
ISSN: 2321-2152
IJMECE
International Journal of modern
electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

ISSN2321-2152www.ijmece .com

Vol 9, Issuse.2June 2021

CNN-BASED AUTOMATIC TRAFFIC SIGN DETECTION AND NUMBER PLATE RECOGNIZATION

DR SITANATH BISWASS

.

Abstract

Modern roadways are equipped with traffic signs to alert drivers to hazards including posted speed limits, upcoming road repairs, and pedestrian crossings, among others. This study describes A picture segmentation, traffic sign detection, and input image classification system for real-time Traffic Sign Recognition and classification. For this effect, we use the color boost technique to zero in on the reds in the image. Traffic sign material is identified using Convolutional Neural Networks (CNN) for detection, classification, and recognition. Signs that forewarn drivers of upcoming roadwork, sharp bends, and pedestrian crossings have greatly improved drivers' safety. The three stages of this research are all about recognizing and categorizing traffic signs in real time: image classification, input image segmentation, and traffic sign detection. The colour improvement method is used to isolate the red areas of the picture. Convolutional Neural Networks (CNN), such as Faster R-CNN, Retina Net, YOLO V4, and YOLO V5, are used to detect, classify, and recognise the traffic sign content.

The number of cars and trucks on the road has been growing at a staggering rate in recent decades. Typically, the transit regulation that oversees parking garages requires that you check the identification of these cars before granting them permission to park there. Physically inspecting such a massive fleet would be a herculean task. Accordingly, it is crucial to construct an accurate automated licence plate identification model integrating character recognition in order to alleviate the aforementioned difficulties. We've built a model using a wide variety of national licence plates. Yolov4, an implementation of CNN architectures, was used to train the dataset of photos. After applying several methods of picture pre-processing and morphological changes, character recognition was performed using the Tesseract OCR. The suggested system successfully detected 92% of licence plates.

index Terms— Traffic sign detection, traffic sign recognition, convolutional neural network, number plate detection.

1. INTRODUCTION

Detection of traffic signs has been important in recent years due to the growth in popularity of self-driving cars and the advancements in autonomous vehicle technology. Drivers will benefit from traffic sign detection (TSD) since it improves situational awareness and makes it easier to see road signs in low light, severe weather, or at night. And a reliable detection system is essential for making it all work. TSD is helpful for these reasons because it helps keep traffic flowing smoothly and reduces the likelihood of accidents. Despite this progress, TSD remains a difficult real-world issue due to factors such as picture quality, illumination, etc. that are more indicative of the actual world. The number of cars on the road has skyrocketed in the last two decades.

ASSOCIATE PROFESSOR, Mtech,Ph.D
Department of CSE
Gandhi Institute for Technology,Bhubaneswar

As the number of vehicles on the road grows, it becomes increasingly challenging to monitor them all for safety and efficiency reasons. Research into Automatic License Plate Recognition (ALPR) has risen in prominence as

a result of its use as a surveillance system for capturing photographs of vehicles and reading their licence plate numbers. License plate recognition is the focus of our research, and we use the License Plates dataset to do so. We used YOLOv4 for the recognition, keeping with the multi task learning method we've used before for character string recognition. The primary goal of this project is to identify a licence plate in a given video feed. We've created a method that works well in a wide range of lighting circumstances to accurately identify a licence plate. The goal of this project is to offer input to the Car License Plate Recognition process by reading licence plate information from a picture. The number plate picture has been extracted and is shown on the screen.

Our primary objective is to successfully implement Convolutional neural networks, which are made up of neurons with learnable weights and biases, to detect and identify traffic signs and license plates in difficult and potentially dangerous environments, all while maintaining a high level of performance. The German Traffic Benchmark's recommended data set will be imported first. Furthermore, it will explore and summarize each data set in order to show them in a unique graphical way. Third, we need to plot out how we'll measure and improve the model's training and testing results. Finally, we may extrapolate current visual data using the model. Both the German Traffic Detection Benchmark (GTSDB) dataset and Google's open car plate photo collection are used.

