REAL TIME DRIVER ASSISTANCE TECHNIQUE TO REDUCE ROAD ACCIDENT PROBABILITY USING COMPUTER VISION

Ranjana S.Jadhav[1]
PhD Scholar SPPU
ranjana.jadhav@vit.edu

Himanshu Pandey[2]
himanshu.pandey18@vit.edu

Bhargav Pawar[3]
bhargav.pawar18@vit.edu

Nishant Bhat[4]
nishant.bhat18@vit.edu

Abhijit Gawai[5]
Abhijit.gawai18@vit.edu

Shyam Kawale[6]
Shyam.kawale18@vit.edu

Abstract:-Discussion statistics says that more than 1.25 million road traffic deaths occur worldwide every year and most of them occur due to the negligence of the driver. These accidents are increasing day by day and there seems to be only one possible approach to reduce these accidents that is by following the traffic rules. Traffic rules are often shown on sign boards. To follow traffic rules a particular individual must keep track of all the upcoming traffic sign boards on their way. Driver might miss to look at a sign board. Sometimes a driver might not understand the meaning of sign board. Thus understanding every sign board on road prove to be a challenging task. Thus to assist and reducing the work of the driver we have introduced 'REAL TIME DRIVER ASSISTANCE TECHNIQUE TO REDUCE ROAD ACCIDENT PROBABILITY USING COMPUTER VISION'. This system uses image processing, neural networks and Google Map API to alert the driver of the upcoming sign board or any turn. This system will also alert about sign meaning to driver by sound. This system provides an mobile application to be installed on mobile phone. These road signs can be detected by using a phone camera fitted at the dashboard. Also driver will get alerts regarding upcoming turns on road. The main goal of this project is to detect and recognize traffic signs boards and alert the driver which will help in reducing the worldwide traffic deaths.

Keywords:-driver alert system, image processing, neural networks, *google map API*

1 INTRODUCTION

On average 1214 accidents take place in India per day. 377 people lose their lives in accidents per day.
Around half a million casualties are recorded every year. These accidents are avoidable barring the negligence to which the larger society pertains to. Ignoring the sign boards, poor roads and city planning are some of the very avoidable reasons responsible for a large portion of the road mishaps. 70 % of people involved in these accidents are the young, productive population, the next generation. A report by World Bank states that the GDP could increase by a mammoth 7% if these accidents are avoided. The severity of this issue is immense. Following traffic signs can play a huge role in avoiding accidents and the further consequences it causes. Traffic signs provide valuable information about the rules that are there to protect you. Traffic signs provide information about sharp turns, speed limits, and they help maintain safe driving environment. In order to solve these concerns surrounding the road mishaps we plan to introduce a REAL TIME DRIVER ASSISTANCE TECHNIQUE. It can detect the traffic signs from a distance with the images that are captured with the cameras. It can detect traffic signs and also classify them in real time giving the driver a head start. Image processing plays a very important part in this process. Image processing processes images with the help of a computer and a developed algorithm. It is useful in extracting the needed information from the image as in this case we need it to alert the driver regarding the traffic signs. The system also makes the use of neural networks and Google map API to alert the driver of any acute turns. This system thus aims to develop a solution which aims to reduce the number of mishaps occurring due to the above mentioned reasons.

2 Literature Review:-

Safat B. Wali et al [1] developed a traffic sign detection and recognition (TSDR). They got the accuracy of 95.71%, false positive rate of 0.9% and processing time of 0.43s. They collected sample images using a less expensive in carcamera linked to a laptop. They have taken into consideration shape matching, color segments and other few features.
Anju Manjooran et al [2] used the image processing by SURF algorithm in MATLAB to identify the signboards. They installed a camera in the vehicle to capture the signboard and provide assistance to driver in traffic sign detection with the help of voice alert accompanying speed control.

Anushree. A.S. et al [3] developed a Autonomous Driving Assistance System (ADAS) using a raspberry Pi camera module to detect traffic signs. They have used a LCD screen and speakers to display or provide alert messages.
Ford [4] has developed Ford Traffic Sign Recognition System. They have installed a camera which detects traffic sign and displays it on the LCD. They have used an aging algorithm such that newer signs appear brighter than the older ones.

