Road Sign Detection

Toshiba Ansari
Student, AIT-CSE
Chandigarh University
Mohali, Punjab
India
20bcs6671@cuchd.in

Sarthak Jain
Student, AIT-CSE
Chandigarh University
Mohali, Punjab
India
20bcs6689@cuchd.in

Abstract **- Humans are becoming more reliant on technology in this age of artificial intelligence. With the boosted technology, multinational corporations like Google, Tesla, Uber, Ford, Audi, Toyota, Mercedes-Benz, and many more are developing automatic automobiles. They are working to develop more precise autonomous or driverless vehicles.**

Traffic sign detection and recognition plays an important role in expert systems, such as traffic assistance driving systems and automatic driving systems. It instantly assists drivers or automatic driving systems in detecting and recognizing traffic signs effectively. This paper presents the design and implementation of a road sign detection system for autonomous driving applications. The motivation behind this system is to improve the safety of autonomous vehicles by accurately detecting and recognizing road signs.

The system is based on a deep learning approach specifically a convolutional neural network (CNN) which is trained on a dataset of German Traffic Sign Recognition Benchmark (GTSRB) to learn the features that distinguish different types of signs. The CNN is integrated with a detection algorithm to identify road signs in images. The results show that the proposed approach achieves high accuracy and robustness in detecting road signs with a low false positive rate.

Overall, the system demonstrates the potential of deep learning techniques for improving road sign detection in autonomous driving systems and provides a foundation for future research in this area. The system can contribute to the development of safer and more efficient autonomous vehicles with potential applications in a variety of industries including transportation, logistics and public safety. It covers detailed intuition about architecture and how we reach the solution and increase the accuracy.

Keywords: - Artificial Intelligence, Machine learning, Deep Learning, Autonomous vehicle, Neural Network, GTSRB dataset, CNN.

I. INTRODUCTION

According to global road crash data, roughly 1.3 million people die in traffic accidents each year [1]. Unfortunately, drunk driving, speeding, reckless driving, fatigue, and driver distraction continue to be the leading causes of road deaths. With developing traffic control technologies, there is a good chance the auto mobilist will miss part of the traffic. A computer vision system that can descry and identify traffic signs could help motorists avoid accidents in a variety of ways. The computer vision technology might condense reality by displaying forthcoming warning signs ahead of time, or indeed keeping them shown on a screen after the sign has history. This would make it less likely that the motorist would miss an important sign.

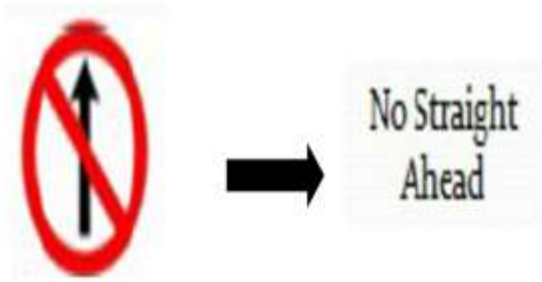


Figure 1: Image of the sign (taken from GTSRB) and its transcription into digital text

Figure 2: Traffic Signs

A. *Machine Learning*

Machine Learning is a data analytics technique that trains computers to learn from experience in the same way that humans and animals do. [2]

Instead of relying on a model built on a preconceived equation, machine learning algorithms use computational techniques to "learn" information straight from data. Machine Learning has emerged as a crucial tool for resolving issues in fields such as finance and healthcare.

Pattern Recognition: Pattern recognition is the process of identifying patterns using machine learning techniques. It categorizes data using statistical data or knowledge derived from patterns and their representation.

Credit scoring and algorithmic trading are two examples of computational finance.

Tumour detection, medication discovery, and DNA sequencing are all examples of biology. Voice recognition applications based on natural language processing.

