Real-Time Traffic Sign Recognition using YOLOv3 based Detector

Shehan P Rajendran*, Linu Shine†, Pradeep R‡, Sajith Vijayaraghavan§
Department of Electronics and Communication Engineering
College of Engineering Trivandrum
Trivandrum, India
Email: *shehanpr@gmail.com, †linu_shine@yahoo.com, ‡pradeep@cet.ac.in, §sajithvijayaraghavan@gmail.com

***Abstract*—Increase in the number of vehicles on road necessitates the use of automated systems for driver assistance. These systems form important components of self-driving vehicles also. Traffic Sign Recognition system is such an automated system which provides the required contextual awareness for the self-driving vehicle. CNN based methods like Faster R-CNN for object detection provide human level accuracy and real time performance and are proven successful in Traffic Sign Recognition systems [1]. Single stage detection systems such as YOLO [2] and SSD [3], despite offering state-of-the-art real-time detection speed, are not preferred for traffic sign detection problem due to its reduced accuracy and small object detection issues. The enhanced YOLO versions, YOLOv2 and YOLOv3 have shown promising results with respect to accuracy and speed required for object detection problems. YOLOv3 uses specialized network architecture inspired from feature pyramid network and has several design changes over previous versions to tackle the low accuracy and small object detection problems. In this paper, an approach for traffic sign recognition system based on YOLOv3 is presented with comparative analysis of its performance with Faster R-CNN based sign detector [1]. YOLOv3 forms the traffic sign detection network and a CNN-based classifier forms the traffic sign class recognizer. The network training and evaluation are done using the German Traffic Sign Detection Benchmark (GTSDB) [5] dataset and the classifier performance is verified using German Traffic Sign Recognition Benchmark (GTSRB) [6] dataset.**

Keywords— Traffic sign recognition, CNN, Faster R-CNN, YOLOv3

I. INTRODUCTION

Systems for assisting the drivers to avoid accidents are becoming more and more important as the number of vehicles on road is on an exponential increase. Advanced Driver Assistance Systems (ADAS) are being effectively used in automobiles for providing lane keep assistance, forward collision warning, pedestrian warning, driver drowsiness detection, traffic sign assist system etc. These form essential systems in autonomous cars for contextual awareness and road attribute mapping in order to control the vehicle motion trajectory. Traffic Sign Recognition (TSR) is the core component of traffic sign assist system for providing timely instructions and warnings to the drivers regarding traffic restrictions and information. In self driving cars, the inputs from traffic sign recognition system are used to make suitable decisions by the car, for example, to reduce speed or prepare for a detour etc.

Traffic sign recognition involves traffic sign detection and classification. Several studies have been conducted to address the traffic sign recognition problem. Even though some of the existing approaches have demonstrated good results on various benchmark datasets, most do not work very well in adverse conditions like motion blur, poor illumination, rainy or foggy weather, small sized signs etc. Recent studies use deep learning techniques, especially Convolutional Neural Networks (CNNs) for solving the traffic sign detection and classification problems. Majority of such studies use the German Traffic sign Detection Benchmark (GTSDB) [5] and German Traffic Sign Recognition Benchmark (GTSRB) [6] datasets for training and evaluation of their detection and classification networks. Improvements are made in the general CNN based object detection networks for traffic sign detection. A fine-tuned version of Faster-RCNN, a two-stage object detection network comprising of Region Proposal Network, bounding box regressor and classifier networks, has shown good accuracy and speed and it is found to be promising for real world problems like self-driving cars. Single stage detectors like You Only Look Once (YOLO) [2] offer real time speed in detection, however these suffer from foreground-background class imbalance problems and does not provide the required detection accuracy. Also, these detectors are not efficient in detecting small objects, which is critical for traffic sign recognition. The recent improvements made as part of YOLOv2 [7] and YOLOv3 [8] have tackled the problems to some extent.

In this paper, YOLOv3 object detection network is tuned and used for the traffic sign detection purpose. Very good detection result with real time performance on the GTSDB database is observed using this approach. The CNN based traffic sign classifier proposed by Li and Wang [1] is used for classification of the traffic signs.

