TSRSY - Traffic Sign Recognition System using Deep Learning

Hiral Visaria
Information Technology
Don Bosco Institute of Technology
Mumbai, India
hiralvisaria23@gmail.com

Shruti Chaube
Information Technology
Don Bosco Institute of Technology
Mumbai, India
shrutichaube2@gmail.com

Franky John
Information Technology
Don Bosco Institute of Technology
Mumbai, India
frankyjohn2255@gmail.com

Prof. Aruna Khubalkar
Assistant Professor, IT
Don Bosco Institute of Technology
Mumbai, India
aruna@dbit.in

***Abstract*—Most of the traffic injuries are triggered because of carelessness, lack of knowledge of the rules and neglecting traffic signboards. The occurrence of street accidents in our country is alarming. When a person disobeys the traffic signs and signals, he/she is putting one's life at risk in addition to the life of the pedestrians. The traffic signs and signals are designed to regulate traffic and thus reduces the severity of traffic injuries. Today, drivers are distracted a lot while driving. For example, talking to someone on the passenger seat, listening to music or news, or using the phone which increases the chances of an accident. To tackle this issue, we had designed Traffic Sign Recognition System using YOLOv5 (TSRSY), which detects traffic signs, classifies them, and provides an appropriate audio alert message.**

Keywords— Traffic sign recognition, YOLOv5, gTTS, Image Processing, Deep Learning, Text to Speech Translation

I. Introduction

India ranks #1 among 199 nations in terms of road fatalities, accounting for over 11% of all accident-related fatalities worldwide [1]. In the calendar year 2019, there were 449,002 accidents in the country, with 151,113 people killed and 451,361 wounded [1]. The number of fatal traffic accidents increased by 10% in 2021 compared to 2020 [2]. When the number of fatal accidents is compared to the total number of accidents, the case fatality rate is high. In 2020, approximately 43 percent of all accidents resulted in death, but by 2021, the figure had risen to 45 percent [2].

The Government of India regulates traffic and creates road signs to warn of road conditions such as roundabouts, speed limits, and the presence of schools, hospitals, etc to ensure the safety of people inside the car and pedestrians.

Road signs should be placed such that they are visible to all road users. Traffic signs should be placed in relation to the site or scenario to which they apply to help in transmitting accurate meaning. The position and readability of the road sign should be such that road users have enough time to read and react at the safe speed [3].

There are three major classes of traffic signs in India, which are:

a. Regulatory Signs: These signs are circular and show rules and regulations.

b. Warnings: These signs are triangular.

c. Informational: These signs are rectangular.

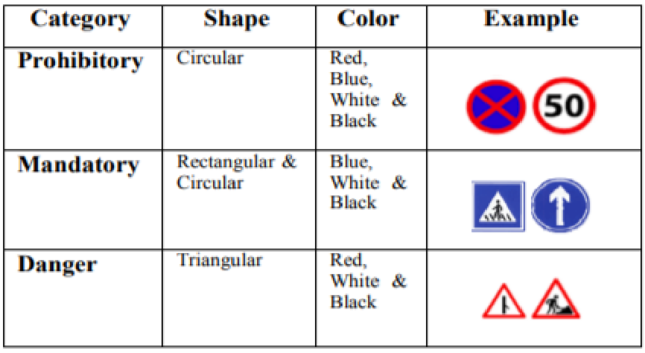
There are further more two sign boards:

a. Give Way: It's shape is an upside-down triangle

b. Stop: It's shape is an octagon

A sign with circle having a slash shows prohibited signs and circles while no slashes show rules. Triangles are pointy and show danger. Circle with blue color depict positive signs and there are for a particular classes of vehicles. The ordinary color of traffic sign is red and white as shown in TABLE 1.

TABLE I. Traffic Signs Categories



Category	Shape	Color	Example
Prohibitory	Circular	Red, Blue, White & Black	50
Mandatory	Rectangular & Circular	Blue, White & Black	
Danger	Triangular	Red, White & Black	

Traffic sign recognition is a requirement for getting a license, but most of us ignore while actually driving. Many people don't follow traffic rules like a "No U-turn" or "No free left sign", etc. Even if one is a sensitized road user and wishes to follow the rules, more often than not the poor visibility of such signage, makes it very difficult to do so.

This paper proposes a solution for detecting traffic signs with the help of a camera and give the appropriate voice alert messages to grab the driver's attention and thus minimize the chances of the accident.