3 METHODOLOGY

3.1 Brief overview of the system

1 NEURAL NETWORK ARCHITECTURE

```
layer     filters    size              input                output
    0 conv     16  3 x 3 / 1   416 x 416 x   3   ->   416 x 416 x  16  0.150 BFLOPs
    1 max          2 x 2 / 2   416 x 416 x  16   ->   208 x 208 x  16
    2 conv     32  3 x 3 / 1   208 x 208 x  16   ->   208 x 208 x  32  0.399 BFLOPs
    3 max          2 x 2 / 2   208 x 208 x  32   ->   104 x 104 x  32
    4 conv     64  3 x 3 / 1   104 x 104 x  32   ->   104 x 104 x  64  0.399 BFLOPs
    5 max          2 x 2 / 2   104 x 104 x  64   ->    52 x  52 x  64
    6 conv    128  3 x 3 / 1    52 x  52 x  64   ->    52 x  52 x 128  0.399 BFLOPs
    7 max          2 x 2 / 2    52 x  52 x 128   ->    26 x  26 x 128
    8 conv    256  3 x 3 / 1    26 x  26 x 128   ->    26 x  26 x 256  0.399 BFLOPs
    9 max          2 x 2 / 2    26 x  26 x 256   ->    13 x  13 x 256
   10 conv    512  3 x 3 / 1    13 x  13 x 256   ->    13 x  13 x 512  0.399 BFLOPs
   11 max          2 x 2 / 1    13 x  13 x 512   ->    13 x  13 x 512
   12 conv   1024  3 x 3 / 1    13 x  13 x 512   ->    13 x  13 x1024  1.595 BFLOPs
   13 conv    512  3 x 3 / 1    13 x  13 x1024   ->    13 x  13 x 512  1.595 BFLOPs
   14 conv     45  1 x 1 / 1    13 x  13 x 512   ->    13 x  13 x  45  0.008 BFLOPs
   15 detection
```

Fig.1. Neural Network Architecture

Keywords:

Bounding box - In a given image, coordinates of a box around interested area.

Ground truth - Original bounding box of image.

IoU - Measure of accuracy of the predicted bounding box with respect to ground truth.

Area of Overlap - Overlapping area between predicted bounding box and ground truth.

Area of Union – Predicted bounding box and ground truth.

Accuracy - The fraction of right predictions $\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Number Of Examples}}$

Recall - Measure of actual positives identified correctly $\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$

Iteration - Update of a parameters in model while training.

Batch - The number of examples used in one iteration.

Training - The process of adjusting value to parameters.

Dataset – A collection of Examples.

Class – label of target values.

Neuron - A node in a neural network which give a output value from set of input values.

Layer- A group of neuron in a neural network which takes input and gives output.

Input layer - The layer that gets the input in a neural network (First layer in neural network).

Output layer - The layer that give output in a neural network (Final layer in neural network).

```
Loaded: 0.000074 seconds
Region Avg IOU: 0.923101, Class: 0.999858, Obj: 0.999473, No Obj: 0.002197, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.886735, Class: 0.998481, Obj: 0.996786, No Obj: 0.004255, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.804176, Class: 0.999873, Obj: 0.699929, No Obj: 0.002012, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.898169, Class: 0.999985, Obj: 0.999931, No Obj: 0.004155, Avg Recall: 1.000000,  count: 2
11678: 1.198501, 0.329463 avg, 0.001000 rate, 0.285405 seconds, 93424 images
```

Fig.2. TRAINING ANALYSIS

Fig.3. SUBDIVISION OUTPUTS

IOU (Intersection Over Union) is a great measurement for accuracy of prediction done for each image. It is also known as Jaccard Index. Formula to calculate this measure is-

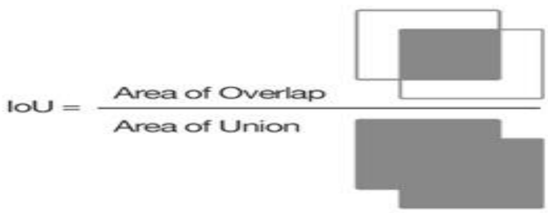

Fig.4. IOU formula

This formula calculates the ratio of the area of overlapping area of predicted bounding box for the image and the original bounding box of the image (ground truth). This gives us the idea about the accuracy of prediction. More the overlapping of the predicted and original image's bounding box, the better is the model. Hence , if IOU closer to one, better is the model.

