Traffic Sign Recognition using YOLOv8 Algorithm extended with CNN

Yashank Singh[1], Dev Singh[2], and Gufran Ali[3]

[1, 2, 3]Sharda University, Greater Noida, Uttar Pradesh, India
[1]2021338973.yashank@ug.sharda.ac.in
[2]2021303194.dev@ug.sharda.ac.in
[3]2021430498.gufran@ug.sharda.ac.in

Abstract. In the realm of computer vision, traffic sign recognition is a captivating problem as it plays an indispensable function in ensuring road safety and improving driving experience. AI models have been a conventional method for traffic sign recognition due to their high accuracy and capability to process images in real-time. This paper presents a software that utilizes the YOLO algorithm combined with a CNN for detecting traffic signs and classification. The YOLO network employed in this study is trained exclusively for detecting traffic signs using a pre-annotated dataset based on a small portion of GTSDB. The detected traffic signs are processed by a CNN model trained on GTSRB dataset for classification. The program detects traffic signs and classifies into 43 classes in real-time with decent accuracy.

Keywords: AI – Artificial Intelligence, CNN – Convolutional Neural Network, YOLO – You Only Look Once, GTSDB – German Traffic Sign Detection Benchmark, GTSRB – German Traffic Sign Recognition Benchmark.

1 Introduction

The detection of traffic signs is an essential element of intelligent transportation systems, which is critical for maintaining road safety. With the rapid advancements in artificial intelligence and computer vision technologies, traffic sign detection has become more sophisticated and efficient, enabling the development of advanced driver assistance systems.

Traffic sign recognition involves detecting and classifying traffic signs in real-time images or videos taken by cameras mounted on the vehicles, road-side units or other sources. The challenge in traffic sign detection lies in the large variability of sign shapes, colors, and conditions, such as varying lighting and weather conditions.

The use of AI-based traffic sign detection systems has the potential to greatly enhance road safety and improve overall driving experience by alerting drivers to traffic signs, reducing the risk of accidents. In recent years, impressive strides have been taken in the field of AI-based traffic sign detection, and various algorithms and models have been proposed and evaluated.

This research paper intends to provide a thorough investigation on AI-based traffic sign detection techniques and to assess their effectiveness in practical situations. The document will offer a synopsis of the present state-of-the-art methods for traffic sign detection and will analyse the strengths and weaknesses of different AI-based approaches. The paper will also present the results of extensive experiments that were conducted to gauge the performance of these methods in applied scenarios.
Additionally, it is crucial to emphasize the relevance of traffic sign detection in the context of emerging trends in intelligent transportation systems. With the increasing focus on autonomous vehicles and connected cars, the demand for reliable and efficient traffic sign detection systems is growing rapidly.
However, despite the progress that has been made in the field of AI-based traffic sign detection, there are still a number of challenges that require attention in this domain, such as detecting traffic signs in complex and cluttered environments, handling occlusions and partial sign visibility, and ensuring robustness in the presence of various environmental conditions.

Therefore, this research paper aims to provide a comprehensive exploration of AI-based traffic sign detection, to analyse the strengths and weaknesses of different approaches, and to address the challenges and opportunities in this field. The results of this study will contribute to the advancement of the field and will be beneficial for scholars and researchers in the development of intelligent transportation systems.

2 Workflow

The GTSRB dataset was pre-processed and divided into training and validation sets. Using the training and validation sets, the CNN model is trained, which is then saved and exported for classification later.
Pre-annotated dataset based on a small portion of GTSDB dataset is used for YOLO model training. After training, the best trained weights are saved and used for localization later.

Real-time video stream from a camera mounted on the vehicle or a pre-recorded video can be used as input for the YOLO model which give a prediction of traffic signs in every frame. Bounding boxes are drawn at these detections. These bounding boxes are then cropped from the frame and used as input for the CNN model which will classify the detected traffic sign into one of 43 categories.

