

Indian Traffic Sign Recognition as a Two Stag Approach using YOLOv4

Muhammad Jaseer KP¹ and Sruthy Manmadhan²

¹⁻²NSS College of Engineering, Palakkad, India

Email: iamjaseer20@gmail.com, sruthym.88@gmail.com

Abstract—In real-world applications such as autonomous driving and driver safety, Traffic Sign Recognition (TSR) systems play a critical role. In a country like India, where new technology is quickly adopted, automatic vehicles will adapt quickly. Even significant corporate in India Tata Motors is working on bringing self-driving cars to India. Tesla, the biggest developers of self-driven cars is granted the right to sell cars in India. According to reports, Tesla will be selling cars in India by the end of 2022. The state-of-the-art of object detection models such as YOLOv3, Tiny-YOLOv4 and YOLOv4 is examined in this work. Transfer learning is used to tailor these models to the domain of traffic sign detection. As we are focusing on India, we finetune these state-of-art object detection model trained on COCO dataset using Indian Traffic Sign Detection (ITSD) Dataset. There is no such database, hence this work contribute a new Indian Traffic Sign Detection dataset obtained from random Indian road scenes. The task of recognition is made up of two parts. Detection and classification are two steps in the process. In this work, detection module is implemented using three different YOLO (You Look Only Once) versions and classification using custom Convolutional Neural Network (CNN) model. The German Traffic Sign Detection Benchmark (GTSDB) dataset is also utilized to evaluate the system because it is the most widely used dataset. With a threshold of 0.5, the detectors made with YOLOv4, YOLOv3, and Tiny-YOLOv4 were tested on the ITSD dataset and achieved mean average precision (mAP) of 99.35%, 97.20%, and 80.89%, respectively.

Index Terms— Traffic Sign Recognition, YOLOv3, YOLOv4, Tiny-YOLOv4, CNN.

I. INTRODUCTION

Advanced Driving Assistance Systems (ADAS) technology is advancing at a rapid pace. Object detection plays a crucial role in the approaching trend of self-governing cars in order to continuously enhancing Intelligent Driving and Traffic Safety. TSD (Traffic Sign Detection) can help drivers become more aware of their surroundings and recognize signals in the distance, particularly in bad weather or at night when vision is limited. A good detecting system is required to make it all possible. Many researchers have suggested somewhat realistic traffic sign identification systems, which often rely on traffic sign color, shape, and machine learning. Color-based detection algorithms commonly segment the image using color space. It can delete countless non-interested areas as long as there is less calculation, resulting in good real-time performance. Several research have been carried out to solve the issue of traffic sign recognition. Even while several existing techniques have shown promising results on a variety of benchmark datasets, the majority of them fail miserably in bad situations such as motion blur, poor illumination, foggy weather, and so on. The GTSDB database is the most commonly utilized for traffic sign detection. This paper uses three distinct YOLO models to implement the system on both

the GTSDDB and the newly constructed Indian Traffic Sign Detection (ITSD) datasets, and it evaluates the performance of each dataset using all three models. In the classification phase, as we are focusing on Indian traffic signs and there is no Indian traffic sign recognition dataset available, a new recognition dataset with 5 classes is created complete system pipeline and also German Traffic Sign Recognition Benchmark (GTSRB) dataset is used to complete the system using GTSDDB dataset.

II. RELATED WORKS

A. Traffic Sign Detection

Traffic sign domain is one of the most researched areas in object detection. Even in recent years, there are many studies conducted on real time traffic sign recognition. YOLO is one of the most advanced object detection algorithms available now. In [1], YOLOv3 and YOLOv4 were compared using the traffic sign data generated using different Generative Adversial Network (GAN). A cross layer connection is added to the YOLOv4 network in [2] to improve the YOLOv4 model to enhance feature extraction ability of the YOLOv4 network. YOLO models are capable of detecting and classifying traffic signs in one go, [3] and [4] used YOLOv4 network for both classification and detection. For improving the detection accuracy, in [5] to improve the accuracy of YOLOv3 network, DenseNet of YOLOv3 is replaced using ResNet. Detection of small traffic signs were addressed in [6], network pruning is introduced to minimize the size of network and network redundancy. Traffic sign recognition is always a challenging area because of the complexity of system to detect the traffic signs in different lighting, weather, etc. Addressing all the challenges at once is very difficult. In [7], they focused on detection small traffic signs using YOLOv3 network. A spacial pyramid pooling is introduced in the YOLOv3 network to enhance the feature extraction.

