Traffic Light Classification Using YOLO Detection

Nadar Rajeshwari[1], Harshitha Aditham[2], Prof. Anita Morey[3]
[1, 2, 3] Dept of Information Technology
[1, 2, 3] Usha Mittal Institute of Technology

Abstract- *The aim of this research paper is to classify the traffic lights on the road to guide the self driving car. Using Yolo (You Only Look Once) object detection which has higher speed and relatively accurate. Analyzing objects and figuring out the lane lines on a given roadway to follow the traffic rules. Considering all these competencies with the self driving car obstacles coming in the road are also need to be taken care of, for which the yolo object detection with sliding window techniques are considered. These methods are solved along with deep leaning models (CNN) for autonomous vehicles.*

Keywords- YOLO object detection, Annotations, Self driving car, Traffic light, classification.

I. INTRODUCTION

Object detection may sound like a pinnacle in Artificial Intelligence but it co-exists with us in our lives hiding in plain sides. We often fail to notice the simple applications of object detection around us. Object detection is a technology that includes computer vision and image processing used to detect objects, images or videos. Self driving cars make use of the moving object detection technology along with computer vision to determine the distance between the car and the moving object also to create alerts and guide the car. The object detection and recognition are considered to be one of the most important tasks as this is what helps the vehicle detect the obstacles in-front of the car. Deep learning and computer vision both are now helping self driving cars to figure out the competencies faced by the autonomous cars. The competencies can be to predict the other cars on the road and the pedestrians around the road so that to avoid them and have a safe drive. Deep learning is able to make face recognition work much better than ever before so that now we are able to unlock our computer screens and phone screens just by using our face. Deep learning is moving so advance that companies are able to build applications on our phones with most attractive, interactive and most beautiful pictures and features to attract the customers. Therefore, it is necessary for the object detection algorithms to be highly accurate. Since machines cannot detect the objects in an image instantly like humans, it is really necessary for the algorithms to be fast and accurate and to detect the objects in real-time, so that the vehicle controllers solve optimization problems at least at a frequency of one per second. In this research paper traffic light prediction for autonomous vehicles and the object detection on the roads is done with the help of Yolo object detection.

II. LITERATURE REVIEW

Computer vision research community has been so inventive is coming up with new neural network architectures, new algorithms. Image classification sometimes also called image recognition where you might take as input image 64 X 64 image and try to figure out if static. The another problem for a computer vision problem is object detection, suppose in self-driving car we need to figure out if there are other cars in this image but instead the position of the other cars need to be figured out so that the car can avoid other cars. By drawing bounding boxes it will be able to find out the position where the object is present in the image.

Computer vision is the area of study in which computers are empowered to visualize, recognize and process what they see in a similar way as that of humans. The main aim of computer vision is to generate relevant information from image and video data in order to deduce something about the world. It can be classified as a sub-field of artificial intelligence and machine learning. This is quite different from image processing, which involves manipulating or enhancing visual information and is not concerned about the contents of the image. Applications of computer vision include image classification, visual detection, 3D scene reconstruction from 2Dimages, image retrieval, augmented reality, machine vision and traffic automation. Today, machine learning is a necessary component of many computer vision algorithms. These algorithms are typically a combination of image processing and machine learning techniques. The major requirement of these algorithms is to handle large amounts of image/video data and to be able to perform computation in real-time for wide range of applications.

For example, real-time detection and tracking. There are various types of artificial neural networks that are considered to be very important such as Radial basis function neural network, Feed-forward neural network, Convolutional neural network, Recurrent neural network, Modular neural network etc. Among these types of networks, the convolutional neural networks (CNNs)are effective in applications such as image/video recognition, semantic parsing, natural language processing and paraphrase detection. A convolutional neural network comprises of 3 layers – Convolutional neural layer, Pooling layer and Fully-connected layer.A convolutional layer generally executes tasks that require heavy computation. It comprises of a set of filters that have the ability to learn. Though the filters are small in size, they reach to the entire depth of the input.

III. YOLO OBJECT DETECTION

Yolo object detection is one of the best algorithms used for object detection. It is an improvement over other region based algorithms because of the speed, accuracy and the ability to deal with multiple objects in the same frame. YOLO is also referred as YOU ONLY LOOK ONCE. It requires only one forward propagation on the image to make the predictions. The features used by these algorithms passed on the entire image and make bounding boxes for objects. The features help to predict all the bounding boxes across the image and make classes of similar objects. It simultaneously predicts the output after recognizing the object with a process called non-max suppression.