2. Related Work

The Google License Plate dataset was used to identify vehicles' registration numbers, while the German Traffic Detection Benchmark (GTSDB) dataset was utilised to identify traffic signs. There are a number of advantages to using this dataset instead of others, and it is generally acknowledged and used for testing traffic sign identification methods in the literature. In addition, the GSTDB authors and organisation host a public challenge in which researchers from a wide range of disciplines submit their findings and put the GSTDB dataset to the test. Images of actual traffic captured on highways, country roads, and city streets at different times of day and in different lighting and weather situations may be found in the GTSDB collection. There are a total of 900 photos in the dataset, contains 1206 photos totalling 1206 traffic signs; 600 images serving as instruction (846 traffic signs), and 300 serving as testing (360 traffic signs).

We classify detection approaches into the following broad categories: color-based (based on the colour space), shape-based, and learning-based in order to better understand how each kind of method is used to pinpoint areas of interest including traffic signs (including deep learning). Furthermore, we classify techniques in two broad groups: those that rely on artificially generated features (HOG, LBP, SIFT, SURF, BRISK), and those that use deep learning techniques.

Access and egress points on Japanese tollways might one day be operated automatically, without human intervention. It has been decided that reading the classification number on the top side of the licence plates is the most efficient approach to categorise the different cars at the toll gates. There is now a sophisticated technology available for use in commercial facilities that makes use of picture analysis. As many as 50 entrance toll gates are already using it successfully. An unprecedented high recognition accuracy of over 99% has been obtained.

3. METHODOF IMPLEMENTATION

For this model, we take advantage of the fact that each neuron in a convolutional neural network is locally connected to the inputs of the layer below it. This layer functions similarly to a 2D convolution with a specific filter, and its activation can be determined as the result of a nonlinear transformation using the equation (1). Here, x is the activity of the input neurons, and f is the result of the Convolutional filter's HW weight matrix being multiplied together. The activation function for the connections between the input neurons and the neuron I j) is represented by the ai,j in the Convolutional layer. The symbol (•), sometimes known as a sigmoid or hyperbolic tangent, represents a non-linear activation function, while the letter b stands for a bias. The convolution operator is represented by the symbol. Taking the maximum response operation and ignoring the convolution to the activations of the neurons in a non-overlapping rectangular segment is the second part of CNN, known as max pooling.the preceding layer, as described by the equation.

$$O_{i,j} = \max(y) = \max_{1\le i'\le H', 1\le j'\le W'} y_{i+i',j+j'}$$

When using a max pooling layer, the output is represented by Oi, j, the maximum response operator is denoted by max (•), and the variable y is the product of H'W', where H and W are patches of neurons connected to neuron I j). Since the maximum pooling is non-overlapping and has a stride equal to the size of the neurone patch, this equation ensures dimensional reductions with the product factor of H' and W' along with each direction. Since the combination of CNN's three architectural qualities substantially restricts the applicability of many parameters previously used by other neural networks, back-propagation is used for CNN training rather than the other deep learning models. As a result, the architecture's parameters are concurrently optimized via the back-propagation system.

The use of CNN for both feature extraction and classification [6, 7] is not new. The limited and sub-optimal generalization ability of CNN stems from the fact that its fully-connected layers constitute a classical neural network classifier trained via gradient descent -based implementations. Nonetheless, such approaches can yield impressive results, albeit typically on the basis of an extremely small training set.huge and complex network.

With Yolo V4 and Yolo V5, object detection works as follows.

Step 1:

Produces a set of SS grids based on an image. K boundary boxes are generated for each grid by pinpointing where the anchor boxes should go. A B boundary box forecast and confidence score are generated for each grid cell.

Step 2:

B only picks up one thing, no matter how many containers it scans. Further, it estimates the likelihood of classes C under certain conditions (one per class for the similarity of the object class).

Step 3:

The b = [bx, by, bw, bh, bc] T and the class = [class1, class2,class] T are predicted after the system runs through the CNN layers to extract all features from the picture.

Step 4:

evaluates the threshold IoUthres against the optimal confidence IoUtruth pred of the K bounding boxes. The item is within the bounding box if IoUtruth pred > IoUthres. If the item were larger, the bounding box would not contain it.