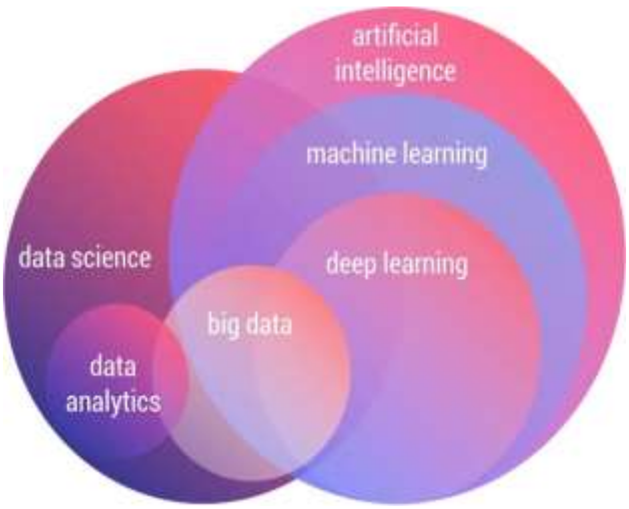


Figure 3: - Machine Learning Representation

B. *Deep Learning*

Feature Engineering is an important step that involves selecting and extracting relevant features from the input data. Early Feature Extraction is required by Machine Learning for features, and classification is conducted on it. Deep Learning, on the other hand, functions as a "black box," extracting and classifying features on its own.

The basic job is to classify the supplied image as face or non-facial, as shown in the diagram above. Machine learning requires an image attribute such as edges, colour, shape, and so on, and does classification on its own.

Deep learning, on the other hand, extracts feature and classifies them on their own. This given image is an example of how Convolutional Neural Networks work. Each layer of CNN takes a feature, and the Fully Connected Layer does categorization.

The major conclusion is that Deep Learning uses deep neural networks to extract data and classify itself. It works on complex problems that require large amounts of data.

As we can see the representation of deep learning in the given figure below [3], using the example of object recognition and its feature extraction and classification.

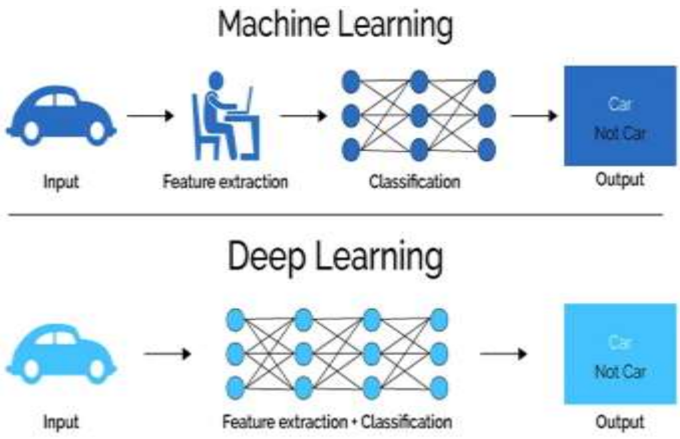


Figure 4: Deep Learning VS Machine Learning

C. *Convolutional Neural Network (CNN)*

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take an image as input, give significance (learning weights and biases) to different aspects of the image, and distinguish one from the other. When compared to other classification algorithms, the amount of pre-processing required by a ConvNet is significantly less. [4]

Whereas primitive methods require hand-engineering of filters, ConvNets can learn these filters/characteristics with enough training. A ConvNet's structure is modelled after the way the visual cortex is set up and resembles the way neurons are connected in the human brain. Individual neurons can only react to changes in a small area of the field of vision called the Receptive Field.

CNN are composed of several layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers perform the feature extraction by applying filters to the input image. Pooling layers reduce the dimensionality of the output from the convolutional layers down-sampling the features. Fully connected layers perform the classification by combining features from the previous layers and making a prediction. They are widely used in industries such as automotive, healthcare, and entertainment.

