IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

DESIGN AND IMPLEMENTATION OF TRAFFIC SIGN DETECTION AND RECOGNITION SYSTEM USING CONVOLUTION NEURAL NETWORK

A Novel Deep Learning Approach

[1]Priyanka Kumbhalkar, [2]Daksh Meshram, [3]Shrunkal Shendekar, [4]Prajakta Kasnawar, [5]Anurag Mishra
[1-5]B.E 4th year Student
[1-5]Department of Computer Technology,
[1-5]Kavikulguru Institute of Technology and Science, Ramtek, Nagpur, Maharashtra, India

Abstract: This study describes the design and implementation of a convolutional neural network (CNN)-based traffic sign detection and identification system. The number of accidents caused by failure to observe traffic signs and follow traffic laws has been steadily growing. The use of synthesized training data generated from road traffic sign pictures enables for the resolution of difficulties with traffic sign detection databases that differ between nations and regions. This technology is used to create a library of synthesized images for detecting traffic signs under varied lighting situations. With this data set and a perfect CNN, a data driven can develop, traffic sign recognition detection system that performs well throughout training and recognition operations and has a high detection accuracy. This reduces the likelihood of accidents and allows the driver to focus on driving rather than studying every traffic sign. The goal of this work is to present a practical approach for detecting and recognizing traffic signs in India. The proposed system comprises two main stages that is traffic sign detection and traffic sign recognition. In the detection stage, the system utilizes a CNN to identify regions of interest (ROIs) that potentially contain traffic signs. The recognition stage involves the classification of the detected ROIs using a CNN based classifier. The proposed system was trained and tested using a publicly available traffic sign dataset, achieving high accuracy rates in both detection and recognition stages. The system's performance was evaluated using various metrics, including accuracy, precision, recall and F1 score, demonstrating its effectiveness and robustness for traffic sign detection and recognition tasks. The results demonstrate the effectiveness and efficiency of the proposed system, which can be used for various real world applications, such as advanced driver assistance systems (ADAS), autonomous driving and traffic management.

Index Terms **– Traffic sign, detection, recognition, classification, feature extraction, convolution neural network, machine learning, deep learning.**

I. INTRODUCTION

Traffic sign detection and recognition is an essential task in the field of computer vision, with significant potential for improving road safety and enabling more efficient and reliable driving. Accurate and reliable detection and recognition of traffic signs are critical for safe and efficient operation of vehicles, particularly in situations where the driver may be distracted or unable to see the signs due to poor weather or lighting conditions. Traditional methods for traffic sign detection and recognition, such as template matching and feature based methods, have limitations in handling variations in the input, such as lighting, occlusions and deformations. Convolution neural networks have emerged as a powerful tool for traffic sign detection and recognition, with the ability to automatically learn features from raw data and handle complex variations in the input. The success of CNN in traffic sign detection and recognition has led to their applications in various domains, including autonomous driving, driver assistance systems and road safety.

Traffic sign detection and recognition systems using convolutional neural networks have become increasingly popular in recent years due to their ability to accurately identify and classify traffic signs in real time. Advanced driver assistance systems and driverless vehicles must be able to detect and recognize traffic signs. With a growing number of automobiles on the road, it is essential to ensure safe and efficient transportation. Convolutional neural networks have been popular in the detection and recognition of traffic signs because they have been shown to be efficient in image recognition tasks. The system uses computer vision techniques to detect and recognize traffic signs in real time, providing valuable information to the driver or autonomous vehicle. Convolutional neural networks are the best option for this application because they have been proven to be very successful for tasks involving the detection and recognition of objects. In this system, the CNN is trained using a large dataset of annotated traffic signs to learn the visual characteristics of different sign types.

The design and implementation of a traffic sign detection and recognition system using CNN involves several steps. The first step is to collect a dataset of traffic sign images. This dataset should be diverse and include various types of traffic signs in different lighting and weather conditions. The next step is to preprocess the dataset by resizing and augmenting the images to improve the model's performance. The next step is to design and train a CNN model using the preprocessed dataset. The model should be designed to detect and recognize traffic signs accurately and efficiently. It should include layers such as convolutional layers, pooling layers, and fully connected layers. The training process involves feeding the model with the preprocessed dataset and adjusting the weights of the model to minimize the loss function. After training the CNN model, the next step is to test it on a separate dataset to evaluate its performance. This dataset should also be diverse and include different types of traffic signs in different conditions. The model's performance can be evaluated by measuring metrics such as precision, recall and F1 score.

Once the model's performance is satisfactory, it can be integrated into a traffic sign detection and recognition system. This system should be able to detect and recognize traffic signs in real time and provide the driver with the necessary information. The system should be designed to be robust and efficient to handle various environmental conditions and different types of traffic signs. Overall, the design and implementation of a traffic sign detection and recognition system using CNN requires a combination of expertise in computer vision, machine learning and system integration. With proper design and implementation, such a system can significantly improve road safety by helping drivers to stay informed and alert about traffic signs on the road.