Step 5:

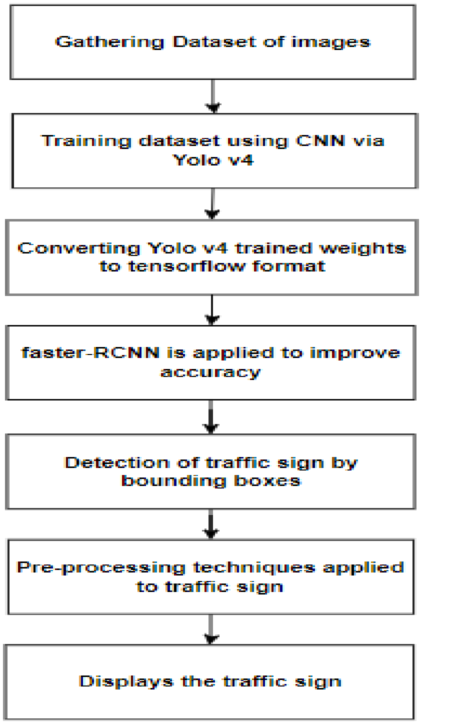
The computer will identify the object after it has determined a likely classification for it. In our investigation, we use Non-Maximum Suppression (NMS) to look for and find possible trouble spots including drop boxes, redundant output, and object detection finds. In addition, the NMS is put into action in the following ways:

1. Arrange forecasts by their level of certainty.(2) If we evaluate the same class predictions and IoU > 0.5 According to the latest prognosis, we should ignore all predictions and instead focus on the highest-scoring options.

Test the validity of the predictions by going back to (2) and doing it again. Our designs were fine-tuned using pre-trained prohibitory sign weights, which significantly reduces training time, before training had even finished.

Step 6:

The last step results in a categorized image labeled with the class.

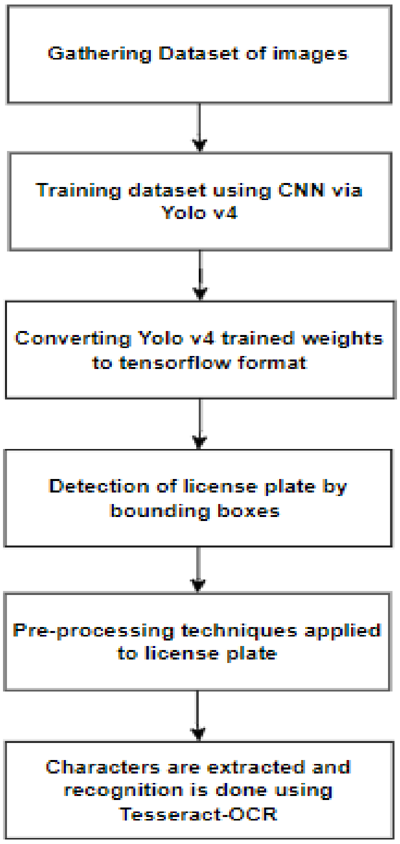


Traffic sign detection flowchart

Included in the 4,500-photo archive are scenes of regular urban life. The jpg images were captured with three distinct cameras. We used 60% of the data for training, 30% for testing, and 10% for evaluation (for validation). In order to forecast where a licence plate would be in a video or still picture, we turned to the YOLOv5 algorithm.

In the first stage, we use the YOLOv4 bounding box coordinates to extract the portion of the subimage that is inside the box. Since this picture is often rather little, we utilise cv2.resize() to increase its size by a factor of three. The next step is to grayscale the picture and blur it somewhat using a Gaussian function. After that, Otsu's approach is implemented, and the picture is thresholder such that the text is white on a black backdrop. White writing on a black backdrop is useful for identifying an image's edges. OpenCV is used to magnify the picture so that the edges may be seen and then detected. The next step is to use OpenCV to identify and arrange all the rectangular shapes in the picture.

Fig 3.1Example for mosaic augmentation for license plate detection



License plate recognition flowchart

The model may be put to better use with a user interface. To create the UI, we combine Flask with HTML. Results from the aforementioned measures indicate that yolo V5 is capable of making reliable traffic sign and vehicle licence plate predictions. Hence, we utilised yolo v5 in our project.