The You Only Look Once (YOLO) method, which is a specialised Single Shot Detector (SSD) technique, is the basis for the solution proposed in this paper. The YOLO method is unique as it processes the entire image using a single neural network. The picture is partitioned into regions, with each region's bounding box and probability determined. YOLO is a SSD that ensures that a picture's bounding box and class are predicted in a single pass. Because of these characteristics, the YOLO technique is one of the fastest object recognition algorithms, hundreds of times faster than Fast R-CNN. The YOLO algorithm suffers from an increase in localization errors [4].

The following is the outline of the paper. Section II provides a synopsis of the relevant material. The method for recognising traffic signs is described in Section III. Section IV explains the training method for the dataset created by us as well as the proposed solution, while Section V contains the evaluations and conclusion are found in Section VI.

II. Related Work

Traffic signs have different properties in different parts of the world. Some road signs are country-specific, having similar meaning but an extremely different design. Furthermore, some countries have their own traffic signs, such as "Hand Cart Prohibited" in India that reflect a situation specific to that country.

Because traffic sign identification is a crucial subject, there are numerous traffic sign datasets available, such as the Belgian Traffic Sign Image Dataset [5] and German Traffic Sign Recognition Benchmark (GTSRB) [6]. This paper focuses on Indian road signs, and because there was no existing dataset for Indian traffic signs, we created our own.

A. *Identification of Traffic Signage*

Over the last decade, the computer vision and machine learning community members are working to solve the challenge of autonomous traffic sign recognition and identification. Traffic sign recognition is problematic because of non-uniform scene lighting, smearing caused by the vehicle-mounted camera moving in relation to traffic signs, and traffic sign blockage caused by other automobiles, trees, and other barriers

Y Yang and others [7] propose traffic sign identification based on extracting sign ideas with a colour probability model and a Histogram of Oriented Gradients (HOG). Manual features such as HOG, on the other hand, fail because to the mentioned difficulties.

Convolutional Neural Networks (CNNs) have recently been shown to be more successful at identifying a range of objects. Computer vision research for intelligent transportation systems is following suit, with many of these studies finding practical applications in increased driver assistance and self-driving automobiles. Several academics have used CNN-based image recognition techniques to try to tackle the challenge of traffic sign identification.

For traffic sign identification, Y Zhu et al. [8] suggested a solution based on deep learning components. To develop proposals, it employed a Fully Convolutional Network (FCN), and to categorise characters, it used CNN. Then, Zhu et al. [9] introduce a traffic sign detection system that uses an end-to-end multi-class CNN to identify and categorise traffic signs concurrently. They've also produced a dataset for performance analysis called the Tsinghua Tencent 100K dataset. They also demonstrated a traffic sign identification network using a single classifier, as well as a separate classifier for classifying traffic signs that have been detected.

Zhu et al. investigate the performance of Fast-RCNN [10] for traffic sign recognition. Peng [11] investigated the use of Faster R-CNN for traffic sign recognition. In comparison to prior research, this technique showed promise, but it was unable to attain consistent accuracy and detection speed. Another actual traffic sign recognition system built on Faster R-CNN [12] and adopting the Mobilenet [13] architecture recently developed by Li and Wang [14] may be proven, achieving the requisite real-time robustness and precision for situations such as driverless cars. Based on the German Traffic Sign Detection Benchmark (GTSDB) dataset outcomes. This methodology also introduced a classifying model that uses an asymmetrical kernel to categorise characters into 43 classes.

B. *Categorization of Signages*

In traffic sign classification, CNN-based classification methods trained on the GTSRB dataset might obtain better accuracy. Ciregan et al. [15] and Sermanet et al. [16] developed classifiers based on Multi Column Deep Neural Network (MCDNN) and Multiscale CNN could obtain precisions of 99.17 percent and 99.65 percent, respectively. On the other hand, such networks are enormous and so must understand a significant set of parameters. T L Yuan [17] attained 99.59 percent accuracy using Spatial Transformer Networks (STN).

When tested on the GTSRB dataset, the CNN-based classifier network [14] with the asymmetric kernel suggested by Li and Wang achieved 99.66 percent accuracy. Modern traffic sign recognition systems have been shown to be inferior to the combination of FRCNN-based detectors and CNN-based classification methods. For the proposed traffic sign recognition pipeline, the CNN-based classifier with the asymmetric kernel presented in [14] is employed as the classifier.

