Traffic Sign Recognition System (TSRS): Convolution Neural Network

[1]Sanjana Saxena (21903045) [2]Jahnavi (21903044) [3]Suneeta Singh (21203028) [4]Rohit Kumar (21203019)

[1234]*Department of Computer Science & Engineering*

Dr B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India.

Abstract— In modern age, mostly everything around is automated and therefore TSRS plays an essential role in self-driven vehicles, traffic control as well as traffic safety. While driving, one tries to focus on the road and most of the time drivers fail to recognize the signs along the road, which could be harmful as it can cause a serious accident. This problem can be solved if there is an effective way. A Traffic Sign Recognition System (TSRS) plays a crucial role here. The system, recognize and detect a sign, thus giving notification to the driver about the speed of car, school ahead, work in progress, turn left or right etc. It ensures road safety as well as becomes more comfortable while driving on new roads. In this project, we propose to design Traffic Sign Recognition System by using the Convolutional Neural Network (CNN). The only reason of using CNN is because of its high recognition rate.

The main objective of our project is to design computer-based system which can detect the traffic sign automatically thus providing assistance to machine so that it can take the appropriate action. The approach is to build a model using convolutional neural network (CNN) by extracting traffic sign from images used as the dataset. We have used CNN model to classify the traffic sign and Color based segmentation to extract signs from images.

Keywords—Traffic Sign Recognition System, CNN.

I. INTRODUCTION

In the world of Artificial Intelligence and advancement in technologies, many scientist and big companies like Tesla, Uber, Google, Toyota, Audi, etc. are working on autonomous vehicles and self-driving cars. So, for achieving accuracy in this technology, the vehicles should be able to interpret traffic signs and make decisions accordingly. There are several algorithms which gives the ability to detect traffic signs.

Traffic Sign Recognition

Knowledge of Traffic signs play very essential role in today's livelihood. It contains several kinds of information which ensures the safety of people. Without the traffic sign, drivers might be clueless that what might be ahead of them. The annual global roach crash statistics say that approx three thousand people die every day in road accident and figure is much bigger in case there is no traffic sign and in other hand, it's very important having ability of driving. There are some rules that have given by government of driving ability.

In our project, we are going to implement a model to detect Different traffic signs using CNN and then drivers can take their directions accordingly. Through this model, people are able to read and understand traffic signs which are a very important task for all autonomous vehicles. This issue has some positivity like visibilities of different traffic signs are being unique which gives small variations in object and Signs are clearly visible to driver while driving. In other hand there are still some problems with low light and bad weather conditions.

The Dataset we will be taking here is the German traffic Sign Recognition Benchmark (GTSRB) dataset. However, the traffic signs are pre-cropped for us, implying that the dataset creators have manually labeled the signs within the pictures and extracted the traffic sign Region of Interest (ROI) for us, thereby simplifying the project. There are a several varieties of traffic signs sort of a stop, a right curve, no parking, speed restrictions, side arrow, up-down arrow then finish.

- **Techniques used**

CNN model

A Convolutional Neural Network (ConvNet/CNN) is one among the Deep Learning algorithmic rule during which image works as an input and allot significance to totally different aspects/objects among the image and be ready to distinguish one from the opposite.

The pre-processing of ConvNet is such a lot lower comparison from alternative algorithms.

When we point out unstructured information like pictures, text, voice, videos, hand designed options square measure time overwhelming, and not climbable in follow. that is why Neural Networks become more and more acknowledge a lot of appreciated to their capability to mechanically notice the representations needed for feature detection or classification from raw data. Manual feature planning is currently replaced by CNN and permits a machine to each learn the options and utilize them to perform a specific assignment.

CNN's were to begin with created and utilized around the 1980s. The foremost that a CNN seem do at that time was recognize manually written digits. It was generally utilized within the postal segments to studied zip codes, stick codes, etc. The vital thing to keep in mind almost any profound learning show is that it requires a huge sum of information to prepare and also requires a part of computing assets. This was a major downside for CNNs at that period and subsequently CNNs were as it were constrained to the postal segments and it failed to enter the world of machine learning.

