (ITS) as traffic signs assist the drivers to drive more safely and efficiently.

**1.1 Introduction to CNN**

In this paper CNN(Convolutional Neural Networks) is the algorithm that is used for the recognition process. The major reason for selecting this algorithm is that CNNs have achieved phenomenal accuracy and efﬁciency in image-classiﬁcation.

An analysis of the dataset reveals that, there are only a ﬁnite number of challenge-types and sign-types to be considered, and the type of noise in each image is distinct and unique. Therefore, a neural-network based classiﬁcation gives good results .Also, CNN models are easy and faster to train on images comparatively to the other models.

It is also very important to consider the processing time of an individual image and object detection accuracy (objects can be of traffic signs and other traffic-related items).

The processing time is correlated with the network architecture(i.e), more time is required when more layers are used and the number of generated features grows with the input image size along with the needed computational power and memory.

On the other hand,trafﬁc-sign recognition is an interesting topic in computer-vision and it is especially important in the context of autonomous vehicle technology.This paper will give details about the dataset,implementation,and also the results obtained after testing the model.

Convolution neural networks(CNN) can take images as their input.It is a deep learning algorithm.It is capable of taking image input and assigning importance to various aspects or objects thus enabling it to differentiate from one image to another.

The architecture of a CNN is very similar to that of the connectivity pattern of Neurons in the Human Brain.In general, an image is nothing but a matrix of pixel values.

**Chapter 2 - Other Related Works**

**2.1**  S. Gunal *et al.* [Subspace based feature selection for pattern recognition](https://www.sciencedirect.com/science/article/pii/S0020025512001673) Inf. Sci. (2008) , has proposed thatthe aim of pattern recognition is to classify objects of interest, or patterns, into appropriate categories or classes. A pattern recognition system consists of a feature extraction mechanism that computes numeric or symbolic information from the patterns, and a classifier that executes the classification process of patterns using the extracted features. The number of features used by the recognition system directly affects the classification accuracy and processing time.

Several methodologies exist for feature selection, ranging from filter to wrapper approaches . Among all of these possibilities, only the exhaustive search and branch-and-bound algorithms can yield optimal results. They are, however, very expensive computationally for even moderate numbers of features, and this leads us to consider suboptimal methods. Widely used criteria include probability of classification error and probabilistic distance measures, such as Divergence, Bhattacharyya, Transformed Divergence and Jeffries–Matusita. Experimental results indicate that the new measures outperform the classic ones in terms of classification accuracy and dimension reduction rate.

**2.2** Alturki, A.S. Traffic Sign Detection and Recognition Using Adaptive Threshold Segmentation with Fuzzy Neural Network Classification. In Proceedings of the 2018 International Symposium on Networks, Computers and Communications (ISNCC), Rome, Italy, 19–21 June 2018; pp. 1–7 has proposed in his paper the Traffic sign detection and recognition using Adaptive Threshold Segmentation with Fuzzy Neural Network classification. It includes three stages. Segmenting the images to extract ROIs in the First stage.To overcome the color segmentation problems, the segmentation is performed usually based on Adaptive thresholding. This geometric information is used to identify traffic shapes from ROIs provided by the first stage. Thus the second stage detects traffic shapes. Based on the information included in their pictograms the traffic signs can be recognized in the Third stage.

Traffic signs give information for navigation of a vehicle For the shape classification a simple invariant geometric moment’s based method is used. Here, six types of features are extracted and these feature are provided to the FNN classifier for performing the recognition.As a classifier, Fuzzy Neural Network (FNN), Artificial Neural Network (ANN) and Support Vector Machine (SVM) classifiers have been tested together with the new descriptor. In this paper, a FNN classifier was presented to recognize the traffic sign. Initially, In this process, the input traffic image is obtained from different highway roads of Bangladesh. Then, it has been processed by the mentioned three steps to recognize the signs. Finally, the performance results have been measured in terms of detection accuracy. A correctly detected sign is considered to be true positive if the corresponding bounding box overlaps with at least 50% of the area covered by the right traffic sign which is present in the image. The evaluation of the detection accuracy is performed based on precision–recall curve. Thus, in this paper because of the effective segmentation and feature extraction, the experimental results show that the proposed FNN has attained 98.2% of accuracy when compared to existing ANN and SVM algorithms.

**2.3** Vokhidov, H.; Hong, H.; Kang, J.; Hoang, T.; Park, K. Recognition of damaged arrow-road markings by visible light camera sensor based on convolutional neural network. Sensors 2016, 16, 2160 has stated in the paper that the road signs are also an important input to the Automated Advanced Driver Assistance System (ADAS) which is installed in many automobiles. Road markings exist on the surface of the road and they can be easily damaged by vehicles. Failure to identify an arrow-road marker properly leads to accidents or injuries to pedestrians. In this paper, a method is proposed that uses a convolutional neural network (CNN) to recognize six types of arrow-road markings that are possibly damaged by visible light camera sensors.

Experimental results with six databases of Road marking dataset, KITTI dataset, Málaga dataset 2009, Málaga urban dataset, Naver street view dataset, and Road/Lane detection evaluation 2013 dataset has been used to generate the training and testing data. In this method,in the first step an arrow-road marking image is provided as the input , and the size of the image is normalized into the image of 265 × 137 pixels in height and width in the second step because the size of the input image to CNN should be the same. And then the normalized image is used as input to pre-trained CNN, the input arrow-road marking image is decided as one of six arrow-road markings ,based on the output of CNN.

