Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp105 -119*

Deep Learning-based Traffic Sign Recognition for Autonomous Driverless Vehicles

K. Smita[1], B. Pranika[2], Ch. Meghana[2], B. Sneha[2]

[1]Assistant Professor,[2]UG Students, Department of Information Technology

[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India.

To Cite this Article

K. Smita, B. Pranika, Ch. Meghana, B. Sneha**, "Deep Learning-based Traffic Sign Recognition for Autonomous Driverless Vehicles"** *Journal of Science and Technology, Vol. 08, Issue 12 - Dec 2023, pp105-119*

Article Info

Received: 20-11-2023 Revised: 30 -11-2023 Accepted: 10-12-2023 Published: 20-12-2023

Abstract

Traffic sign detection and recognition play a crucial part in driver assistance systems and autonomous vehicle technology. One of the major prerequisites of safe and widespread implementation of this technology is a TSDR algorithm that is not only accurate but also robust and reliable in a variety of real-world scenarios. However, in addition to the large variation among the traffic signs to detect, the traffic images that are captured in the wild are not ideal and often obscured by different adverse weather conditions and motion artifacts that substantially increase the difficulty level of this task. Robust traffic sign detection and recognition (TSDR) is of paramount importance for the successful realization of autonomous vehicle technology. The importance of this task has led to a vast amount of research efforts and many promising methods have been proposed in the existing literature. However, the machine learning methods have been evaluated on clean and challenge-free datasets and overlooked the performance deterioration associated with different challenging conditions (CCs) that obscure the traffic images captured in the wild. In this paper, we look at the TSDR problem under CCs and focus on the performance degradation associated with them. To overcome this, we propose a Convolutional Neural Network (CNN) based TSDR framework with prior enhancement.

Keywords: traffic sign, driverless vehicles, TSDR algorithm, Convolutional neural network.

1. INTRODUCTION

Traffic sign recognition (TSR) is a critical component of the technology stack for autonomous driverless vehicles, contributing significantly to their ability to navigate safely and make informed driving decisions. TSR systems are designed to detect and interpret traffic signs and signals, providing essential information to the vehicle's control system. These systems typically employ a combination of computer vision, machine learning, and deep neural networks to process data from onboard cameras and sensors.

The process begins with cameras capturing real-time images of the vehicle's surroundings, including road signs, traffic lights, and other relevant traffic-related information. These images are then processed through sophisticated image recognition algorithms that can identify and classify various types of traffic signs, such as speed limits, stop signs, yield signs, and more. This recognition process often involves complex image preprocessing techniques to enhance the visibility of signs under various lighting and weather conditions.

Once a traffic sign is detected and classified, the TSR system interprets its meaning and relays this information to the vehicle's control system. For instance, if a speed limit sign is recognized, the vehicle can adjust its speed accordingly, ensuring compliance with traffic laws and safety regulations. TSR also plays a crucial role in identifying stop signs and traffic signals, enabling the vehicle to come to a complete stop when required and proceed safely through intersections.

To achieve high accuracy and robustness, TSR systems continually improve through machine learning techniques. They are trained on vast datasets containing various traffic sign images captured under different conditions and scenarios, allowing the algorithms to learn and adapt to diverse real-world environments. Moreover, TSR systems can benefit from advancements in hardware, including high-resolution cameras, powerful GPUs, and specialized AI chips, which enhance their speed and accuracy.

So, traffic sign recognition is a pivotal technology for autonomous driverless vehicles, as it enhances their ability to comprehend and respond to traffic regulations, ensuring the safety of passengers, pedestrians, and other road users. As autonomous vehicles become more prevalent on our roads, TSR systems will continue to evolve and play a crucial role in making self-driving transportation a reality.

The research motivation for Traffic Sign Recognition (TSR) in autonomous driverless vehicles is rooted in the pursuit of safer, more efficient, and sustainable transportation solutions. Several compelling factors drive the need for extensive research and development in this area.

Firstly, road traffic accidents remain a global concern, leading to significant loss of life and property. Many of these accidents are caused by human error, including misinterpretation of traffic signs or failure to comply with traffic regulations. The integration of TSR technology in autonomous vehicles offers a promising solution to mitigate such errors by providing an extra layer of vigilance and ensuring strict adherence to traffic rules. Reducing accidents and improving road safety is a primary motivation for this research.

Secondly, the advent of autonomous vehicles has the potential to revolutionize urban mobility and reduce traffic congestion. However, for these vehicles to operate safely and efficiently in shared road spaces, they must be equipped with reliable TSR systems. Such systems can help autonomous vehicles make informed decisions in real-time, such as adjusting speed to match speed limits, navigating complex intersections, and responding to temporary road signs. Consequently, TSR research contributes to the broader goal of improving traffic flow and reducing congestion in urban areas.