The basic idea in yolo object prediction is to take any input image and divide the image into blocks also called as “grids” with equal sizes. Then these grids are used to make a bounding box along with the label. The co-ordinates of the image are also processed and stored. The co-ordinates are the exact location of the objects in the image. Predict a class and bounding box of the objects present in grid of original image.
Working of YOLO:

- In YOLO we provide input as an image to a Convolutional neural network which outputs a particular vector which consists of the class probabilities for that object. The class to which that particular object belongs to and the coordinates of the bounding box for the object.
- In sliding window approach take a particular dimension of window and slide it over the input image stride by stride. Stride allows to decide what is the overlap of each sliding window. In YOLO instead of using stride of finite number, stride with 0 is used. Stride with 0 is basically to show no overlap on each sliding window. These results in the image divided into grids. There are no overlaps in the whole image and it is divided simply into grids.

We split the image into an S*S grid. Each cell predicts B number of bounding boxes with a confidence score for each of the boxes. Confidence score means how confident the model is that the box contains an object and how accurately the box has predicted the object boundaries means how accurately these (x, y, h, w) are estimated.

A) YOLO OBJECT DETECTION INPUT AND OUTPUT:

- Suppose a grid in YOLO CNN is of size 16 X16.
- In the output, each bounding box is represented by 6 numbers(pc, x, y, h, w, c)
- Here class probability c is a 1-hot vector of size 20.
- Every bounding box that is predicted called as anchor box.
- Thus each anchor box will be represented by a vector of size 25.
- If we get b = 4, then YOLO produces 3 anchor boxes for each grid cell.

Then the YOLO architecture has the following input/ output for each batch and the flowchart is shown in figure 1.

IMAGE (M, 304, 304, 3) →YOLO CNN → ENCODING (m, 19, 19, 2, 25).

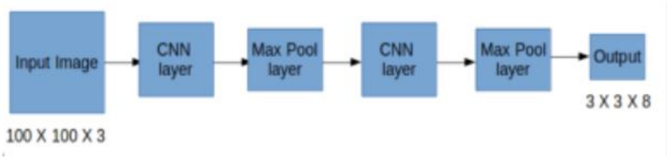


Figure 1: Flow chart for YOLO

YOLO’s PREDICTION:

- In each of the 19 x 19 grids, the grids that have maximum probability score are choosen.
- The maximum probability score is predicted by taking any 2 anchor boxes and finding max between them which are across different class.
- Color that grid cell according to what object that grid cell considers the most likely.

DEALING WITH ANCHOR BOXES:

- Two stage filtering out of anchor boxes.
- Set a threshold on confidence of a box detecting a class.
- Ignore boxes with a low score, that is, what the box is not very confident about detecting a class

BOX CONFIDENCE:

- Anchor box confidence depends upon two factors:
- How confident the model is that the box contains an object and
- How accurate it thinks the box is that it predicts.
- Intersection Over Union-It is a measure of overlap between the actual (ground truth) bounding box and the predicted bounding box.

First Level of Filtering Out (Boxes):

Remove all those boxes whose scores are less than threshold.

Second Level of Filtering Out (Non Max Suppression)

1. Select any bounding box from the image that has the highest score.
2. Compute its overlap with all other boxes, and remove boxes that overlap it more than the threshold set for IOU (Intersection Over Union)

YOLO CNN Specification:

- The features from the input image are extracted initially by the Convolutional layers and the probalities are predicted by the fully connected layers of the CNN
- The CNN network architecture is inspired by the GoogleNet model for image classification.
- The 24 Convolutional layers are followed by two fully connected layers of CNN.

Yolo is better than region-based algorithm, because it was much slower and error was more as compared to YOLO algorithm. Yolo prioritize the speed; this is particular are used in application like self-driving car.

The self-driving car is an excepted to detect the objects on its path with much more speed and accuracy, then the current system allows. A self-driving car can look at an object and recognize the object whether, the image or a video are able to identify whether the object is present in that and detect what kind of object is present.