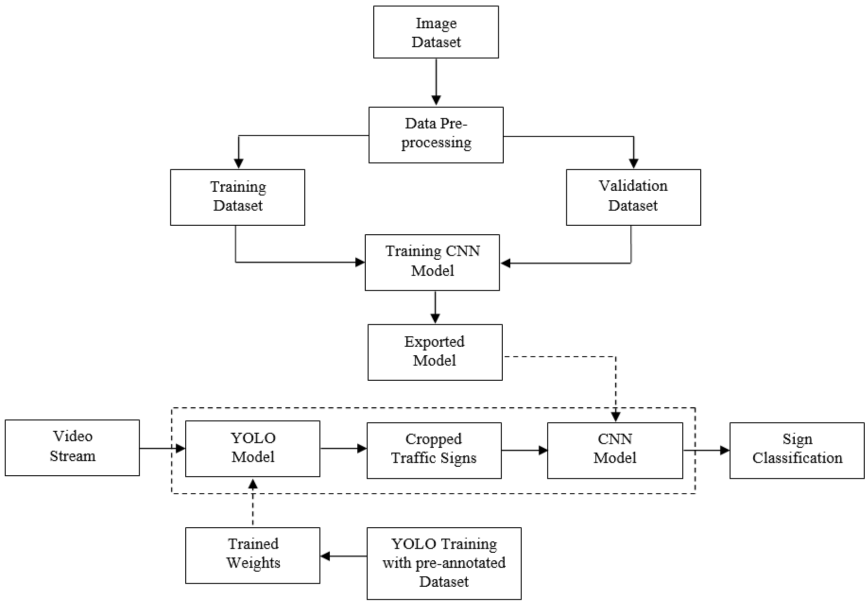

Fig. 1. Workflow of the program

3 Methodology

This paper uses a two-stage implementation for traffic sign recognition. On the first stage, real-time video stream from the cameras is processed by the trained YOLO model. Results are processed and bounding boxes are drawn around detections with confidence over a pre-defined threshold. On the second stage, these detections are croppped and are further processed by the trained CNN model which classifies the traffic signs into 43 categories.

2.1 You Only Look Once

Training. A pre-annotated dataset based on a portion of GTSDB was used to train the model. Random rotation augmentation between -15 and +15 degrees was applied to create two versions of a source image. Training was done with 100 epochs over the above stated dataset with image size of 640x640 pixels and batch size of 16. After training, best weights are used for inference further in this paper. Ultralytics' YOLOv8 python library was used for training of the network.

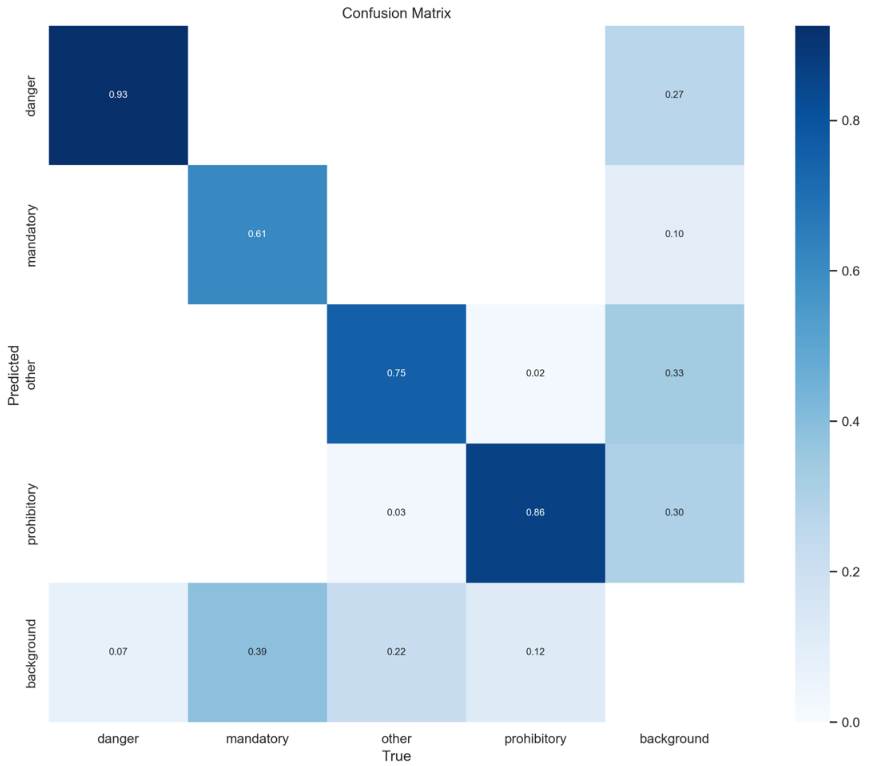

Fig. 2. Confusion Matrix

Calculation. The precision, recall, F1 score and mAP can be evaluated using the following equations:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$f_1 = 2 \cdot \frac{precision \times recall}{precision + recall}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