Furthermore, the motivation for TSR research is closely tied to the increasing demand for sustainable transportation options. Autonomous vehicles hold promise in reducing fuel consumption and emissions through more efficient driving behaviors. TSR plays a pivotal role in achieving these goals by optimizing speed and providing timely information about road conditions, which can lead to more eco-friendly driving practices.

Additionally, the global deployment of autonomous vehicles raises regulatory and standardization concerns. Different regions and countries have unique traffic sign designs, colors, and meanings. As such, research in TSR aims to create adaptable systems capable of recognizing and interpreting a wide range of regional traffic signs, contributing to the harmonization of autonomous vehicle technology on a global scale.

Lastly, the continuous evolution of technology, machine learning, and computer vision presents opportunities to enhance TSR systems further. The motivation lies in leveraging these advancements to improve accuracy, robustness, and real-time processing capabilities, ensuring that TSR keeps pace with the dynamic nature of traffic environments.

So, the research motivation for TSR in autonomous driverless vehicles is multifaceted. It encompasses the overarching goals of enhancing road safety, reducing traffic congestion, promoting sustainability, addressing regulatory challenges, and harnessing technological advancements to create more capable and reliable TSR systems. As autonomous vehicles become an integral part of future transportation ecosystems, the importance of TSR research cannot be overstated in realizing the full potential of this transformative technology.

2. LITERATURE SURVEY

Mannan et. al [4] presented a novel flexible Gaussian mixture model-based technique with automatic split and merge strategy. This adaptive scheme works as a preprocessing step which facilitates locating traffic signs under a certain degree of degradation in a real-world scenario. A multiscale convolutional neural network augmented with dimensionality reduction layer is proposed to recognize contents of the sign. Since, there is no available benchmark dataset for this purpose, we collected a number of images containing partially degraded signs from the famous German Traffic Sign Detection Benchmark (GTSDB) and augmented it with manually and naturally degraded traffic sign images taken from the longest highway in the country of authors' residence. Experimental results show that our proposed technique outperforms many states of the art and recent methods for detection and recognition of degraded traffic signs.

Jin et. al [5] proposed an improved (Single Shot Detector) SSD algorithm via multi-feature fusion and enhancement, named MF-SSD, for traffic sign recognition. First, low-level features are fused into high-level features to improve the detection performance of small targets in the SSD. We then enhance the features in different channels to detect the target by enhancing effective channel features and suppressing invalid channel features. Our algorithm gets good results in domestic real-time traffic signs. The proposed MF-SSD algorithm is evaluated with the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The experimental results show that the MF-SSD algorithm has advantages in detecting small traffic signs. Compared with existing methods, it achieves higher detection accuracy, better efficiency, and better robustness in complex traffic environment.

Gámez Serna et. al [6] introduced a real-world European dataset for traffic sign classification. The dataset is composed of traffic sings from six European countries: Belgium, Croatia, France, Germany, The Netherlands, and Sweden. It gathers publically available datasets and complements French traffic signs with images acquired in Belfort with the equipped university autonomous vehicle. It is composed of more than 80000 images divided in 164 classes that at the same time belong to four main categories following the Vienna Convention of Road Signs. We analyzed the intra variability of classes and compared the classification performance of five convolutional neural network architectures.

Varytimidis et. al [7] investigates a number of feature extraction methods in combination with several machine learning algorithms to build knowledge on how to automatically detect the action and intention of pedestrians in urban traffic. We focus on the motion and head orientation to predict whether the pedestrian is about to cross the street or not. The work is based on the Joint Attention for Autonomous Driving (JAAD) dataset, which contains 346 videoclips of various traffic scenarios captured with cameras mounted in the windshield of a car. An accuracy of 72% for head orientation estimation and 85% for motion detection is obtained in our experiments.

Bangquan et. al [8] introduced a new efficient TSC network called ENet (efficient network) and a TSD network called EmdNet (efficient network using multiscale operation and depthwise separable convolution). We used data mining and multiscale operation to improve the accuracy and generalization ability and used depthwise separable convolution (DSC) to improve the speed. The resulting ENet possesses 0.9 M parameters (1/15 the parameters of the start-of-the-art method) while still achieving an accuracy of 98.6 % on the German Traffic Sign Recognition benchmark (GTSRB). In addition, we design EmdNet' s backbone network according to the principles of ENet. The EmdNet with the SDD Framework possesses only 6.3 M parameters, which is similar to MobileNet's scale.

Tabernik et. al [9] address the issue of detecting and recognizing a large number of traffic-sign categories suitable for automating traffic-sign inventory management. We adopt a convolutional neural network (CNN) approach, the mask R-CNN, to address the full pipeline of detection and recognition with automatic end-to-end learning. We propose several improvements that are evaluated on the detection of traffic signs and result in an improved overall performance. This approach is applied to detection of 200 traffic-sign categories represented in our novel dataset. The results are reported on highly challenging traffic-sign categories that have not yet been considered in previous works. We provide comprehensive analysis of the deep learning method for the detection of traffic signs with a large intra-category appearance variation and show below 3% error rates with the proposed approach, which is sufficient for deployment in practical applications of the traffic-sign inventory management.