**Chapter 3 - Basic Requirements**

**3.1 Dataset**

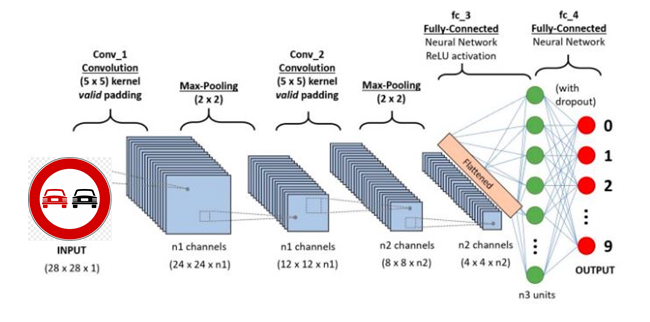
The dataset used is German Traffic Sign Recognition Benchmark.This dataset is available in Kaggle and is of size 300MB.This dataset has more than 50,000 images in total.It is further divided into more than 40 classes.It has two three folders in total where the train folder has around 43 folders numbered from 0 to 42. Each folder contains a variety of images which can be used for training our model. Apart from this there is a folder called test which can be used to test our model and finally helps in also determining or predicting the accuracy of our model.Size of images varies from 15x15 to 222x193.



Sample image of the dataset

**3.2 Convolution Neural Networks**

Convolution neural networks(CNN) can take images as their input.It is a deep learning algorithm.It is capable of taking image input and assigning importance to various aspects or objects thus enabling it to differentiate from one image to another.The architecture of a CNN is very similar to that of the connectivity pattern of Neurons in the Human Brain.In general, an image is nothing but a matrix of pixel values.CNN is capable of successfully capturing the Spatial and Temporal dependencies in an image through the application of relevant filters.The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. On the whole, the network can be trained to understand the sophistication of the image better.



Better view of the overall architecture of CNN used in this model

**3.3 Opencv:**

OpenCV (Open Source Computer Vision Library is a platform through which we can develop real time computer vision applications.It maily looks into image processing,analysing the features like face detection and object detection.) It is an open source computer vision and machine learning software library. It can run on Linux, Windows, Android and Mac.

The library has more than 2500 optimized algorithms, which includes a set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, and many more.

Some applications of openCV are Vehicle counting on highways along with their speeds,Video/image search and retrieval,Medical image analysis,TV Channels advertisement recognition etc.

**3.4 Deep learning:**

The deep-learning based detection uses an ensemble of feature extraction network models.. Here, network models can perform localization and classification. Only a few color spaces have been applied in deep-learning based traffic light recognition. Because color information plays an important role in the performance of recognition, it is necessary to select the color space carefully. It is also required to apply more sophisticated and efficient deep-learning network models to traffic sign recognition.

In deep learning, models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. A large set of labeled data is used for training the model along with the neural network architectures that contain many layers.

Deep learning is a key technology behind driverless cars, which allows them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers.

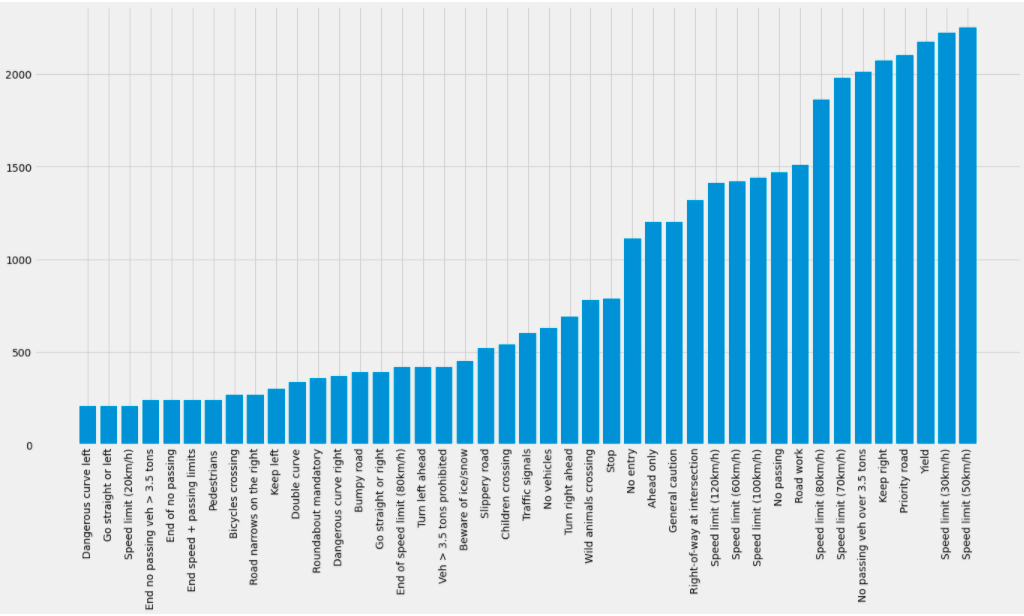
**3.5 Computer Vision:**

Computer vision is the field of computer science that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that humans do. Cameras capture video from different angles around the car and feed it to computer vision software, which then processes the images in real-time to find the extremities of roads, read traffic signs, detect other cars, objects and pedestrians.

**Chapter 4 - Experiment and Analysis**

**4.1 Visualization:**

Initially the dataset was visualized after the label overview which is particular to specific classIds.The dataset sorted on the basis of number of images in each class.After sorting it is plotted as a graph as shown below



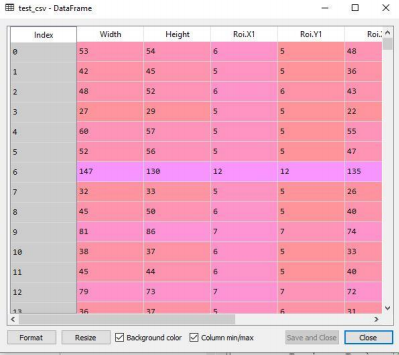
After this various analyses were done to find if there is any relationship between the attributes present in the dataset by plotting various types of graphs like scatterplot, histogram, heat map and many more.