Average IOU is the average of each IOUs of predicted bounding box in a subdivision. In our case since the dataset is small batch size is of 8 images, and there are 4 subdivisions of the batch. Hence, there are 2 images in each subdivision. The average of IOUs, calculated for each of the two images is the average IOU.

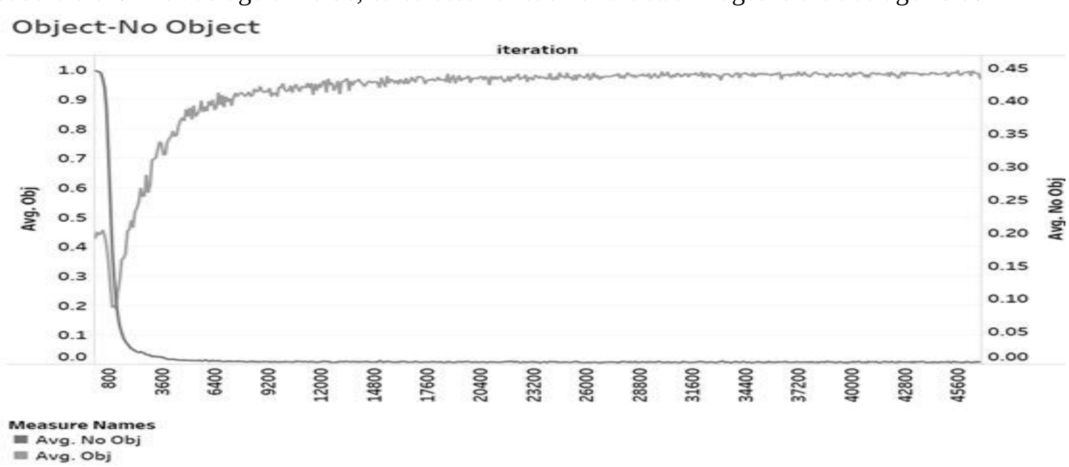

Fig.5. dual axis graph showing average object and average no object.

Average Object is the average of confidence levels of each image in a subdivision that the predicted image contains the labelled object of the original image. In this case each subdivision has 2 images, so for each image the model gives us the probability of being the object in the predicted image as labelled in the data, and then average of those probabilities is taken. More the average object value close to 1, better is the model.
Average No Object is the average of confidence levels of each image in a subdivision that the predicted image does not contains the labelled object of the original image. More the average object value close to 0 better is the model.

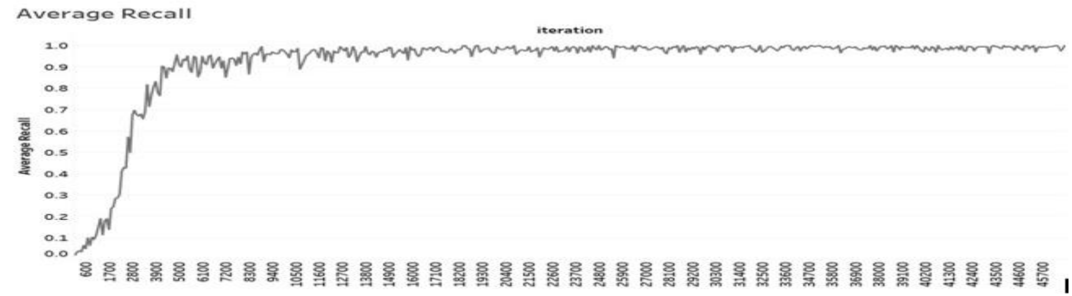

Fig.6. Average Recall

Average Recall is defined as recall/count in the code. This is the ratio of total number of predicted positive images and total number of positive images in the subdivision. If there are 16 images in a subdivision and our model predicted 8 images correct out of 16 then Recall will be 0.5. More the recall closer to 1 better is the model.