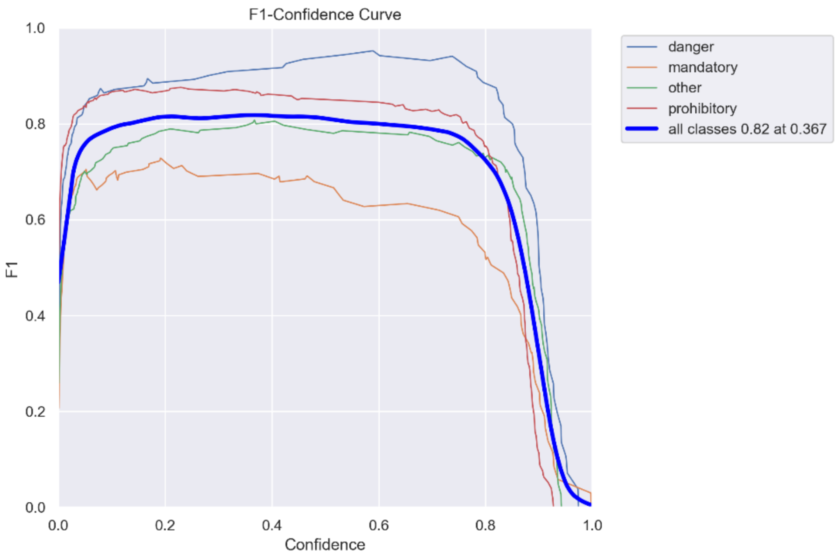

Fig. 3. F1 – Confidence Graph

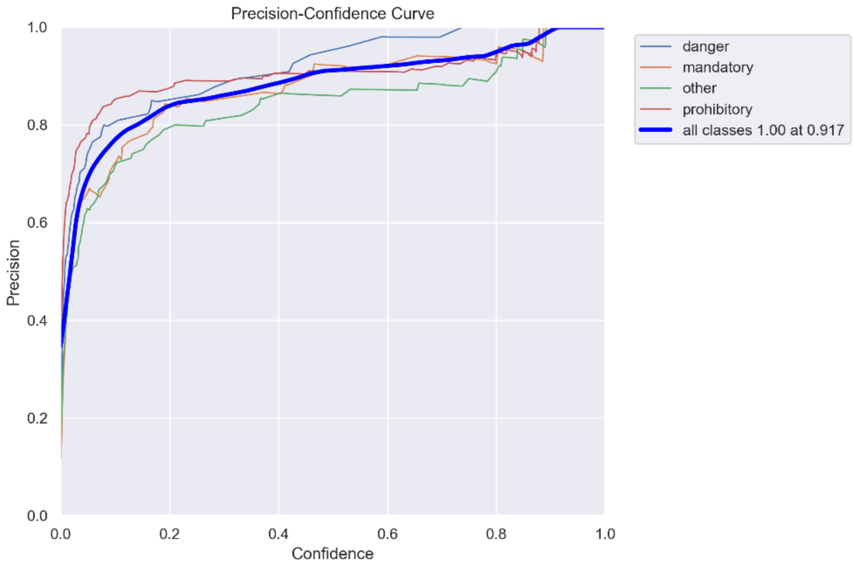

Fig. 4. Precision – Confidence Graph

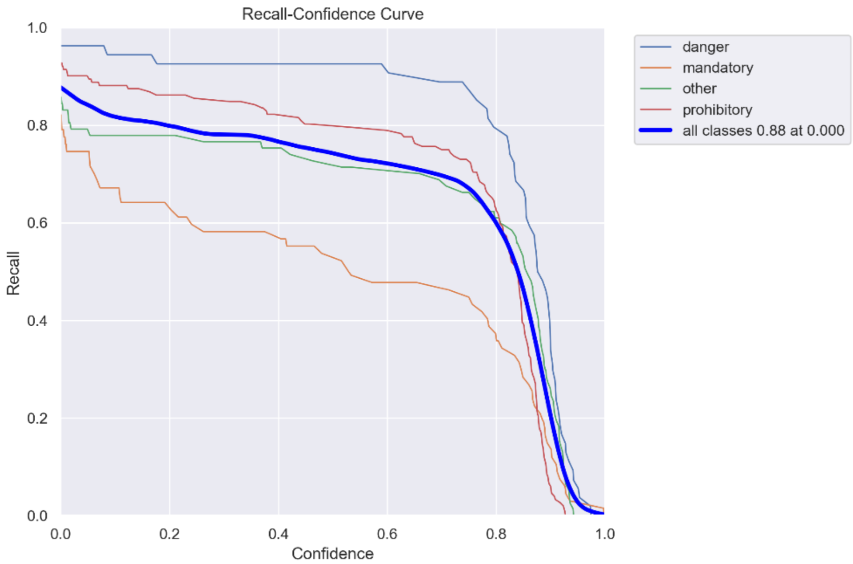

Fig. 5. Recall – Confidence Graph

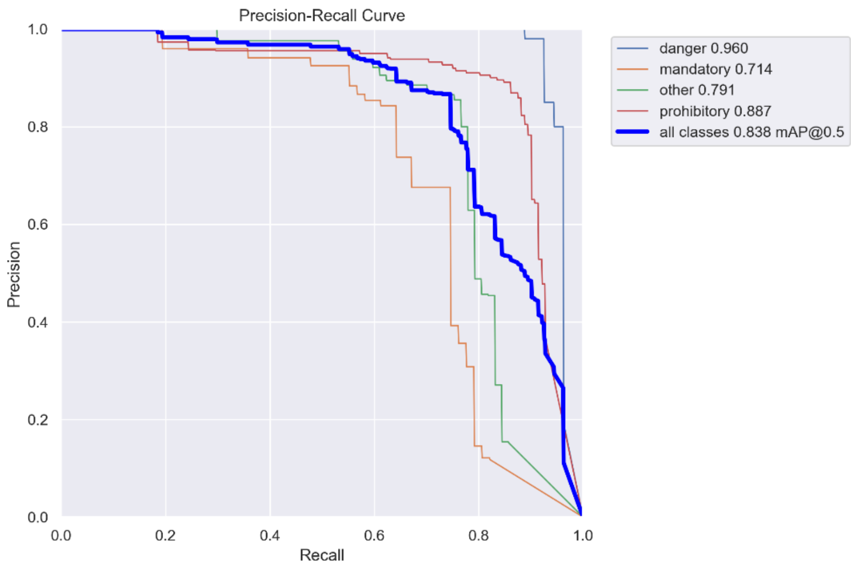

Fig. 6. Precision – Recall Graph

Results. Decent accuracy was obtained with 91.7% precision at 1.0 confidence, and 88% recall at 0.0 confidence. Mean Average Precision (mAP) of 83.8% was obtained at 0.50 confidence. Average inference time recorded is 45ms which allows inference at over 20 frames per second. By utilizing the F1 curve, the confidence value that maximizes precision and recall is 0.367 or 36.7% with the score of 0.82.