Convolutional Neural Network (CNN) works

Firstly, image is taken as an input by the CNN model and then using the forward propagation it is sent to the network to obtain the desired output. The weight and offset of the network are updated by predicting the error of the obtained output and label. The principle followed is as: To start, it takes N inputs randomly from the dataset and assign each input weight and offset to initialize the learning rate in order to obtain actual output vector of the network. To calculate the output error, it takes picture an input and then performs comparison of the output vector with the target vector. Adjustment amount of each weight and offset is calculated to adjust the weight and threshold. At last it is observed that whether the indicator meets the accuracy requirement, if yes then goes to the next step otherwise continue the iteration. On the completion of training, the weight and offset are saved in the file which indicates that the stabilization of weight and formation of classifier.

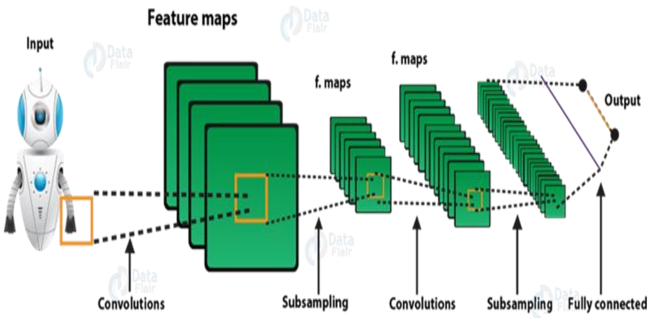


Fig:1 Working of CNN

Traffic Sign Classifier Graphical User Interface (GUI)

Here we have a tendency to ar planning to build a graphical program for our classifiers of Traffic signs with Tkinter. Tkinter could be a graphical user interface toolkit within the commonplace python library. Produce a replacement goes in the project folder and duplicate the code. Reserve it as graphical user interface.py and you'll run the code by typewriting python graphical user interface.py within the statement.

In this file, we've got 1st loaded the trained model exploitation Keras. So, for uploading the image; we have a tendency to produce the graphical user interface and a button is employed to classify that calls the classify () operate. The classify () operate is changing the image into the scale. This is often as a result of to predict the traffic sign we've got to produce a similar dimension we've got used once building the model. Then we have a tendency to predict the category, provide output range |variety} between (which is one in all our number of images) that represents the category it belongs to. We have a tendency to use the lexicon to induce the knowledge regarding the category.

Tools used: python, scikit-learn, keras, numpy.

II. Motivation

The first and foremost reason of choosing this particular work is to reduce the number of accidents which take place every day due to poor infrastructure of the pavement. By developing such system, one would underneath the bar of number of deaths, thus saving life of millions. Even the self-driven vehicles can extract huge benefits from the TSR system which could help the them to recognize and understand the traffic signs. Apart from this, already conducted researches on the same topic are also one of the reasons of choosing Traffic sign detection System as the Project. Numerous good ranked Research Papers have been published which motivated us to develop such a system that can overcome all the traffic related difficulties and even can guide the driver on the safer route. Most of the authors have built such a system by using different Machine Learning concepts and of all those, the one which fascinated us the most, is the use of CNN model to develop the Traffic Sign Detection System. This is because of its ability to detect the essential features without any human intervention and it can work efficiently with the image classification problem which is counted as a positive aspect when working with the set of images as the dataset. Overall, the motivation behind developing such a system is crystal clear due to benefits it offers by saving lives of the population around as well as saving cost.

III. Previous Works

Nowadays, Traffic Sign Recognition System had gained popularity because of its image processing ability and many others benefits that it provides as the system. An ample amount of research has been already done on Traffic Sign Recognition System (TSRS) using different ML techniques.

In [1], Nazmul Hasan proposed an approach in a paper which was published in 2020 to recognize the traffic sign using two Machine Learning algorithms that is SVM and CNN.A real dataset was prepared from random videos in which essential traffic sign images were cropped. In total 1200 images were captured, that were further classified into training and validation set. In order to obtain good result, 80% data was considered for training purpose while remaining 20% for testing. After inputting these datasets in SVM and CNN, the 98.33% accuracy was achieved in case of SVM while 99.56% training accuracy and 96.40% accuracy was obtained from CNN.