II. RELATED WORK

A. Traffic Sign Recognition

Extensive research is being done by the computer vision and machine learning communities to address the problem of automatic traffic sign detection and recognition in the past decade. Issues such as non-uniform scene illumination, blurring due to motion of vehicle-mounted camera with respect to traffic signs, traffic sign occlusion due to other vehicles, trees etc. make the traffic sign recognition task very challenging. Traffic sign detection based on extracting sign proposals and classifying based on a color probability model and Histogram of Oriented Gradients (HOG) is proposed by Y Yang et.al [9]. However manual features such as HOG fails due to the challenges mentioned above. Recently, several

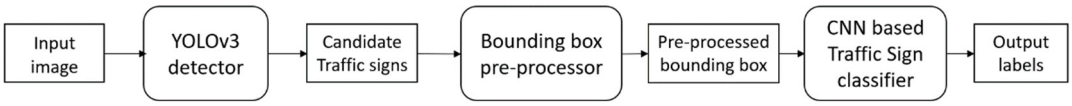

Fig. 1. YOLOv3 based traffic sign recognition pipeline

object detection problems are being better addressed using Convolutional Neural Networks (CNNs). Computer vision research for intelligent transportation systems are also following the trends and many such work are finding practical use in advanced driver assistance applications as well as in autonomous driving vehicles. In line with that, several researchers have attempted to solve the problem of traffic sign recognition using CNN based object detection frameworks.

Y Zhu et al [10] proposed a method based on deep learning components for traffic sign recognition. It consisted of a Fully Convolutional Network (FCN) for proposal generation and a CNN for classifying signs. Later Zhu et al [11] proposed a traffic sign recognition system using an end-to-end multi-class CNN for simultaneous traffic sign detection and classification. They have also created a new dataset called Tsinghua-Tencent 100K dataset for performance analysis. They also proposed a single class detection network for the traffic sign detection and using a separate classifier for classifying the detected traffic signs.

The performance of Fast-RCNN [12] for traffic sign recognition is studied by Zhu et al [11]. Also, Peng et al. [13] analysed Faster R-CNN for traffic sign detection. Even though this approach was promising as compared to the previous studies, consistent accuracy and detection speed could not be achieved. Another real time traffic sign recognizer based on Faster R-CNN [14] following the Mobilenet [15] structure proposed recently by Li and Wang [1] could show the real-time performance and accuracy required for applications such as self-driving car, based on their evaluation using GTSDB dataset. In this approach they have also proposed a classifier network using asymmetric kernels for classifying the signs into 43 classes.

B. Traffic Sign Classification

CNN based classifiers trained with GTSRB dataset could achieve high classification accuracy in classifying traffic signs. The classifier based on Multi Column Deep Neural Network (MCDNN) proposed by Ciregan et al. [16] and Multi-scale CNN by Sermanet et al [17] could achieve accuracies of 99.17% and 99.65% respectively. These networks however are large and have to learn a huge number of parameters. Another approach by T L Yuan [18] using Spatial Transformer Networks (STN) could achieve an accuracy of 99.59%.

Classifier network based on CNN using asymmetric kernels proposed by Li and Wang [1] reported an accuracy of 99.66% when evaluated with GTSRB dataset. Their FRCNN based detector and CNN based classifier combination has proved superior to the state-of-the-art traffic sign recognizers. The CNN based classifier employing asymmetric kernel proposed in [1] is used as the classifier in the proposed traffic sign recognition pipeline.

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. This paper presents a study on the usage of the new YOLOv3 [8] network as the detector for the traffic sign recognition system.

III. APPROACH

In this section, our approach for traffic sign recognition based on YOLOv3 is presented. The traffic sign recognition pipeline, which is illustrated in Fig. 1 consists of YOLOv3 detector trained for detecting the candidate traffic signs, a bounding box pre-processor, which enlarges the detected bounding box, crops and resizes the boxes containing candidate traffic signs, and a CNN based classifier which classifies the candidate traffic sign as belonging to one of the 43 classes.