C. *Motivation*

Two stage detectors like Faster R-CNN have been studied well for the traffic sign detection problem. Single state detectors such as YOLO suffer from less accuracy and have difficulty in detecting small objects. Traffic signs which are far from the vehicle would appear small in the image and hence single stage detectors are not considered suitable for this. The recent advancements made with respect to multi scale detection in single stage detectors, the small object detection issues have been tackled to a large extent. This makes it worth studying the performance of the new single stage detectors for traffic sign detection.

After reviewing previous work on this issue, we found that voice alert messages when the traffic sign was recognized are missing. The system itself may distract the driver who has to read text message displayed by the system.

This paper presents our project where we have used the new YOLOv5 [18] network as the detector for the traffic sign recognition system and a new improvement in this system is a voice module that issues a voice warning when a traffic sign is detected.

III. Approach

Based on YOLOv5 and voice translation, this section provides a method for recognising traffic signs. A YOLOv5 detector has been trained to recognise candidate traffic signs and offer its speech translation, as shown in the traffic sign recognition flowchart in Fig. 1. The model was trained to recognise potential traffic signage corresponding to any one of the 75 classes in our dataset.

The footage collected by the camera will be sent into our traffic sign recognition algorithm, which will pre-process it and, if any traffic signs are recognised, extract the frame and categorise it according to its class. Later, our speech translation module will play an alert message for the driver.

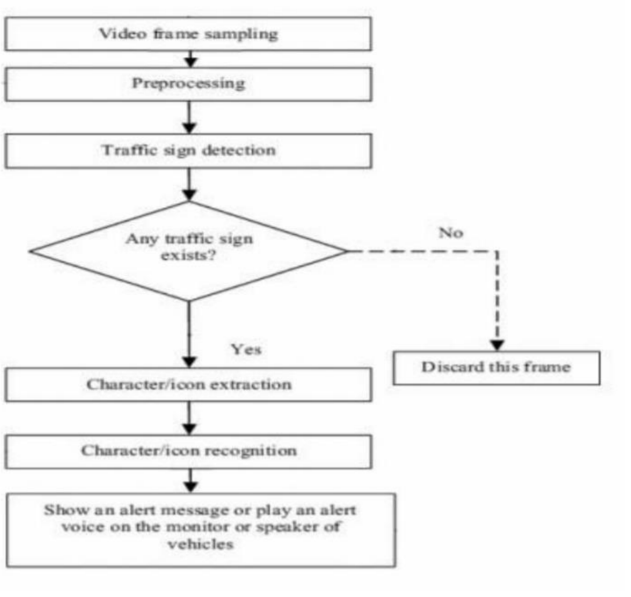

Video frame sampling
Preprocessing
Traffic sign detection
Any traffic sign exists?
No
Yes
Discard this frame
Character/icon extraction
Character/icon recognition
Show an alert message or play an alert voice on the monitor or speaker of vehicles

Fig. 1. Flowchart of Traffic Sign Recognition Model.

A. *Traffic Sign Identification Using YOLOv5*

Yolov5 is based on Yolov1 through Yolov4. Yolo is a cutting-edge real-time object detection model. It was able to attain the best results using officially recognized object recognition datasets, Pascal VOC (Visual Object Class) [20] and Microsoft COCO (Common Objects in Context) [21], via continual development. A comparison of Microsoft COCO object identification with multiple YOLO model versions [22] is shown in Fig. 2.