B. Traffic Sign Classification

Traffic sign classification complete the pipeline of a traffic sign recognition system. In [18], they proposed an approach based on the deep convolutional network for classifying traffic signs. In [9], they separate traffic signs into hierarchical structures based on the types of features, and then use a combination CNNs to adapt the hierarchical traffic signs, where the probability of superclass and subclass, the sign belongs to are determined using two CNNs and a simple network. GTSRB dataset with 43 classes is commonly used for classification process. In [10], to train the detection of traffic signs into super classes, a multi class Support Vector Machine (SVM) with Histogram Oriented Gradient (HOG) is used. Following detection, three CNN networks for three super classes are formed. Except for the number of nodes in the last full-connected layer, which equals the number of sub classes in each super class, all three CNNs have the same structure. In [11], a deep network is made up of spatial transformer layers, and a modified version of the inception module is used to capture both local and global features. The use of a spatial transformer layer makes this network more resistant to input image deformations including translation, rotation, and scaling.

III. PROPOSED SYSTEM

This section discusses about the modules included in the proposed system, different configuration parameters and how the system works. This work proposes a TSR system that can be used to detect and classify traffic signs from real traffic scene images using YOLO and CNN. The system design of proposed system is shown in Fig. 1. Two different datasets are used for implementing detector module. One newly created ITSD dataset with one class and GTSDDB dataset with 4 classes. Both datasets are trained on three different YOLO versions and performances are evaluated. In classification phase, GTSRB dataset is used.

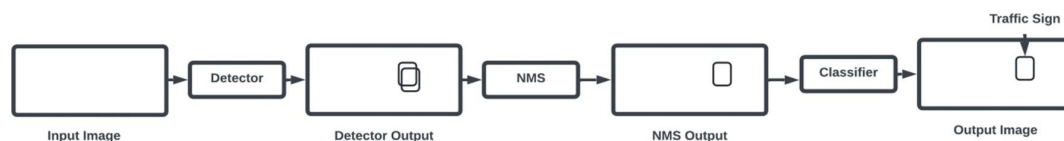


Figure 1. TSR system pipeline

The traffic scene images are given as input to the system. At first the YOLO based detector module detect all the traffic signs present in the image and detect bounding boxes. After detecting bounding boxes non maximum suppression is performed on detected bounding boxes to find bounding boxes with IOU value greater than

threshold. Bounding boxes left after non-maximum suppression is cut into fragments and passed to CNN based classifier to classify these signs to predefined classes.

A. Input

Input of the system can be images, videos and real time camera frames. The system is developed based on images and can be used in loop to implement the system in video and camera inputs. In video and real time inputs, each frame is detected and passed to the system and finally combine all the frames to produce the output.

B. Detector

Detectors play the major role in traffic sign recognition system. The first process of TSR is localizing where the traffic sign is in the frame. In this work, we used three different versions of YOLO algorithms to implement the detector module. YOLOv3, YOLOv4 and Tiny-YOLOv4 is used for implementing the detector module on both newly created ITSD dataset and GTSDDB dataset. There are few crucial files that must be handled when training any YOLO model. Each YOLO model has its own set of configuration files that were trained using the COCO dataset [12], which has 80 classes. To finetune the model to our domain, these configuration files must be updated. Depending on the dataset, various critical parameters must be replaced.

In YOLOv3, the network size becomes 416*416 instead of 608*608 in YOLOv4. In Tiny-YOLOv4, the size of network is 416*416 and the learning rate changes from .001 to .00261 and also there is only two YOLO layers in Tiny-YOLOv4. Changes such as max batches, steps, and filters should be made in accordance with the number of classes the detector has been trained on. Max batches = (# of classes * 2000) is used to do updates. However, max batches should not be less than 6000; filters = (# of classes + 5) * 3 and steps = (80% of max batches), (90% of max batches). Filters and classes should be replaced in all the convolutional layers just above the YOLO layers.

