International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

AI based Driver Assistant system

Akshay Naikwadi[1], Hitisha Pichad[2], Bhupesh Patil[3], Snehal Andhare[4]

[1,2,3]*Student, Dept. of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India*
[4]*Professor, Dept. of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India*

Abstract - *With the increase in number of road accidents happening every year, there has been a need to develop a system that contributes to the safety of the drivers, pedestrians and vehicles. Traffic sign detection and recognition plays an integral role for driver assistant system as well as autonomous driving vehicles. Although traffic sign detection has been studied for years and great progress has been made with the rise of deep learning techniques, there are still many problems remaining to be addressed. For real-world and complicated traffic scenarios, there are two main challenges. Firstly, traffic signs are usually small size object, which makes them more difficult to detect than large ones; secondly, the driver unwittingly may miss the traffic sign or misinterpret it while driving. Also, it is hard to distinguish false signs which may resemble an actual traffic sign. In the recent advancement of object detection, algorithms like R-CCN, fast R-CCN and faster R-CCN have provided result yet not efficient enough to be implemented in real-time. In this paper, an approach to assist the driver through traffic sign recognition with much more faster detection in conjunction with human-like general voice feedback has been presented.*

Key Words: You Only Look Once (YOLOv3), Region Convolutional Neural Network (R-CNN), traffic sign detection, traffic sign recognition, advanced driving systems, voice feedback.

1. INTRODUCTION

Artificial intelligence (AI) and self-driving cars are bilateral topics in technology. Traffic sign detection and recognition plays a crucial role in such expert systems, such as advanced driving systems as it instantly assists drivers or automatic driving systems to detect and recognize traffic signs effectively. However, traffic sign recognition turns to be a bit difficult task since there are many adverse factors, such as bad weather, viewpoint variation, wore out traffic signs, physical damage, etc. The difficulties in this area that we can face, are as follows:

(i) As the cameras mounted on vehicles are not always perpendicular to the traffic signs, and the shape of traffic signs are often distorted in road scenes, the shape information of traffic signs is no longer fully reliable.

(ii) Traffic signs on some road often gets blocked by buildings, trees, and other vehicles; therefore, we needed to recognize the traffic signs with incomplete information.

(iii) Traffic sign discoloration, traffic sign damage, rain, snow, fog, and other problems, are also given as challenges in the process of traffic sign detection and classification.

In this research, we evaluated the performance of real time traffic sign detection with 44 different classes of German traffic signs using YOLOv3 which has proven to be one of the fastest object detection algorithms. In this, we created a custom dataset of traffic signs by dividing traffic video into numerous frames such that each frame contains atleast one traffic sign from our custom Meta data. Then the model training was performed over Google Colaboratory (free Jupyter notebook environment provided by Google) as it provides free GPU over the interface which reduces the time required for model training. Finally, we enhanced the ability of the model to adapt to real-world environment. Using this approach, we observed that a real time performance on our custom database which gave very good and legitimate classification results.

2. BACKGROUND & RELATED WORK

This section of the paper depicts the different work carried out by others in areas which are relevant to our research. The sub-parts below are the most important key aspects in our research.

Object detection is a computer vision technique for spotting instances of objects in images or videos. Object detection algorithms generally use machine learning or deep learning to produce insightful results. When humans look at images or video, we can recognize and locate objects of interest within a blink of an eye. The goal of object detection is to imitate this intelligence using a computer.

Traffic Sign Recognition: Conventionally, most of the traffic sign recognition algorithms followed a two stage convolution neural network architecture. The first stage consisted of detecting a traffic sign from the image frame using shape detection, contour detection, etc. and the following stage classified the detected traffic sign on the basis of the model trained on the traffic signs dataset. Some of these techniques can be implemented using algorithms such as R-CNN, fast R-CNN, faster R-CNN, etc [5]. These R-CNN based algorithms need two shots, one for generating region proposals and the other for detecting the object of each proposal.

YOLO object detection: YOLO v3 algorithm is a cutting edge technology which consists of fully CNN and also used for post-processing outputs from neural network. CNNs are peculiar architecture of neural networks suitable for processing topology which is grid-like in a frame. This

outstanding feature of CNNs which has relevance in object detection is parameter distributing. Its architecture also contains residual layers, upsampling layers, and skip (shortcut) connections besides using convolutional layers [2]. This feature plays important role in capturing whole scene on the road. YOLO uses a totally different approach. A single neural network is applied to the entire image. The image is divided into different regions by the network and it predicts bounding boxes and probabilities for these divided regions.

Voice feedback: After successful detection of any traffic sign, the next step is to provide assistance to the driver in the form of vocal feedback. However, there are few challenges to do so such as playing the audio which is associated to the same frame, moreover if the frame containing traffic sign is detected then the frame may wait for the voice feedback to completely speak out and then proceed to the next frame for processing which may cause a significant delay resulting in asynchronization with real time. Also, one more problem one could face is when there are multiple traffic signs in the same frame which may result in echo or irritation to the driver.