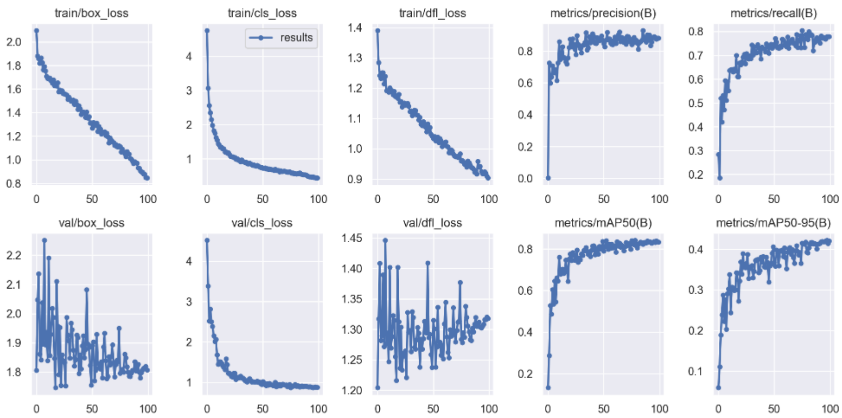

Fig. 7. Training and Validation Results

2.2 Convolutional Neural Network

Dataset. The CNN model was training with GTSRB dataset consisting over 51000 images with a 75-25 training and validation split accounting for 39209 training images and 12630 validation images. The dataset consists of 43 classes of traffic signs.

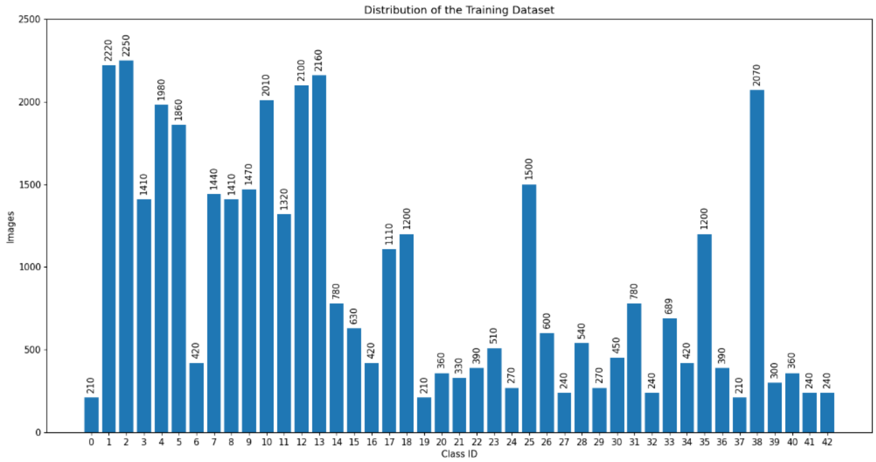

Fig. 8. Distribution of training images

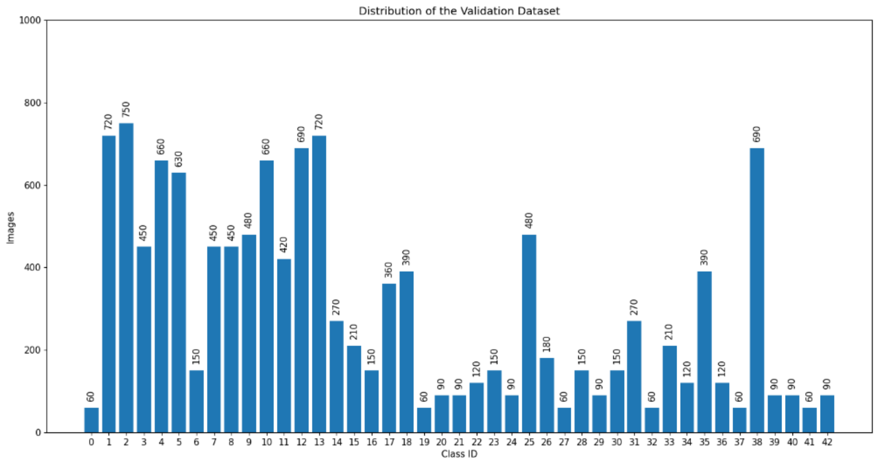

Fig. 9. Distribution of validation images

Model. There are 6 convolutional layers in the neural network, with 32, 64, 64, 128, 128, and 128 neurons assigned to each, respectively. Following each convolutional layer, there is an activation layer of Rectified Linear Unit (ReLU) and a layer of batch normalization. With each change in the number of neurons in the convolutional layers, a max pooling layer is applied.
Finally, flatten layer is applied before connecting with a dense layer with 128 neurons followed with ReLU activation, batch normalization and dropout layers. This is done twice once with a dropout of 0.7 and then with a dropout of 0.5.
Lastly, a dense layer is added with 43 neurons, as there are 43 categories for classification, followed up with a softmax activation layer.