C. Non Maximum Suppression

NMS (Non Maximum Suppression) [13] is a computer vision approach that is employed in many tasks. It is a set of methods for selecting one entity from a large number of overlapping entities, primarily Bounding Boxes. To get the desired results, we can choose the criteria for selection. Most of the time, the requirements are a probability number and an overlap measure, such as Intersection Over Union (IoU). The IOU metric is used to evaluate an object detector's accuracy on a given dataset.

NMS filters the bounding boxes according to the IoU threshold. Default IoU threshold is 50%. The output of NMS network is the input frame with the best bounding box with highest IoU value. This detected traffic sign is cut into fragments and passed to classifier module for recognizing traffic sign class. Fig. 2 shows how IoU is calculated using bounding boxes.

D. Classifier

A convolutional neural network (CNN) is a form of artificial neural network that is specifically intended to process pixel input and is used in image recognition and processing [14]. CNNs are sophisticated image processing AI systems that employ deep learning to accomplish both generative and descriptive tasks. CNN training is done using GTSDDB dataset with 43 classes. As there is no recognition dataset available for India, for completing the Indian recognition system, we create a new Indian recognition dataset with 5 classes for completing the system pipeline. The future work, the recognition dataset including all variety of Indian traffic signs will be created with more complex images.

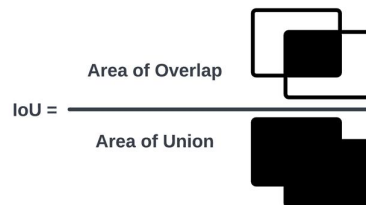


Figure 2. IoU Calculation

The model architecture consists of two conv2D layer with ReLU activation function, kernel size 3 and filters 32 and 64 respectively. Each cov2D layer is followed by maxpool2D layers with kernel size 2. To tackle overfitting Dropout layers are used after each maxpool2D layer with rate .25 and .5 respectively. Finally, two Dense layers

with units 512 and 43 is used respectively. Final Dense layer is output layer with the number of classes used for classification.

E. Output

Output of the system is the input frame with traffic sign detected labelled with the confidence score of the class. For video and real time inputs, the outputs are a video with detected traffic signs in every frame of the input.

IV. EXPERIMENTAL SETUP

The system is trained on Google Colab Pro. Colab Pro provide high speed 25GB of RAM. Tesla T4 GPU is used in this work. Also, Colab Pro provide over 100+GB of disk space.

A. Dataset

For the detection module, a new Indian Traffic Sign Detection (ITSD) dataset is generated. For evaluating the performance of the newly produced dataset, the most commonly used GTSDDB dataset is also employed. There are 1264 images in total. All of the images are grouped together in a single class called Traffic Sign. Morning, Afternoon, and Night are the three times when images are collected. Motion blur, car light reflection, and other difficult conditions were also addressed.

From the dataset, 10% is used for testing and rest for training. Dataset is trained on three different YOLO versions - YOLOv3, YOLOv4 and Tiny-YOLOv4. GTSDDB dataset has 4 different classes - Prohibitory, Mandatory, Warning and Others. GTSDDB is implemented using 4 classes to study about the improvement of training time compared to number of classes.

All the traffic sign images in the dataset should be annotated into YOLO format for training the detector module. LabelImg tool used for annotating dataset images to YOLO format. After annotation, each image in the dataset will have a text file having data in the format- ClassID, centerX, centerY, width and height.

There is no recognition dataset available for India, German Traffic Sign Recognition Dataset (GTSRB) is used to complete the TSR pipeline.

V. RESULTS AND DISCUSSIONS

In this section, results of detector models, comparison of different YOLO versions and results of classifier are discussed.

A. Detector Results

The performance of detection module using YOLO is evaluated using Mean Average Precision (mAP). The newly created ITSD dataset and GTSDDB test dataset is used for testing detector. The detector only detects or localize the traffic sign from traffic scenes. The detector created using YOLOv4, YOLOv3 and Tiny-YOLOv4 was tested on ITSD dataset with a threshold of 0.5 and achieved a mean average precision of 99.35%, 97.20% and 80.89% respectively. IoU used for the detection purpose. Threshold value for IOU to find best bounding boxes.