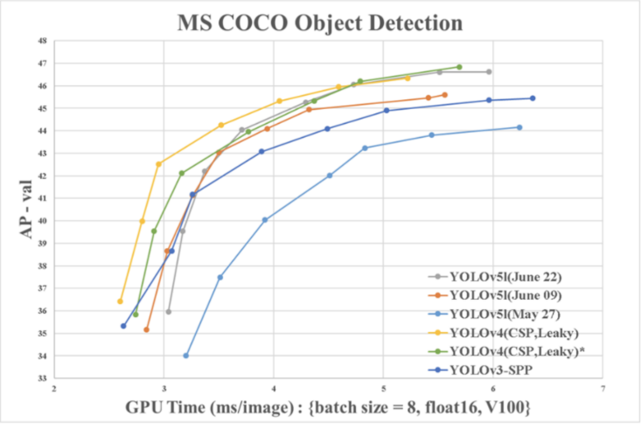


Fig. 2. Comparison of MS COCO Object Detection against different YOLO version

Fig. 3 shows the Yolov5 network architecture. There are three reasons why we chose Yolov5 as our detection model. First, Yolov5 integrated the Cross-Stage Partial Network (CSPNet) [23] into the Dark Web, creating CSP Darknet as its backbone. The problem of gradient information repeating over prolonged foundations by incorporating gradient modifications into feature maps, which minimizes parameters of the model and FLOPS (floating point operations per second) is overcame by CSPNet. As illustrated in Fig. 4 [26], CSPNet segregates the base layer's feature map into two parts: one section passes via a dense block and a transitioning layer, while the other half is combined with the transmission feature map of other stage. Fig. 4 depicts the CSPDenseNet's flow. This not only assures inference precision and agility, but it also minimizes the model complexity. Detection speed and accuracy are essential for traffic sign detection, and compact model size also determines the conclusion efficiency of resource-poor edge devices.

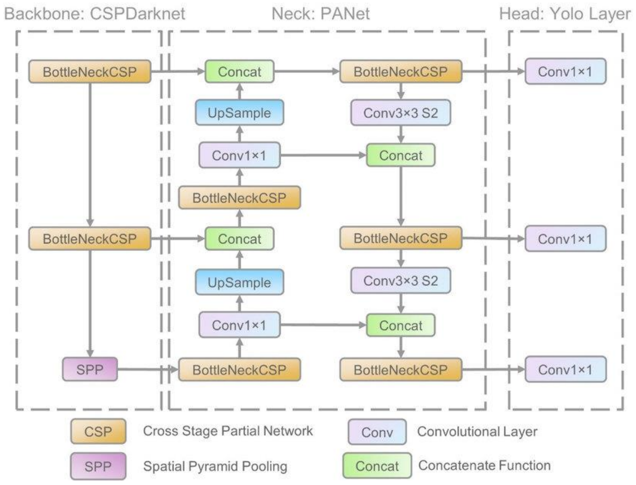


Fig. 3. Yolov5's network architecture It is divided into three sections: (1) CSPDarknet as the core, (2) PANet as neck, and (3) Yolo Layer as the head. The data is initially fed to CSPDarknet for extraction of features before being passed into PANet for feature fusion. Finally, the Yolo Layer produces detection statistics (class, location, score, size).

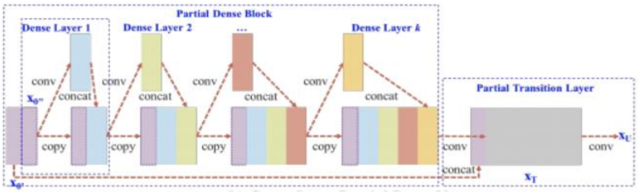


Fig. 4. Cross Stage Partial DenseNet (CSPDenseNet)

Yolov5 then employed a bottleneck called the Path Aggregation Network (PANet) [21] to improve information flow. PANet makes use of a novel Functional Pyramid Network (FPN) topology that has a better bottom-up approach for low-level feature propagation. Adaptive feature pooling, which connects the feature grid to all feature levels, is used at the same time to convey important information for each feature level straight to the following subnetworks. At the lowest levels, PANet enhances the utilisation of precise localization signals. As an outcome, the precision of the object's position will be increased. The FPN, PANet, NAS-FPN, and BiFPN network architectures are shown in Fig. 5.

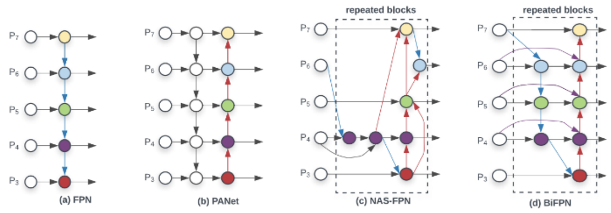


Fig. 5. Functionality of network structure – (a) FPN [24] exposes a top-down roadmap to fuse multi-scale functionalities from stage 3 to 7 (P3 - P7); (b) PANet [25] includes an extra bottom-up route on top of FPN; (c) NAS-FPN [26] utilises neural architecture search to identify an unusual feature topology of the network and then recurrently refers the same block; and (d) is BiFPN with greater accuracy and performance trade-offs.