In [2], This Paper was printed in 2020; it contains implementation on Traffic sign Detection and recognition mistreatment Image process. Image analysis consists of 3 steps: Detection, Segmentation and classification. This paper centered on correct and high economical results of detection and recognition. Deep Convolutional neural network square measure used for training then classifying traffic sign. Here some methodology square measure accustomed phase traffic sign by applying different techniques like filtering, edge detection, DWT Feature segmentation (color segmentation). The YCBCR conversion method square measure used for comparison. Proposed feature selection extraction, wherever the 2 datasets of GTSRB and GTSDB square measure used for implementation. These datasets contain several advanced traffic signs. Final result is showed in screen by notification crop up.

In [3], Autonomous Traffic Sign Detection (ATSR) system was made using Deep CNN model. This paper was published in 2019 in 16th International Learning & Technology Conference and it carried out two main tasks; one was to create a database on its own and second one was to develop and design CNN architecture. The dataset consisted of 2728 RGB images which required preprocessing before inputting it to CNN network. The process of preprocessing converted it to gray scale images and then the CNN was designed which achieved the accuracy of 100 % and was successful in detecting traffic symbols.

In [4], Traffic Sign Detection and Classification has been done using Deep Learning. The paper was published in May 2019 in IRJET and author of paper is Shalinia. The author proposed the system which used the feature extraction method in order to obtain the efficient results. The German Traffic sign detection benchmark dataset has been used in the project which contains 39209 images in total. For the preprocessing, Adam was used which basically includes all the aspects of CNN. In this System, RGB images were preprocessed using Local Histogram Equalization and Normalization and the data in hand was trained and tested using TensorFlow with LeNet architecture. To improve the quality of RGB images, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied on images given as input. The accuracy of 98.3% was achieved with RGB which concluded that RGB were more accurate in comparison with Gray images.

In [5] Oct 2018, Citlalli Gámez Serna and Yassine Ruichek had planned the implementation on classification of real-world traffic signs on European dataset. This dataset combined with six European countries traffic sign. This implementation had taken quite 80,000 pictures divided in 164 categories. Models square measure trained in GPU mode employing a NVIDIA GeForce GTX1080Ti with 11GB of memory. For this implementation they were getting to perform dataset comparison between classification performances of 5 CNN design. Comparison had taken on GTSDB and European Dataset. After performing training and validation on each dataset, analyzed accuracy of model. Some techniques like Preprocessing and data-augmentation square measure accustomed improve accuracy. All these implementation work on TensorFlow Framework. Here CNN asymmetric and CNN-8 layers model was enforced on GTSRB and European information set with original and augmented data. Accuracy goes to 99.37% on GTSRB dataset and 98.99% on European dataset.

In [6], April 2017, Traffic sign recognition task had done by Alexander Shustanova and Pavel Yakimova.This paper contains implementation of classification formula for the real- time traffic sign recognition task with preprocessing and localization steps. In which they used Deep learning library TensorFlow. After some implementation they normalize all the layers. Due to sizable number of layers, training taken batch of 50 pictures, result not provide excellent accuracy then they scale back convolutional layer. After reducing range of iterations, they found that result was worse than before, it results that one convolutional layer isn't enough to urge excellent accuracy. To judge the classification, execution time is 20ms. Once many implementations they found that it's most effective methodology for TSR mistreatment GTSDB and GTSRB is mistreatment convolutional neural network for each localizing and classifying traffic signs and accuracy goes to 99.89% once police detecting a symbol and 99.55% once classifying it. It acknowledges traffic sign distance up to 50m.This methodology was enforced on device with Nvidia Tegra K1 processor.

IV. Proposed Solution

Two recognize traffic sign we focused on machine learning algorithms. Many research works used CNN method to classify traffic sign. Here we used this algorithm to detect and recognize traffic signs. In this project, we will used Convolutional Neural Network to build and test this model. By using TensorFlow and Keras, we build this traffic sign classification model. It is a multiclass classification problem. There are some steps, by which we proposed this model using CNN.

Step 1: First we find Data on Kaggle and Loading it into Collab.

Step 2: Preprocessing the images and Visualizing them.

Step 3: Finding out the mean of the dimensions and resizing all images accordingly.

Step 4: Converting the images into a numpy array and normalize them.

Step 5: Checking class imbalance.

Step 6: Splitting the data and performing one hot encoding.

Step 7: Creating the Model Architecture, Compiling the Model and then fitting it.

Step 8: Plotting the Accuracy and loss against each epoch.

Step 9: Preprocessing the test data and make Predictions on it.

Step 10: Now Visualizes the Actual and Predicted labels for the given test images.