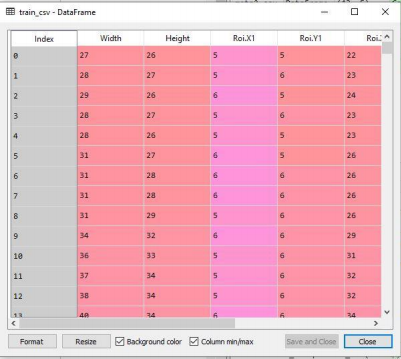
**DATASET:**

https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

**CODE AND OUTPUT:**

meta\_csv=pd.read\_csv('/content/drive/MyDrive/DataScience/Meta.csv') 

test\_csv=pd.read\_csv('/content/drive/MyDrive/DataScience/Test.csv') 

train\_csv=pd.read\_csv('/content/drive/MyDrive/DataScience/Train.csv') 

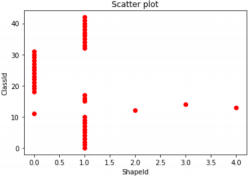
**Scatter Plot :**

plt.scatter(meta\_csv['ShapeId'],meta\_csv['ClassId'],c='red') plt.title('Scatter plot')

plt.xlabel('ShapeId')

plt.ylabel('ClassId')

plt.show()



**Histogram**

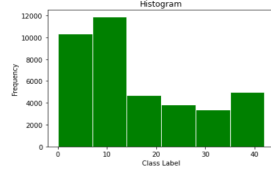
plt.hist(train\_csv['ClassId'])

plt.hist(train\_csv['ClassId'],color='green',edgecolor='white',bins=6) plt.title('Histogram')

plt.xlabel('Class Label')

plt.ylabel('Frequency')

plt.show()



**Bar graph**

plt.bar(index,y,color=['red','blue'])

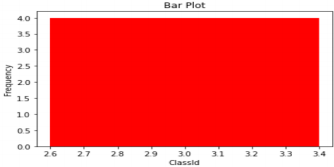
plt.title('Bar Plot')

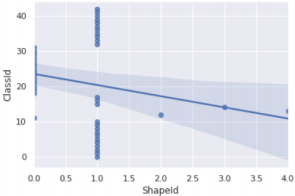
plt.xlabel('ClassId')

plt.ylabel('Frequency')

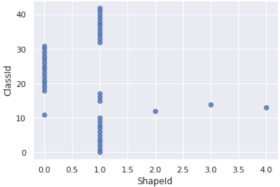
plt.xticks(index,ClassId,rotation=90)

plt.show()

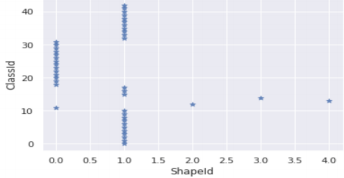


**SNS-** sns.regplot(x=meta\_csv['ShapeId'],y=meta\_csv['ClassId'])

**SNS**

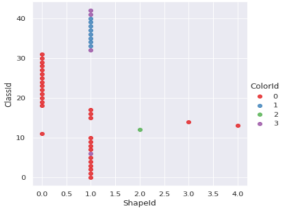
sns.regplot(x=meta\_csv['ShapeId'],y=meta\_csv['ClassId'],fit\_reg=False 

**SNS**

sns.regplot(x=meta\_csv['ShapeId'],y=meta\_csv['ClassId'],fit\_reg=False,marker="\*")

**SNS**

sns.lmplot(x='ShapeId',y='ClassId',data=meta\_csv,fit\_reg=False,hue='ColorId',legend=True,palet te='Set1')

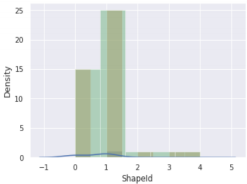


**SNS-Histogram**

sns.distplot(meta\_csv['ShapeId'])

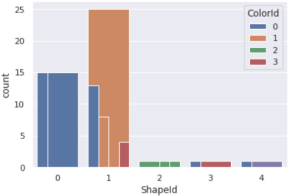
sns.distplot(meta\_csv['ShapeId'],kde=False)

sns.distplot(meta\_csv['ShapeId'],kde=False,bins=5)



**SNS- Grouped bar plot**

sns.countplot(x="ShapeId",data=meta\_csv)

sns.countplot(x="ShapeId",data=meta\_csv,hue="ColorId") 

**Scatter Plot :**

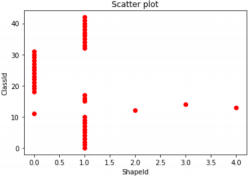
plt.scatter(meta\_csv['ShapeId'],meta\_csv['ClassId'],c='red')

plt.title('Scatter plot')

plt.xlabel('ShapeId')

plt.ylabel('ClassId')

plt.show()



**Histogram**

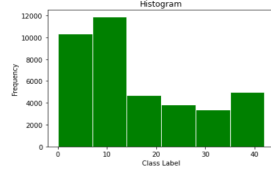
plt.hist(train\_csv['ClassId'])

plt.hist(train\_csv['ClassId'],color='green',edgecolor='white',bins=6) plt.title('Histogram')

plt.xlabel('Class Label')

plt.ylabel('Frequency')

plt.show()



**Bar graph**

plt.bar(index,y,color=['red','blue'])

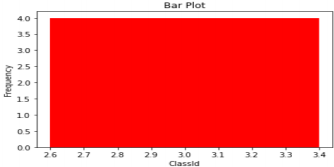
plt.title('Bar Plot')

plt.xlabel('ClassId')

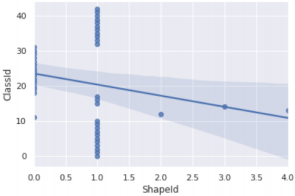
plt.ylabel('Frequency')

plt.xticks(index,ClassId,rotation=90)

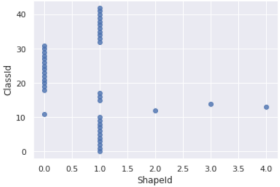
plt.show()



