IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Traffic Goggles – Real Time Traffic Sign Recognition for Driving Assistance

Abhishek Singh Army Institute of Technology
Pune, India

Akash Mall
Army Institute of Technology Pune, India

Anil Mudgal
Army Institute of Technology Pune, India

Akash Sangwan
Army Institute of Technology Pune, India

***Abstract*— Traffic Sign Recognition (TSR) is a crucial component of Advance Driving System Assistance System (ADAS). A TSR system helps the driver in safe driving as traffic signs provide invaluable information about the rules and regulation to follow while driving. This paper brings forth a proposal of a system that detects different types of traffic signs from camera feed in real time and provides output in audio format. YOLO v3 is used for real time recognition. The output is then converted to audio format using text to speech (TTS) API.**

Keywords*— YOLOv3, Traffic sign, ADAS, Machine learning, Convolutional Network, , Text to Speech, Darknet, Voice Feedback.

I. INTRODUCTION

Traffic sign detection and recognition is a very important field of computer vision which is used in Advanced Driving Assistance System (ADAS). Traffic signs or Road signs are visual cautionary representations erected at the side or above roads that give instruction or provide road guidance to road users. Traffic signs helps in maintaining order on roads and reducing accidents by provide information to motorists and pedestrians. They are the rules that are there for everyone to follow them, and they help to communicate between fellow drivers and pedestrians. They define statutory rights granted to vehicles, allow or prohibit certain actions or directions, warn about road conditions, define speed limit etc. Ignoring them is dangerous.

Most signs make use of pictures, rather than words, so that they can be easily interpreted by people who speak different languages. They consist of various shapes and colors which can be used to distinguish and classify them easily by people as well as computers. Red, blue, green, yellow are the common colors that are associated with traffic signs and the shapes are mostly circular, triangular or rectangular.

Various methods for traffic-sign detection and recognition have been proposed. However, recognition with respect to various viewing angles, different weather conditions and factors like occlusion still remains a challenging task and is an open research problem in the field of computer vision.

In this paper, we are using You Only Look Once (YOLO) v3 for real time object detection.

II. RELATED WORK

Extensive research has been done by the computer vision and deep learning enthusiasts in this field.

[1] and [2] shows that accuracies of 99.17% and 99.65% can be achieved using the classifier which use Multi Column Deep Neural Network (MCDNN) but the networks are too large and are required to learn a large number of parameters. [3] shows that accuracy and reliability of the application which is working in real time can be improved. To achieve that, it uses the Faster R-CNN Inception-V2 model via transfer learning for traffic light detection and recognition for self -driving cars.

[4] presents a new traffic sign detection approach by integrating both the Adaboost algorithm and SVR together which achieves a detection time of 0.05 - 0.5 seconds for each image, precision of 94% and recall of 80%.

On the other hand, [5] uses YOLO v3 detector along with custom CNN based classifier to obtain a performance of 92.2% for traffic sign detection and classification. There are two types of object detection methods

1.Two stage object detection- In the first stage object location is determined. In the second stage the object classification is done. Examples of such CNN models are RCNN, F-RCNN etc.

2.One stage Object detection – Such networks process the input image from end-to-end to detect objects in it. Examples of such models are YOLO and SSD.

YOLO along with mobile platforms is suitable for advanced driver assistance systems due to low consumption and affordability [6].

We are providing input to YOLO directly from the live camera feed. In [7] author achieved recall of 65.4% and precision of 96.4%, even if the application is android based and the phone's camera is used there, the frame rate of 11 to 17 frames per second (FPS) is achieved with image analysis in real time where continuous input was provided to it. It is built over SSD topology.

[8] proposes a tool for real time traffic signs recognition for mobile devices using which achieved good results. It uses Tensorflow Lite quantized model, which is compatible with low computational capacity mobile devices and 4 tim1es faster detection is shown by the quantized model compared to the float model on the mobile device. This tool is also based on SSD topology.

We are proposing an application that would be built over YOLOv3 architecture and incorporated to mobile devices with the help of tensorflow Lite.

III. METHODOLOGY

A. *Preprocessing*

● Prepare YOLOv3 Darknet Custom Data.

We will prepare custom Indian traffic sign dataset and will use Roboflow to convert it to Darknet annotation format, automatically.

● Train YOLOv3 Darknet.

In order to get weights file for our custom dataset, we will first install Darknet YOLOv3 on a laptop/PC. We will then customize training configurations (for eg. Setting epochs) to get least loss in the training phase.

We will then train the dataset on the yolov3 detector which will give us the weights of our model for our custom dataset.

● Convert Darknet Model to TensorFlow Lite

As we have yolov3. weights after training the model. We will convert yolov3.weights to .pb configuration using mistic123's implementation to get the yolo-v3.pb.

We will use utility tflite_convert which is the part of tensorflow 1.10 (or higher) package to convert yolov3.pb to yolov3.tflite format.