II. Problem Statement

The problem statement for the design and implementation of a traffic sign detection and recognition system using an convolutional neural network is to develop a computer vision system that can accurately detect and classify different types of traffic signs from real world images or a video feeds. The system should be able to identify and recognize various traffic signs, such as stop signs, yield signs, speed limit signs, pedestrian crossing signs and other regulatory signs. The system should also be able to detect multiple traffic signs in a single image or a video frame and accurately classify them.

The main goal is to create a reliable and robust system that can be deployed in various scenarios, such as in autonomous vehicles, traffic management systems, and other applications that require accurate and real time traffic sign detection and recognition. The system should be able to operate in different lighting conditions and weather conditions, and be able to handle various challenges, such as occlusions, partial obstructions, and variations in size, orientation, and distance of the traffic signs.

III. Aim and Objectives

The aim of the traffic sign detection and recognition system using convolutional neural network is to develop a robust and accurate computer vision system that can automatically detect and classify traffic signs in real-world images or video feeds.

The main objectives of the system are:

- Design and develop a CNN model that can accurately detect and classify traffic signs.
- Collect and preprocess a large dataset of traffic sign images and labels for training and testing the CNN model.
- Evaluate the performance of the CNN model using various metrics, such as accuracy, precision, recall, and F1 score.
- Implement the CNN model in a real-world application, such as an autonomous vehicle or traffic management system, and test its performance in different scenarios and conditions.
- Improve the CNN model's performance by fine-tuning the hyperparameters, optimizing the architecture, or incorporating additional techniques, such as data augmentation, transfer learning, or assembling.
- Compare the performance of the CNN model with other state-of-the-art methods for traffic sign detection and recognition, and identify its strengths and weaknesses.
- Provide a user-friendly interface for interacting with the system and visualizing the results, such as the detected traffic signs, their locations, and their classifications.

IV. Literature Review

Traffic sign detection and recognition using convolutional neural networks by Ankit Agrawal and Shivam Gupta (2019) proposed a traffic sign detection and recognition system based on convolutional neural networks. The proposed system was tested on the German traffic sign recognition benchmark dataset and achieved high accuracy. The authors also discussed the importance of data augmentation and the trade-off between accuracy and computational efficiency.

Traffic sign detection and recognition using deep learning techniques by S. V. Aruna and P. Kannan (2018) presented a traffic sign detection and recognition system based on deep learning techniques. The proposed system was tested on the German traffic sign recognition benchmark dataset and achieved high accuracy. The authors also discussed the importance of feature extraction and the impact of different network architectures on the performance of the system.

Traffic sign detection using deep convolutional neural networks by G. Ramesh Babu and K. R. Santhi (2017) proposed a traffic sign detection system based on deep convolutional neural networks. The proposed system was tested on the Belgium traffic sign recognition benchmark dataset and achieved high accuracy. The authors also discussed the importance of pre-processing techniques such as image normalization and thresholds.

Real time traffic sign recognition using a deep convolutional neural network by Sanghyun Woo and Jun Young Lee (2016) presented a real time traffic sign recognition system based on a deep convolutional neural network. The proposed system was tested on the German traffic sign recognition benchmark dataset and achieved high accuracy and real time performance. The authors also discussed the impact of different network architectures and hyperparameters on the performance of the system.

Traffic sign recognition with multiscale convolutional neural networks by Pierre Sermanet and Yann LeCun (2011) proposed a traffic sign recognition system based on multiscale convolutional neural networks. The proposed system achieved high accuracy on the German traffic sign recognition benchmark dataset. The authors also discussed the importance of data augmentation and the trade-off between accuracy and computational efficiency.

V. Proposed Approach and System Architecture

The proposed approach of traffic sign detection and recognition system consists of two main stages. The first stage is the detection stage which takes the input image or video to identify and rectify the traffic sign from the captured image. The collected picture is used as an input in the detecting phase, when the noise or undesired background is eliminated. In the second stage that is recognition phase the image generated from the last phase is compared and identified from the existing database to uniquely recognize the traffic sign. This phase extracts sign information by utilizing numerous properties of the traffic sign such as color, shape, size, texture, and so on. Once the sign is identified the system generates a voice alert to the vehicle driver by guiding the driver about the sign and provides the information about what action to be performed.

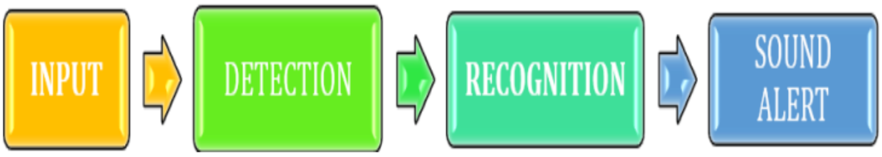

Figure 1 Simple block diagram

Figure 2 depicts the suggested method's system architecture of the proposed method. In this initially a traffic sign is captured which will further undergo through the pre-processing stage, detection phase, classification phase and finally an audio message is generated which denotes the classified traffic sign. The different modules of traffic sign detection system are image acquisition, image pre-processing, sign detector, text detector, text translator, sign-text comparator, sign classifier, sign to text translator, audio generator. The brief descriptions about these modules are given below.