3. PROPOSED SYSTEM

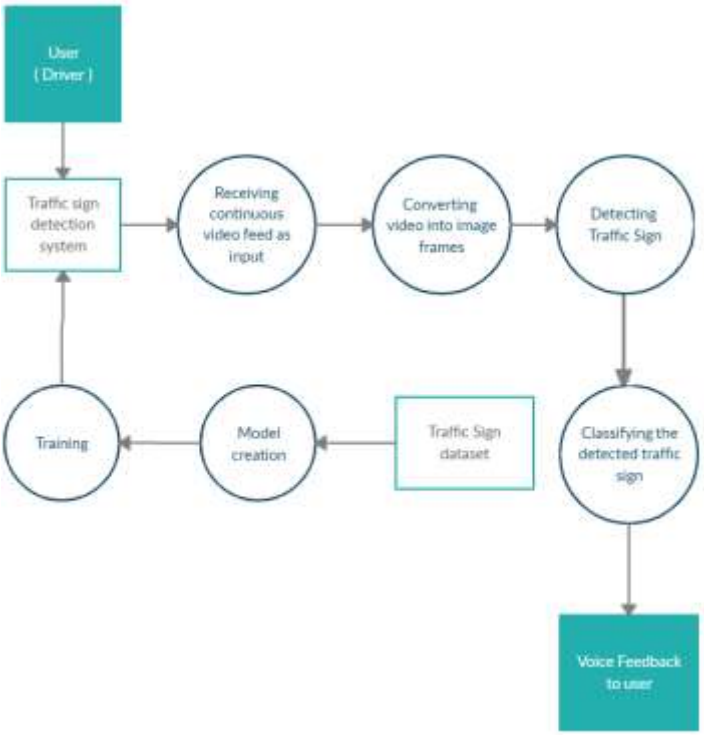

Fig -1: Work flow of system

The above diagram shows the overall workflow of our Driver Assistant system.

Following are the steps for our proposed system.

Step 1: Collecting sample videos containing German traffic signs matching to our custom dataset.

Step 2: Trimming the video such that when it is converted to frames there exists no frame without any traffic sign in it.

Step 3: Preprocessing the frames and discarding the noisy data.

Step 4: Creating bounding boxes manually around the traffic signs and labelling them with their respective class name in every frame.

Step 5: Training the model with the labelled frames such that the loss between the consecutive epochs is close to each other and as low as possible.

Step 6: Testing the model with user input video. As soon as the traffic sign is detected, the corresponding class label is called out using voice assistant system.

4. OVERVIEW OF SYSTEM

In this section, we will be focusing more on all the important stages such as data collection, bounding box mechanism, YOLOv3 algorithm, training, voice feedback, and testing.

4.1 Data collection

As our meta data is mostly focusing on German Traffic signs, videos related to same are collected. As the video collected is raw, some processing has to be performed in order to obtain clean dataset. Software tools like Filmora and video-to-jpeg convertor are used. During the preprocessing, the part of the video that does not contain traffic sign is trimmed as it may add to the noisy data. After the entire video is trimmed, the frames are obtained at the rate 10 frames per second and each frame is saved as a jpeg file.

Fig -2: Examples of traffic signs

4.2 Bounding Box mechanism

Now, we have to draw bounding boxes for traffic signs on every frame. To accomplish this we used a tool OpenLabelling written in Python to draw bounding box for each traffic sign. YOLOv2 requires annotation text in XML file format while YOLOv3 requires the same in TXT file format. So, this tool generates a txt file for every image. The format of storing the annotation data in the txt file is as follows:

[class_id] [x] [y] [width] [height]

Where:

- [class_id] : integer number of object class from 0 to (classes-1)

- [x] [y] [width] [height] : float values according to width and height of image, it ranges from 0.0 to 1.0
- [x] [y]: is the centre of rectangle (are not top-left corner).