Google's open vehicle dataset, a domestic urban traffic sign scene photo collection, and an overseas traffic sign detection data set are used in this project (GTSDB). The international traffic sign detection data set is made up of 900 photographs taken in 1360x800 resolution with the provided data on the locations of traffic signs in the photos. A total of 125 pictures representing different landscapes and five images depicting common urban traffic signals made up the data set, which was divided into 600 training images and 300 test images. To get a larger sample size, we rotated the dataset by 5 degrees and 10 degrees, and then we downsized it to two different pixel dimensions (1920 x 1080 and 800 x 600), for a grand total of 750 pictures. The leave-out strategy is used to analyse training effects by splitting the dataset into two groups: the training set and the testing set. Since there weren't enough datasets to go around for the home cities, we used all the photos for the training set and randomly selected 225 for the test. The Training Image Labeler is used to manually label a database of images of U.S. cities.

It is helpful to frame the traffic sign in the scene shot by drawing a rectangle around it. The coordinates, as well as the length and width, of the rectangular frame are recorded by the software. Adjusting and Assessing Training Settings 4.2. The training period is set to 10, the batch size is often set to 128, and the learning rate is fixed to 2.48105 for the purpose of training datasets for the identification of foreign traffic signs. Each time the parameters are adjusted, the effect of the previous value should be lessened; for example, if the value of is 0.82, if 128 samples are used to train each iteration, and if one cycle uses all of the data to retrain, then the impact of the previous value of should be reduced.

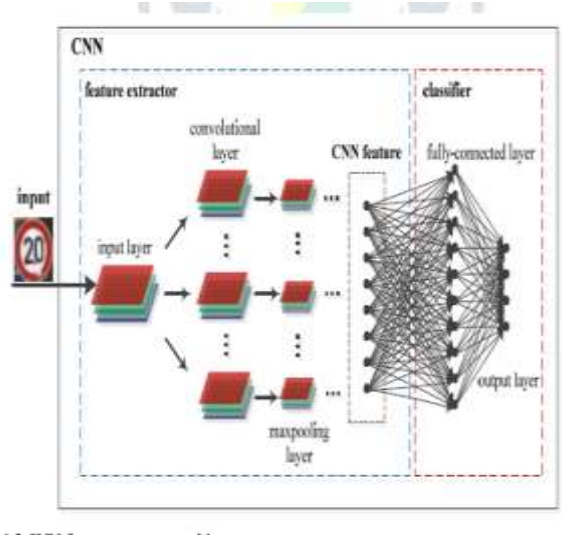
Since Alex Net is a pre-trained deep convolutional neural network model, the data obtained by the model during the initial training is very close to the objective, and the loss value is small. Continuous training resulted in a rapid improvement in the network's signal-recognition abilities, as it defining features gradually morphed over time. The weight characteristics of the first convolutional layer of the combined convolutional network are shown in (a) and (b) of Figure 5 after training on international and domestic datasets, respectively.Learned characteristics, such as colours and edges, are mostly represented in the graph as a result of the neural network's processing. Considering the underlying characteristics are only a high-level summary of the picture, the weights seldom change. Lines of varying lengths and orientations in the weight feature map stand in for the features of extracted textures, each of which represents an individual texture's size and extraction. Background feature integration is seen on the slurred weight feature map.

For a shared convolutional network, the first convolutional layer's attribute weights After the training phase are complete, the trained model is put to the test on the test set's data. All regional boundaries with a score higher than 0.5 will be shown in the test results. The more highly rated border will often have better target detection. Area overlaps ratio Iowa, the ratio between the intersection and union areas of the true target area and the detection area in the picture, is used as the assessment index for the model detection. If Iowa >= 0.5, then the findings are valid. The findings of the two test models are shown in Table 1. Table 1: The Standard Error of the Results from Two Test Sets Accuracy on average, 1 52% 2 65% The detection outcomes of international and local test sets are shown in Figures 6 and 7, respectively. The detection model's output is represented by a white border. The traffic signal's likelihood is shown in the digital display box's picture.

3.1Algorithms

Convolutional Neural Networks (CNN)

Three distinguishing features of convolutional neural network (CNN) models for deep learning are their reliance on locally linked neurones, the calculation of shared weight, and the provision of spatial or temporal sub-sampling. There are essentially two components that make up CNN the first layer has a series of alternating Convolutional layers, while the second features a carpooling layer. Each layer receives as input the data that was generated by the layer above it. As a consequence, a hierarchical feature extractor is formed to convert the raw input photos into feature vectors. The fully-connected layers, characteristic of a feed-forward neural network, then categorise the generated features vectors.