**SNS-** sns.regplot(x=meta\_csv['ShapeId'],y=meta\_csv['ClassId'])

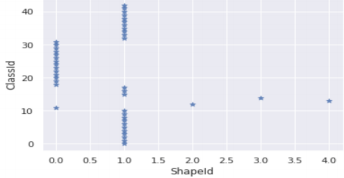


**SNS**

sns.regplot(x=meta\_csv['ShapeId'],y=meta\_csv['ClassId'],fit\_reg=False) 

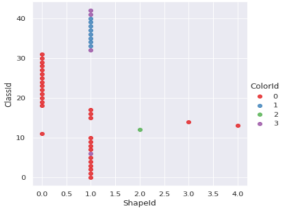
**SNS**

sns.regplot(x=meta\_csv['ShapeId'],y=meta\_csv['ClassId'],fit\_reg=False,marker="\*")



**SNS**

sns.lmplot(x='ShapeId',y='ClassId',data=meta\_csv,fit\_reg=False,hue='ColorId',legend=True,palet te='Set1')

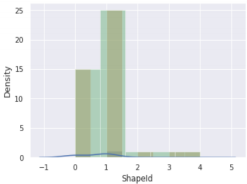


**SNS-Histogram**

sns.distplot(meta\_csv['ShapeId'])

sns.distplot(meta\_csv['ShapeId'],kde=False)

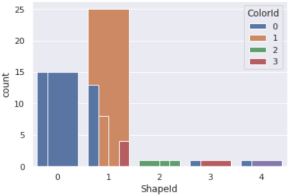
sns.distplot(meta\_csv['ShapeId'],kde=False,bins=5)

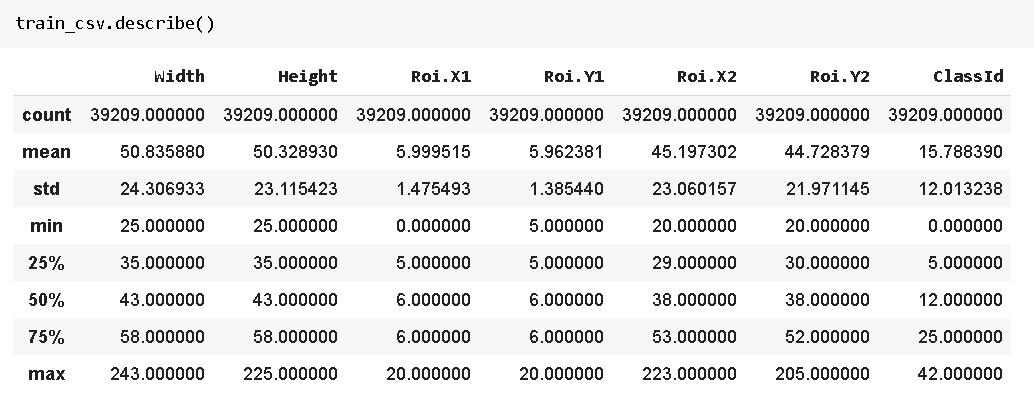


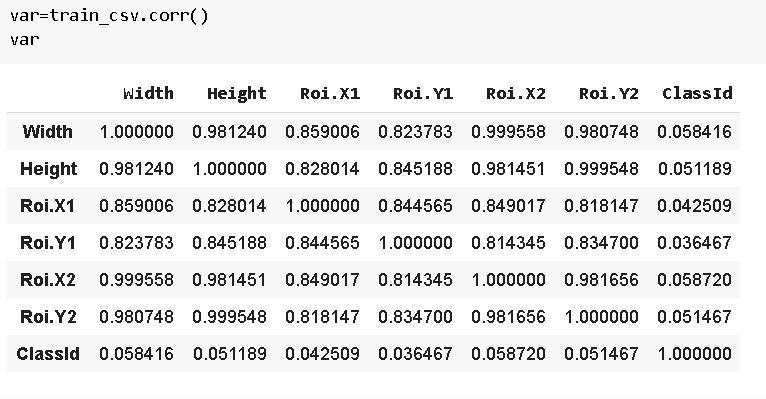
**SNS- Grouped bar plot**

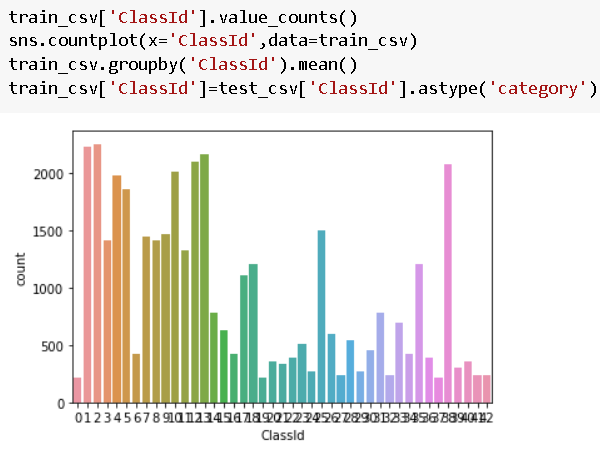
sns.countplot(x="ShapeId",data=meta\_csv)

sns.countplot(x="ShapeId",data=meta\_csv,hue="ColorId")