Cao et. al [10] improved traffic sign detection and recognition algorithm is proposed for intelligent vehicles. Firstly, the HSV color space is used for spatial threshold segmentation, and traffic signs are effectively detected based on the shape features. Secondly, this model is considerably improved on the basis of the classical LeNet-5 CNN model by using Gabor kernel as the initial convolutional kernel, adding the BN processing after the pooling layer, selecting Adam method as the optimizer algorithm. Finally, the traffic sign classification and recognition experiments are conducted based on the GTSRB. The favorable prediction and accurate recognition of traffic signs are achieved through the continuous training and testing of the network model. The experimental results show that the accurate recognition rate of traffic signs reaches 99.75%, and the average processing time per frame is 5.4 ms. The proposed algorithm has more admirable accuracy, better real-time performance, stronger generalization ability and higher training efficiency than other algorithms. The accurate recognition rate and average processing time are significantly improved.

Wali et. al [11] provides a comprehensive survey on traffic sign detection, tracking and classification. The details of algorithms, methods and their specifications on detection, tracking and classification are investigated and summarized in the tables along with the corresponding key references. A comparative study on each section has been provided to evaluate the TSDR data, performance metrics and their availability. Current issues and challenges of the existing technologies are illustrated with brief suggestions and a discussion on the progress of driver assistance system research in the future. This review will hopefully lead to increasing efforts towards the development of future vision-based TSDR system.

Liu et. al [12] proposed a new transfer learning structure based on two novel methods of supplemental boosting and cascaded ConvNet to address this shortcoming. The supplemental boosting method is proposed to supplementally retrain an AdaBoost-based detector for the purpose of transferring a detector to adapt to unknown application scenes. The cascaded ConvNet is designed and attached to the end of the AdaBoost-based detector for improving the detection rate and collecting supplemental training samples. With the added supplemental training samples provided by the cascaded ConvNet, the AdaBoost-based detector can be retrained with the supplemental boosting method. The detector combined with the retrained boosted detector and cascaded ConvNet detector can achieve high accuracy and a short detection time. As a representative object detection problem in intelligent transportation systems, the traffic sign detection problem is chosen to show our method. Through experiments with the public datasets from different countries, we show that the proposed framework can quickly detect objects in unknown application scenes.

Zhang et. al [13] presented a new way to evaluate traffic sign visual recognizability in each lane. The proposed model not only quantitatively expresses the visibility and recognizability of a traffic sign from a viewpoint, but also continuously expresses visibility and recognizability, within sight distance, over the entire road surface. Unlike the existing methods for studying visibility and recognizability limited by position of viewpoint in 2D space or cannot be applied in the real road environment, we proposed a new way to evaluate visibility and recognizability in 3D space conquered those problem. Based on traffic sign detection method [31], our algorithm can automatically process more than (92.61% in [31]) traffic signs. The rest traffic signs can be manually detected and processed by our algorithm. Our methods also can be used to detect occlusion and inspect spatial installation information of traffic signs for inventory purposes. Moreover, our model, because it has a process similar to traffic signs, can be easily expanded to other traffic devices, such as traffic lights.

3. PROPOSED SYSTEM

3.1 Overview

This project is aimed at improving traffic sign recognition for autonomous driverless vehicles operating under adverse weather conditions. It involves a sequence of steps, starting with the acquisition of hazy traffic sign images and progressing through haze removal, DLCNN -based sign detection, and performance evaluation through loss and accuracy calculations. Ultimately, the goal is to develop a robust system that can reliably recognize and interpret traffic signs in challenging environmental conditions, contributing to the safety and efficiency of autonomous driving systems. Figure 4.1 shows the proposed system model. The detailed operation illustrated as follows:

step 1. Hazy Traffic Sign Image: This is the initial phase of the project where you start with hazy or degraded traffic sign images. These images are likely to be affected by adverse weather conditions, such as fog, rain, or haze, which can obscure the visibility of traffic signs.

step 2. Deep Learning Haze Removal: In the second step, you employ deep learning techniques to perform haze removal from the hazy traffic sign images. This involves using deep learning convolutional neural networks (DLCNNs) or similar deep learning architectures to enhance the clarity and visibility of the traffic signs in the images by mitigating the effects of haze.

step 3. DLCNN Traffic Sign Detection: After removing the haze, you proceed to the core task of traffic sign recognition. In this step, a DLCNN -based model is employed to detect and recognize traffic signs within the processed images. The DLCNN is trained to identify various traffic sign types, including speed limits, stop signs, yield signs, etc.

step 4. Accuracy and Loss estimation: To train and evaluate the DLCNN model's performance, calculate the loss and accuracy during the training process. The loss function measures the difference between the predicted traffic sign labels and the ground truth labels.