3.2 Reader familiarity with Neural networks is expected.

Superior performance in Machine Learning is achieved via Artificial Neural Networks. Images, sounds, and even language may all be classified with the help of artificial neural networks. Recurrent Neural Networks, or more specifically a Long Short-Term Memory (LSTM) network, are used to predict the sequence of words, whereas Convolutional Neural Networks are used to classify images. Here on the blog, we'll be putting together CNN's foundational pieces.

YOLO V5

Object detection in photographs is regarded to be a common activity for the human brain, but is challenging for a computer. The task of "object detection" is one of computer vision's primary functions, since it seeks for and points out certain objects in a picture. To solve this problem, many strategies have emerged in recent years. Among the many real-time object recognition methods available, YOLO (You Only Look Once) stands out as a popular choice. The original publication was done by Redmond et al.

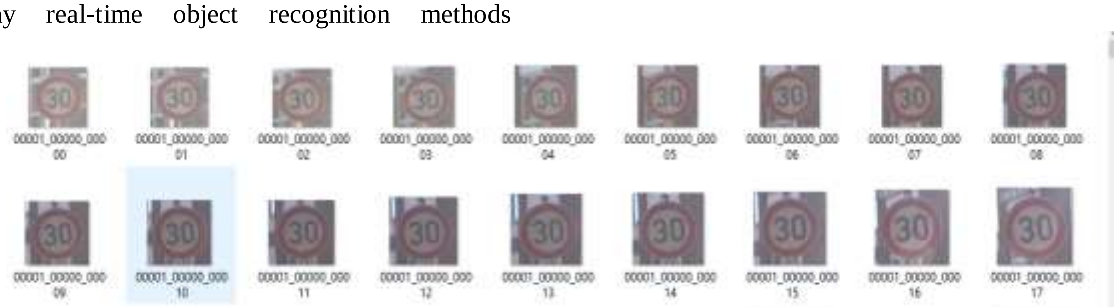
This article will show you how to conduct a whole object identification project using a personal dataset and the latest YOLOv5 implementation developed by Ultralights [2]. We're going totrain our own model using transfer-learning methods, assess its effectiveness, apply it for inference, and even convert it to various file formats like ONNX and TensorORT.

The course is designed for those who have a theoretical understanding of object identification algorithms and are looking for advice on how to put that knowledge into practise. For your convenience, the whole code is included below along with a simple Jupiter notebook.

by manually labelling around 250 web-based photos and video frames of penguins. I used the user-friendly and cost-free Roboflow platform, which took me a few hours. It is advised to train with more than 10,000 instances per class and over 1500 photos per class to produce a strong YOLOv5 model. Additionally, adding background pictures up to 10% is advised to lower false-positive error rates. Since my dataset is so little, I'll use transfer learning strategies to streamline the training procedure.

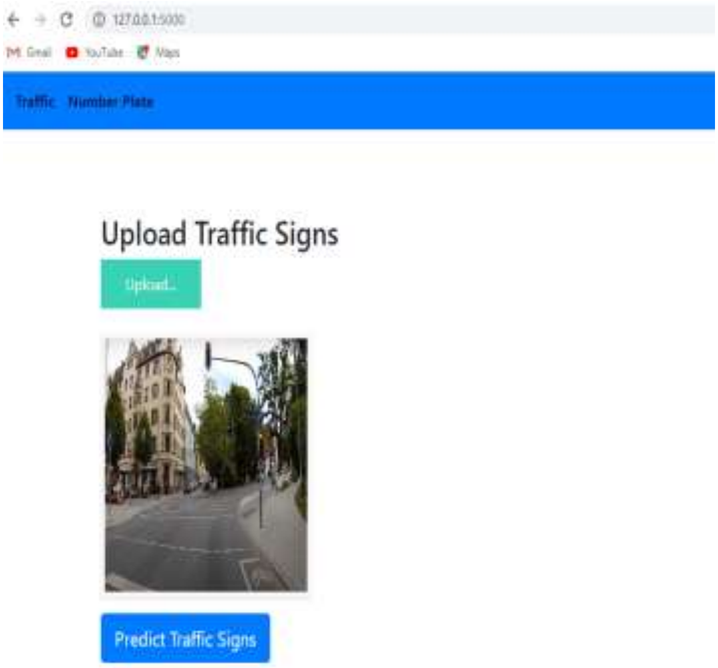
3.2 SAMPLE DATASET

For the purpose of traffic sign identification, this collection includes 877 photos split over 4 categories. As PASCAL VOC was the format in which I received the bounding box annotations, I converted them to YoloV4 for use in the current endeavour (using roboflow),