Data collection

To build this model, here we collect some traffic signs from different traffic sign areas. And we categorized the whole dataset into training and validation. We have total 39209 images to propose CNN Model.

We Considered only 42 different classes of dataset after preprocessing of whole dataset. Split the whole dataset for training and validation purpose. To build this model we consider, 80% of data for training and 20% of data for testing purpose. Below this table describe about the traffic signs.

Description	Traffic Sign	Description	Traffic Sign
Turn Left		Danger	
Stop		40km/h	
Only Left		30km/h	
Only Right		Turn Right	
Road Merges Ahead		Pedestrian	
Speed Breaker		Bike	

Fig:2 Table of dataset

In the next part ,we are going to elaborate the procedure of approach to recognize traffic signs.

Convolution Neural Network

CNN is a deep learning approach. It gives highest accurate result that is why used by many of research works. CNN Model contains following functions which is used:

- Relu function
- Softmax function

"Relu" function used for evaluating non-liner function. It performs operation on pixels and keep 0 in place of all negative values."Relu" gives best result among all different non-linear function which is used in CNN. And "Softmax" function mainly used for multi-classes problems, it assigns decimal probabilities to each different classes. It helps in training convergence more fast than other one. SoftMax executed through neural network layer before the output layer.

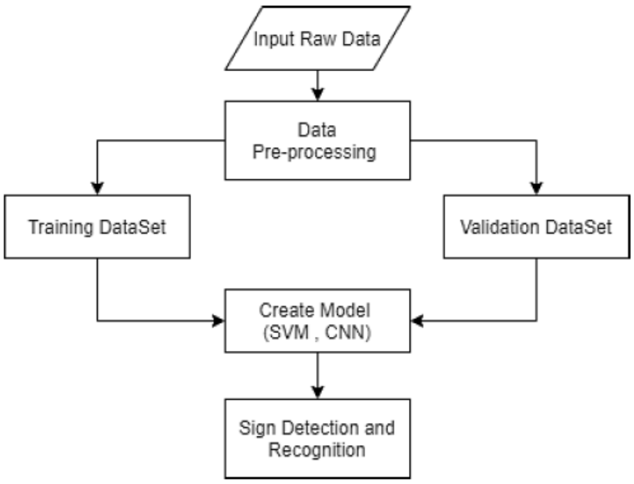


Fig:3 Flowchart

V. Experimental Setup and Results

In this project, we will use Convolutional Neural Network to build train and test a traffic sign classification model. We will build this model using TensorFlow and keras. It is a multiclass classification problem.

Import the data

We will start by connecting to Kaggle using Kaggle API which can be downloaded from your Kaggle account's settings and uploading it in the upload box.

We run our first cell and select the *kaggle.json* file downloaded from the Kaggle account settings page.

```
from google.colab import files
files.upload()
```

```
Choose Files kaggle.json
• kaggle.json(application/json) - 70 bytes, last modified: 11/30/2021 - 100% done
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username":"sanjanasaxena5","key":"8b40755f9a11cd9533aa93c51d06e4f7"}'}
```

Setting up Kaggle APIs

Next, install the Kaggle API client.

```
!pip install -q kaggle
```

Installing kaggle api using pip

Import the libraries

Import the basic libraries padas, os, numpy.
Other libraries are matplotlib.pyplot, imread, seaborn, random, and tensorflow.

Data visualization

Visualizing some images of traffic sign from the test dataset. we can see here that the dimension of images is uneven.

Fig:4 Some images from the dataset

For further processing we will require the images of same dimension. So, we will start storing the dimension of all the images from training dataset from all 43 classes.

Now we will reshape the images into (50,50) and also store their label ids.

Now we will convert all the images into numpy array and normalize them.

Checking the shape of the images. Here we find that there are 39209 images with a shape of (50,50,3.)

Now we will observe images per class for checking whether the data is balanced or not.

From the result we find that data is balanced.

Splitting the data

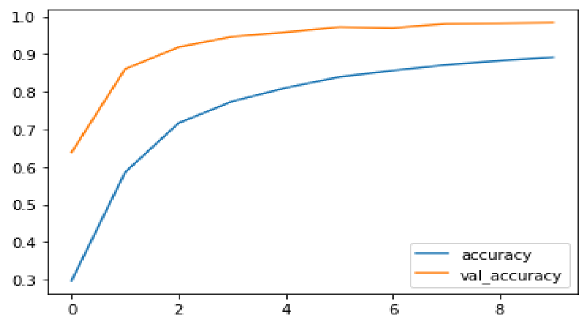
The next step would be to split the data into training and validation with 80% of training data and 20% of validation data.