Third, the Yolov5 head, or Yolo layer, generates feature maps of three different sizes (18x18, 36x36, 72x72) for multiscale prediction [27] and makes the model smaller. Make it a medium or large object. Multiscale detection ensures that the model can keep up with changes in size.

B. Pre-processing of Bounding Box

Boundary Boxes Pre-processing retrieves and prepares eligible traffic sign boxes for categorization. To guarantee that the traffic sign is completely covered by the area, the centre of the regressed bounding box is identified and the box is expanded by 25% to compensate for any regression errors. Before being transmitted to the classification system for traffic sign identification, the expanded boxes are trimmed and shrunk to 36 x 36 size.

C. Classifier for Traffic Signage

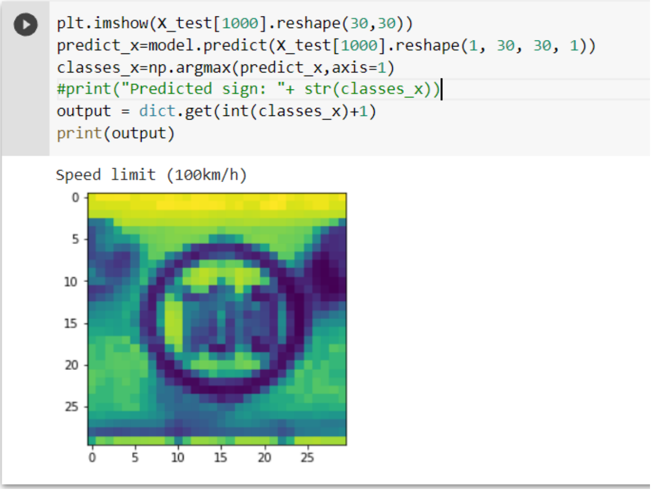
With a mAP of 36.7, YOLOv5s features 224 layers and 7.2 million trainable variables and is as rapid as 2 milliseconds (FLOPs or floating-point operations numbering roughly 17 billion). The network architecture introduced in [23] has been used to develop a rapid and precise traffic sign classifier. An n x n convolution is substituted with a n x 1 convolution accompanied by a 1 x n convolution in this model, minimizing the likelihood of convolution operations as well as the network parameters. As a consequence, computational expenses are decreased while performance is increased.

Except for the last dense layer, all layers are followed by Normalization of Batches and Leaky ReLU Layers. An inception module is provided by the sixth layer in which kernels of varied sizes collect information from the preceding layer's feature map output. To merge the feature maps, the inception layer's output feature maps are concatenated. To ensure that the final steps are always active Dropout layers are used. As the final layer, a fully-connected layer with an output size of 75 and Softmax activation is employed to detect the 75 traffic sign classes.

D. Speech Translation of Detected Signage

After recognising the traffic sign, YOLOv5 extracts the label, which is then translated into an audio file (mp3) by gTTS (Google Text-to-Speech) translator [28], a Python library and CLI utility that interfaces with Google Translate's text-to-speech API, and provided to the driver as an alert message.

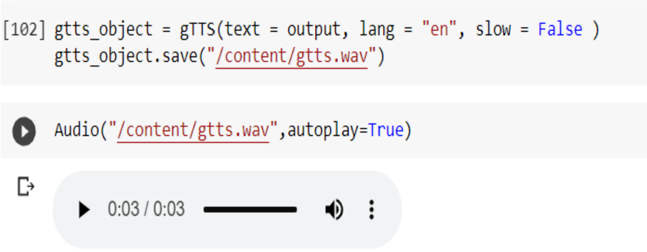
Fig. 6 illustrates an image of written output for the provided sign to our system, and Fig. 7 presents an example of audio output.



```
plt.imshow(X_test[1000].reshape(30,30))
predict_x=model.predict(X_test[1000].reshape(1, 30, 30, 1))
classes_x=np.argmax(predict_x,axis=1)
#print("Predicted sign: "+ str(classes_x))
output = dict.get(int(classes_x)+1)
print(output)
```

```
Speed limit (100km/h)
```

Fig. 6. Text output of the traffic sign given as input (Speed Limit 100km/h).