● Deploy on Device

The last step before using our model on android devices is to deploy it using tensorflowlite package.

B. *Android App Flow*

Our application will give users option to choose a preferred language and will give the voice assistance in the desired language.

It will further get access of the camera and start recording the video. Each frame of the video will then be tested on the model using the tensorflow lite api and probability for different classes will be returned by the model.

If any class probability will cross the threshold value then that class of sign will be read out as voice output to the user.

C. *YOLOv3 Framework*

Yolov3 has darknet-53 as its backbone network. Darknet-53 is a network with 53 convolutional layers. It is a classifier network.

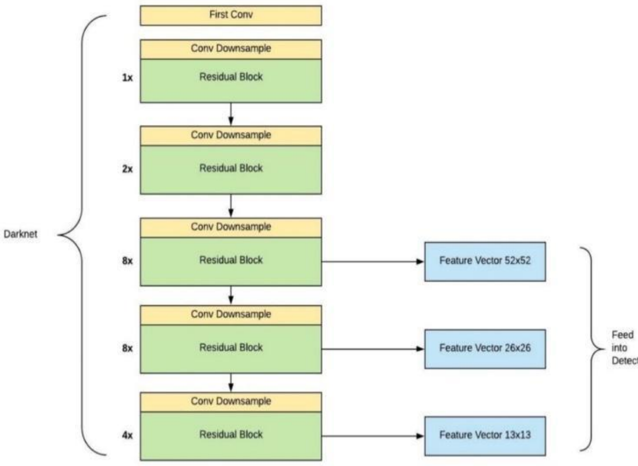

Fig. 1. Darknet-53 Architecture

Yolov3 uses upsampling layers for detecting objects at different scales which are branched from the upsampling network.

In Fig. 1. input is assumed as an image of 416 * 416 dimension and detection occurs at three scales namely, 13*13, 26*26 and 52*52.

The detection happens at three different scales, which makes detection of small objects possible.

We will be training our dataset on the yolov3 architecture hence performing Transfer Learning for our application.

IV. COVER FLOWCHART

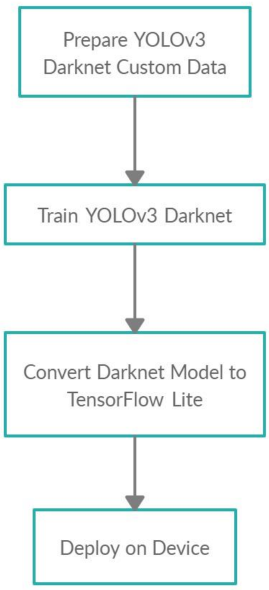

Fig. 2. Preprocessing

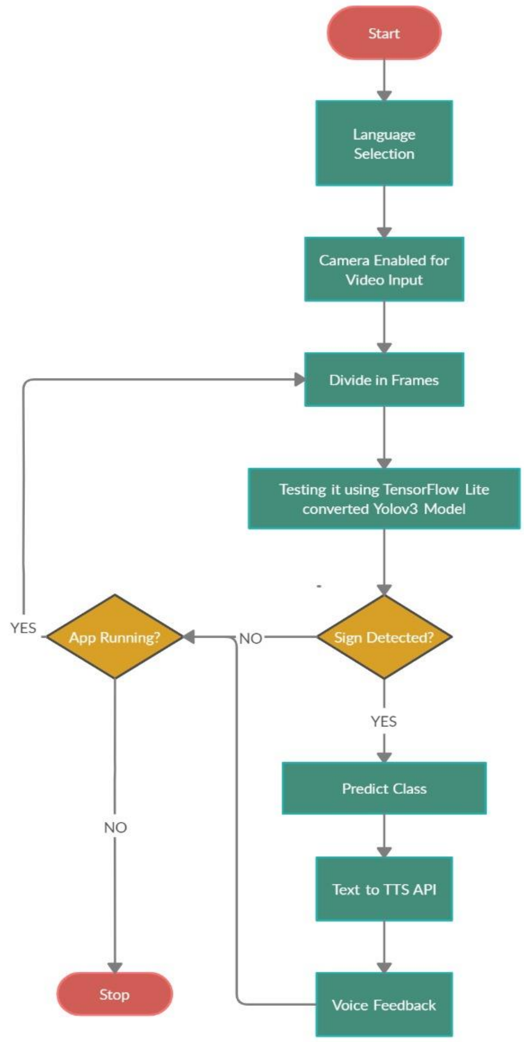

Fig. 3. Android App Flow

V. ALGORITHM COMPARISON

- R-CNN

R-CNN combines region proposals with CNN, it was introduced in 2014. Selective search is used to generate regions and detect the object. Time taken to train the network is more as 2000 region proposals per image are classified [9]. Its shortcomings are that it is consumes large memory, slower and harder to train. It cannot be used in real time, because 47 seconds are taken for each test image.