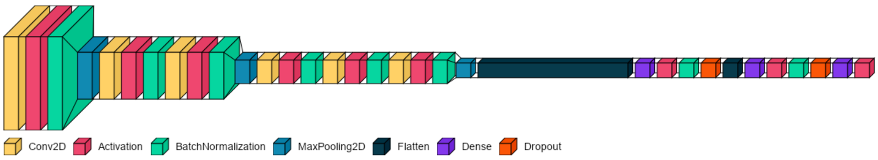

Fig. 10. Layers in the neural network

Training. The images loaded from the dataset were resized into 32x32 pixels, Histogram Equalization was applied to standardize lighting and image values were normalized between 0 and 1.

The following image augmentation were applied:

random rotation range between -10 and +10 degrees.
zoom in and zoom out by 20%.
shear magnitude angle of 10 degrees.
height and width shift range of 10%

Several models were trained with varying amount of epochs and batch sizes. In some models over-fitting was observed while in others under-fitting was observed. Model version 5 was selected for inference trained with 10 epochs and a batch size of 32.

Class IDs	Class Names
0	Speed limit (20km/h)
1	Speed limit (30km/h)
2	Speed limit (50km/h)
3	Speed limit (60km/h)
4	Speed limit (70km/h)
5	Speed limit (80km/h)
6	End of speed limit (80km/h)
7	Speed limit (100km/h)
8	Speed limit (120km/h)
9	No Overtaking
10	No Overtaking for Heavy Vehicles
11	Right-of-Way at next Intersection
12	Priority Road
13	Yield
14	Stop
15	No Vehicles
16	Heavy Vehicles Prohibited
17	No Entry
18	General Caution
19	Dangerous Left Curve
20	Dangerous Right Curve
21	Double Curve
22	Bumpy Road
23	Slippery Road
24	Narrowing Road
25	Road Work
26	Traffic Signals
27	Pedestrian
28	Children
29	Bike
30	Snow
31	Deer
32	End of Limits
33	Turn Right Ahead
34	Turn Left Ahead
35	Ahead Only
36	Go Straight or Right
37	Go Straight or Left
38	Keep Right
39	Keep Left
40	Roundabout Mandatory
41	End of No Overtaking
42	End of No Overtaking for Heavy Vehicles

Fig. 11. Class IDs and Classes Names

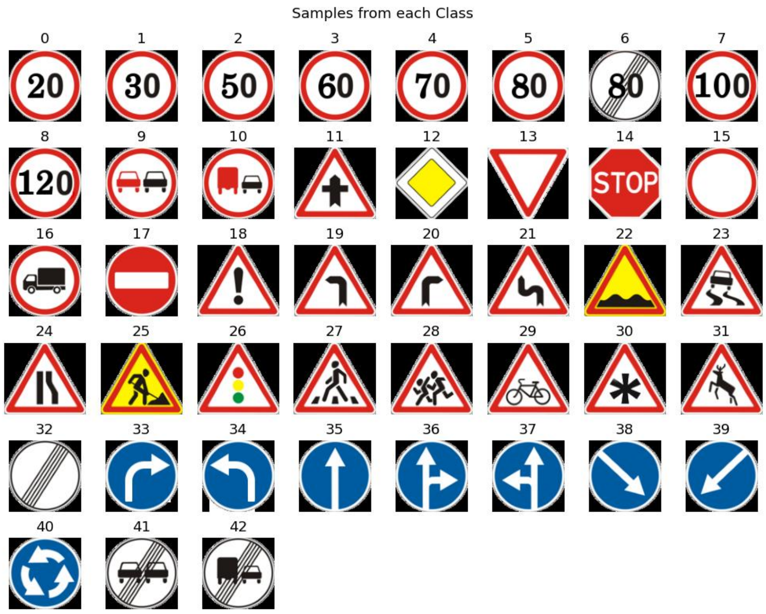

Fig. 12. Unprocessed samples from each class

Fig. 13. Processed samples from each class

Results. Decent accuracy was observed in model v5 with 97.73% training accuracy and 97.90% validation accuracy.