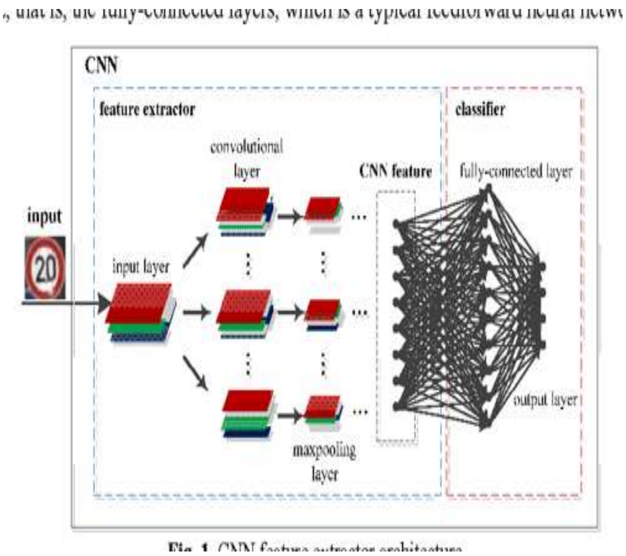


Figure 5: CNN representation

The challenges we encountered in this area were as follows:

- Although the color of the same type of traffic signs is mostly consistent, in outdoor settings, illumination and light direction have a significant impact on the color of the traffic signs. As a result, the color information is not entirely trustworthy.

- The shape information of traffic signs is no longer completely valid because vehicle mounted cameras are not always perpendicular to the traffic signs and the shape of traffic signs is frequently altered in road scenes.

- Because buildings, trees, and other cars frequently block the view of traffic signs in some road scenarios, we had to recognize them using just partial information. Other issues, such as traffic sign damage, rain, snow, and fog, are also mentioned.

- Traffic sign detection typically involves detecting multiple classes of signs, such as speed limit signs, stop signs, and yield signs, which can make it more challenging to design a detection algorithm that can accurately detect and recognize all classes of signs.

Some challenging examples are shown in Figure 6.

Figure 6: Road Signs

In this research, we describe a traffic sign detection and classification system for traffic sign recognition. The following is a summary of the paper's main contributions:

- We propose CNN model with convolutional layer, pooling layer, max layer and activation function.

- Convolutional layer performs a set of convolution operations on the input feature map and produces an output feature map with a reduced spatial size.

- The pooling layer then takes the output feature map from the convolutional layer and reduces its spatial size even further by replacing a set of adjacent values with a single value.

- ReLU activation function used in the hidden layers of a neural network to learn discriminative features from the input image. These features then be used by the subsequent layers to classify the image into the correct traffic sign category.

- Our method performs only one feature extraction through the detection and classification stage, which causes feature sharing throughout the two stages.

- Compared with algorithms used in the different feature extraction methods, in the detection and classification stage, this saves a lot of processing time and makes it feasible for use in real time applications.

II. Literature Survey

Traffic sign detection is the first stage of the process, which is concerned with the location and size of the traffic signs in the images of traffic scenes. Traffic sign recognition is the second stage of the process, which pays close attention to the classification of what exact class the traffic signs belong to. Traffic sign recognition is frequently employed with classifiers like convolutional neural networks (CNNs) and SVM with discriminative features. Traffic sign detection is typically based on the form and colour aspects of traffic signs. It is not difficult for human to recognize the sign but for computer it is.

J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel [5] in their paper compare the traffic sign recognition performance of human to that of the state-of-the-art machine learning algorithm. They have used the GTSRB dataset with 51,480 images and 43 classes. The testing is done on unseen images. The results showed that the best performing algorithm was a convolutional neural network (CNN) with an accuracy of 99.46%. The human participants had an average accuracy of 99.2%, which was slightly better than the CNN but not statistically significant. The study also found that the performance of the machine learning algorithms was highly dependent on the quality of the input images. In particular, images with low contrast and poor lighting conditions proved difficult for the algorithms to recognize.

Brogi, Alberto, et al. proposed a paper "Real Time Road Signs Recognition" [6] that presents a system for real-time recognition of road signs using a machine learning approach. The system uses a camera to capture images of the road signs and then processes them using a convolutional neural network (CNN). The authors used a dataset of over 3,000 road sign images and trained the CNN to recognize 43 different types of road signs. The system was then tested on a set of previously unseen images and achieved an accuracy of 96.4%. The system also includes a real-time detection algorithm that can detect road signs in the camera feed and classify them using the trained CNN. The system is capable of recognizing road signs in real-time with a processing time of less than 20 milliseconds per frame. They suggest that the system could be used in autonomous vehicles or as an assistive technology for drivers to improve road safety. They also note that the system could be improved by using larger datasets and more advanced machine learning algorithms.