Fig.7. Class

Class is the mark for appropriate rate of classification of objects. Value of class more closer to 1, better is the model.

BATCH OUTPUT

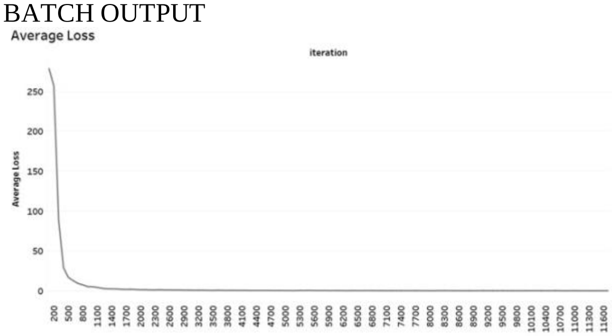

Fig.8. Average Loss

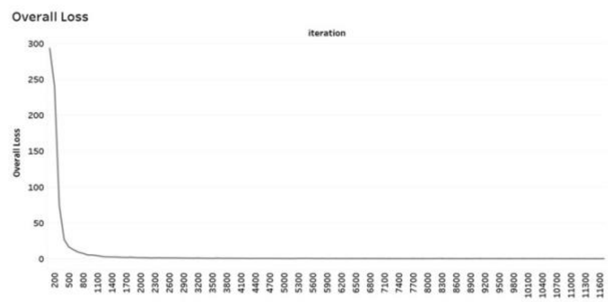

Fig.9. Overall Loss

2 YOLO:

YOLO stands for You Only Look Once. YOLO is a neural network designed for object detection. It detects objects in given image in real time. Yolo is used to solve problem of object detection. Object detection is an extension of Image Classification. Image classification is a subset of computer vision. Image classification is used to recognize presence of object in a given image. Image Classification recognizes only one class. If we use Image Classification on this image-

Fig.10. Result of Image Classification Dog(60%) ,Bicycle(25%), Car(15%)

Here Image Classification tells that only object present in image is Dog with 60% confidence. Bicycle and Car will not be considered as object within image as we got low confidence score for them. But we know that image also contains bicycle and car, but Such problem occurs in Image Classification when multiple objects are present inside image. Image classification recognizes object inside image. But it doesn't talk about location of object inside image.
Object Detection is a becomes difficult than Image classification, because it requires object location in results. In Object Detection we recognize objects and locate object inside an image and recognizes more than one object within an image. For example if we run Object detection for this Image

Fig.11. Dog, bicycle and car detected and located by object detector

Object Detection recognizes dog, bicycle and car. So we got results as expected.

How YOLO works:

We can take a image classifier and change image classifier into an object detector. We slide a small grid across the given image. After every small movement of a sliding window will give several predictions of image classification of that image, but we will consider results with which image classifier has more prediction score. This approach will run image classifier many times. It will also take longer time. YOLO takes a completely different approach. YOLO use input image for only once (hence its name). YOLO divides the given image in a grid structure. Generally we set grid of 13 * 13 cells.

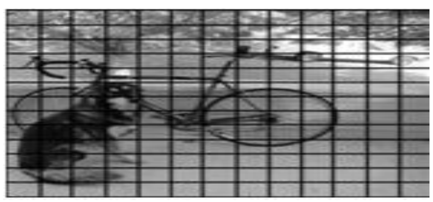

Fig.12. Image divided into 13*13 grid cells

For each of these small grid cells, we predict 5 bounding boxes. A bounding box is just a rectangle that has an object inside it. YOLO gives output as a number called as confidence score. This is output we get from last layer of neural network. This output tells probability of an object available inside bounding box. From this score we cannot tell which object is there in that box.