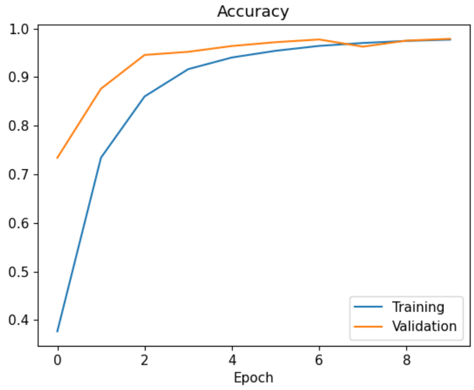

Fig. 14. Model v5 Training and Validation Accuracy – Epochs Curve

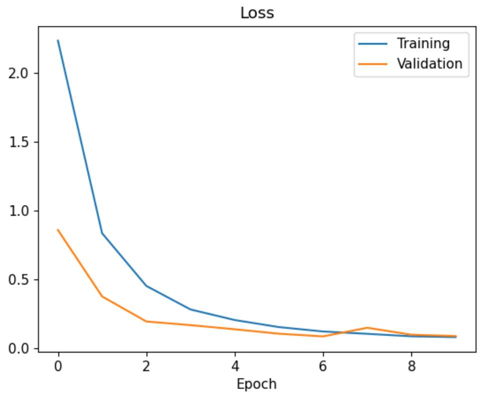

Fig. 15. Model v5 Training and Validation Loss – Epochs Curve

Table 1. Effect of Epochs and Batch Size on Accuracy

Version	Epochs	Batch Size	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	5	64	20.60	94.25	15.03	95.56
2	10	64	06.91	98.05	10.41	97.35
3	30	64	02.55	99.30	16.83	96.65
4	5	32	19.21	94.73	12.08	96.62
5	10	32	08.29	97.73	09.04	97.90
6	30	256	02.90	99.22	13.73	97.25
7	30	8	06.26	98.31	15.27	97.37

Conclusion

In this paper, an application is presented where YOLOv8 algorithm is applied for real-time sign detection and additional CNN is deployed for categorization of traffic signs.
Sign Localization is based on an already annotated dataset. YOLOv8 alone can be trained to identify numerous categories but adding new images to the dataset and relabeling the entire collection would require a considerable investment of time and effort.
Furthermore, the CNN model v5 was trained in approximately three minutes which takes considerably less time comparing to 65 minutes that what was required for training the YOLOv8 model. It is important to consider that this prototype algorithm operates at a slower pace but provides a more accurate solution for traffic sign recognition.
Considering this, it is worth noting that this research provides ample room for future development. As a means of advancing this integrated solution, it is crucial for the CNN model to function in real-time in conjunction with YOLOv8 to facilitate its use in the development of autonomous driving systems. Additionally, the integration of CNN for recognizing traffic signs with other sensor data can enhance the accuracy of the system and minimize the chances of false predictions.

Acknowledgement

The research described in this paper was made possible by the faculty of Computer Science and Engineering, Sharda University, Greater Noida by providing experimental facilities to conduct research and development. We would like to thank Dr. Mandeep Kaur for assisting us in various stages of this project and conducting research.

References

1. Manawadu, M., Wijenayake, U.: Voice-Assisted Real-Time Traffic Sign Recognition System Using Convolutional Neural Network. (2021)
2. Ellahyani, A., Jaafari, I. E., Charfi, S.: Traffic Sign Detection for Intelligent Transportation Systems: A Survey. (2020)
3. Fernando, W. H. D., Sotheeswaran, S.: Automatic Road Traffic Signs Detection and Recognition using 'You Only Look Once' version 4 (YOLOv4). (2021)
4. Novak, B., Ilić, V., Pavković, B.: YOLOv3 Algorithm with Additional Convolutional Neural Network Trained for Traffic Sign Recognition.
5. Huang, Z., Yu, Y., Gu, J., Liu, H.: An Efficient Method for Traffic Sign Recognition based on Extreme Learning Machine.
6. Zhu, Y., Yan, W. Q.: Traffic Sign Recognition based on Deep Learning.