N. Barnes, A. Zelinsky and L. S. Fletcher presented a paper "Real-Time Speed Sign Detection Using the Radial Symmetry Detector" [7] that presents a system for real-time detection of speed limit signs in driving videos. The system uses a radial symmetry detector (RSD) to detect circular shapes in the video frames and then applies a set of rules to determine if the circular shape is a speed limit sign. The authors used a dataset of over 5,000 images of speed limit signs and trained the RSD to detect circular shapes that resemble speed limit signs. The system was tested on a set of previously unseen video frames and achieved an accuracy of 94.8%. The system also includes a real-time processing algorithm that can detect speed limit signs in driving videos with a processing time of less than 40 milliseconds per frame. The authors suggest that the system could be integrated into advanced driver assistance systems to improve road safety and reduce the number of speeding violations. They note that the system could be improved by incorporating more advanced machine learning algorithms and using larger datasets of speed limit signs to train the RSD. They also suggest that the system could be extended to detect other types of road signs using similar techniques.

A. Møgelmose et al. worked on a paper "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey" [8] that provides an overview of vision-based traffic sign detection systems and their potential applications in intelligent driver assistance systems (IDAS). The authors discuss the various components of a typical traffic sign detection system, including image acquisition, pre-processing, feature extraction, and classification. The paper also reviews several existing traffic sign detection systems, including those that use traditional computer vision techniques and those that use machine learning algorithms. They highlight the strengths and weaknesses of each approach and suggest that a combination of both techniques may yield the best results. They also discuss the potential applications of traffic sign detection systems in IDAS, such as speed limit recognition, stop sign detection, and pedestrian crossing detection. They suggest that these systems could help to improve road safety and reduce the number of accidents caused by driver error. The paper concludes with a discussion of future directions for research in this area, including the use of deep learning algorithms and the development of real-time systems that can operate under challenging lighting and weather conditions. They suggest that continued improvements in traffic sign detection systems will be critical for the development of safe and reliable IDAS.

Yihui Wu et al. presented a paper “Traffic sign detection based on convolutional neural network” [9]. They trained a CNN using a dataset of over 51,000 traffic sign images, and the system has multiple stages, including image pre-processing, candidate region generation, feature extraction, and classification. They used a combination of region proposal algorithms and CNN-based object detection algorithms to generate candidate regions in the input images. They then used a CNN to extract features from these regions and classify them into one of the 43 traffic sign categories. They evaluated the performance of their system on a set of previously unseen images and achieved an accuracy of over 99.73% on “Danger” images and 97.62% in “Mandatory” images. They also compared the performance of their system with other state-of-the-art traffic sign detection systems and found that their system outperformed these systems in terms of both accuracy and processing time. Their research paper provides a detailed explanation of the system architecture and evaluation metrics, making it an essential resource for researchers and developers working in the field of traffic sign detection. They also highlight the potential applications of their system in autonomous driving and advanced driver assistance systems.

Jianming Zhang et al. proposed a "A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2" [10] paper that proposes a real-time traffic sign detection algorithm based on the modified YOLOv2 (You Only Look Once version 2) architecture. They used a dataset of more than 17,000 Chinese traffic sign images, and they modified the YOLOv2 architecture to improve the detection accuracy and reduce false positives.

The modified YOLOv2 architecture has four main components: a feature extractor, a region proposal network, a region of interest pooling layer, and a fully connected layer. The feature extractor is based on the Darknet-19 architecture, and it generates a feature map from the input image. The region proposal network generates candidate regions in the feature map, and the region of interest pooling layer extracts features from these regions. The fully connected layer classifies these features into one of the 47 traffic sign categories. They evaluated the performance of their system on a dataset of real-world traffic sign images, and they achieved an accuracy of over 98%. They also compared the performance of their system with other state-of-the-art traffic sign detection systems and found that their system outperformed these systems in terms of both accuracy and processing time. Their research paper provides a comprehensive description of the modified YOLOv2 architecture and evaluation metrics, making it a valuable resource for researchers and developers working in the field of traffic sign detection. They also illustrate how their system might be used in autonomous driving and intelligent transportation systems.