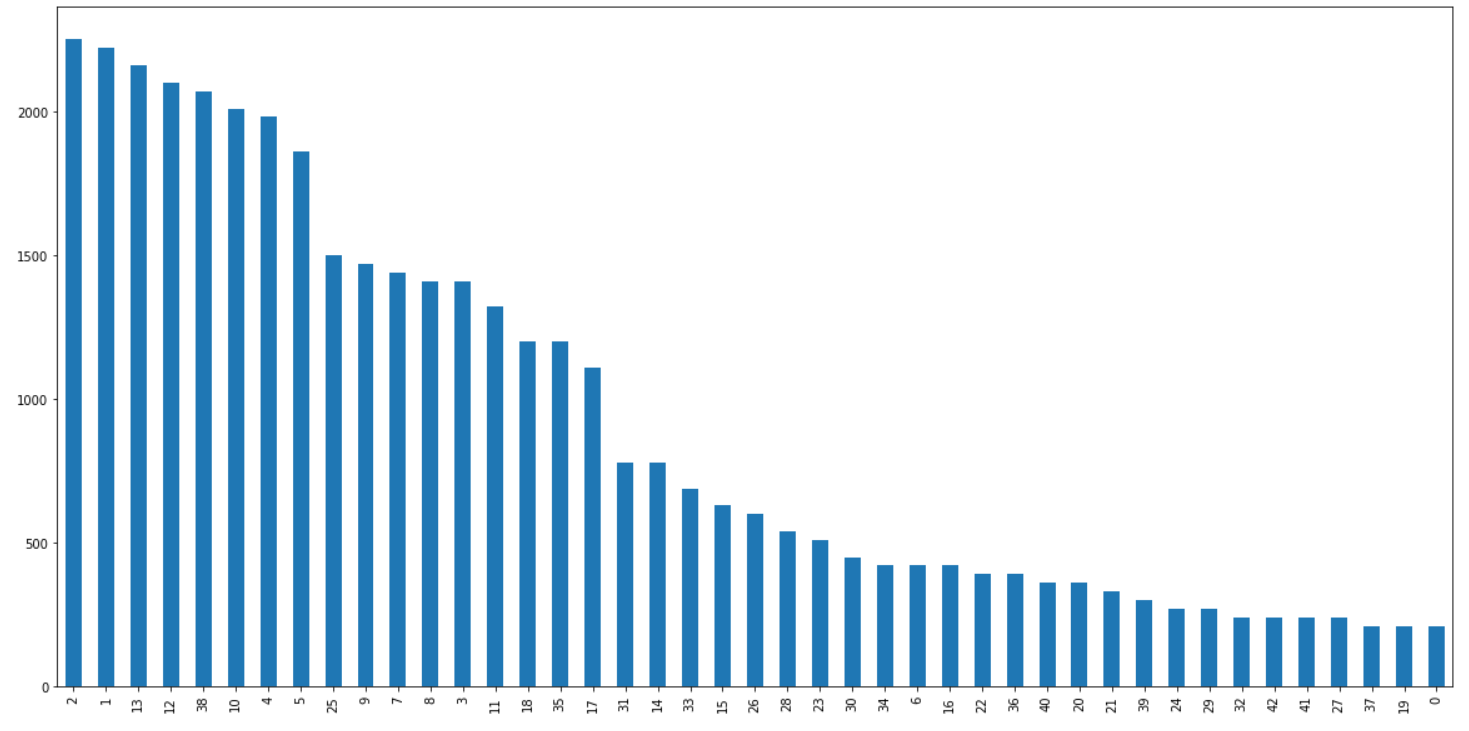








train\_df['ClassId'].value\_counts().plot.bar(figsize=(20, 10))train\_df['ClassId'].value\_counts().median()



dir = '/content/drive/MyDrive/DataScience'

plt.figure(figsize=(10, 10))

for i in range (0,43):

plt.subplot(7,7,i+1)

plt.xticks([])

plt.yticks([])

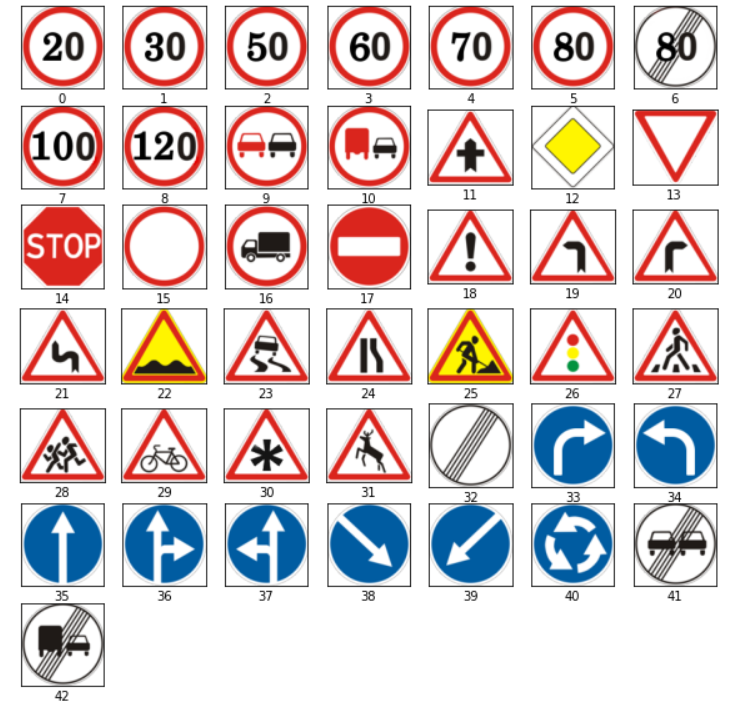
plt.grid(False)

path = dir + "/Meta/{0}.png".format(i)

img = plt.imread(path)

plt.imshow(img)

plt.xlabel(i)



import seaborn as sns

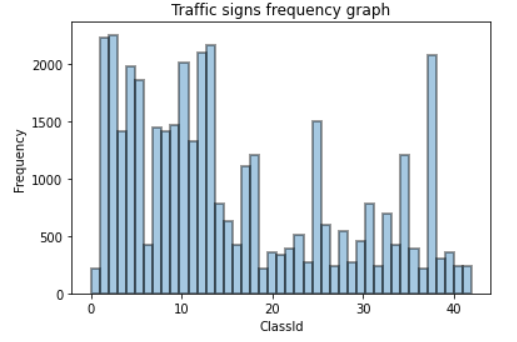
df=train\_csv['ClassId'].copy()