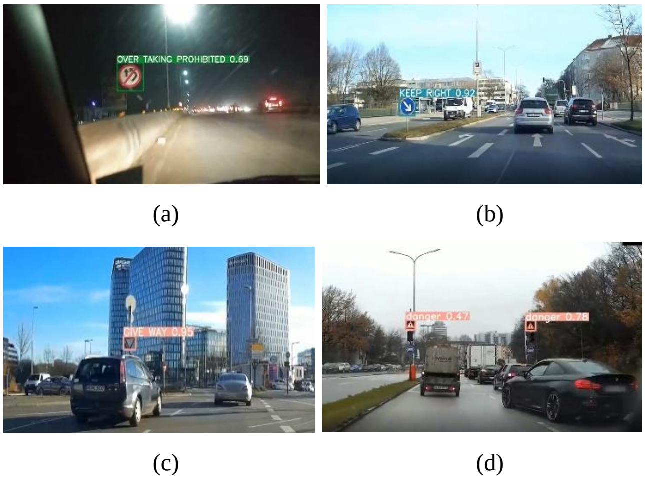
```
[102] gtts_object = gTTS(text = output, lang = "en", slow = False )
      gtts_object.save("/content/gtts.wav")
```

```
Audio("/content/gtts.wav",autoplay=True)
```

Fig. 7. Audio output of the traffic sign given as input (Speed Limit 100km/h) using gTTS [28].

IV. The Proposed System

By capturing urban traffic and integrating photographs with annotated traffic signs, we created a new dataset for our project, Traffic Sign Recognition System using YOLOv5 (TSRSY). Images were acquired in a range of weather conditions, including sunlight, clouds, drizzle, and even night, allowing the recommended solution to be evaluated in a variety of scenarios and test against the YOLOv5 model, which was trained on our data. The solution for identifying this traffic signage, as well as the training and testing technique, are also discussed. For training and testing, we used Python3 Google compute engine backend as GPU and 8GB of GPU RAM.



(a) (b)

(c) (d)

Fig. 8. Instances of image sequences taken in various weather: (a) at night, (b) gloomy, (c) sunny, and (d) rain.

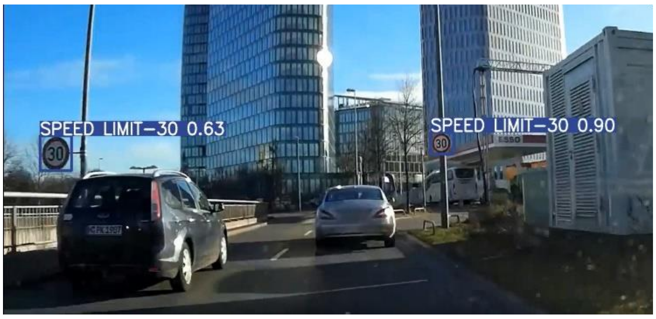


Fig. 9. Traffic sign detection illustration from the video sequence.

By combining the components, a comprehensive traffic sign identification pipeline employing a YOLOv5 based detector, bounding box pre-processing, and classification model with speech translation has been created. Fig. 8 shows a range of detection outcomes in varying weather conditions, and Fig. 9 shows an example of our approach detecting several signs.

A. Our Dataset

Traffic signs that have a direct influence on the vehicle while driving were chosen for this task. When a vehicle notices any of these indicators, he or she must respond quickly on the road (for example, turn left or right, stop moving, or change the desired speed). This approach is appropriate for various types of traffic signs, in Mumbai and its surrounding areas. The regularity with which these traffic signals were placed beside the road was also a significant consideration while selecting traffic signs for this approach. Fig. 10 displays few of the traffic signs that the proposed system should be able to recognise from our dataset.

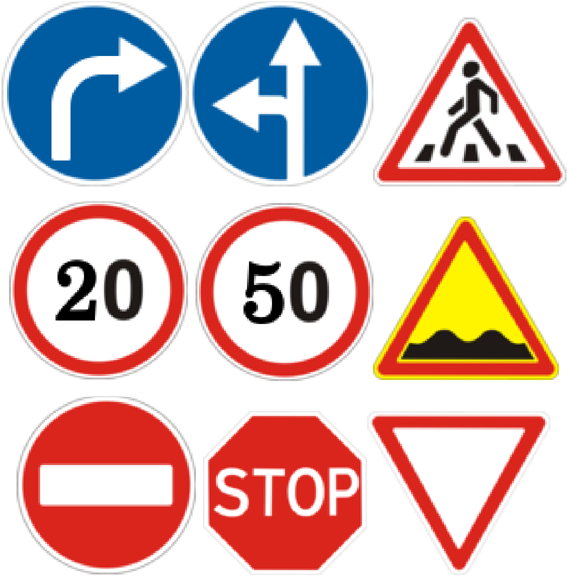


Fig. 10. Traffic Sign Images included in our dataset.