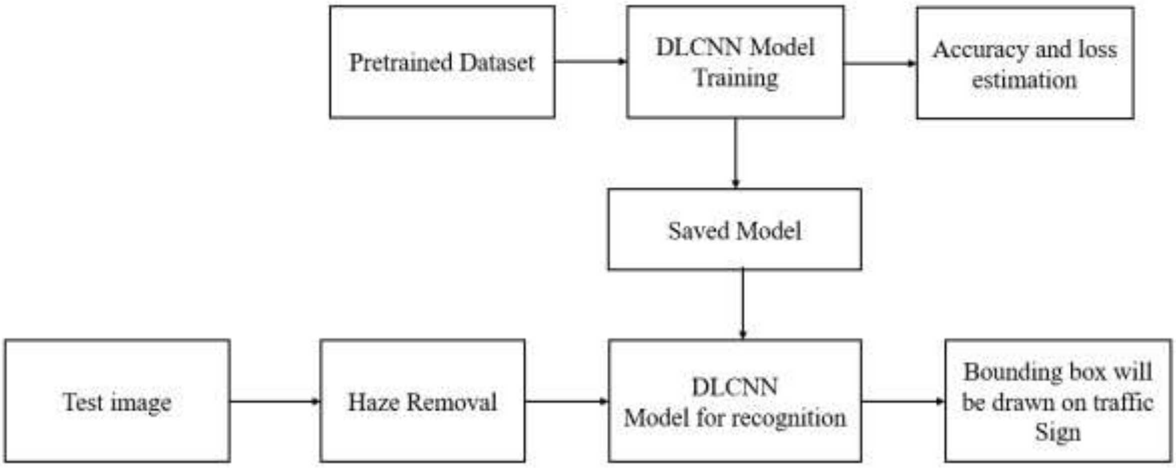


Figure. 1: Block diagram of proposed system.

3.2 Haze Removal

This project revolves around the development of a deep learning-based system for image dehazing, particularly with a focus on enhancing the clarity of hazy or obscured images, which is a common problem in computer vision and image processing.

step 1. Global Variable Initialization: Global variables like dehaze_model, saver, RGB, and MAX are declared. These variables play essential roles in storing and managing components used in the deep learning model.

step 2. Data Reading and Preprocessing: The project begins by gathering a dataset consisting of two types of images: "clear" and "haze." These images likely represent pairs of original (clear) and hazy versions of the same scenes, which are essential for training and evaluating the model.

step 3. TensorFlow Graph Reset: TensorFlow, a popular deep learning framework, is used. Resetting the TensorFlow graph ensures a clean slate for defining and running subsequent computations without interference from previous operations.

step 4. Data Generation: The collected data is preprocessed and organized for model training and evaluation. This step involves tasks like resizing, normalizing, and splitting the dataset into training and testing subsets.

step 5. TensorFlow Placeholders: TensorFlow placeholders, RGB and MAX, are introduced. These placeholders serve as symbolic inputs to the deep learning model. RGB likely represents input hazy images, while MAX represents the target clear images.

Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp105 -119*

step 6. Model Initialization: The core of the project involves building and initializing a deep learning model, referred to as dehaze_model. This model is likely based on an encoder-decoder architecture, specialized in the task of image dehazing.

step 7. Loss Calculation: To quantify how well the model is performing, a loss function is defined. The project utilizes the mean squared error (trainingLoss) as the loss metric. It measures the disparity between the dehazed images predicted by the model and the actual clear images.

step 8. Optimization and Gradients: The model's parameters are optimized using an Adam optimizer, a common optimization algorithm in deep learning. Gradients of the loss with respect to the trainable model parameters are computed.

step 9. Gradient Clipping: Gradient clipping is applied to ensure stable training. This technique limits the magnitude of gradients during optimization, preventing them from becoming too large and causing instability.

step 10. Gradient Descent: The computed and clipped gradients are employed to update the model's parameters, iteratively minimizing the loss. This process is essential for training the model to perform the dehazing task effectively.

step 11. Model Saving: Finally, a TensorFlow Saver object (saver) is instantiated. This object allows the trained model's parameters to be saved to disk. This step is crucial for preserving the trained model for future use or deployment in real-world applications.

3.3 Traffic Sign Detection

According to the facts, training and testing of DLCNN involves in allowing every source feature via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from [0,1]. Convolution layer is the primary layer to extract the features from a source feature and maintains the relationship between pixels by learning the features of feature by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source feature $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an feature (here $d = 3$, since the source feature is RGB) and a filter or kernel with similar size of input feature and can be denoted as $F(k_x, k_y, d)$.