A. YOLOv3 based Traffic Sign Detection

YOLO is a CNN-based object detection network. It offers real-time object detection performance by considering detection as a regression problem rather than several classification problems. A single stage detection network is used to take in image pixels as input and to predict bounding box coordinates and class probabilities. The whole detection process is performed in a single evaluation of the image. YOLO divides the input image into grids and predicts bounding boxes, confidence of those boxes, and class probabilities simultaneously. YOLOv2 and YOLOv3 are improved versions of the base YOLO network with several enhancements to increase the accuracy and detection speed. One of the important improvements in YOLOv3 is the multi-scale prediction, which helped in overcoming the difficulty in small object detection existed in its predecessors. Also, it uses logistic regression to predict the objectness score and uses multiple independent logistic classifier per class.

YOLOv3 is used as the traffic sign detector in the proposed method. YOLOv3 uses a base network which can be considered as a hybrid between YOLOv2 network, DarkNet-19 and a Residual network. The resulting network consists of 53 convolutional layers and therefore called DarkNet-53, which is the feature extractor. YOLOv3 network considers the image as an S × S grid of sub-images or cells and predicts bounding boxes and class probabilities for each sub-image.

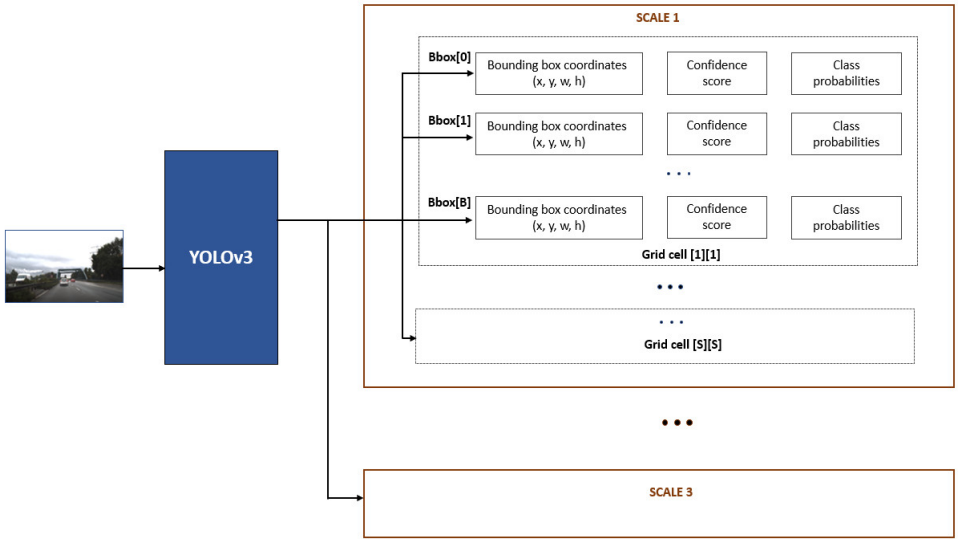

Fig. 2. YOLOv3 output visualization

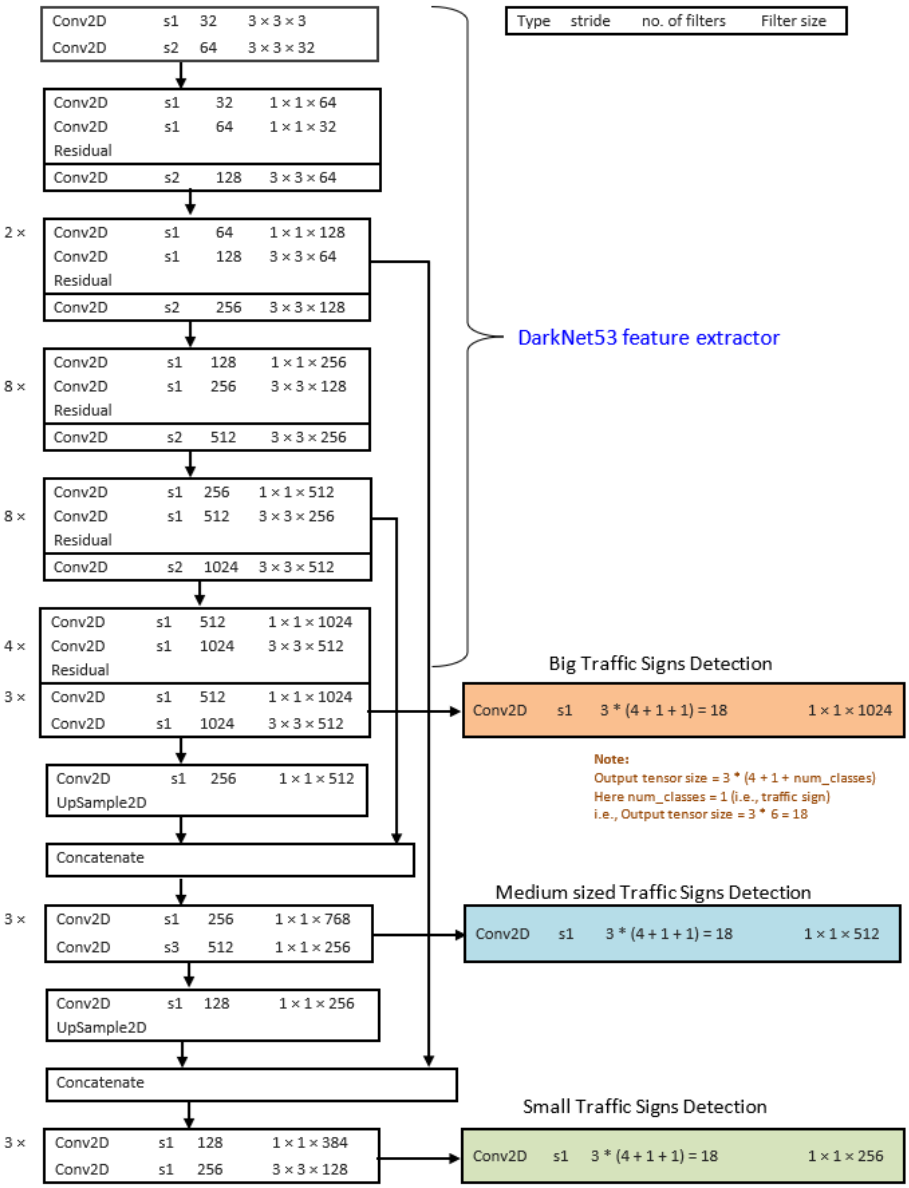

Fig. 3. YOLOv3 traffic sign detector network structure