Ayoub Ellahyani et al. presented a paper on "Traffic sign detection and recognition based on random forests” [11]. They propose a traffic sign detection and recognition system based on the Random Forests algorithm. They used a dataset of more than 10,000 traffic sign images and trained a Random Forests classifier to detect and recognize traffic signs. The system has two main stages: detection and recognition. In the detection stage, the authors used a sliding window approach to scan the input image for potential traffic sign regions. They extracted features from these regions and used a Random Forests classifier to determine whether each region contains a traffic sign or not. In the recognition stage, the authors used a template matching approach to match the detected traffic signs to their corresponding templates. They evaluated the performance of their system on a dataset of real-world traffic sign images, and they achieved an accuracy of over 95%. They also compared the performance of their system with other state-of-the-art traffic sign detection and recognition systems and found that their system outperformed these systems in terms of both accuracy and processing time. Their research paper provides a detailed description of the system architecture and evaluation metrics, making it a valuable resource for researchers and developers working in the field of traffic sign detection and recognition.

III. Proposed Work

The proposed model is based on Deep Learning. This project aims to ensure that correct sign should be recognized. We prepared this model using Python libraries such as Keras, TensorFlow, and OpenCV. The model used here is CNN (convolutional neural network). It has effective feature extraction capability, that it extracts the relevant features automatically. It is composed of 4 main layers that plays a crucial role in traffic sign detection [12].

1. **Convolutional layers:** These layers apply a series of filters (kernels) to the input image, extracting features that capture spatial relationships between adjacent pixels. In our model, we have used the Conv2D, Conv2D_1, Conv2D_2, Conv2D_3 layers, that takes the input images of traffic sign as a 3D tensor and set the hyperparameters to extract the important features.

2. **Pooling layers:** These layers down sample the output of the convolutional layers, reducing the dimensionality of the data and making it easier to process. We used the max pooling layer, that divides the feature into no-overlapping region and finds the maximum value. It removes the noise like uneven lighting, fog, snow etc.

3. **Activation layers:** These layers apply an activation function to the output of a previous layer, introducing nonlinearity into the model. In our model, we have used the ReLU activation function as it allows the network to introduce non-linearity into the output. Non-linearity is important in traffic sign detection because the relationship between the input image and the output class labels is often non-linear and complex.

4. **Fully connected layers:** These layers take the output of the previous layer and produce a final classification or regression output. The dense layer learns the high-level features from the traffic sign images and predict the correct image for the autonomous driving system. The dropout layer is used to prevent overfitting and it is applied after the dense layer.