B. Yolo object detection implementation for self driving car

In yolo input image is provided to a convolutional neural network, which outputs a particular vector which consist of class probability of that object, the class to which the particular object belongs to and coordinates of the bounding box.

a) Sliding window approach

In a sliding window approach, particular dimension of a window and slide it over an image stride by stride. Stride allows us to decide, what is the overlap of each sliding window. In Yolo instead of using stride of finite number, stride of 0 is been used, which basically gives no overlap between each sliding window and it result into image been divide into grids, so imagine that there are no overlap between sliding window and the whole image is divided into grids of particular dimensions as shown in figure 2.

Figure 2: Sliding window approach

b) Anchor box

If two or more object are present in same grid or if a single object is present in two or more grids, then angular box is used. Two anchor boxes are used, one vertically and one horizontally, ideally, we can choose as many as anchor box .which are required; hence it is particular aspect ratio. When anchor boxes are used then, the output vectors are also changed according to the output of two anchor boxes for a particular grid cell. It is shown in figure 3.

Figure 3: Anchor Box approach

C. Equations

1. pc be the probability of class present.
2. c1, c2…. cn be the probability of each class present.
3. (bh, bw) - it detects height and width of the bounding box
4. (bx, by) -co-ordinates of the bounding box.

pc, bh, bw, bx, by are the five fixed parameters.

Let n number of class is present, m number of anchor box we are using and for above example there are 3 x 3 grids. So, y = (5 + n) x m x 3 x 3

The grid object detection is shown in figure 4.

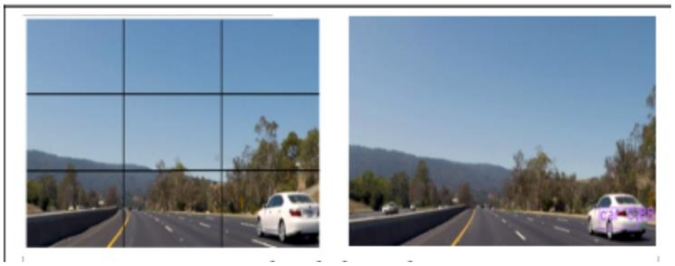
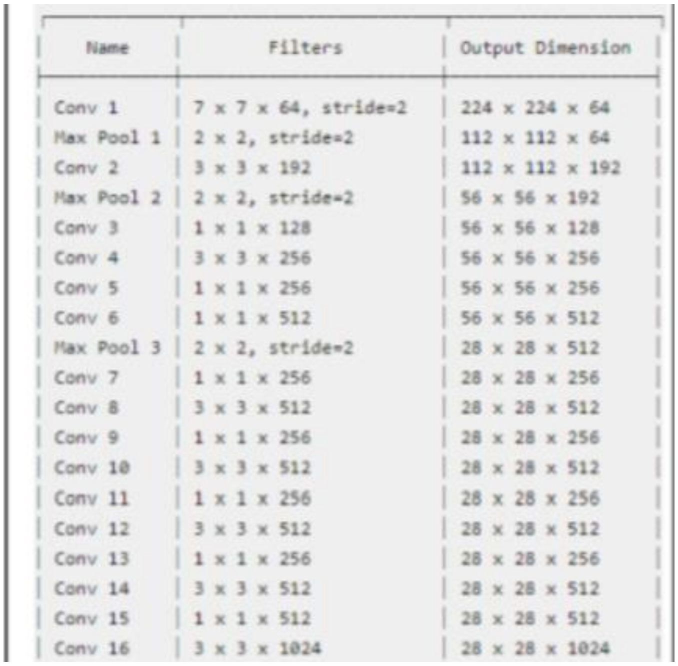


Figure 4: Grid and object detection

D. Network

The network structure of CNN model in yolo, with convolutional and max pooling layers, for 2 fully convoluted layers are, Calculate the x-coordinate, y-coordinate, height and width of the object detected. After analyzing for many frames of the video, it is found the x-coordinate and the height of the object changes instantly, so while analyzing the for these specific image, it was found that, giving value x-coordinate < 900 and height > 300 will be the static value for getting the object in same lane where the actual car mounted camera is placed. The result of CNN model is shown in following figure 5.