Converting the classes column into categorical using to_categorical() function.

Defining the model architecture. In this we will define all the layers with their input shape kernel size, activation, etc.

After that, we will fit the model and observe how our is getting trained on each epoch.

Next we will visualize the accuracy and loss per epoch. For this we will store the model history in the pandas data frame and plot them.



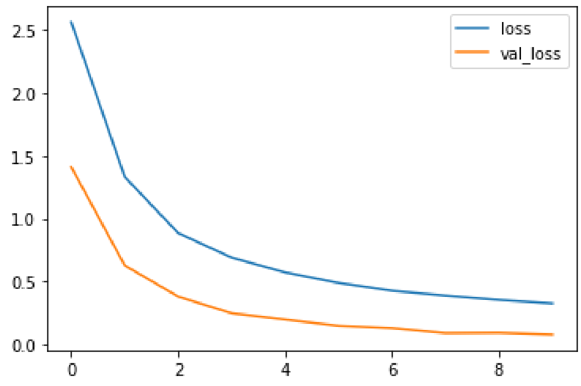


Fig:5 Visualizing the accuracy and loss per epoch.

Next, we storing the labels according to the image classes.

```
all_lables = ['Speed limit (20km/h)',
              'Speed limit (30km/h)',
              'Speed limit (50km/h)',
              'Speed limit (60km/h)',
              'Speed limit (70km/h)',
              'Speed limit (80km/h)',
              'End of speed limit (80km/h)',
              'Speed limit (100km/h)',
              'Speed limit (120km/h)',
              'No passing',
              'No passing for vechiles over 3.5 metric tons',
              'Right-of-way at the next intersection',
              'Priority road','Yield','Stop','No vechiles',
              'Vechiles over 3.5 metric tons prohibited',
              'No entry','General caution',
              'Dangerous curve to the left',
              'Dangerous curve to the right','Double curve',
              'Bumpy road','Slippery road',
              'Road narrows on the right',
              'Road work','Traffic signals','Pedestrians',
              'Children crossing','Bicycles crossing',
              'Beware of ice/snow',
              'Wild animals crossing',
              'End of all speed and passing limits',
              'Turn right ahead','Turn left ahead',
              'Ahead only','Go straight or right',
              'Go straight or left','Keep right',
              'Keep left','Roundabout mandatory',
              'End of no passing',
              'End of no passing by vechiles over 3.5 metric']
```

Visualize the test image

Let's visualize test image

```
img = Image.open(test_path + '/00001.png')
img
```

After visualizing the image, find out original label for the image above.

And we find;

```
Original label :  Speed limit (30km/h)
```

And in the last; find out the predicted label for the image above.

And the output is;

```
print("Predicted label : ",all_lables[y_pred[1]])
```

```
Predicted label :  Speed limit (30km/h)
```

VI. Comparative Analtsis

The issues which are being discussed in the report are the main reasons which are hindering the growth of the Traffic Sign Recognition System but by applying various other Machine Learning algorithms, the system can be made more robust. Numerous amounts of researches and studies had been conducted in past. Among all, one was carried by Escalera et al and Sheng et al where the system was built using Probabilistic neural network in which the Traffic Signs were detected in noisy, blur background and even in variant lightning.

In order to make system work in real environment, somewhat fast algorithm is required. Therefore to fulfill this need, a detailed comparison was carried between SVM, Decision tree and MLP.The result obtained depicted that the highest accuracy rate of 94.5% was achieved when Decision Tree was used whereas nearly 84.2% was obtained in case of SVM and 89.3% in case of MLP.If we consider the computational time ,then studies revealed that for adding new classes, the whole amount of dataset was needed to be changed in which neural network took more computational time as compared to SVM.