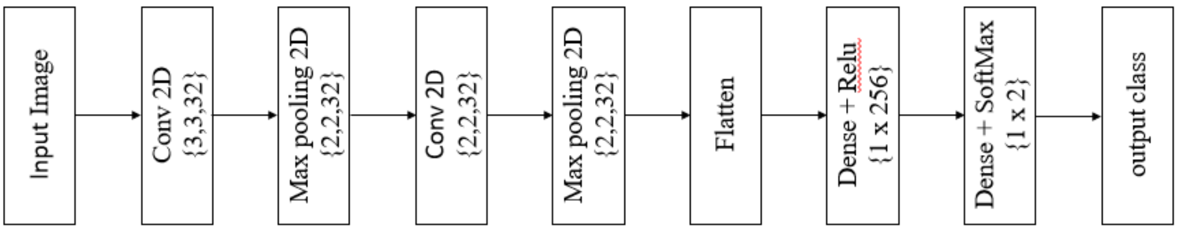


Figure 2: Representation of convolution layer process

The output obtained from convolution process of input feature and filter has a size of $C\left((x - k_x + 1), (y - k_y + 1), 1\right)$, which is referred as feature map. An example of convolution procedure is demonstrated in Fig. 2(a). Let us assume an input feature with a size of 5×5 and the filter having the

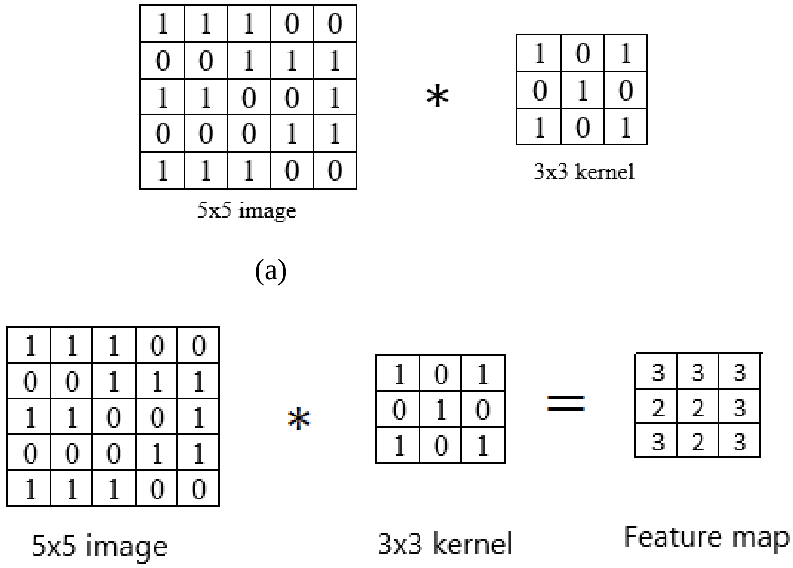
Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp105 -119*

size of 3×3. The feature map of input feature is obtained by multiplying the input feature values with the filter values as given in Fig. 2(b).

ReLU layer: Networks that utilize the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

Max pooing layer: This layer mitigates the number of parameters when there are larger size featurees. This can be called subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.



(b)

Figure 3: Example of convolution layer process. (a) a feature with size $\mathbf{5} \times \mathbf{5}$ is convolving with $\mathbf{3} \times \mathbf{3}$ kernel. (b) Convolved feature map.

SoftMax classifier

Generally, SoftMax function is added at the end of the output since it is the place where the nodes are meet finally and thus, they can be classified. Here, X is the input of all the models and the layers between X and Y are the hidden layers and the data is passed from X to all the layers and Received by Y. Suppose, we have 10 classes, and we predict for which class the given input belongs to. So, for this what we do is allot each class with a particular predicted output. Which means that we have 10 outputs corresponding to 10 different class and predict the class by the highest probability.

ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp105 -119*

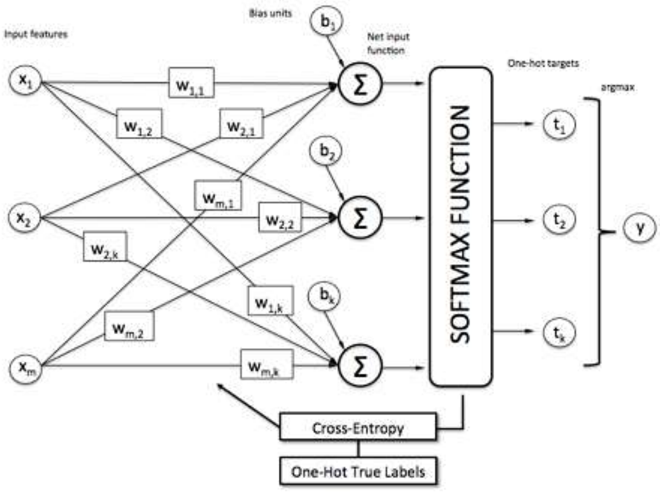


Figure 4: SoftMax classifier.