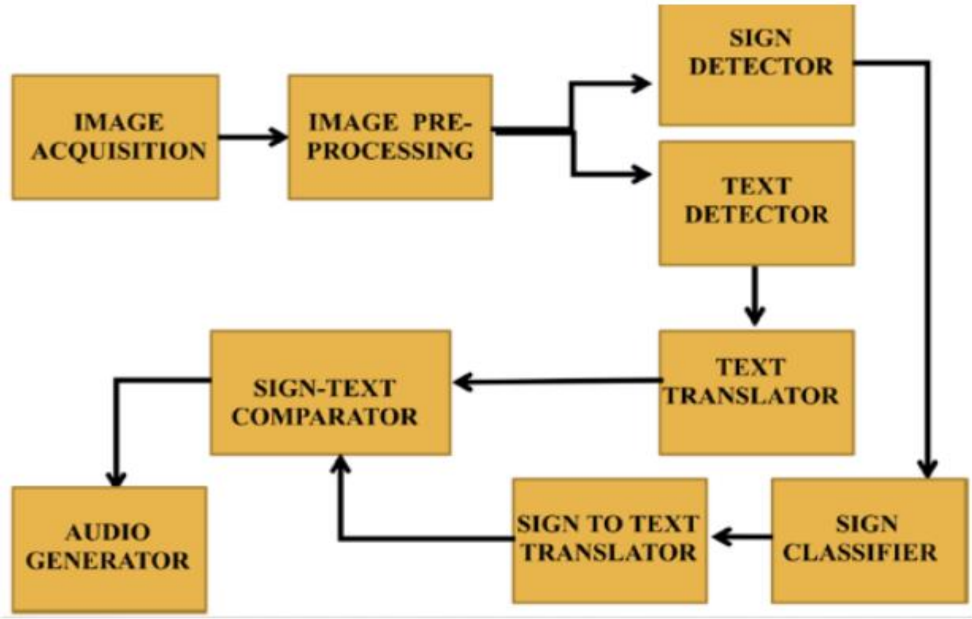

Figure 2 System architecture

1. **Image acquisition:** Image acquisition can be defined as the process of retrieving or obtaining an image using some source usually it may be hardware based source. In this stage with the help of camera the traffic scenes are obtained.
2. **Image preprocessing**: Digital image processing is the process of using algorithms to perform image processing on digital images. Digital image processing, as a subsection of digital signal processing, has numerous benefits over analogue image processing. It enables the application of a considerably broader range of algorithms to the supplied data. The goal of digital image processing is to improve picture data by removing undesired distortions or enhancing some crucial image properties so that our AI computer vision can perform better models can benefit from this improved data. In this proposed system CLAH E algorithm is used for the preprocessing of image. It will help to improve the contrast in images.
3. **Sign detector:** The preprocessed traffic scene image is given to the sign detector algorithm which will detect whether any traffic signs are present in the captured image or not. If it detects any traffic sign, then the sign will be fed into the classifier for further classification process. It is implemented using transfer learning method in googles object detection API toolkit.
4. **Sign classifier:** The sign classifier will classify the traffic signs into appropriate classes. There are 43 different classes in the dataset that is GTSRB for classification. In this proposed method a convolution neural network is used for the purpose of classification of traffic signs.
5. **Text detector:** Text detector uses the concept of OCR. OCR is a reader that recognizes computer text characters, which can be printed or written. Along with the sign detector, text detector is used for obtaining the text written on the sign board, if any.
6. **Text translator:** It converts the text to natural language specified by the user. This also uses the concept of OCR. OCR is a type of reader that can identify computer text characters, whether they are typed or printed.
7. **Sign to text translator:** It is used to translate the classified sign to corresponding textual data. In this system, planning is to take the dataset that has labels as textual representation of the classified images. Input is obtained from the sign classifier and the output is fed to the sign text comparator.
8. **Sign text comparator:** The sign text comparator module of the proposed system architecture is intended for the purpose of comparing the text obtained from the text translator module and the text obtained from the sign translator module. This contributes to improved accuracy in the proposed system for traffic sign detection, identification, and conversion into natural language audio output. Hence, it can be used as an end to end application.

9. **Audio generator:** It uses the concept of natural language processing. NLP is a subset of linguistics, computer science and information science, information engineering and artificial intelligence which include how to program computers and analyze large amounts of data.

VI. METHODOLOGY

The design and implementation of a traffic sign detection and recognition system using convolutional neural networks typically involves the following steps.