The subimage where an object's centre is located is responsible for detecting that specific object. For each cell, the information for three bounding boxes are predicted. For each box, the predicted information consists of the location of the box, a box confidence score and probability of object classes for the box. The box coordinates consist of four values viz., two values corresponding to the offset to the centre co-ordinates of the object, width and height – (x, y, w, h). These values are normalized using image width and height. Hence these values fall in the range of [0, 1]. Box confidence score is an indicator of the likelihood of an object in the box. The class probabilities indicate the probabilities that the object in the box belonging to each available class. In YOLOv3, detection results at three scales are generated. Fig. 2 shows a visualization of YOLOv3 outputs.

The network structure of YOLOv3 based traffic sign detector is shown in the Fig. 3. It consists a total of 75 convolutional layers with no pooling or fully connected layers. Instead of pooling layers, convolutional layers with stride 2 are used to down-sample the feature map. The structure allows to predict traffic signs from images of any size. Residual connections similar to ResNet [19] and a multiscale detection mechanism similar to Feature Pyramid Network (FPN) [4] are used for accuracy improvement and small object detection in YOLOv3. Through this architecture, network becomes capable of identifying traffic signs of different sizes present in the image. For this, detection is done in feature maps of different scales in YOLOv3. To make the features suitable for detection at different scales, the feature maps from deeper levels are up-sampled and merged before detection at a particular scale. In YOLOv3 detector, a 1 × 1 convolutional layer with logistic regression is used for enabling multi-label classification, instead of SoftMax.

As shown in the Fig. 3, YOLOv3 uses DarkNet53 based feature extractor network and fully convolutional detector networks at three different scales. Each block shows the type of the layer, the stride, number of filters and the filter size.