Name	Filters	Output Dimension
Conv 1	7 x 7 x 64, stride=2	224 x 224 x 64
Max Pool 1	2 x 2, stride=2	112 x 112 x 64
Conv 2	3 x 3 x 192	112 x 112 x 192
Max Pool 2	2 x 2, stride=2	56 x 56 x 192
Conv 3	1 x 1 x 128	56 x 56 x 128
Conv 4	3 x 3 x 256	56 x 56 x 256
Conv 5	1 x 1 x 256	56 x 56 x 256
Conv 6	1 x 1 x 512	56 x 56 x 512
Max Pool 3	2 x 2, stride=2	28 x 28 x 512
Conv 7	1 x 1 x 256	28 x 28 x 256
Conv 8	3 x 3 x 512	28 x 28 x 512
Conv 9	1 x 1 x 256	28 x 28 x 256
Conv 10	3 x 3 x 512	28 x 28 x 512
Conv 11	1 x 1 x 256	28 x 28 x 256
Conv 12	3 x 3 x 512	28 x 28 x 512
Conv 13	1 x 1 x 256	28 x 28 x 256
Conv 14	3 x 3 x 512	28 x 28 x 512
Conv 15	1 x 1 x 512	28 x 28 x 512
Conv 16	3 x 3 x 1024	28 x 28 x 1024

Figure 5: CNN Output

YOLO Features:

- Computationally Very Fast, can be used on real time environment.
- Globally processing the entire image once only with a single CNN.
- Learn generalized representations
- Maintains a high accuracy range.
- The object detection with YOLO is possible with all kinds of input files such as images, video files and webcam real-time capturing.

Yolo Algorithm Limitations:

- YOLO inflicts strong spatial constraints on the bounding box predictions as each grid cell predicts two boxes at a time and it can only have one class.
- Due to this spatial constraint the number of near by objects is also limited that the given model can predict.
- This object detection model struggles with small objects that appear in groups within the same grid.

It also struggles to generalize to objects in new or unusual aspect ratios.

IV. TRAFFIC LIGHT DETECTION

Traffic light is one the most important traffic rule which everyone should maintain, so in self driving car also the processor should automatically detect the traffic light signals. Each frame which is captured by the camera should be send to a specific algorithm to detect whether that frame contains the traffic light or any traffic symbol, the yolo object which was discussed, used coco model. This model can detect car, person, animals etc., so if the frame contains any object present in coco.names, it can be detect otherwise it cannot detect it, in coco model there is also object know as traffic light, but it can only detect the traffic light, but cannot classify whether the traffic light isgreen or red, to make decision whether the processor should stop the car or else keep moving. So, in order to consider that condition, lots of dataset of red and green images are needed, so that, a separate model can be created.

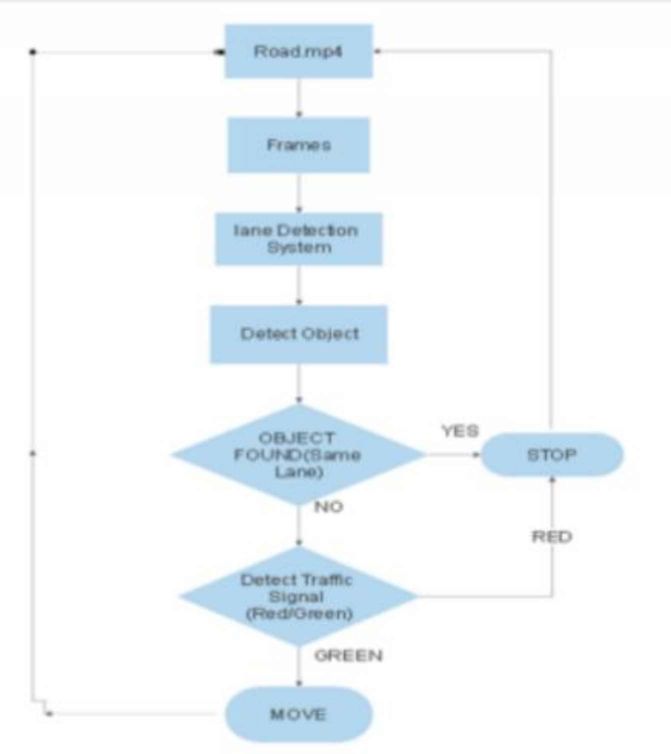


Figure 6: Flow chart for Traffic light Detection

The flowchart shown in figure 6 gives a glimpse about how the traffic light is detected, if traffic light detected is “GREEN” then keeps moving and if it is “RED”, stop the car and wait for the next frame to be processed. So, to make the model to classify “RED” and “GREEN” signal, following steps are present: -

A. COLLECT IMAGES OF RED AND GREEN LIGHT:

Collect 500 to 1000 images from Google or any online site, in order to train the model using the images, more the number of images more the accuracy of classifying the traffic signal.