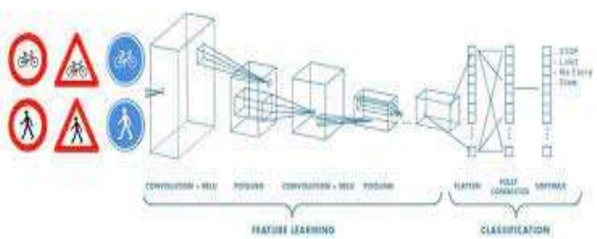
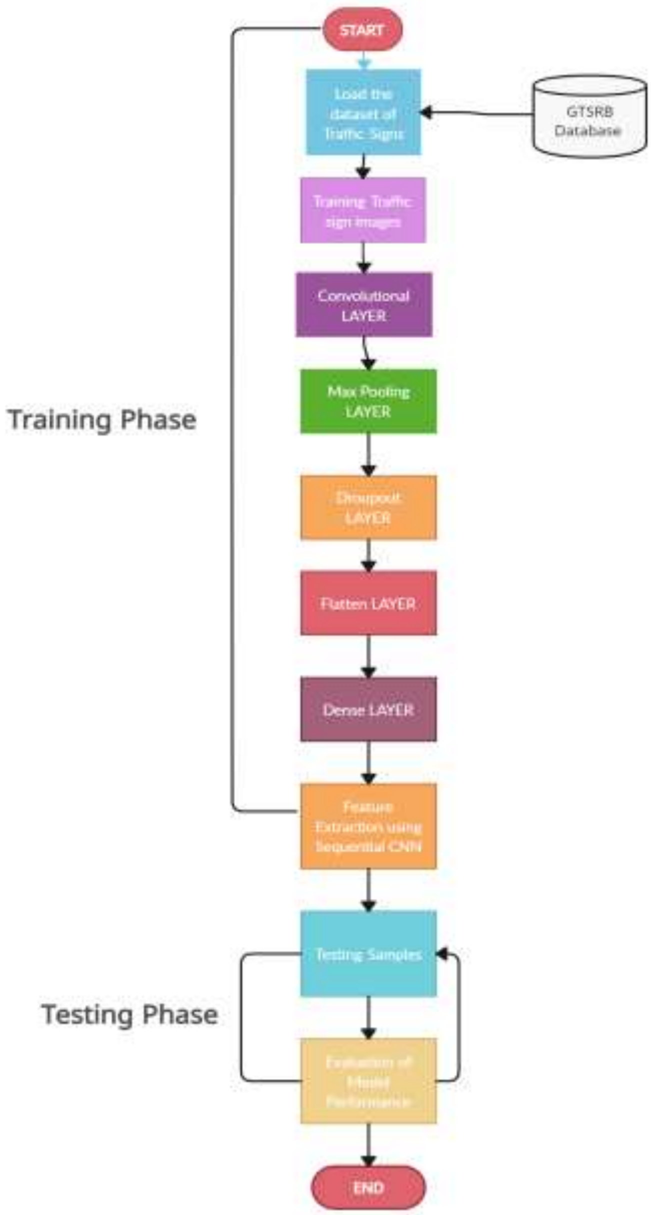


Figure 7: CNN in Traffic Sign Recognition

To detect traffic sign in this study, we use the CNN method. We have selected the GTSRB dataset with 50000+ images and given them as input as shown in figure 8 [13]. The dataset includes numerous types of complex traffic signs similar as sign cock, uneven lighting, and traffic sign with distraction, occlusion and analogous background colours, as well as factual scene charts.

Figure 8: GTSRB Dataset

The output is then the digital text of the sign in the image from the same dataset. Following is the flowchart of our proposed model:



The Traffic sign detection involves acquiring and pre-processing an image, detecting the location of traffic signs using a detection algorithm, classifying the detected signs using an image classification algorithm, and applying post-processing techniques to refine the final output. The specific algorithms and techniques used in traffic sign detection may vary depending on the specific application and requirements.

IV. Results & Discussion

Our proposed model has been trained and tested on the dataset of around 7842 images. The summary of the model has been shown in figure 9.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        2432

 conv2d_1 (Conv2D)           (None, 22, 22, 32)        25632

 max_pooling2d (MaxPooling2D  (None, 11, 11, 32)       0
 )

 dropout (Dropout)           (None, 11, 11, 32)        0

 conv2d_2 (Conv2D)           (None, 9, 9, 64)          18496

 conv2d_3 (Conv2D)           (None, 7, 7, 64)          36928

 max_pooling2d_1 (MaxPooling  (None, 3, 3, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 3, 3, 64)          0

 flatten (Flatten)           (None, 576)               0

 dense (Dense)               (None, 256)               147712

 dropout_2 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 43)                11051

=================================================================
Total params: 242,251
Trainable params: 242,251
Non-trainable params: 0
```

Figure 9: Model Summary

From the evaluation results, it is evident that accuracy is around 95.42%. This shows that the proposed model is effective in determining road sign even if the signs are occluded due to tilt, orientation, and rotation etc.

Accuracy with the test data

```
In [21]: from sklearn.metrics import accuracy_score
         print(accuracy_score(label, Y_pred))