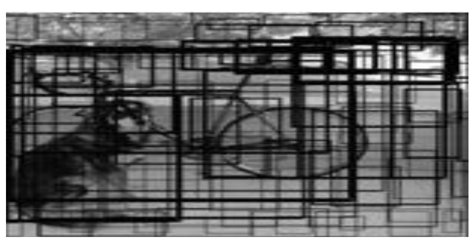

Fig.13. Bounding Box (the higher the confidence score, the fatter the bounding box is drawn)

For every box we got, there is a class predicted for every cell. We do image classification, we get a confidence score of all the available classes for that box. From this confidence score and bounding box we finalize the score as final score. This final score tells us probability of bounding box containing a available object of a class. In below image on left hand side, the huge box is 60% sure that it contains the object of class "dog"

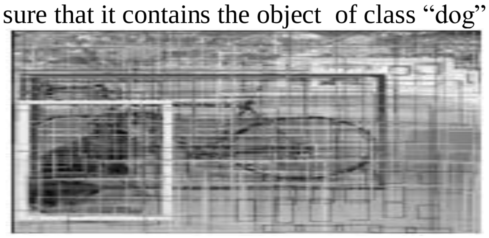

Fig.14. Class prediction for bounding box

Since we have divided image into 13 × 13, we got total 169 grid cells. All of these grid cells cell is predicted for number of classes we have (5 classes). So we get 169×5= 845 different bounding boxes. To discard repeated predictions of same object we have to discard some bounding boxes. Here we choose to discard boxes whose final score is less than 40% (we can change this threshold if we still get some repeated results). After removing class with final score less than threshold we get final prediction.

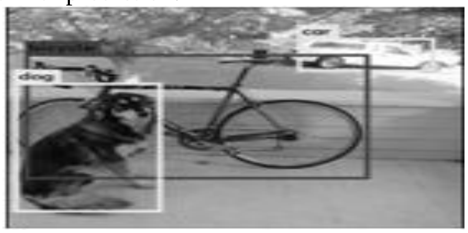

Fig.15. Result of Object Detector (Dog, bicycle, car) with location

Here we get 845 bounding boxes. Out of that we consider only three bounding boxes because we got highest predictions for that boxes. Although we got 845 different predictions, but we got all predictions in one go. Our neural network made all predictions at same time. That is why YOLO makes faster predictions.

The neural network of YOLO:

The architecture of YOLO consists of a convolutional neural network.

Layer	kernel	stride	output shape
Input			(416, 416, 3)
Convolution	3×3	1	(416, 416, 16)
MaxPooling	2×2	2	(208, 208, 16)
Convolution	3×3	1	(208, 208, 32)
MaxPooling	2×2	2	(104, 104, 32)
Convolution	3×3	1	(104, 104, 64)
MaxPooling	2×2	2	(52, 52, 64)
Convolution	3×3	1	(52, 52, 128)
MaxPooling	2×2	2	(26, 26, 128)
Convolution	3×3	1	(26, 26, 256)
MaxPooling	2×2	2	(13, 13, 256)
Convolution	3×3	1	(13, 13, 512)
MaxPooling	2×2	1	(13, 13, 512)
Convolution	3×3	1	(13, 13, 1024)
Convolution	3×3	1	(13, 13, 1024)

Fig.16. Neural Network Architecture of YOLO

3 DRIVER ALERT SYSTEMAPP DESCRIPTION:

Our app is all-in-one platform designed for all types of travellers, drivers, and also for regular person who is associated with driving in some way.

Our app basically provides 3 main features to our user i.e. Road Safety Sign Detection, Obstacle Detection, and Acute Turn Detection. We have integrated Image processing (YOLO-Model) for detecting sign board and Obstacles. Also we have integrated all features with user friendly and very interactive mobile UI. All this features can be accessed with only one click on app!!

Other features:

-Detect and Announces Road Signs using camera,

-Map Navigation,

-Different Views on map(Hybrid, Satellite, Terrain),

-GPS tracker for getting location,

We have also provided one feature for checking if camera is working properly without any problems.

External plugins that we have used in our apps are:

Camera:

It is a Flutter plugin to access phone camera. As we are using flutter environment it will support both Android and IOS platform. It supports capturing images, capturing video, previewing the camera feed, and streaming image buffers to dart. We used this plugin for accessing mobile camera to capture real-time video stream in front of driver’s vehicle and also to check whether camera is working properly without any errors.

tflite:

A Flutter plugin for accessing Tensor Flow Lite API. This API supports many machine learning functions like image classification, object detection (YOLO and SSD), Pix2Pix, Deeplab and PoseNet on both platform Android and iOS devices. We used this plugin for integrating image processing model in our app to detect road signs, and obstacles in front of vehicle.