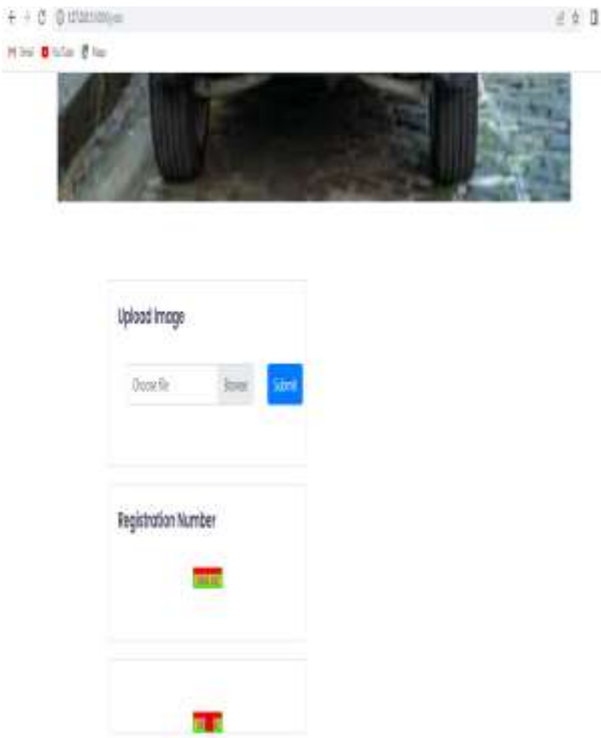
4. Results

The model may be put to better use with a user interface. To create the UI, we combine Flask with HTML. Results from the aforementioned measures indicate that yolo V5 is capable of making reliable traffic sign and vehicle licence plate predictions. As a result, yolo v5 was included into this project.



Traffic sign Image upload preview

Prediction of traffic Sign



Home page of Vehicle Licence Plate Detection

	Faster R-CNN	RetinaNet	YOLO V4	YOLO V5
Run1	55.7%	75.5%	90.0%	90.7%
Run2	44.2%	81.8%	90.5%	91.5%
Run3	53.0%	82.1%	91.6%	92.4%
Average	51.0%	79.5%	90.7%	91.5%

Table 6.1Accuracy of CNN models

	Faster R-CNN	RetinaNet	YOLO V4	YOLO V5
Training (batch size, epochs, learning rate)	(2,100,0.0001)	(8,1200,0.01)	(8,1200,0.01)	(16,1200,0.0032)
Training Loss	0.136	0.020	0.017	0.015
mAP@0.5-0.95	43.4%	44.6%	45.4%	58.9%
Inference speed: Image resolution (1774 × 2365)	0.047 s	0.032 s	0.03 s	0.009 s
Inference speed: Image resolution (204 × 170)	0.052 s	0.0032 s	0.03 s	0.009 s
Model Size (MB)	624.84	219.8	134.5	14.8

Table: 6.2 Metrics of CNN models

Output of Traffic sign test data

5. CONCLUSION

The yolo V4 target identification technique is based on the concept of migration learning; it employs previously learned neural network models to extract visual characteristics and may be used to effectively train a target detection model system with a little quantity of data. The method proposes an RPN module that vastly improves the efficiency of the target area suggestion box's selection process. By enhancing yolo V4 and omitting the need for its repetitive extraction, the algorithm speeds up the detection process. In this study, we show that the yolo V4 training target detection model significantly outperforms previous versions, and that it can be easily adapted to new scenes via extensive testing on domestic and international traffic sign scene graphs. More scene photos need to be collected, the model's detection accuracy has to be boosted, and traffic class labels need to be added before the target detection model can understand the meaning of the symbols used in traffic signs.