Step 1: Data collection

There are several datasets available for training and testing convolution neural networks for traffic sign detection and recognition. These datasets are commonly used for training and testing CNN models for traffic sign detection and recognition. They can also be augmented and preprocessed to increase the datasets size and diversity and improve the CNN models performance. German traffic sign recognition benchmark is the dataset considered for this experimentation process. This dataset was introduced in international joint conference on neural networks. It is a benchmark dataset for traffic sign recognition, which is multi class and used for single image classification. The dataset was entered into the IJ CNN competition, and good classification accuracy was attained using both CNN models and hand classification. The dataset is fairly diverse. Some classes contain many photographs, while others have few images. The dataset is separated into two parts: train and test. Figure 3 depicts an example traffic sign dataset.

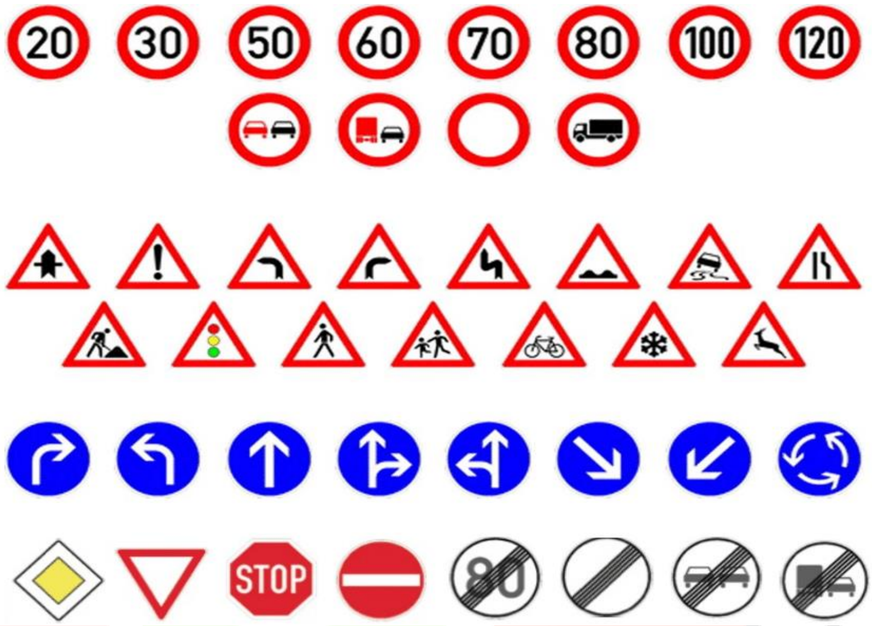

Figure 3 Sample traffic sign dataset

Step 2: Data pre-processing

Pre-processing the dataset images is an essential step for traffic sign detection and recognition using convolution neural networks. The pre-processing step aims to standardize the images, reduce noise and artifacts, and enhance relevant features that are important for traffic sign detection and recognition. Some common pre-processing techniques used in traffic sign detection and recognition include the following.

1. **Resizing:** Resizing the images to a standard size can help reduce the computational cost and memory requirements of the CNN model. The images should be resized to the same aspect ratio to avoid distortion.
2. **Normalization:** Normalizing the pixel values of the images can help standardize the images and make them more compatible with the CNN model. Normalization can be done by subtracting the mean and dividing by the standard deviation of the pixel values.
3. **Color conversion:** Converting the images to grayscale or other color spaces can help reduce the computational cost and make the images more robust to variations in lighting conditions.
4. **Contrast enhancement:** Enhancing the contrast of the images can help improve the visibility of the traffic signs and make them easier to detect and recognize.
5. **Edge detection:** Detecting the edges of the traffic signs can help extract relevant features and reduce noise and artifacts in the images. This can be done using edge detection algorithms, such as Sobel, Canny, or Laplacian.
6. **Histogram equalization:** Equalizing the histogram of the images can help improve the contrast and visibility of the traffic signs, especially in low light conditions.
7. **Augmentation:** As mentioned, data augmentation techniques can also be applied during preprocessing to increase the datasets size and diversity.

The choice of pre-processing techniques depends on the characteristics of the dataset and the requirements of the CNN model. Pre-processing should be done carefully to avoid introducing artifacts or distorting the images. It is also important to validate the pre-processing techniques effectiveness on the dataset to ensure that they improve the model's performance.

After splitting the dataset for training and testing, the preprocessing techniques are applied on training and testing datasets, respectively, by iterating through the images. Initially, the image is converted into an array of pixels. The array of pixels is sent as input to gray scale method. Gray scale method is used to convert colorful image to monochrome gray scale image. Now the converted gray scaled array is sent as input to the histogram equalization method. This method is used to improve contrast in images by spreading out the most frequent intensity values. Initially, the pixel values are higher in some areas and lower in some areas. What histogram equalization does is to stretch out the pixel range by uniformly distributing the pixel values respectively. The figure 4 shows sample pre-processed images.