Because not every traffic sign is evenly frequent in road traffic, such as, "pedestrian crossing signs" appear considerably more frequently than "100 km/hr speed limit signs", the lighting and resolution of some video frames were augmented to enhance them. Those video frames that rotate but the message of the sign does not change were mirrored to capture more images without further recording. "Turn left" and "Turn right" video frames were likewise mirrored. "Turn left" becomes "Turn right," and the other way around.

Each connected traffic sign was manually labelled after gathering all of the requisite image sequences and data augmentation, producing a combination of 7336 images. Each and every traffic sign used in this project was labelled using open source photos and video labeller, named Make Sense [19].

TABLE II. LIST OF HANDFULS OF THE TRAFFIC SIGN LABELS IN OUR DATASET, ALONG WITH THE NUMBER OF IMAGES USED FOR TRAINING AND TESTING.

Traffic sign labels	***Number of Labels***
Compulsory Ahead	175
Compulsory Ahead or Turn Right	111
Compulsory Right Turn	123
Compulsory Turn Left	111
Pedestrian Crossing	128
Slippery Road	152
Speed Limit – 20	210
Speed Limit – 50	324
School Ahead	127
Left Hand Curve	83
Hump or Rough Road	100
Narrow Road Ahead	65
Give Way	146
Go slow	178
No Entry	216
Stop	100

B. Evalution Criteria

Intersection over Union (IoU) between the ground truth box and the predicted bounding box > 0.5 is considered as a positive detection. Mean Average Precision (mAP) is used for evaluating the detector performance. Speed of detection per image, measured in milliseconds (or number of frames per second) is also used for evaluating the detector. These performance measures are used for comparing the performance of the proposed detector with YOLOv5 based detector. We have using mAP as the measure for evaluating the classifier performance and confusion matrix-based evaluation for analyzing the mAP.

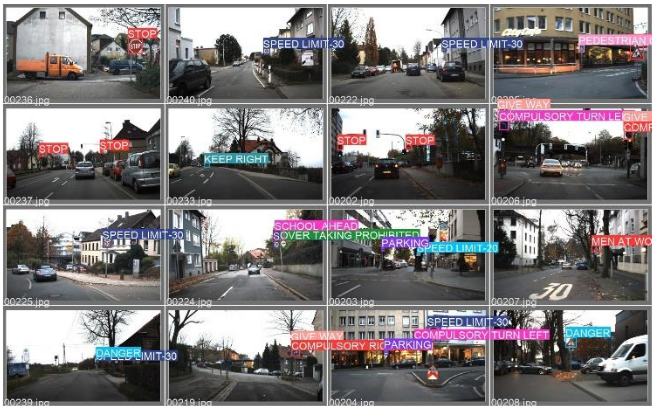


Fig. 11. Instances of image sequences during validation.

C. Training for Classification Model

Our dataset with 6713 training images was used to train the classification model as well as we included 623 testing images. The model was trained for 20 epochs, under yolom version with each label having several images created by different changes such as tilting, cutting, resizing, and spinning.

V. Evaluation

The detector's performance is assessed using Mean Average Precision (mAP) and on our dataset, the traffic sign detector achieved a mAP of 70 percent.

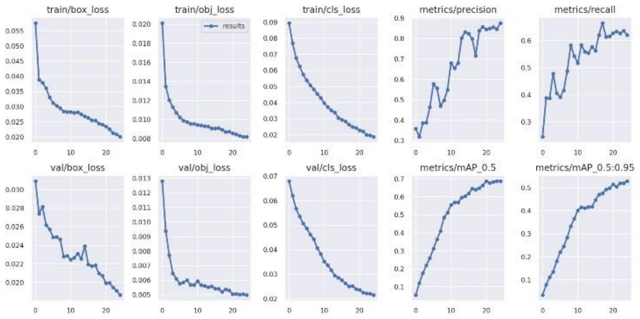


Fig. 12. Graph Instances of loss while training.

There were a few false positive detections as well, when some marketing signs that looked suspiciously like traffic signage were captured.