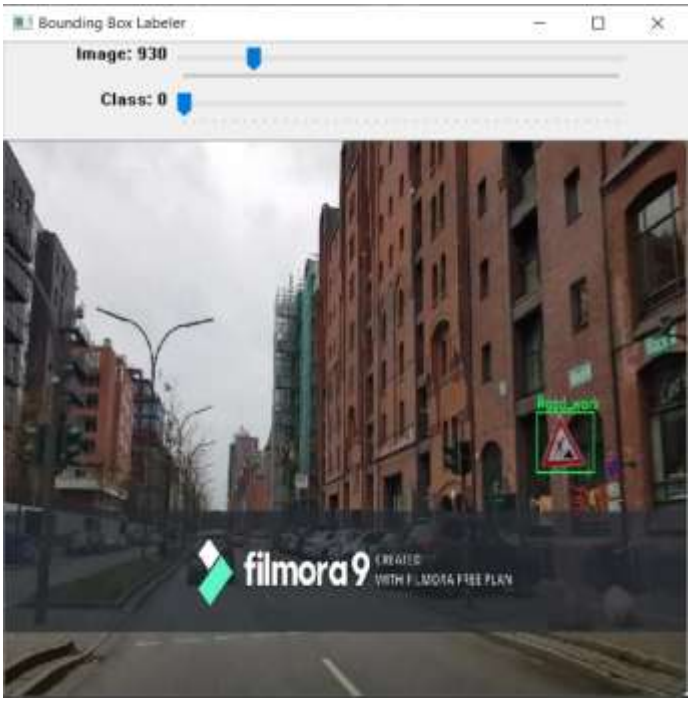

Fig -3: Bounding box

The following image shows annotation for the traffic sign "Road Work".

Fig -4: Annotation file

4.3 YOLOv3 algorithm & architecture

You Only Look Once (YOLO) is used for real-time object detection and is based Convolutional Neural Network (CNN). YOLOv3 uses a single step detecting neural network where input is given in the form of pixels of image. It is used to predict the bounding boxes and also calculate their confidence score which denote how much the system is certain about the predicted object. A significant advancement in YOLOv3 is the multi-scale prediction which is useful during detecting objects of small size that existed in its previous versions. [1]

In our proposed method, we used YOLOv3 algorithm to detect traffic sign. The base network of YOLOv3 is considered as the combination of YOLOv2 network, DarkNet-19 and a residual network [1]. This entire architecture results into 53 convolution layers and hence it is called DarkNet-53 which stands for feature extractor.Yolov3 takes an image as an input in form of n x n grids. Each grid corresponds to sub image in which bounding box is predicted according to the class with maximum confidence for that box.

No.	Layer	Filter	Size	Input	Output
0	Convolution	16	3 x 3 / 1	416 x 416 x 3	416 x 416 x 16
1	Max Pooling		2 x 2 / 2	416 x 416 x 16	208 x 208 x 16
2	Convolution	32	3 x 3 / 1	208 x 208 x 16	208 x 208 x 32
3	Max Pooling		2 x 2 / 2	208 x 208 x 32	104 x 104 x 32
4	Convolution	64	3 x 3 / 1	104 x 104 x 32	104 x 104 x 64
5	Max Pooling		2 x 2 / 2	104 x 104 x 64	52 x 52 x 64
6	Convolution	128	3 x 3 / 1	52 x 52 x 64	52 x 52 x 128
7	Max Pooling		2 x 2 / 2	52 x 52 x 128	26 x 26 x 128
8	Convolution	512	3 x 3 / 1	26 x 26 x 128	26 x 26 x 256
9	Max Pooling		2 x 2 / 2	26 x 26 x 256	13 x 13 x 256
10	Convolution	1024	3 x 3 / 1	13 x 13 x 256	13 x 13 x 512
11	Max Pooling		2 x 2 /2	13 x 13 x 512	13 x 13 x 512
12	Convolution	1024	3 x 3/ 1	13 x 13 x 512	13 x 13 x1024
13	Convolution	256	1 x 1/ 1	13 x 13 x1024	13 x 13 x 256
14	Convolution	512	3 x 3/ 1	13 x 13 x 256	13 x 13 x 512
15	Convolution	255	1 x 1/ 1	13 x 13 x 512	13 x 13 x 255
16	yolo				
17	route	13			13 x 13 x 256
18	Convolution	128	1 x 1/ 1	13 x 13 x 256	13 x 13 x 128
19	Up sample		2x	13 x 13 x 128	26 x 26 x 128
20	route				26 x 26 x 384
21	Convolution	256	3 x 3/ 1	26 x 26 x 384	26 x 26 x 256
22	Convolution	255	1 x 1/ 1	26 x 26 x 256	26 x 26 x 255
23	yolo				

Fig -5: YOLOv3 Layers architecture

YOLOv3 makes prediction on the basis of darknet-53 at 3 different scales. Each location is being predicted 3 times by YOLOv3. Each prediction takes into account a boundary box, an object score and 44 class scores, i.e. N × N × [3 × (4 + 1 + 44)] predictions [9]. Each block displays the following things that is the type of layer, the stride, number of filters and filter size. The detection layer used for detection at feature maps are of three different sizes, having the strides 32, 16, and 8 respectively. This means, with an input of 416 x 416, we make detections on scales 13 x 13, 26 x 26 and 52 x 52 [2]. Each cell predicts 3 bounding boxes using 3 anchors at each scale which makes the total number of anchors used as 9 (The anchors are different for different scales).