Another study came forward where after carrying out RGB segmentation, a genetic algorithm was used for detecting the Traffic signs. The positive aspect of the algorithm was that it does not depend on the error and can easily deal with multi-dimensional as well as multi directional problems and it was figured out that the detection rate of the genetic algorithm is quite high as compared to other algorithms. As noted earlier that, Support Vector Machine (SVM) is another known technique which can be used to construct a robust Traffic sign detection system. Gil-Jimenez-et al and Prisacariu et al both made the use of SVM to detect the Speed Limit Traffic Sign but using different techniques. The former one used Gaussian Kernel to recognize the speed limit sign whereas the latter one used Haar like feature. The result obtained from both the researches were remarkable and can be seen in the image below.

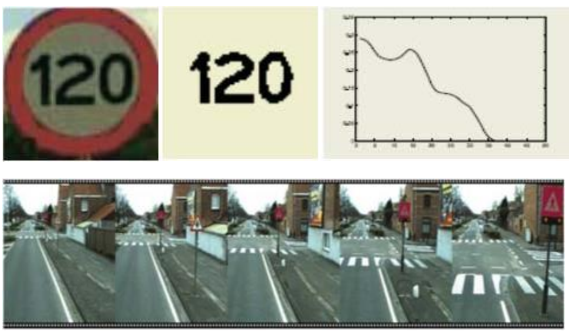
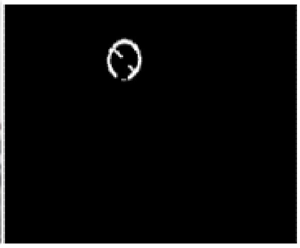
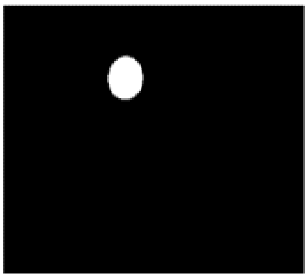


Fig:6 Speed limit sign detection using SVM

Liet al made use of HIS transformation technique and the fuzzy shape detector for recognizing the Traffic signs. The interesting point about this research was that it was carried forward in different weather such as sunny, rainy and cloudy. The outcome obtained from the process of segmentation was different for different weather. The picture shown below of occluded Sign is captured during sunny weather.



[A] **[B]**



[C]

Fig:7 Segmentation process on the occluded Traffic sign

Haung et al used Adaboosted technique which was proven to be fast as compared to others and can even capture blur image in any lightening condition. In order to determine the accuracy of system using the aforementioned method 925 images of speed limit were taken along with 2040 random images with no road sign and result was approximately 95.5 %.

Ref.	Methods	irregular lighting	Blur & fade image	Multiple appearance of traffic sign
[1]	Probabilistic Neural Network	√	√	
[2]	Support Vector Machine	√	√	
[3]	Genetic Algorithm + Probabilistic NN	√	√	
[4]	Adaptive shape recognizer+ Probabilistic Neural Network	√		
[5]	Haar like features + SVM			√
[6]	HSI Transformation +Fuzzy shape recognizer			√
[7]	AdaBoost	√	√	√

Ref.	Methods	Partial obscured sign	Fast algorithm for Real-time	Blur effect
[1]	Probabilistic Neural Network		√	
[2]	Support Vector Machine		√	
[3]	Genetic Algorithm + Probabilistic NN		√	
[4]	Adaptive shape recognizer+ Probabilistic Neural Network		√	
[5]	Haar like features + SVM	√		√
[6]	HSI Transformation +Fuzzy shape recognizer	√		
[7]	AdaBoost		√	

Ref.	Methods	Rotation scaling	Noisy background	Viewing angle
[1]	Probabilistic Neural Network		√	
[2]	Support Vector Machine	√	√	√
[3]	Genetic Algorithm + Probabilistic NN		√	
[4]	Adaptive shape recognizer+ Probabilistic Neural Network		√	
[5]	Haar like features + SVM			√
[6]	HSI Transformation +Fuzzy shape recognizer	√		√
[7]	AdaBoost			

Fig:8 Comparative study between different methods

If we want to compare this project with previous one which was published in 2019 by ELSEVIER publication, titled as “Autonomous traffic sign Detection and recognition using CNN “, then we found that previous project was using CNN as Feature extraction and reduction technique and after that split the data for training and testing purpose. And It had taken traffic sign via camera which was attached in vehicles, recognize them and classifies their sign and after that gave voice notification to driver or automatically take decisions in autonomous cars. This project contains datasets of Arabic traffic and road signs, many images and different classes.