In Figure.4, and we must predict what is the object that is present in the picture. In the normal case, we predict whether the crop is A. But in this case, we must predict what is the object that is present in the picture. This is the place where softmax comes in handy. As the model is already trained on some data. So, as soon as the picture is given, the model processes the pictures, send it to the hidden layers and then finally send to softmax for classifying the picture. The softmax uses a One-Hot encoding Technique to calculate the cross-entropy loss and get the max. One-Hot Encoding is the technique that is used to categorize the data. In the previous example, if softmax predicts that the object is class A then the One-Hot Encoding for:

Class A will be [1 0 0]

Class B will be [0 1 0]

Class C will be [0 0 1]

From the diagram, we see that the predictions are occurred. But generally, we don't know the predictions. But the machine must choose the correct predicted object. So, for machine to identify an object correctly, it uses a function called cross-entropy function. So, we choose more similar value by using the below cross-entropy formula.

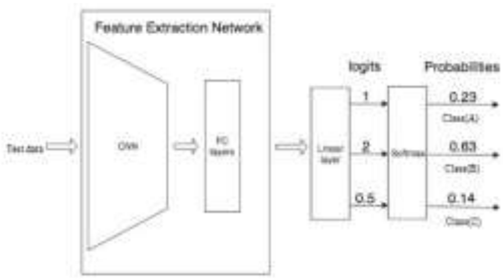


Figure 5: Example of SoftMax classifier.

In Figure.5, we see that 0.462 is the loss of the function for class specific classifier. In the same way, we find loss for remaining classifiers. The lowest the loss function, the better the prediction is. The mathematical representation for loss function can be represented as: -

$$LOSS = np.sum(-Y * np.log(Y\_pred))$$

4. RESULTS AND DISCUSSION

Figure 6 illustrates the process of generating and loading a Convolutional Neural Network (CNN) model for traffic sign detection. It might include components like model architecture, layers, and details of how the model is constructed and prepared for use.

Figure 7 provides a visual demonstration of the performance of a dehazing model. It consists of two images side by side. The first image is a "weather-affected" image, which could be hazy, cloudy, rainy, or captured in poor lighting conditions. This image is typically challenging to interpret due to the weather-related issues. The second image is the "clean" image, which serves as a reference or ground truth. It represents what the image should ideally look like without any weather-induced degradation.

Figure 6. Generate & Load Traffic Sign CNN Model

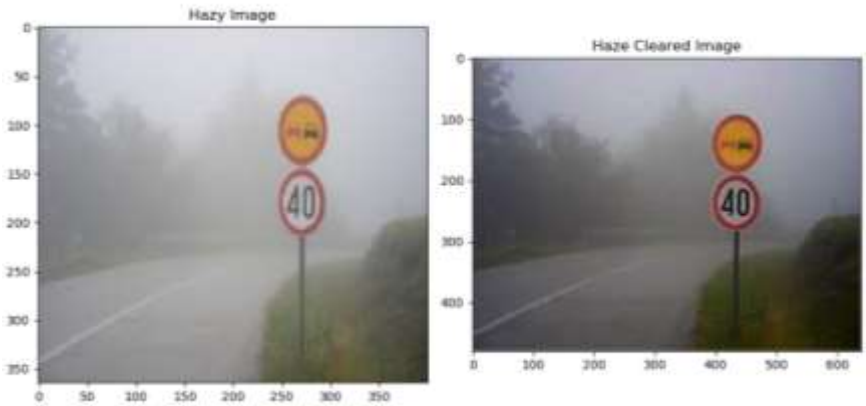


Figure 7. Dehazed performance.

In this figure 8, the focus is on traffic sign detection using a CNN model. It likely shows an image with detected traffic signs, and bounding boxes may be drawn around them to indicate their locations. This figure demonstrates the successful application of the CNN model for traffic sign recognition.

Similar to Figure 7, this figure 9 demonstrates the dehazing performance, but it likely features a different "weather-affected" image as the first image. The purpose is to showcase how the dehazing model performs on a different sample, addressing different weather conditions or image quality issues.

Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp105 -119*

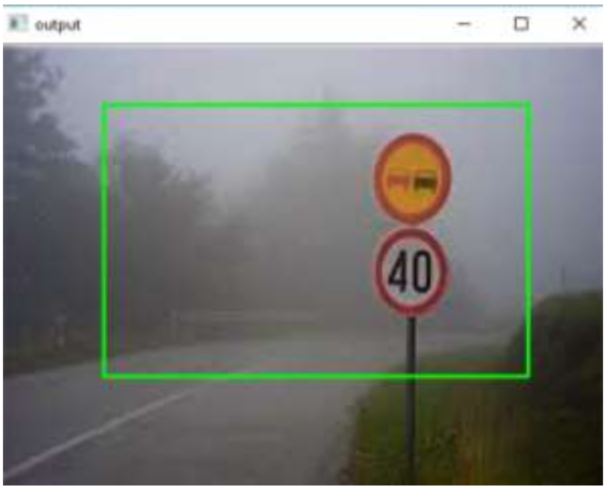


Figure 8. Traffic Sign detected image.