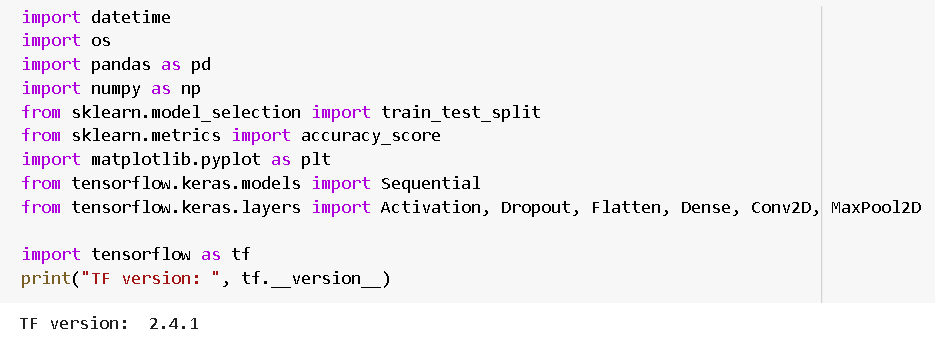
fig = sns.distplot(df, kde=False, bins = 43, hist = True, hist\_kws=dict(edgecolor="black", linewidth=2))

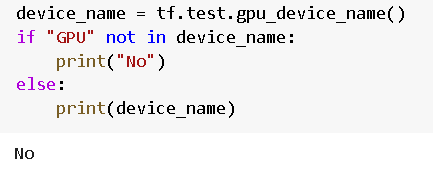
fig.set(title = "Traffic signs frequency graph",

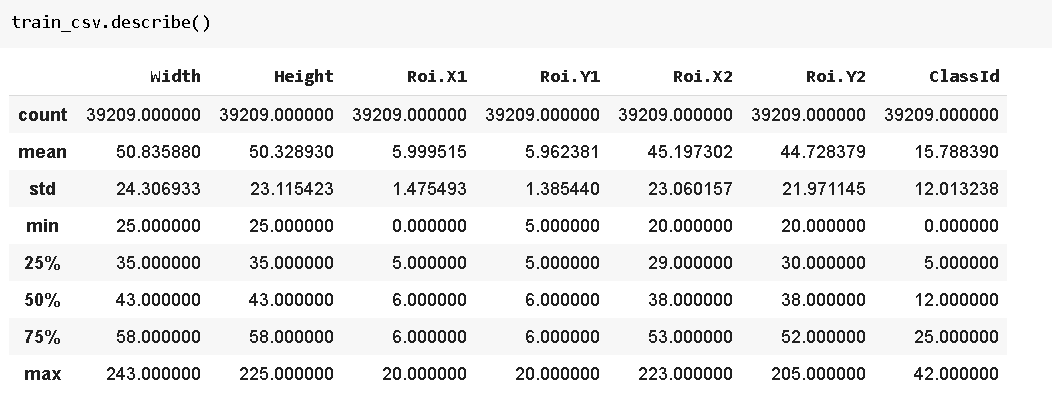
xlabel = "ClassId",

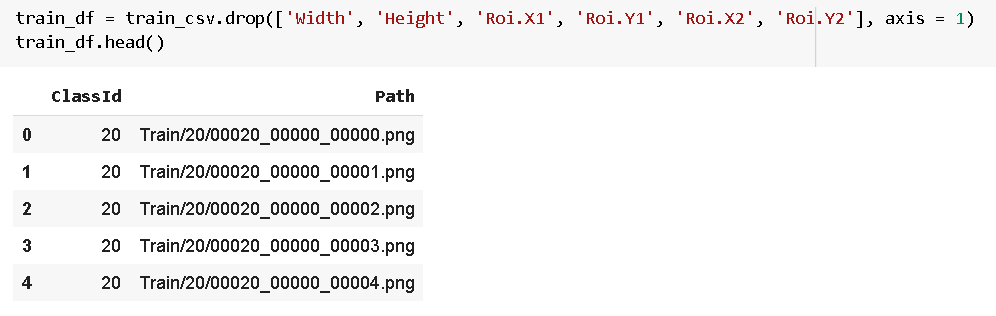
ylabel = "Frequency")