The detector created using YOLOv4, YOLOv3 and Tiny-YOLOv4 was tested on GTSDDB dataset with a threshold of 0.5 and achieved a mean average precision of 99.96%, 94.82% and 61.91% respectively. The mAP values from different iterations are shown in Table 2. YOLOv4 model performed better in all three models with highest mAP value.

TABLE II. MEAN AVERAGE PRECISION ON ITSD

Iterations	YOLOv3	Tiny-YOLOv4	YOLOv4
1000	71.44	24.47	90.46
2000	95.18	66.81	99.35
3000	92.70	66.02	98.77
4000	97.20	69.79	99.30
5000	96.21	78.76	99.28
6000	96.20	80.89	99.25

B. Models Comparison

Mean Average Precision: From the table II, it is clearly visible that, using ITSD dataset YOLOv4 achieved best mAP value after 2000 iterations, YOLOv3 and Tiny-YOLOv4 achieved higher mAP after 4000 and 6000 iterations respectively. By training the models on GTSDDB dataset, YOLOv4 achieved higher mAP at 8000 iterations, YOLOv3 and Tiny-YOLOv4 achieved higher mAP after 7000 iterations.



Figure 4. Outputs of YOLO Detector

Precision - Recall - IOU Average - F1-Score: Along with mean average precision, other metrics used for evaluating an object detection model is precision, recall, F1-score and IOU average.

In GTSDb and ITSD dataset, YOLOv3 achieved highest precision of 96% and 99.24%. Highest Recall of 99.24% was achieved by YOLOv3 on ITSD and 99.24% by YOLOv4 on GTSDb dataset. IOU value is higher in YOLOv4 on GTSDb and YOLOv3 on ITSD dataset. F1-score is also higher in YOLOv4 on GTSDb dataset and YOLOv3 on ITSD dataset. Table III shows the values obtained for different models.

Frames Per Second (FPS): In this section, speed of processing, that is FPS rate is discussed. FPS is the major metric used while developing a real time application. If FPS rate is very low, then the model cannot be used for real time applications. FPS rate obtained in both dataset is higher with the Tiny-YOLOv4 model and the value is 42 and 38 in ITSD and GTSDb dataset respectively. Both YOLOv3 and YOLOv4 model achieved FPS rate of 32 in ITSD dataset and 25 frames in GTSDb dataset.

C. Classifier Results

Classifier performance is evaluated using accuracy as evaluation metrics. The classifier module is trained on GTSRB dataset with 43 classes and also with a newly created Indian Traffic Sign Recognition dataset with 5 classes - Speed Limit(20km), Speed Limit(30km), Danger Curve to Left, Danger Curve to Right and Pedestrian Crossing. The CNN model trained using GTSRB were able to achieve an accuracy of 98.9% on training data and 92.4% on testing data. The newly created Indian recognition dataset with 5 classes were able to achieve 98.6% on training data and 95.6% on test dataset.

D. Dataset Comparison

Two datasets are used to train the detector system, one German and other a newly created Indian dataset. German dataset is commonly used and standard dataset. Even though in some situations like motion blur and reflection from headlight situations ITSD dataset detected the traffic signs better. From the images above, in right reflected traffic signs ITSD dataset performed better than GTSDb dataset and also detected all 4 traffic signs in the input. Whereas the model using GTSDb dataset was only able to detect the sign speed limit (40km), which have a decent lighting. On the other hand, in motion blur situation, the model using GTSDb gave a prediction score of 68% and ITSD dataset was able to achieve 98% accuracy.