1) DarkNet53 feature extractor: DarkNet53 network is used for feature extraction purpose. It uses a fully convolutional network with residual connections. Every convolutional layer is followed with a Batch Normalization layer and uses Leaky ReLU activation. Residual blocks with convolutional layers and shortcut paths make the network learn features when the network goes deeper. DarkNet53 layers are indicated in the network structure.

2) Detector subnetwork at various scales: The extracted features are used by convolutional subnetworks for detection of traffic signs at various scales, which correspond to big, small and medium sized signs. The last stage feature maps are combined with feature map outputs from earlier stages and the concatenated activations are used by the convolutional networks for different scales to generate the output tensors of size given by:

$$\textit{Number of bounding boxes} \times (4 + 1 + \textit{Number of classes}) = 3 \times (4 + 1 + 1) = 18 \quad (1)$$

3) Loss Function: YOLOv3 loss function is a multi-task loss function given by:

$$L = L_{loc} + L_{cls} \quad (2)$$

$$\begin{aligned} L = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{3} 1_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{3} 1_{ij}^{obj} \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{3} 1_{ij}^{obj} \left[-\hat{C}_i \log C_i - (1 - \hat{C}_i) \log(1 - C_i)\right] \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{3} 1_{ij}^{noobj} \left[-\hat{C}_i \log C_i - (1 - \hat{C}_i) \log(1 - C_i)\right] \\ & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} \left[-\hat{p}_i(c) \log p_i(c) - (1 - \hat{p}_i(c)) \log(1 - p_i(c))\right] \end{aligned} \quad (3)$$

The first two terms of the loss function signify the localization loss (regression loss). The last three terms constitute the classification loss. Localization loss is the squared error between predicted bounding box coordinates, $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$ and the ground truth box coordinates, (x_i, y_i, w_i, h_i). Classification losses are cross entropy loss terms. Of the last three terms, the first one penalizes the objectness score prediction for those bounding boxes responsible for predicting objects, the second one for bounding box with no objects and the last one penalizes the class prediction for bounding box which predicts the objects. In the loss function equation, λ_{coord} is the scale parameter which controls how much to increase loss from bounding box coordinate predictions, λ_{noobj} is the scale parameter which controls how much to decrease the loss of confidence score predications for boxes with no objects 1_i^{obj} indicates whether the grid cell i contains an object, 1_{ij}^{obj} indicates whether j^{th} bounding box of i^{th} grid cell is responsible for prediction of object, C_i is the confidence score of i^{th} cell, $\hat{C}_i$ is the predicted confidence score, *classes* indicate the set of all classes (for our case, it is 1 since only traffic sign class is available), $p_i(c)$ is the conditional probability of whether i^{th} cell contains object of class $c \in classes$, $\hat{p}_i(c)$ is the predicted conditional class probability.

B. Bounding Box Pre-processor

Bounding Box Pre-processor stage extracts and prepares the detected candidate traffic sign boxes for classification. The center of the regressed bounding box is determined and the box is enlarged by 25% to compensate for any regression errors to make sure the traffic sign is completely enclosed in the region. The enlarged boxes are cropped and resized to 48 × 48 size and fed to the classifier network for recognition of the traffic sign class.

C. Traffic Sign Classifier

A fast and accurate traffic sign classifier network architecture proposed in [1] is used here.

The classifier network structure is in Table I. In this architecture, an n × n convolution is replaced by an n × 1 convolution followed by a 1 × n convolution, which reduces both the number of convolution operations and the network parameters. This leads to computational cost reduction and increased speed.

TABLE I. TRAFFIC SIGN CLASSIFIER ARCHITECTURE [1]