B. ANNOTATE THE IMAGE:

Use a software know as “LabelImg”, this will help to label the image, to give “RED” or “GREEN”. The annotation of image for traffic light detection is shown in figure 7.

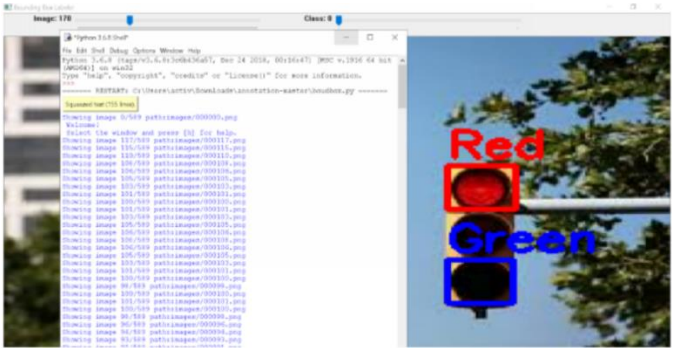
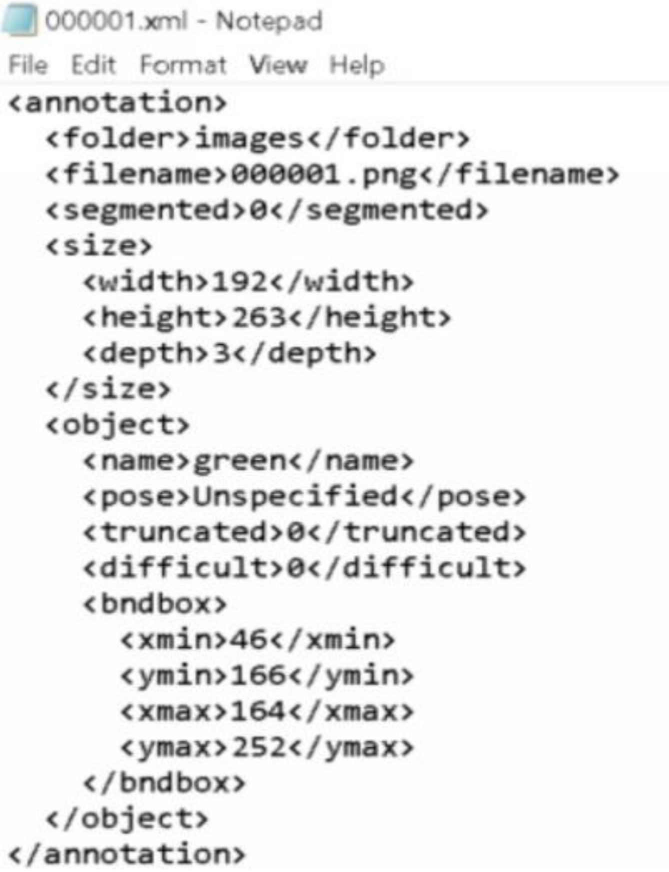


Figure 7: Annotate the image

C. CREATE XML FILE

After giving naming using annotation, each annotated image will have a separate xml, which says about the folder name, image name, image width, height and the label image



```
<annotation>
  <folder>images</folder>
  <filename>000001.png</filename>
  <segmented>0</segmented>
  <size>
    <width>192</width>
    <height>263</height>
    <depth>3</depth>
  </size>
  <object>
    <name>green</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>46</xmin>
      <ymin>166</ymin>
      <xmax>164</xmax>
      <ymax>252</ymax>
    </bndbox>
  </object>
</annotation>
```

Figure 8: XML File

D. TRAIN MODEL

This xml will be sending for training using “Yolo method”, and there will be separate name file to train the model, which consist of only “RED and GREEN”. After this procedure, a yolo.h5 modelfile is created and this file will be used to classify the traffic light. To train the model we used steps which are represented by a flowchart as given in figure 9.