Google_maps_flutter:

A Flutter plugin for integrating Google Maps api on both Android and iOS applications. We integrated Google maps in our app using this plugin and also build different views on map which could help driver to check for acute turn ahead him.

Location:

This plugin for Flutter for getting and handling location on both Android and iOS applications. It also provides call-backs when there is change in location. We used this plugin for getting live location of our vehicle.

So, Put your phone into car holder. Drive and get real time updates on what sign, obstacles and turns are ahead of you!!!!

Our main motto is Drive Safe and live tension free with all updates on your mobile phone.

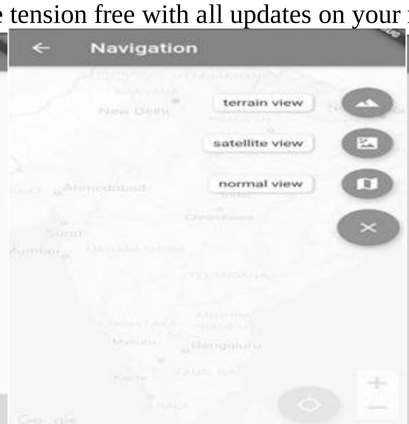

Fig.17 **Fig.18** **Fig.19**

4 Result

By using our models in application we evaluated the accuracy of the traffic sign detection and classification through experiments. This are some photos of result the we got:

Fig.20 **Fig.21**

Fig.20. and Fig.21. are the screenshots of app while detecting traffic road signs.

We also did some experiments on map navigation and gps tracker:

Fig.22. shows real time GPS tracking of driver while driving also showing acute turns in front of vehicle.

Our detection accuracy result and map navigation and GPS tracking result are very promising and we intend to improve it as further as we can.

5 Limitations

It becomes difficult to detect traffic board signs incase of fog or any other visual disturbance as we are using a mobile camera at present. This also leads to lower range so driver gets lesser time to react to a particular board sign.

6 Future Scope

All weather cameras can be used in place of a mobile camera which will highly increase the quality of the whole system.
Automatic speed control system can be added which will help in reducing number of accidents
Obstacle detection feature can be added.

7 Conclusion

As we are aware that about a million traffic deaths occurs worldwide and most of them occur due to the sheer negligence of the driver. By realizing this problem we decided to come up with a driver assistance technique to reduce road accident probability using computer vision and thus successfully implemented the same.This system alerts the driver of the upcoming traffic board sign or a upcoming turn thus making him more aware of his surroundings which will in turn help in reducing the number of traffic deaths worldwide by a great margin.

8 Acknowledgment

Our team want to seize this opportunity to express appreciationtowards the following professionals: Firstly, we would like to thank the Honourable Director of Vishwakarma Institute of Technology, Prof. (Dr.) R.M. Jalnekar Sir for including the concept of Engineering Design and Innovation (EDI) in our syllabus, and our HOD Premanand Ghadekar for giving us this opportunity to work on this innovative project . Finally, we would specially like to thank our guide Mrs. Ranjana Jadhav for her valuable support and guidance.

9 References

[1]. "Traffic Sign Board Detection and Voice Alert System Along with Speed Control Anju Manjooran, Anphy Varghese, Annmariya Seby and Krishnadas J Asian Journal of Applied Science and Technology (AJAST) Volume 2, Issue 1, Pages 28 1-286, 2018"
[2].https://www.researchgate.net/publication/331319583_REVIEW_ON_CLASSIFICATION_ALGORITHMS_IN_IMAGE_PROCESSING
[3]. "Traffic Sign Board Detection" by Annmariya Seby.
[4]. "Detection and Recognition of Alert Traffic Signs" by ChiaHsiung Chen, Marcus Chen, and Tianshi Gao.
[5].https://pjreddie.com/darknet/yolo/
[6].https://arxiv.org/pdf/1506.02640.pdf
[7].https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
[8].https://www.mdpi.com/1424-8220/19/9/2093
[9]. https://pub.dev