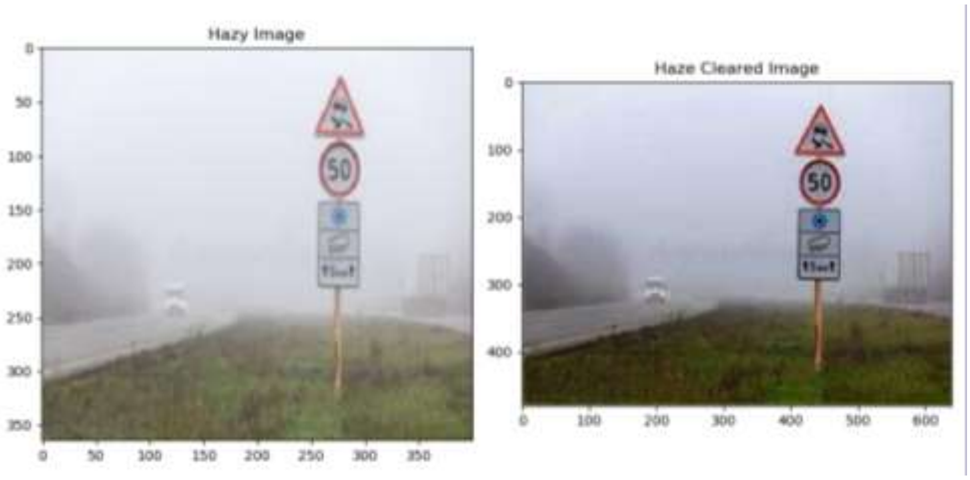


Figure 9. Dehazed performance on sample 2.

Figure 10 shows the results of traffic sign detection on the second sample image mentioned in Figure 9. It may display the detected traffic signs and any bounding boxes or annotations associated with them. This is a continuation of the traffic sign detection process.

Figure 11 likely represents a line plot or a graph that visualizes the performance of the proposed CNN model over the course of training. It typically consists of two subplots:

- One subplot showing the model's accuracy on the training and validation datasets over the course of 10 training epochs. This helps assess how well the model is learning to recognize traffic signs.
- Another subplot showing the model's loss (e.g., mean squared error or cross-entropy loss) on the training and validation datasets across the 10 epochs. This helps monitor how well the model is converging during training and whether it's overfitting or underfitting.

Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp105 -119*

Figure 10 Traffic Sign detected from sample 2.

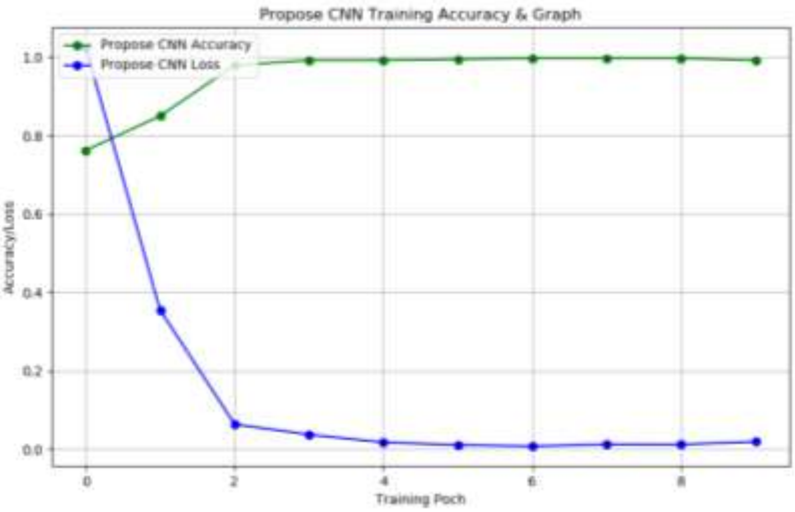


Figure 11. Proposed CNN accuracy and loss for 10 epochs.

In this table 1, performance metrics are compared across three different models: the "Existing SVM" model, the "Existing RFC" (Random Forest Classifier) model, and the "Proposed CNN" (Convolutional Neural Network) model. The table provides values for two key performance metrics, namely "Accuracy (%)" and "Loss," for each of these models. Let's explain the table:

- **Metric:** This column specifies the performance metric being measured, which is categorized into three models: "Existing SVM," "Existing RFC," and "Proposed CNN."
- **Existing SVM:**
  - **Accuracy (%):** The "Existing SVM" model achieves an accuracy of 96.91%. This means that the SVM model correctly classifies approximately 96.91% of the test data samples, indicating a high level of accuracy.
  - **Loss:** The loss for the "Existing SVM" model is 0.142. The loss is a measure of how well the model's predictions align with the true labels. A lower loss value indicates better alignment.