- ResNet

Initially ResNet was implemented as VGG network in 2015, it has accomplished state-of-the-art performance and is one of the most powerful deep learning neural network. The accuracy is decreased as a drawback, even though ResNet-101 layer has more depth than VGG-16 layer. Mainly it can undergo a deeper layer.

- Fast R-CNN

Fast R-CNN [10] was introduced in the year 2015 the same year as ResNet. It has higher accuracy as compared to previous versions and also it detects in less amount of time. A single method is used by F-RCNN to divide the extracted features into different classes from the regions as well as to provide the bounding boxes for the detected classes.

- Faster R-CNN

Selective search algorithm is slow and time consuming.

In the previous methods it was used to detect the region proposals. A new method is used to classify the image in the selected region which uses RoI pool layer, to overcome the slowness. Bounding box values can also be calculated using this method. It is faster as well as reliable as it can detect smaller objects as well.

- YOLO

YOLO was introduced in 2015 by Redmon et al, it is a single-stage detector. Major benefit compared to above methods are that the fast-moving objects are very fastly captured. Purpose of this is mainly speed. It is quicker compared to other methods.

- YOLOv3

YOLOv3 a better version was released in the year 2018. It can detect multi-labels predictions as well as small objects which was not possible with earlier version of YOLO. It uses a new network for feature extraction, Darknet-53. There are 53 convolutional layers in Darknet-53 and it is comparable with the existing classifiers but with lesser floating point operations and greater speed

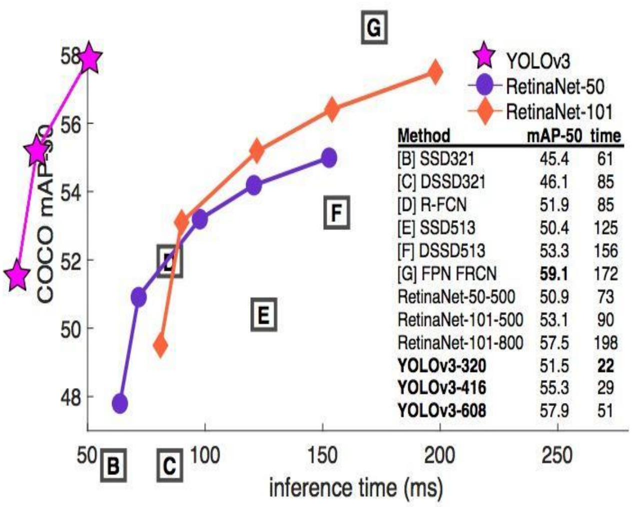

Fig. 4. Model Comparison

VI. CONCLUSION

Based on algorithm comparisons we can conclude that with mAP of 55.3, which is slightly less than the best among the models, the time taken, 22 ms, is way less than the other models. So, with slight trade-off of mAP for time, YOLOv3 is most suitable for the real time application. Hence we will use YOLOv3 for our Traffic Goggles Application.

REFERENCES

[1] D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp. 3642-3649, doi: 10.1109/CVPR.2012.6248110.

[2] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," The 2011 International Joint Conference on Neural Networks, San Jose, CA, 2011, pp. 2809-2813, doi: 10.1109/IJCNN.2011.6033589.

[3] R. Kulkarni, S. Dhavalikar and S. Bangar, "Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697819.

[4] T. Chen and S. Lu, "Accurate and Efficient Traffic Sign Detection Using Discriminative AdaBoost and Support Vector Regression," in IEEE Transactions on Vehicular Technology, vol. 65, no. 6, pp. 4006-4015, June 2016, doi: 10.1109/TVT.2015.2500275.

[5] S. P. Rajendran, L. Shine, R. Pradeep and S. Vijayaraghavan, "Real-Time Traffic Sign Recognition using YOLOv3 based Detector," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-7, doi: 10.1109/ICCCNT45670.2019.8944890.

[6] Towards Real-Time Traffic Sign Recognition via YOLO on a Mobile GPU. N S Artamonov and P Y Yakimov 2018 J. Phys.: Conf. Ser. 1096 012086.

[7] Z. Domozi, D. Stojcsics, A. Benhamida, M. Kozlovszky and A. Molnar, "Real time object detection for aerial search and rescue missions for missing persons," 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2020, pp. 000519-000524, doi: 10.1109/SoSE50414.2020.9130475.

[8] A. Benhamida, A. R. Várkonyi-Kóczy and M. Kozlovszky, "Traffic Signs Recognition in a mobile-based application using TensorFlow and Transfer Learning technics," 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2020, pp. 000537-000542, doi: 10.1109/SoSE50414.2020.9130519.

[9] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587). http://openaccess.thecvf.com/menu.py

[10] Girshick, R. (2015). Fast R-CNN. Proceedings of the IEEE international conference on computer vision (pp. 1440-1448). http://openaccess.thecvf.com/menu.py.