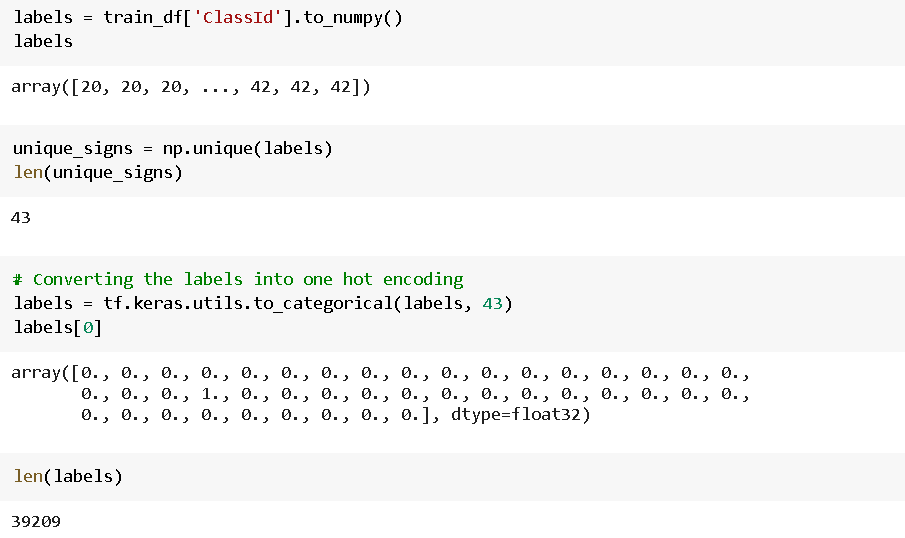


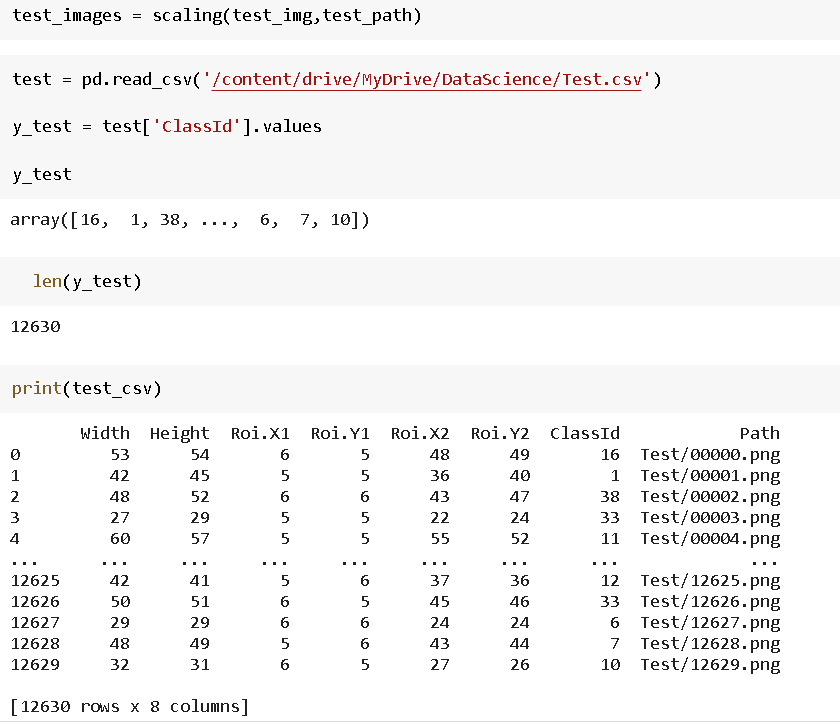


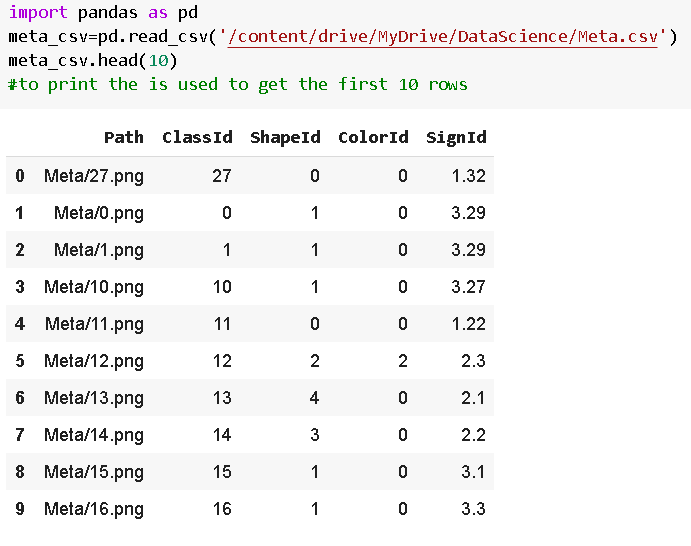


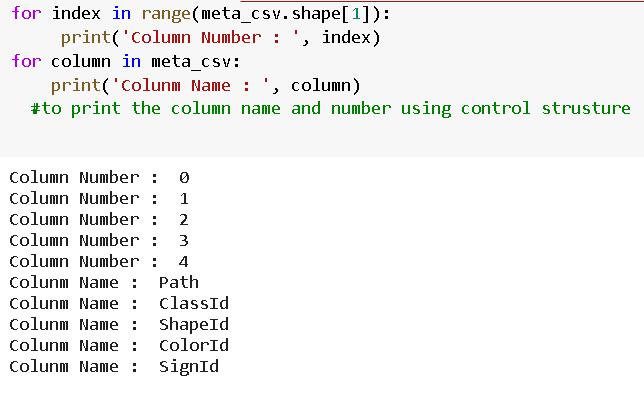


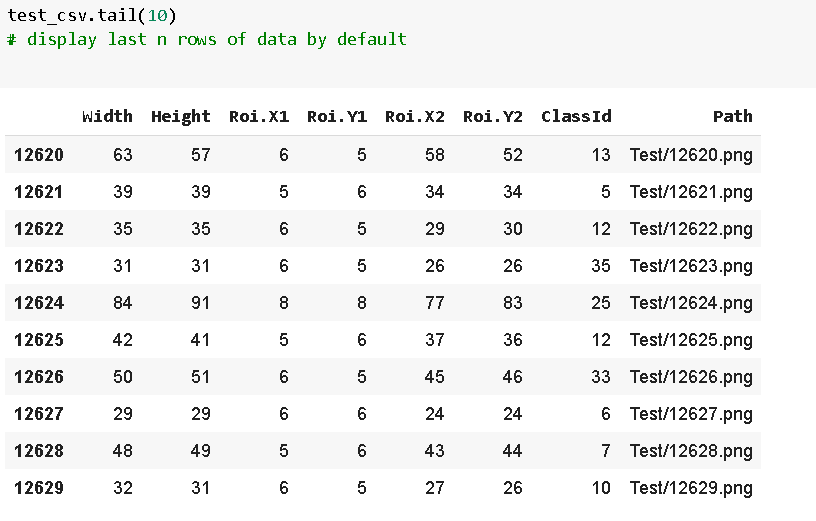


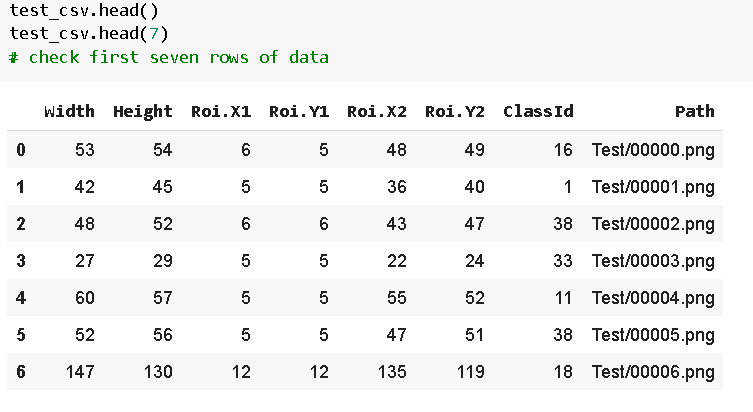






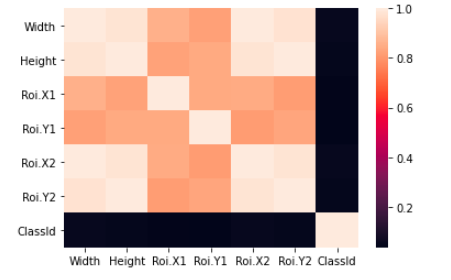






**4.2 Identifying the Correlation**

A heatmap is a graphical representation of data that uses a system of color-coding to represent different values[8].This paper also represents the correlation between the features using the heatmap.In general, correlation heatmap is used to compare the measurement of each pair of dimensions.The below figure represents the heat map of this dataset,



This implies that there is positive correlation between features.Here, the dark color implies stronger correlation and light color implies weaker correlation.

**4.3 Testing on Dataset**

The dataset had undergone various kinds of tests like the normality test like Shapiro testing, then correlation test like Pearson's Correlation test then stationary test like Kwiatkowski-Phillips-Schmidt-Shin test,Parametric Statistical Hypothesis test like Student’s t-test, non parametric statistical hypothesis test like Mann-Whitney U Test and conclusions were drawn from it. The overall result is displayed in the table below:

|  |  |
| --- | --- |
| **Shapiro test** | **Probably not Gausian at every level** |
| **Pearson's Correlation test** | **The attributes are dependent on each other** |
| **Kwiatkowski-Phillips-Schmidt-Shin test** | **The attributes are stationary** |
| **Student’s t-test** | **The attributes have probably different distributions.** |
| **Mann-Whitney U Test** | **The attributes have probably different distributions.** |

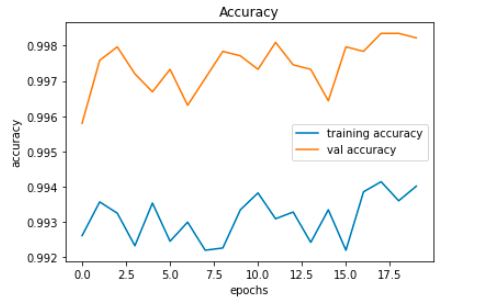
**Chapter 5 - Proposed Methodology**

The GTSRB(German Traffic Sign Recognition Benchmark) data set is used to train and test CNN. CNN is a multi-layered network, it is very similar to brain.Each layer of CNN is composed of multiple neurons. Each neuron receives an input to perform a task, and some operations and outputs pass as input to the next neuron. Convolutional layer is the core part of convolutional neural network, and its main function is feature selection. Compared with traditional machine learning, CNN has good classification performance.

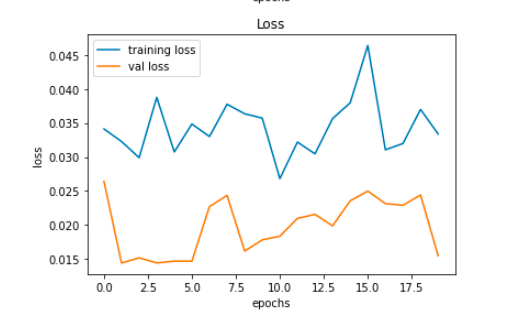
In this proposed methodology, the model makes use of an Keras, a neural