VI. Conclusion

TSRSY proposes a technique for recognizing a subset of traffic signs. For training and testing purposes, a subset of 75 traffic signs was chosen, and a dataset of 7336 photos tagged with 6713 of these traffic signs was constructed. Images are obtained from an urban driving environment in Mumbai, India, under variety of weather situations. The YOLOv5 algorithm is used to recognize traffic signage from the image sequence using the front-facing camera. This allows the model to achieve a mAP value of 70%.

This system can be extended to cover all the traffic signs that are followed in any region of the world. We intend to create a similar type of detector in the future to be used by self-driving cars.

Acknowledgment

Golnaz Ghiasi, Tsung-Yi Lin, and Ruoming Peng deserve special gratitude for their contributions to architecture and discussions of YOLO which helped us to develop the detector required for our project. We also thank Google Collab team for providing the execution environment. We value the open source community's efforts.

References

[1] Government of India, "Road accidents in India – 2019", India:Minister of Road Transport & Highways Transports Research Wing, 2019.

[2] TOI. (2022, Jan. 15). Road crashes rise by 10% in 2021, over 400 died in accidents [Online].

Available:https://timesofindia.indiatimes.com/city/gurgaon/road-crashes-rise-by-10-in-2021-over-400-died-in-accidents-in-city/articleshow/88906699.cms

[3] IRC 067, "General - Placement and Operation of Road Signs," in Code of practice for road signs, Third revision, India:Indian Roads Congress, 2012, pp. 13.

[4] J. Redmon, "YOLO: Real-Time Object Detection" pjreddie.com. [Online]. Available: https://pjreddie.com/darknet/yolo/

[5] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German traffic sign detection benchmark," in Proc. IEEE Int. Joint Conf. Neural Netw., pp. 1–8, Aug. 2013.

[6] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in Proc. IEEE Int. Joint Conf. Neural Netw., pp. 1453–1460, Aug. 2011.

[7] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," IEEE Trans. Intell. Transp. Syst., vol. 17, no. 7, pp. 2022–2031, Jul. 2016.

[8] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," Neurocomputing, vol. 214, pp. 758–766, Nov. 2016.

[9] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in Proc. IEEE Conf. Comput.Vis. Pattern Recognit., pp. 2110–2118, Jun. 2016.

[10] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis., pp. 1440–1448, Dec. 2015.

[11] E. Peng, F. Chen, and X. Song, "Traffic sign detection with convolutional neural networks," in Proc. Int. Conf. Cogn. Syst. Signal Process., Singapore: Springer, pp. 214–224, 2016.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[13] A. G. Howard et al. (2017). "MobileNets: Efficient convolutional neural networks for mobile vision applications." [Online]. Available: https://arxiv.org/abs/1704.04861

[14] Jia Li and Zengfu Wang, "Real-Time Traffic Sign Recognition Based on Efficient CNNs in the Wild", IEEE Trans. Intell. Transp. Syst., vol. 20, no. 3, pp. 975–984, Mar. 2019.

[15] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3642–3649, Jun. 2012.

[16] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in Proc. IEEE Int. Joint Conf. Neural Netw., pp. 2809–2813, Aug. 2011

[17] T. L. Yuan. GTSRB_Keras_STN. Accessed: Nov. 1, 2017. [Online]

Available: https://github.com/hello2all/GTSRB_Keras_STN

[18] Xingkui Zhu, Shuchang Lyu, Xu Wang, Qi Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios", [Online] Available: https://arxiv.org/abs/2108.11539, 2021

[19] Piotr Skalski, Make Sense, 2019. Available: https://github.com/SkalskiP/make-sense/

[20] Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vis. 2015, 111, 98–136, doi:10.1007/s11263-014-0733-5.

[21] Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the 13th European Conference on Computer Cision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.

[22] Jacob Solawetz (2020, Jun. 29). YOLOv5 New Version - Improvements And Evaluation [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/

[23] Wang, C.Y.; Mark Liao, H.Y.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2020), Washington, DC, USA, 14–19 June 2020; pp. 390–391.

[24] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. CVPR, 2017.

[25] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. CVPR, 2018.

[26] Golnaz Ghiasi, Tsung-Yi Lin, Ruoming Pang, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. CVPR, 2019.

[27] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv: 1804.02767.

[28] Pierre-Nick Durette, gTTS Documentation. 2022.