0.9542359461599367
```

Fig. 10 shows the plot for accuracy for the training and validation dataset. At the beginning of training, the training accuracy is typically very low, as the model's weights are randomly initialized and have not yet been optimized to the task at hand. As training progresses, the model learns to better fit the training data, and the training accuracy improves. In our case, training accuracy started from zero and took a sharp upright turn. This is expected, as the model starts learning patterns in the data.

The validation accuracy, on the other hand, starts from some value and moves forward in a straight line. This is because the model is generalizing well to the validation data, which it has not seen during training. The validation accuracy can be thought of as a proxy for the model's performance on new, unseen data. If the validation accuracy is consistently moving in a straight line, it indicates that the model is not overfitting to the training data, and is able to generalize well to new data.

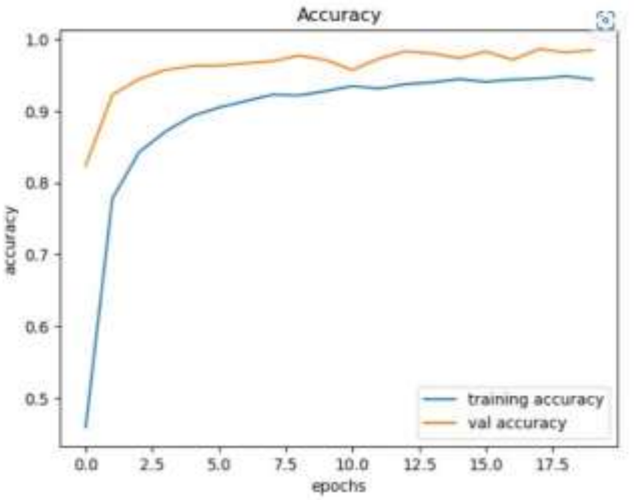


Figure 10: Training and Validation Accuracy

However, the validation accuracy sometimes drops by a little while moving forward with each epoch. This can be an indication of overfitting, where the model becomes too specialized to the training data and does not generalize well to new data. To address this issue, you can try regularization techniques such as dropout, weight decay, or early stopping. These techniques can help prevent overfitting and improve the model's generalization performance.

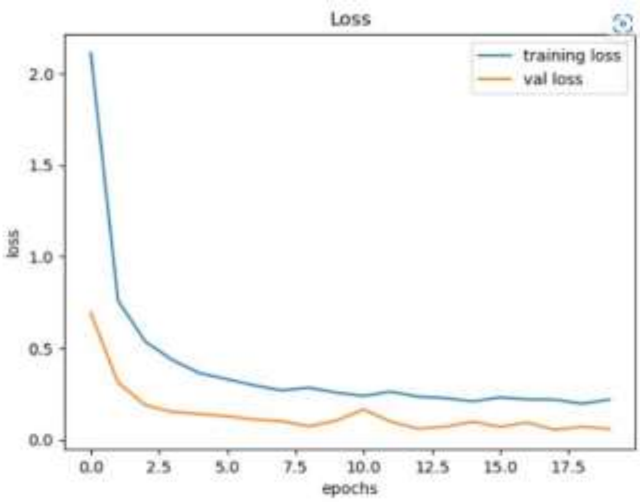


Figure 11: Training and Validation Loss

Fig. 11 shows the plot for training and validation loss. It is clear from Fig. 11 that training loss is decreasing from 30% to around 1% with a change in epochs. Validation loss also decreased from 4% to 1% with a change in epochs**.**

From Fig. 10 and Fig. 11, it is evident that more epochs may increase the calculation time only without making any significant change in accuracy as well as loss. So, epochs can be kept to a minimum to reduce complexity of the model as shown in fig 12 and 13.

Figure 12: Epochs

Figure 13: Epochs

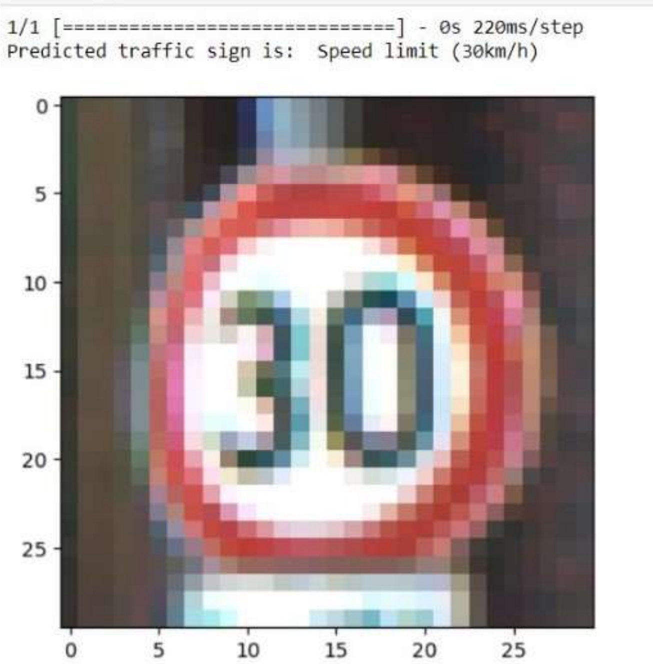


Figure 14: Output

Fig. 14 shows the result of traffic sign detection system from image. From Fig. 14, it is clear that in the case of a blurry image our proposed model is detecting the correct sign with an accuracy of 95.42%.

IV. Conclusion & Future Scope

Traffic sign detection is the process of automatically identifying and localizing traffic signs in images or videos captured by cameras mounted on vehicles or other devices. It is a critical component of advanced driver assistance systems (ADAS) and autonomous vehicles, as it enables the vehicle to interpret and respond to traffic signs and signals.