Layer	Type		Filter Size/ Parameter	Filters/ Stride	Output shape
1	Conv		3 × 3	32, s1	48 × 48 × 32
2	Conv		7 × 1	48, s1	48 × 48 × 48
3	Conv		1 × 7	48, s1	48 × 48 × 48
4	MaxPool		2 × 2	s2	24 × 24 × 48
5	DropOut		0.2		24 × 24 × 48
6-1	Inception	Conv	3 × 1	64, s1	24 × 24 × 64
7-1	Inception	Conv	1 × 3	64, s1	24 × 24 × 64
6-2	Inception	Conv	1 × 7	64, s1	24 × 24 × 64
7-2	Inception	Conv	7 × 1	64, s1	24 × 24 × 64
8	Inception	Concatenate	-	-	24 × 24 × 128
9	MaxPool		2 × 2	s2	12 × 12 × 128
10	DropOut		0.2	-	12 × 12 × 128
11	Conv		3 × 3	128, s1	12 × 12 × 128
12	Conv		3 × 3	256, s1	12 × 12 × 256
13	MaxPool		2 × 2	s2	6 × 6 × 256
14	Dropout		0.3	-	6 × 6 × 256
15	Dense		256	-	256
16	Dropout		0.4	-	256
17	Dense - Softmax		43	-	43

Batch Normalization and ReLU layers follows all layers other than the final dense layer. The sixth layer forms an inception module where kernels of different sizes are used to extract information from feature map output of the previous layer. The output feature maps from the inception layer are concatenated to combine the feature maps. Dropout layers are also used to regularize the activations of the final stages. To recognize the 43 traffic sign classes, fully-connected layer of an output size of 43 with Softmax activation is used as the last layer.

IV. EXPERIMENTS

The YOLOv3 based traffic sign detector and CNN based classifier are implemented using Keras with TensorFlow backend. The detector and classifier are trained and evaluated on the GTSDB and GTSRB datasets respectively. A computer having Nvidia GTX 1060 GPU with 6GB GPU memory was used for training and evaluation of the detector and classifier.

A. Datasets

The GTSDB and GTSRB datasets are very popular datasets for training and evaluating models for traffic sign detection and recognition respectively.

The GTSDB dataset is generated from video sequences recorded near Bochum, Germany. It consists of images from urban, rural and highway scenarios under different weather and lighting conditions making it pretty representative and challenging. The total dataset consists of 900 images of 1360 × 800 pixels in raw PPM format, which are divided into 600 training images and 300 test images. The training images contain 846 traffic signs and test images contain 360 traffic signs. The traffic sign sizes in the images vary between 16 and 128 pixels. Sample images from GTSDB dataset is shown in Fig. 4. The YOLOv3 based detector is trained and evaluated on the GTSDB dataset.

The GTSRB dataset consists of more than 50000 traffic sign images of 43 classes with sizes varying between 15 × 15 pixels and 222 × 193 pixels. The classifier is trained using 39209 training images and evaluated on 12630 test images from GTSRB dataset. Sample images of different traffic sign classes is shown in Fig. 5.

Fig. 4. Traffic scene image samples from GTSDB dataset [5]

Fig. 5. Sample images of 43 traffic sign classes from GTSRB dataset [6]

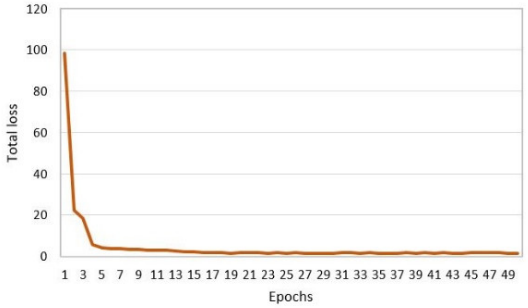

Fig. 6. YOLOv3 detector training - Loss vs Epochs

(a) (b)

Fig. 7. (a) Traffic sign detection results (b) Traffic signs labelled with classification results

B. *Evaluation 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 Faster R-CNN based detector. Accuracy is used as the measure for evaluating the classifier performance. Confusion matrix-based evaluation for analyzing the accuracy of the classifier across various traffic sign classes is also used.

C. *Detector training*

Detector training was done on GTSDB dataset with 600 training images. YOLOv3 network is loaded with DarkNet53 weights pre-trained using ImageNet dataset for transfer learning.

Adam algorithm [20] was used as the loss function optimizer with an initial learning rate of 1e-4 with a learning rate reduction factor of 0.1 on plateauing. Sum squared error loss function is used for bounding box regression and cross entropy loss is used for classification. Mini-batch training using GTSDB training image set with a batch size of 4 is done with the training images for 50 epochs.

The plot showing the total loss vs training epochs for YOLO-v3 based detector is shown in Fig. 6.