TABLE III. PRECISION, RECALL, F1-SCORE AND IOU AVERAGE OF ITSD

Model	Precision	Recall	IOU	F1-Score
YOLOv4	99.16%	90.15%	83.95%	94.44%
YOLOv3	99.24%	99.24%	90.58%	99.24%
Tiny-YOLOv4	81.11%	55.30%	65.56%	65.76%

VI. CONCLUSIONS AND FUTURE WORKS

In this work, a real time Indian Traffic Sign Recognition system is developed using YOLOv4 detector and CNN classifier. As part of this a new Indian Traffic Sign Detection dataset is developed that is trained on more complex situational images and lighting condition than GTSDb dataset. ITSD dataset created have less classes than GTSDb dataset. Even though the ITSDb dataset performed better in headlight reflected and motion blur situations. YOLOv4 performed better in both the dataset and was able to achieve a mAP of 99.35% and 99.96% using ITSD and GTSDb dataset respectively. The CNN classifier is trained on GTSRB and newly created 5 class Indian recognition dataset and achieved an accuracy of 98.6%.

In the future, we develop the Indian recognition dataset and add more classes to both ITSD and recognition dataset. The ITSD dataset will be annotated with more classes other than single class. Create a recognition model using YOLO that can classify all the traffic signs with a better accuracy. Develop a hybrid dataset model using all the available detection datasets and that can be used in any part of the world to detect traffic signs.

REFERENCES

- [1] Dewi, Christine, Rung-Ching Chen, Yan-Ting Liu, Xiaoyi Jiang, and Kristoko Dwi Hartomo. "Yolo v4 for advanced traffic sign recognition with synthetic training data generated by various GAN." *IEEE Access* 9 (2021): 97228-97242.
- [2] Gan, Zhang, Li Wenju, Chu Wanghui, and Su Pan. "Traffic sign recognition based on improved YOLOv4." In *2021 6th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, vol. 6, pp. 51-54. IEEE, 2021.
- [3] Fernando, W. H. D., and S. Sotheeswaran. "Automatic road traffic signs detection and recognition using 'You Only Look Once' version 4 (YOLOv4)." In *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, vol. 4, pp. 38-43. IEEE, 2021.
- [4] Arief, Rezki Wulandari, Ingrid Nurtanio, and Faizal Arya Samman. "Traffic Signs Detection and Recognition System Using the YOLOv4 Algorithm." In *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, pp. 1-6. IEEE, 2021.
- [5] Fan, Wenshuo, Nanqiao Yi, and Yongzhong Hu. "A Traffic Sign Recognition Method Based on Improved YOLOv3." In *International Conference on Intelligent Automation and Soft Computing*, pp. 846- 853. Springer, Cham, 2021.
- [6] Wan, Jixiang, Wei Ding, Hanlin Zhu, Ming Xia, Zunkai Huang, Li Tian, Yongxin Zhu, and Hui Wang. "An efficient small traffic sign detection method based on yolov3." *Journal of Signal Processing Systems* 93, no. 8 (2021): 899-911.
- [7] Dewi, Christine, Rung-Ching Chen, Hui Yu, and Xiaoyi Jiang. "Robust detection method for improving small traffic sign recognition based on spatial pyramid pooling." *Journal of Ambient Intelligence and Humanized Computing* (2021): 1-18.
- [8] Mehta, Smit, Chirag Paunwala, and Bhaumik Vaidya. "CNN based traffic sign classification using Adam optimizer." In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 1293-1298. IEEE, 2019.
- [9] Chen, Lingying, Guanghui Zhao, Junwei Zhou, and Li Kuang. "Realtime traffic sign classification using combined convolutional neural networks." In *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 399-404. IEEE, 2017.
- [10] Yang, Yi, Hengliang Luo, Huarong Xu, and Fuchao Wu. "Towards real-time traffic sign detection and classification." *IEEE Transactions on Intelligent transportation systems* 17, no. 7 (2015): 2022-2031.
- [11] Haloi, Mrinal. "Traffic sign classification using deep inception based convolutional networks." *arXiv preprint arXiv:1511.02992* (2015).
- [12] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. "Microsoft 'coco: Common objects in context." In *European conference on computer vision*, pp. 740-755. Springer, Cham, 2014.
- [13] Neubeck, Alexander, and Luc Van Gool. "Efficient non-maximum suppression." In *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 850-855. IEEE, 2006.
- [14] Lo, Shih-Chung B., Heang-Ping Chan, Jyh-Shyan Lin, Huai Li, Matthew T. Freedman, and Seong K. Mun. "Artificial convolution neural network for medical image pattern recognition." *Neural networks* 8, no. 7-8 (1995): 1201-1214.