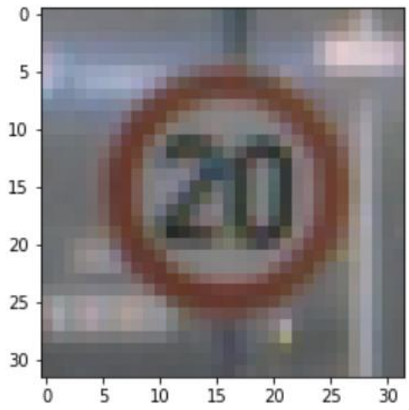

Figure 4 Sample pre-processed image

Step 3: Image augmentation

Image augmentation is a popular technique used to increase the diversity and size of the dataset for traffic sign detection and recognition using convolution neural networks. This technique involves applying various transformations to the original images to create new ones that are similar but not identical to the original. Some common image augmentation techniques used in traffic sign detection and recognition includes the following.

1. **Rotation:** Rotating the image by a certain angle can help increase the datasets size and improve the model's ability to detect and recognize traffic signs at different orientation.
2. **Translation:** Translating the image in the x and y directions can help simulate the effect of the traffic sign being seen from different angles or distances.
3. **Scaling:** Scaling the image up or down can help simulate the effect of the traffic sign being seen at different distances, and it can also help increase the datasets size.
4. **Shearing:** Shearing the image can help simulate the effect of the camera being tilted or rotated, which can occur in real world scenarios.
5. **Flipping:** Flipping the image horizontally can help increase the datasets size and improve the model's ability to recognize traffic signs that are mirrored.
6. **Color jitter:** Modifying the color of the image by adjusting the brightness, contrast and saturation can help simulate different lighting conditions and increase the datasets size.
7. **Gaussian noise:** Adding Gaussian noise to the image can help simulate the effect of the camera being affected by noise or distortion.

Image augmentation is an effective technique to improve the performance of CNN models for traffic sign detection and recognition, especially when the dataset is limited in size or diversity. It can also help improve the model's ability to generalize to new data and be more robust to variations in lighting, angle and occlusions.

Image augmentation is a useful method for building the CNN model by increasing the dataset images. It can create more images from a single image so that the CNN model can have a lot of images for training the model. The idea is simple, it creates multiple duplicate images from a single image with some variations to that. Keras provides a method called image data generator for performing image augmentation. It has some parameters like zoom, rotation etc., and based on these parameters the multiple images are created. The figure 5 shows sample augmented images.

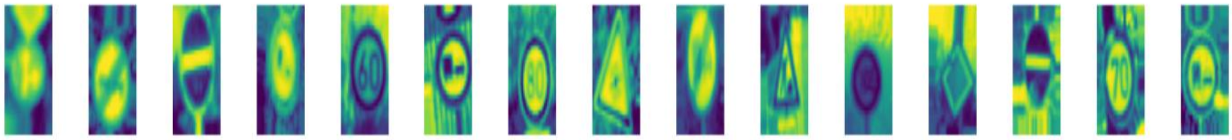

Figure 5 Sample augmented images

Step 4: Building CNN model

In this project, LeNet model has taken and made some changes to it. The LeNet-5 CNN architecture has seven layers. Three convolutional layers, two subsampling layers, and two fully connected layers make up the layer composition. This modified the LeNet model by adding two dropout layers and one additional convolutional layer. So, in this project, it used 4 convolution layers, 2 max-pooling layers, 2 dropout layers, 2 fully connected layers and 1 flatten method. 60 kernels of size 5*5 is used in the first and second convolution layers with ReLU activation. There are 30 kernels of size 3*3 utilized for the third and fourth layers. For both max-pooling layers, pool_size as 2*2 filter is used. 2 drop out layers are used with 0.5 drop that is 50% neurons will shut down at each training step by zeroing out the neuron values. 2 dense layers for which one dense layer performs ReLU activation and other performs soft max.