While in our project, we use CNN where after preprocessing and reshaping the size of datasets, split the data into training and validation part. And to classify traffic sign used “RELU” and “SOFTMAX” Activation function to get more accurate result. The Datasets containing are German traffic sign datasets and 39209 images with 43 classes. This project detects traffic sign with their shape and label on the images for visualization.

VII. CONCLUSION AND FUTURE WORK

The prime objective of the project is to detect different Traffic signs and classify them according to their shape so that visualization can become effortless. We started our work by downloading the appropriate dataset. After downloading it, we preprocessed our data by assigning the same dimensions to all the images that are present in the dataset. At last, we finally obtained the model using TensorFlow and Keras. The resulting model was trained and tested using Convolutional Neural Network (CNN) which reached the accuracy of 95% in just 50 epoch which was quite remarkable. The main merit of the project was that CNN was used with learnable as well as fixed layer to produce a good outcome.

In our future work, we decided to make our system more efficient by training the CNN is such a manner that it can include more traffic sign classes and can even work in bad weather. Another thought that strikes us was to use CNN not only for the purpose of classification but also for detecting the objects. To make Traffic Sign Detection System more reliable, GPS must be able to represent the different Traffic signs for more accurate detection. In addition, the driver safety can be maintained by introducing inter-vehicle communication through Sonar while driving on the road. If all the discussed future work will be accomplished successfully, then the system proposed by us would become stronger and can overcome numerous issues related to the real time application.

REFERENCES

[1] Karthikeyan D, 2Enitha C, 3Bharathi S, 4Durkadevi K, "Traffic Sign Detection and Recognition using Image Processing" , International Journal of Engineering Research & Technology (IJERT), Volume 8, Issue 08, Thottiyam, Trichy, Tamilnadu, India,2020.

[2] Mr.K.Kalaiarasan, J.Akilan, M.Naveen Kumar, M.P.Sukumar, V.Surya Prasath, "Traffic Signs Recognition Using Cnn & Keras," International Journal of Future Generation Communication and Networking Vol. 13, No. 4, (2020), pp. 267–274.

[3] Nazmul Hasan, Tanvir Anzum, and Nusrat Jahan, Computer Science and Engineering Department Daffodil International University, "Traffic Sign Recognition System (TSRS): SVM and Convolutional Neural Network", Research Gate (2020).

[4] Aashrith Vennelakanti, Smriti Shreya, Resmi Rajendran, Debasis Sarkar, Deepak Muddegowda, Phanish Hanagal, "Traffic Sign Detection and Recognition using a CNN Ensemble," 2019 IEEE International Conference on Consumer Electronics (ICCE).

[5] Danyah A. Alghmghama, Ghazanfar Latif, Jaafar Alghazo, Loay Alzubaidi, "Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN", 16th International Learning & Technology Conference 2019 Procedia Computer Science 163 (2019) 266–274.

[6] Shalinia G , Student, Dept of CSE, MSRIT, Karnataka, India, "Traffic Sign Classification and Detection using Deep Learning" , International Research Journal of Engineering and Technology (IRJET) Volume: 06 Issue: 05th May (2019).

[7] CITLALLI GÁMEZ SERNA AND YASSINE RUICHEK, "Classification of Traffic Signs: The European Dataset", IEEE Access ,vol. 6, pp. 78136-78148 ,France,Oct. 2018.

[8] Alexander Shustanova, Pavel Yakimova, "CNN Design for Real-Time Traffic Sign Recognition" ,ELSEVIER ,pp. , 3rd International Conference "Information Technology and Nanotechnology", ITNT-2017,718-725 ,Samara, Russia, April 2017.

[9] Yihui Wu, Yulong Liu, Jianmin Li, Huaping Liu, Xiaolin Hu, "Traffic Sign Detection based on Convolutional Neural Networks", International Joint Conference on Neural Networks, Dallas, Texas, USA, August 4-9, 2013.

[10] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011, July). The German traffic sign recognition benchmark: a multi-class classification competition. In The 2011 international joint conference on neural networks (pp. 1453-1460). IEEE.

[11] B. Hoferlin and K. Zimmermann, "Towards reliable traffic sign recognition," in Intelligent Vehicles Symposium, 2009 IEEE, 2009, pp. 324– 329.

[12] Fleyeh, H., "Traffic and road sign recognition" (Doctoral dissertation 2008).