- **Existing RFC (Random Forest Classifier):**
  - **Accuracy (%):** The "Existing RFC" model achieves a higher accuracy of 98.85% compared to the SVM model. It correctly classifies approximately 98.85% of the test data samples, demonstrating improved accuracy over the SVM.
  - **Loss:** The loss for the "Existing RFC" model is 0.0846, which is lower than that of the SVM model. This indicates that the RFC model's predictions are closer to the true labels, resulting in a lower loss.
- **Proposed CNN (Convolutional Neural Network):**
  - **Accuracy (%):** The "Proposed CNN" model attains a perfect accuracy of 100%. This implies that the CNN model correctly classifies all of the test data samples, indicating a flawless performance in terms of accuracy.
  - **Loss:** The loss for the "Proposed CNN" model is 0.0584, which is the lowest among the three models. This suggests that the CNN model's predictions are very close to the true labels, resulting in the smallest loss.

Table 1. Performance comparison

Metric	Existing SVM	Existing RFC	Proposed CNN
Accuracy (%)	96.91	98.85	100
Loss	0.142	0.0846	0.0584

5. CONCLUSION

In conclusion, traffic sign detection using CNN represents a transformative advancement in the realm of computer vision and autonomous transportation systems. The utilization of CNNs offers a multitude of advantages, including unparalleled accuracy in recognizing and classifying traffic signs, robustness in the face of diverse environmental conditions and sign variations, and real-time processing capabilities crucial for ensuring road safety. These systems reduce human intervention, minimize the risk of errors, and enhance the adaptability of autonomous vehicles to different regions and signage styles. Moreover, they contribute to safer roadways by ensuring that vehicles accurately interpret and respond to traffic signs, ultimately leading to improved road safety and traffic management. As technology continues to evolve, CNN-based traffic sign detection systems are poised to play an increasingly pivotal role in advancing the capabilities of autonomous vehicles, making our roads safer and more efficient.

REFERENCES

[1] H. Li, F. Sun, L. Liu and L. Wang, "A novel traffic sign detection method via color segmentation and robust shape matching", Neurocomputing, vol. 169, pp. 77-88, Dec. 2015.
[2] A. Gudigar, S. Chokkadi, U. Raghavendra and U. R. Acharya, "Multiple thresholding and subspace-based approach for detection and recognition of traffic sign", Multimedia Tools Appl., vol. 76, no. 5, pp. 6973-6991, Feb. 2016.
[3] S. K. Berkaya, H. Gunduz, O. Ozsen, C. Akinlar and S. Gunal, "On circular traffic sign detection and recognition", Expert Syst. Appl., vol. 48, pp. 67-75, Apr. 2016.

[4] A. Mannan, K. Javed, A. Ur Rehman, H. A. Babri and S. K. Noon, "Classification of Degraded Traffic Signs Using Flexible Mixture Model and Transfer Learning," in IEEE Access, vol. 7, pp. 148800-148813, 2019, doi: 10.1109/ACCESS.2019.2947069.

[5] Y. Jin, Y. Fu, W. Wang, J. Guo, C. Ren and X. Xiang, "Multi-Feature Fusion and Enhancement Single Shot Detector for Traffic Sign Recognition," in IEEE Access, vol. 8, pp. 38931-38940, 2020, doi: 10.1109/ACCESS.2020.2975828.

[6] C. Gámez Serna and Y. Ruichek, "Classification of Traffic Signs: The European Dataset," in IEEE Access, vol. 6, pp. 78136-78148, 2018, doi: 10.1109/ACCESS.2018.2884826.

[7] D. Varytimidis, F. Alonso-Fernandez, B. Duran and C. Englund, "Action and Intention Recognition of Pedestrians in Urban Traffic," 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2018, pp. 676-682, doi: 10.1109/SITIS.2018.00109.

[8] X. Bangquan and W. Xiao Xiong, "Real-Time Embedded Traffic Sign Recognition Using Efficient Convolutional Neural Network," in IEEE Access, vol. 7, pp. 53330-53346, 2019, doi: 10.1109/ACCESS.2019.2912311.

[9] D. Tabernik and D. Skočaj, "Deep Learning for Large-Scale Traffic-Sign Detection and Recognition," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1427-1440, April 2020, doi: 10.1109/TITS.2019.2913588.

[10] Cao, J., Song, C., Peng, S., Xiao, F. and Song, S., 2019. Improved traffic sign detection and recognition algorithm for intelligent vehicles. Sensors, 19(18), p.4021.

[11] Wali, S.B., Abdullah, M.A., Hannan, M.A., Hussain, A., Samad, S.A., Ker, P.J. and Mansor, M.B., 2019. Vision-based traffic sign detection and recognition systems: Current trends and challenges. Sensors, 19(9), p.2093.

[12] Liu, C., Li, S., Chang, F. and Dong, W., 2018. Supplemental boosting and cascaded ConvNet based transfer learning structure for fast traffic sign detection in unknown application scenes. Sensors, 18(7), p.2386.

[13] Zhang, S., Wang, C., Lin, L., Wen, C., Yang, C., Zhang, Z. and Li, J., 2019. Automated visual recognizability evaluation of traffic sign based on 3D LiDAR point clouds. Remote Sensing, 11(12), p.1453.