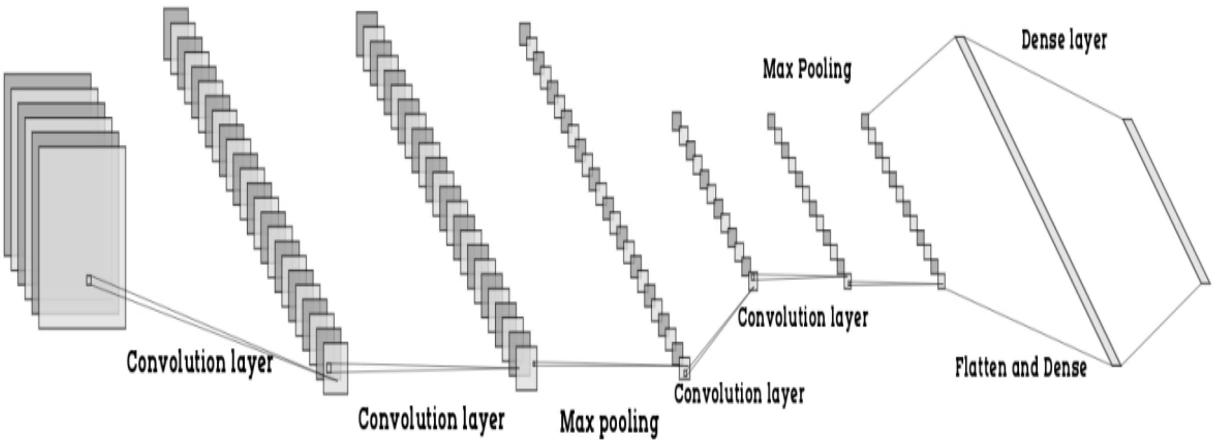

Figure 6 CNN model architecture

1. The preprocessed 3D image array is element wise multiplied in the conv2d layer using a kernel or filter of size 5*5, which adds up the array and updates the sum 43 value in the appropriate spot. Every pixel value in the picture array is updated in the same way. This layer has 60 filters. 28*28 is the new size after this layer. The activation function max (0, x) is then applied to all the pixel values in the output image by the ReLu layer.
2. The conv2d_1 layer is subjected to the same operation using the ReLu activation function. The area will be 24*24 in size.
3. The maximum element from the region of the feature map that the filter covers is then extracted using max pooling, which is used later. Retrieving the salient features of the preceding feature map is the max pool's primary goal. In this case, a 2*2 filter size is provided.
4. Following maximum pooling, two conv_2d layers with 30 filters of size 3*3 and ReLu activation function are applied one after the other.
5. Max-pooling with a 2*2 filter size was once more used. Later, the connections are cut in half using a drop out of 0.5. Drop out is followed by the flatten procedure, which reduces the image array to a single dimension.
6. In order to obtain the final output array, 2 completely connected layers are employed, one with a ReLu activation function and the other with a soft max activation function, with a dropout in between of 0.5. After construction, the model was assembled using the Adam optimizer, accuracy metric, and loss.

Step 5: Training and testing the model

Figure 7 summarizes the steps which are conducted in the project to recognize traffic signs. The steps are mainly classified into two phases.

Phase 1: The constructed CNN model is trained and tested using default settings.

Phase 2: Real-time testing of the model using OpenCV.

Training of the model is done using Adam optimizer with batch size 32 and number of epoch is 20. It followed a simple approach and ran only 20 epochs of the training and observed the validation error trying to set it on a minimum level and also due to limitation of computational power While improving the model, it is critical to keep validation errors in a mind. Only reducing the error with respect to training data might easily lead to model over fitting. After training is completed, testing is performed with the testing dataset.