The experimental findings presented in this research demonstrate the efficacy of the Convolutional neural network as a method of picture categorisation and recognition. Because of its straightforward design, we were able to develop a high-performing classifier for the notoriously challenging job of traffic sign categorisation and identification. The model's strength is also its intrinsic weakness, which shows itself as over-fitting. Therefore, data augmentation is employed as a simple solution to this issue. Using the state-of-the-art YOLOv5 object identification algorithm, we demonstrate a comprehensive real-time end-to-end system, as well as a huge and varied dataset of traffic signs and licence plates. Through enhanced picture generation and algorithm enhancement, our system has the potential to achieve a complete detection rate of 89% on the Google Open Image dataset. In the future, we want to enhance our algorithm and improve accuracy by exploring new CNN architecture. We also think the suggested system's tech can recognise the traffic sign and locate a vehicle in any nation.

REFERENCES

[1]. A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," IEEE Trans. Intell. Transp. Syst., vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[2]. A. Gudigar, C. Shreesha, and U. Raghavendra, "A review on automatic detection and recognition of traffic sign," Multimedia Tools Appl., vol. 75, no. 1, pp. 333–364, 2016.

[3]. S. Xu, "Robust traffic sign shape recognition using geometric matching," IET Intell. Transp. Syst., vol. 3, no. 1, pp. 10–18, Mar. 2009.

[4]. H. Fleyeh and E. Davami, "Eigen-based traffic sign recognition," IET Intell. Transp. Syst., vol. 5, no. 3, pp. 190–196, Sep. 2011.

[5]. Y. Chen, Y. Xie, and Y. Wang, "Detection and recognition of traffic signs based on hsv vision model and shape features," J. Comput., vol. 8, no. 5, pp. 1366–1370, May 2013.

[6]. J. M. Lillo-Castellano, I. Mora-Jiménez, C. Figuera-Pozuelo, and J. L. Rojo-Álvarez, "Traffic sign segmentation and classification using statistical learning methods," Neurocomputing, vol. 153, pp. 286–299, Apr. 2015.

[7]. A. Ellahyani, M. El Ansari, and I. El Jaafari, "Traffic sign detection and recognition based on random forests," Appl. Soft Comput., vol. 46, pp. 805–815, Sep. 2016.

[8]. R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," Mach. Vis. Appl., vol. 25, no. 3, pp. 633–647, Apr. 2014.

[9]. F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition in three stages," Robot. Auto. Syst., vol. 62, no. 1, pp. 16–24, 2014.

[10]. X. W. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, "Recognition of traffic signs based on their colour and shape features extracted using human vision models," J. Vis. Commun.Image Represent., vol. 17, no. 4, pp. 675–685, Aug. 2006.

[11]. X. Gao, K. Hong, P. Passmore, L. Podladchikova, and D. Shaposhnikov, "Colour vision model-based approach for segmentation of traffic signs," J. Image Video Process., vol. 2008, Dec. 2007, Art. no. 386705. [Online]. Available:

https://link.springer.com/content/pdf/10.1155%2F2008%2F386705.pdf

[12]. J. F. Khan, S. M. A. Bhuiyan, and R. R. Adhami, "Image segmentation and shape analysis for road-sign detection," IEEE Trans. Intell. Transp.Syst., vol. 12, no. 1, pp. 83–96, Mar. 2011.

[13]. I. M. Creusen, R. G. J. Wijnhoven, E. Herbschleb, and P. H. N. de With, "Color exploitation in hog-based traffic sign detection," in Proc. IEEEInt. Conf. Image Process., Sep. 2010, pp. 2669–2672.

[14]. M. S. Prieto and A. R. Allen, "Using self-organising maps in the detection and recognition of road signs," Image Vis. Comput., vol. 27, no. 6, pp. 673–683, May 2009.

[15]. J. Canny, "A computational approach to edge detection," in Readings in Computer Vision. Amsterdam, The Netherlands: Elsevier, 1987, pp. 184–203.

[16]. M. Meuter, C. Nunn, S. M. Gormer, S. Müller-Schneiders, and A. Kummert, "A decision fusion and reasoning module for a traffic sign recognition system," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1126–1134, Apr. 2011.

[17]. M. A. Garcia-Garrido, M. Ocaña, D. F. Llorca, M. Sotelo, E. Arroyo, and A. Llamazares, "Robust traffic signs detection by means of vision and v2i communications," in Proc. 14th Int. IEEE Conf. Intell. Transp.Syst. (ITSC), Oct. 2011, pp. 1003–1008.

[18]. Á. González et al., "Automatic traffic signs and panels inspection system using computer vision," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 2, pp. 485–499, Jun. 2011.