The goal of traffic sign detection is to accurately detect and localize all relevant traffic signs in an image or video stream, even under challenging conditions such as occlusions, lighting variations, and perspective distortions. To achieve this, various machine learning algorithms, such as convolutional neural networks (CNNs), have been used. This method significantly speeds up road sign detection while also assisting in reaching high precision. The proposed model is giving an accuracy approaching perfection i.e., around 95.2%. The model can be used in various applications, such as speed limit recognition, stop sign detection, and traffic signal recognition. However, there are still some challenges in developing robust and accurate traffic sign recognition systems. Further research is needed to improve the performance of these systems in these challenging scenarios. In the future, traffic sign recognition systems could be integrated with other intelligent driver assistance systems (IDAS), such as lane departure warning, pedestrian detection, and collision avoidance, to provide more comprehensive and advanced driver assistance. These systems could also be integrated with autonomous vehicles to improve their ability to navigate and operate safely on the roads.

References

[1] Road traffic injuries. (2022, June 20). Road Traffic Injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[2] Lorberfeld, A. (2019, April 25). Machine Learning Algorithms In Layman's Terms, Part 1. Medium. https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b

[3] What is the difference between deep learning and usual machine learning? (n.d.). Quora. https://www.quora.com/What-is-the-difference-between-deep-learning-and-usual-machine-learning

[4] CNN representation https://www.semanticscholar.org/paper/Traffic-Sign-Recognition-Using-Extreme-Learning-Zeng-Xu

[5] Stallkamp, Johannes, et al. "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition." *Neural networks* 32 (2012): 323-332.

[6] Broggi, Alberto, et al. "Real time road signs recognition." *2007 IEEE Intelligent Vehicles Symposium*. IEEE, 2007.

[7] N. Barnes, A. Zelinsky and L. S. Fletcher, "Real-Time Speed Sign Detection Using the Radial Symmetry Detector," in IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 2, pp. 322-332, June 2008, doi: 10.1109/TITS.2008.922935.

[8] A. Mogelmose, M. M. Trivedi and T. B. Moeslund, "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484-1497, Dec. 2012, doi: 10.1109/TITS.2012.2209421.

[9] Y. Wu, Y. Liu, J. Li, H. Liu and X. Hu, "Traffic sign detection based on convolutional neural networks," The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 2013, pp. 1-7, doi: 10.1109/IJCNN.2013.6706811.

[10] Zhang, Jianming, et al. "A real-time Chinese traffic sign detection algorithm based on modified YOLOv2." *Algorithms* 10.4 (2017): 127.

[11] Ellahyani, Ayoub, Mohamed El Ansari, and Ilyas El Jaafari. "Traffic sign detection and recognition based on random forests." Applied Soft Computing 46 (2016): 805-815.

[12] CNN in traffic sign recognition- https://images.app.goo.gl/mkRT14TiSxZRytov6

[13] GTSRB Dataset image- https://images.app.goo.gl/4EPMyxHT7gPLNvFp7