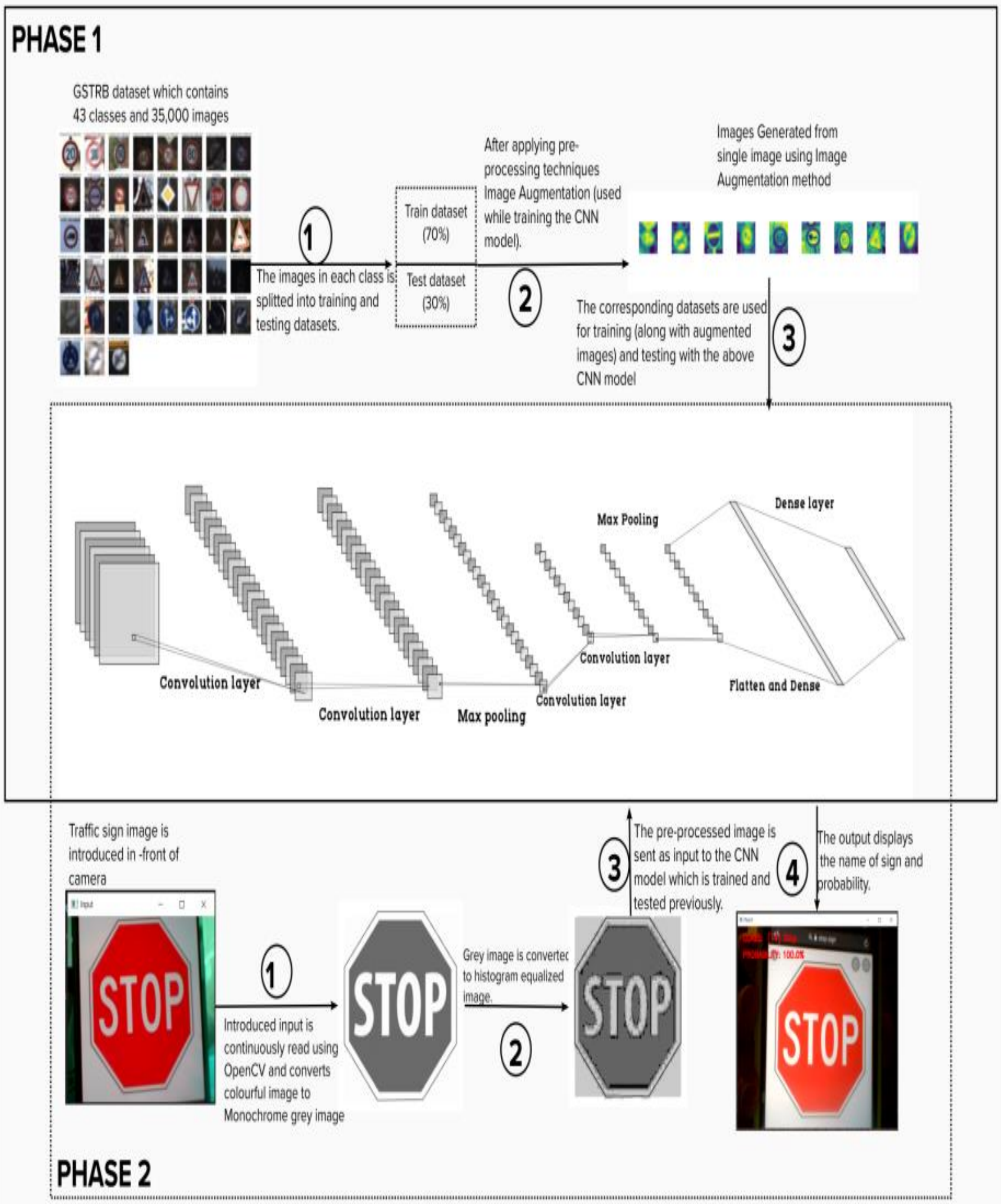

Figure 7 Steps involved in training and testing the model

Step 6: Evaluation and deployment

The CNN should be evaluated on the validation set to determine its performance. The evaluation should include metrics such as accuracy, precision, recall, and F1 score. Use techniques like confusion matrices and ROC curves to analyze model performance. Once the CNN has been trained and evaluated, it can be deployed to perform traffic sign detection and recognition in real time. The system can take input from a camera mounted on a vehicle and use the trained CNN model to detect and recognize traffic signs in real time.

VII. Results and Discussion

In this project, the model is tested on a real time video using a webcam or any external live feed. Traffic sign images introduced to the camera are new images and from training or testing data and dataset. For testing, new photos from each class are utilized. It processes every frame of the video to check for a traffic sign. If the probability value of the prediction is higher than the threshold value, then the class label will be displayed along with the probability The below figures represent various traffic sign which are classified under different classes. The probability we obtained is different for different signs.

Figure 8 Speed limit 20 km/hr

The figure 8 represents speed limit 20 km/hr traffic sign which is classified under class 0 and the probability obtained is 99.63% at distance of 10 meters from the webcam. The likelihood may drop when the distance is increased, which would therefore make it less effective to forecast the traffic sign.

Figure 9 Bumpy road

The figure 9 represents bumpy road traffic sign which is classified under class 22 and the probability obtained is 98.34% at distance of 10 meters from the webcam. The chance may decline when the distance is extended, which would therefore limit the effectiveness of forecasting the traffic sign.

Figure 10 Go straight or right

The figure 10 represents go straight or right traffic sign which is classified under class 36 and the probability obtained is 93.11% at distance of 10 meters from the webcam. Further increasing the distance may result in a decline in probability, which would subsequently lower the accuracy of the traffic sign prediction.

Figure 11 No passing

The figure 11 represents no passing traffic sign which is classified under class 9 and the probability obtained is 99.93% at distance of 10 meters from the webcam. The probability may drop when the distance is extended, which would therefore make it less effective to forecast the traffic sign.

Figure 12 Slippery road

The figure 12 represents slippery road traffic sign which is classified under class 23 and the probability obtained is 99.0% at distance of 10 meters from the webcam. The likelihood may drop when the distance is increased, which would therefore make it less effective to forecast the traffic sign.

VIII. Conclusion

Traffic sign detection and recognition using convolutional neural networks have proven to be effective and efficient in real world applications. CNN based systems have demonstrated high accuracy and robustness in detecting and recognizing traffic signs, making them a valuable tool for improving road safety and developing autonomous vehicles. The end to end learning approach used by CNN eliminates the need for manual feature engineering and allows the system to learn directly from raw image data. This approach is especially useful in handling variations in lighting, weather conditions, and other environmental factors that can affect traffic sign appearance. Furthermore, the real time processing capability of CNN enables them to handle high speed traffic situations and make quick decisions, which is essential for applications such as self-driving cars and advanced driver assistance systems.