D. *Classifier training*

Classifier was trained on GTSRB dataset with 39209 training images. Minibatch training with batch size of 16 was done. Adam optimizer was used with an initial learning rate of 0.001 and learning rate decay of 1e-6 per mini batch. The classifier was trained for 50 epochs initially without any data augmentation and further for an additional 200 epochs with data augmentation, as proposed in [1], using various image transformations like shifting, shearing, scaling and rotation.

V. EVALUATION

A. Detector Performance

The YOLOv3 based traffic sign detector was evaluated using the GTSDB test set having 300 images. Mean Average Precision (mAP) is used as the performance measure for the detector. The proposed detector could achieve a mAP of 92.2% on the GTSDB test set. mAP of 96.1% is observed when verified with the training set images. It took an average of 101ms for processing an image, i.e., an average frame processing rate of approximately 9.87 frames/sec (fps) could be achieved. The detector exhibited very good localization performance for small traffic signs, however missed to accurately localize a few very small traffic signs. A few false positive detections were also observed, where some advertisement signs which closely resembles traffic signs have been misclassified. Some examples of detection results are shown in Fig. 7(a).

The detector performance is compared with traffic sign detector implementation using Faster R-CNN [1], evaluated using the GTSDB test set. The proposed YOLOv3 based detector achieved high accuracy with a mAP of 92.2% on GTSDB test set, with a frame rate performance of nearly 10 frames per second.

A comparison of the detector performances evaluated on Intel i7 based PC with Nvidia GTX 1060 GPU is in Table II.

TABLE II. TRAFFIC SIGN DETECTOR COMPARISON

Detector	Mean Average Precision (mAP)	Time taken per image	Frame Rate
Faster R-CNN [1]	84.5%	261ms	3.82 fps
YOLOv3 (proposed)	**92.2%**	101ms	**9.87fps**

B. Classifier Performance

The CNN based custom traffic sign classifier was evaluated using the GTSRB test set consisting of 12630 images. Model trained without data augmentation could achieve an accuracy of 96.46%. With data augmentation, accuracy has improved to 99.6% as reported in [1].

A complete traffic sign recognition pipeline using YOLOv3 based detector, bounding box pre-processor and CNN based classifier has been setup by integrating the components. The final result images with labels generated by the classifier on bounding boxes from the detector are shown in Fig. 7(b).

VI. CONCLUSION

In this paper, a traffic sign recognition system based on YOLOv3 based detector and a custom CNN based classifier is proposed. The detection performance, with traffic-sign being considered as a single class, is found to be superior to the previous detectors, based on the detection speed. It could achieve a speed of almost 10 frames/second with a high mAP performance of 92.2%. The proposed detector is able to detect almost all categories of traffic signs and could regress accurate bounding boxes for most of the detected signs. A few very small traffic signs could not be accurately localized and also, there were a few false positive detections. The traffic sign classifier used is simple in architecture with very high accuracy. The YOLOv3 based detector and CNN based classifier completes the traffic sign recognition pipeline.

As a future step, simultaneous detection and classification of traffic signs using single stage detectors could be explored. This method could help avoiding the use of an additional traffic sign classification network. Fine tuning of networks to learn the intricacies of similar looking signs is required in this case and it would be challenging to tune the network to give an accuracy performance exhibited by the detector-classifier based recognition pipeline. But it would be worth exploring this aspect in future.

REFERENCES

[1] 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.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput.Vis. Pattern Recognit.*, pp. 779–788, Jun. 2016

[3] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf Comput. Vis.* Cham, Switzerland: Springer, 2016.

[4] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, "Feature Pyramid Networks for Object Detection ", [Online] Available: https://arxiv.org/abs/1612.03144, 2016

[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] Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", [Online] Available: https://arxiv.org/abs/1612.08242, 2016

[8] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", [Online] Available: https://arxiv.org/abs/1804.02767, 2018

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] T. L. Yuan. GTSRB_Keras_STN. Accessed: Nov. 1, 2017. [Online] Available: https://github.com/hello2all/GTSRB_Keras_STN

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", [Online] Available: https://arxiv.org/abs/1512.03385

[20] D. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980