network library that can run on TensorFlow.Apart from this, we import various libraries like Conv2D, MaxPool2D, Dense, Flatten, Dropout for building the model and finally compiling the model.In order to prevent over-fitting, Dropout layer is added to the CNN to suppress over-fitting. Dropout can increase the randomness of the network.

The test result of the trained CNN shows that the accuracy of the model in detecting and recognizing traffic signs is 99.7%..The experimental result shows that the network model has good recognition accuracy for traffic signs recognition.



This image shows the Training accuracy



This image shows the Training loss



The obtained accuracy is displayed

**5.1 Workflow**

****

The overall workflow is is displayed in this flow diagram.Initially libraries are imported then the dataset is loaded after that the input images are read and stored in an array.After this,visualization,identifying correlation,dimensionality reduction and various tests were done on the dataset.After all these a model is built using CNN and compilation of model is done by using nearly 20 epochs.After that graphically training accuracy and loss is displayed.Finally the accuracy is predicted.

On doing all these we find that the training accuracy obtained in this model is 99.7% which means this is an efficient model.

**5.2 Conclusion**

In this paper, an approach based on the Convolutional Neural Network (CNN) for recognising traffic signs is proposed.Compared to many other models the model proposed in this paper shows a much higher accuracy of 99.7%.The goal of this research is to develop an efficient TSR system based on German traffic sign dataset and is almost achieved.

**Recording:**

https://drive.google.com/file/d/18pcGQB4IU4dP6HOYZaTJlvfWvbTIXLI1/view

**References**

i. Alturki, A.S. Traffic Sign Detection and Recognition Using Adaptive Threshold Segmentation with Fuzzy Neural Network Classification. In Proceedings of the 2018 International Symposium on Networks, Computers and Communications (ISNCC), Rome, Italy, 19–21 June 2018; pp. 1–7.

ii. C.C. Chang *et al.* [A fast VQ codebook search with initialization and search order](https://www.sciencedirect.com/science/article/pii/S0020025512006342) Inf. Sci.(2012)

iii. De la Escalera, A.; Armingol, J.M.; Mata, M. Traffic sign recognition and analysis for intelligent vehicles. Image Vis. Comput. 2003, 21, 247–258

iv. Loy, G.; Barnes, N. Fast shape-based road sign detection for a driver assistance system. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2004), Sendai, Japan, 28 September–2 October 2004; pp. 70–75.

v. Pettersson, N.; Petersson, L.; Andersson, L. The histogram feature-a resource-efficient weak classifier. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 678–683.