Convolutional neural networks can significantly improve road safety and reduce the number of accidents caused by driver error. The use of deep learning algorithms and computer vision techniques has allowed for the creation of highly accurate and efficient systems that can detect and classify traffic signs in real time. The implementation of the system involves several steps, including data collection and preparation, network architecture design, training and testing. The CNN is designed to extract features from the input images and classify them into different traffic sign categories. The use of various pre-processing techniques, such as image resizing and normalization, can also improve the accuracy of the system. In terms of the results, the system can achieve high accuracy rates in both detection and recognition tasks. The implementation of the system has also been shown to be efficient, with real time performance achieved on some hardware configurations. Overall, the design and implementation of a traffic sign detection and recognition system using CNNs has great potential for improving road safety and reducing the number of accidents caused by driver error. Further research and development in this field could lead to even more accurate and efficient systems that can be deployed on a large scale.

IX. Limitations of Study

Although traffic sign detection and recognition using convolutional neural networks has shown promising results, there are still some limitations to this approach. Firstly, the performance of CNN models heavily depends on the quality and quantity of training data, which can be a challenging and time-consuming task to collect and label. Moreover, the generalization ability of the trained models to handle various traffic sign designs and shapes may be limited, especially in cases where there are insufficient training samples.

Secondly, the computational complexity of CNN models can be high, which may hinder their real time applicability in some low power embedded systems. Furthermore, CNN models may be susceptible to adversarial attacks, where the input images are intentionally modified to mislead the models output. Thirdly, the accuracy of the detection and recognition performance of CNN models may degrade under challenging conditions, such as low lighting, occlusions, and different camera angles, which can affect the reliability and safety of the system.

Lastly, the CNN based approach may not be suitable for some specific scenarios, such as those with numerous sign classes or where multiple signs are present in the same image, which may require more sophisticated techniques or hybrid approaches. Overall, despite its significant improvements, the limitations of the CNN based approach for traffic sign detection and recognition should be taken into account when designing and implementing real world systems.

X. Future Scopes

The use of convolutional neural networks for traffic sign detection and recognition is a rapidly evolving field, with many promising directions for future research. One potential avenue is to develop more efficient and accurate CNN models that can handle more complex and diverse traffic sign designs and shapes. This could involve the use of more advanced techniques, such as multiscale feature extraction, attention mechanisms and spatial transformer networks.

Another area for future research is to explore the use of unsupervised and semi-supervised learning methods to overcome the challenges of limited training data. This could involve techniques such as domain adaptation, transfer learning, and generative models, which can learn from unlabeled or partially labeled data to improve the robustness and generalization of CNN models. Moreover, the development of more robust and reliable traffic sign detection and recognition systems can be achieved through the integration of multiple sensing modalities, such as Lidar, radar and GPS, which can enhance the accuracy and reliability of the system in challenging conditions.

Lastly, the deployment of CNN based traffic sign detection and recognition systems in real world scenarios requires addressing the challenges of real time processing and low power consumption, which can be achieved through the development of more efficient hardware architectures, such as field programmable gate arrays and application specific integrated circuits. Overall, there is significant potential for future research in the field of traffic sign detection and recognition using CNN, and it is essential to continue exploring new methods and techniques to improve the accuracy, robustness and real world applicability of these systems.

References

[1] Sermanet, P., Kavukcuoglu, K., Chintala, S., & LeCun, Y. (2011). “Traffic sign recognition with multiscale convolutional networks”, In proceedings of the international joint conference on neural networks, IEEE, 2809-2813.

[2] J. Stallkamp, M. Schlipsing, J. Salmen and C. Igel. (2011). “The German traffic sign recognition benchmark: A multi-class classification competition”, IEEE IJCNN, 1453– 1460.

[3] P. Shopa, N. Sumitha and P. S. K. Patra. (2014). “Traffic sign detection and recognition using OpenCV”, International conference on information communication and embedded systems, 1–6.

[4] Dietterich, F., & Wieland, M. (2016). “Traffic sign detection and recognition using convolutional neural networks”, In proceedings of the international joint conference on neural networks, IEEE, 1078-1085.

[5] Zhu, L., Zheng, Y., & Wang, B. (2016). “A new method for traffic sign detection based on convolutional neural network”, Neurocomputing, 657-664.

[6] P. Dhar, M. Z. Abedin, T. Biswas and A. Datta. (2017). “Traffic sign detection - a new approach and recognition using convolution neural network”, IEEE Region 10 humanitarian technology conference, 416–419.

[7] Li, L., Yang, F., & Zhao, G. (2017). “Learning multiscale feature representation with deep convolutional neural network for traffic sign detection”, Neural computing and applications, 815-825.

[8] Saha, S., & Chanda, B. (2018). “Detection and recognition of traffic sign using convolutional neural network”, In proceedings of the international conference on innovations in electronics, signal processing and communication, 327-332.

[9] Pisarevsky, V., & Pshenichny, S. (2018). “Real-time traffic sign recognition using a convolutional neural network”, International journal of applied mathematics, electronics and computers, 8-12.

[10] Nghiem, A. T., Tran, N. T., & Le, A. P. (2020). “Traffic sign detection and recognition using convolutional neural networks with fusion of color and shape features”, Journal of information science and engineering, 1-18.