An image may contain many objects and each object is related with one grid cell. YOLO can work well in such situations where overlapping of centre points of two objects can occur. To allow a grid cell to detect multiple objects, YOLO uses anchor boxes [2]. With the help of anchor boxes, a longer grid cell vector is created and multiple classes with each grid cell can be associated. Anchor boxes have a defined aspect ratio with which they try to detect objects that properly fit into a box with the defined ratio.

4.4 Training

For our proposed model, we have collected upto 8200 images with each image labelled and having its annotation file associated with it such that the name of the txt file is same as that of the image. We trained our model on Google Colaboratory as it provides a single 12GB NVIDIA Tesla K80

GPU for a runtime of about 12 hours. The total number of epochs were 15000 i.e. 15000 iterations with an average loss of 0.5 and a learning rate of 0.001. After every thousand iterations, weights were saved as weights file.

4.5 Voice Feedback

For including voice assistance in our system, we have used Google Text-to-Speech (gTTS) python library [12] which generates sound based on the text. However, while speaking out the label of the detected traffic sign in current frame, the processing of the next frame used to halt. Hence, there was a delay with respect to the actual video. To overcome this problem, we used the concept of Multithreading in which the processing of the frame continues and the voice feedback does not interrupt the processing of the next frame. However, even after handling the issue of delay, another obstacle arose i.e. as the frames were continuous, in each frame the traffic sign detected was called out by the gTTS. Hence, we used a buffer array to check the repetition of consecutive traffic sign in the frames. If the sign detected is not in the buffer then the gTTS will speak the label of the class else it will look for the next frame. Solving these issues helped in increasing the performance of the model on our system.

4.6 Testing

The below image shows the output snapshot of the testing video with the traffic sign as Speed limit 30 and a confidence score of 0.92. However, the overall confidence score is the range of 0.7 to 0.9.

Fig -6: Output snapshot

5. CONCLUSIONS

In this paper, we have presented the idea of traffic sign detection using YOLOv3 algorithm. Also, our system provides vocal feedback which gives the feel of assistant for a driver. The intuition behind implementing our idea was that driver may miss the traffic sign while driving through alert areas or driver may doze off, also, inexperienced drivers may miss multiple signs occurring simultaneously which can cause accidents in several cases. As the number of images in training were in huge number, the model gave more accurate and robust results.

6. FUTURE SCOPE

Our model can be extended to detect traffic signal on the roadways which will also help driver not to miss any signals and violate any traffic rules .This can be done by adding images of traffic signals under various conditions. Similarly, idea can be protracted for driver drowsiness detection, potholes detection and lane detection. However every idea contributes to the safety of the driver and the pedestrians. In some places self-driving cars are still not in wide use, our system can contribute along with other aligned ideas to provide guidance to either driver or the self-driving cars.

REFERENCES

[1] Shehan P Rajendran, Linu Shine, Pradeep R Sajith Vijayaraghavan, "Real-Time Traffic sign Recognition using based Detector" 10th ICCCNT 2019 July 6-8, 2019, IIT - Kanpur,Kanpur, India. IEEE-45670.

[2] Aleksa Ćorović, Velibor Ilić, Siniša Đurić, Mališa Marijan, and Bogdan Pavković "The Real-Time Detection of Traffic Participants Using YOLO Algorithm" 26th Telecommunications forum TELFOR, Serbia, Belgrade, 2018.

[3] Aleksej Avramović, Domen Tabernik, Danijel Skočaj, "Real-time Large Scale Traffic Sign Detection", 14th Symposium on Neural Network and Applications (NEUREL) Serbia, Belgrade, 2018.

[4] Chengji Liu ,Yufan Tao ,Jiawei Liang ,Kai Li ,Yihang Chen, "Object Detection Based on YOLO Network", 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC 2018).

[5] Priya Garg, Debapriyo Roy Chowdhury, Vidya N. More, "Traffic Sign Recognition and Classification Using YOLOv2, Faster RCNN and SSD", 10th ICCCNT 2019 July 6-8, 2019, IIT - Kanpur,Kanpur, India. IEEE-45670.

[6] Felipe Sisido, Jonas Goya, Guilherme S. Bastos and Audeliano W. Li, "Traffic Signs Recognition System with Convolutional Neural Networks", American Robotic Symposium, 2018.

[7] 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.

[8] Towards Data Science blog, "YOLO Object Detection with OpenCV and Python", Available: https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9

[9] "YOLO", Available: https://pjreddie.com/darknet/yolo/

[10] "OpenCV tutorials", Available: https://docs.opencv.org/2.4/doc/tutorials/tutorials.html

[11] "TensorFlow Object Detection API tutorial", Available: https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/

[12] "Google Text-to-Speech", Available: https://gtts.readthedocs.io/en/latest/