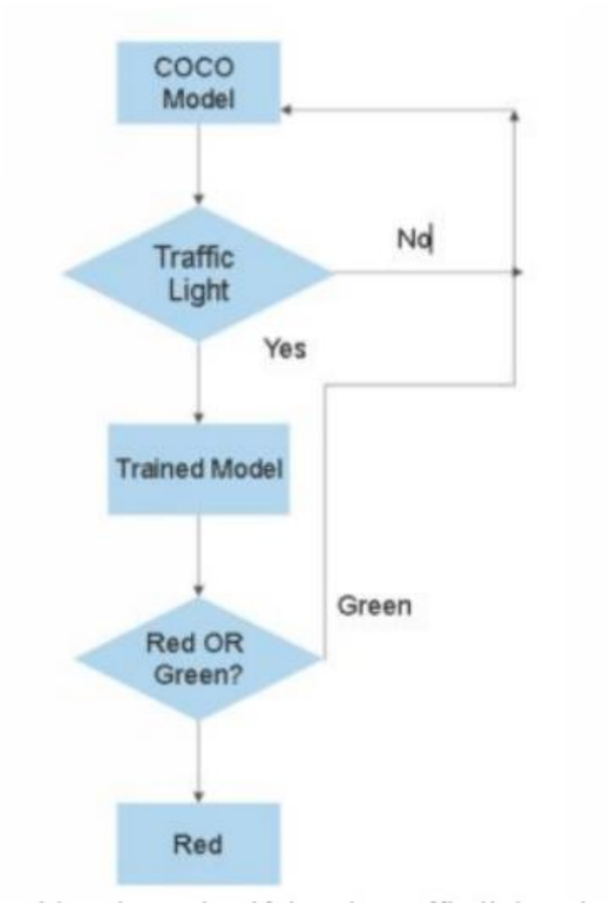


Figure 9: Flow chart for traffic light training model

E. FINAL OUTPUT:

Above flowchart (figure 9), gives idea about classifying the traffic light using the new trained model, whenever the traffic light is detected using the coco model then it is sent to new trained model "yolo.h5" because if single frame is sent to two model at a time, then unnecessary time complexity in classifying the traffic light in a frame which doesn't contain the traffic light will be reduced. So, after the frame enter the coco model to detect the traffic light, if it detects the traffic light then only it enter then it moves trained model, which is used to classify "RED" forward else it will stop. "signal, so if frame detects "GREEN" as shown in figure 10.

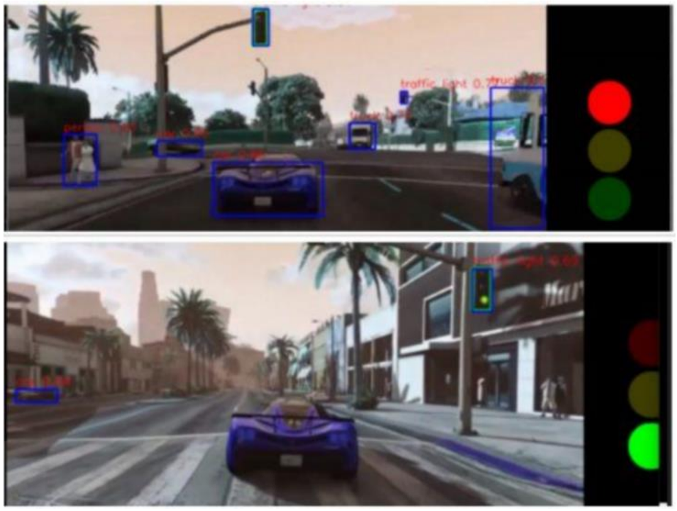


Figure 10: Traffic Light Prediction

V. CONCLUSION

Above flowchart gives idea about classifying the traffic light using the new trained model, whenever the traffic light is detected using the coco model then it is sent to new trained model "yolo.h5" because if single frame is sent to two model at a time, then unnecessary time complexity in classifying the traffic light in a frame which doesn'tcontain the traffic light will be reduced. So, after the frameenter the coco model to detect the traffic light, if itdetects the traffic light then only it enter the trained model, which is used to classify "RED" and "GREEN" signal, so if frame detects "GREEN" then it moves forward else it will stop.

REFERENCES

[1] Liu, C., Tao, Y., Liang, J., Li, K., & Chen, Y. (2018). *Object Detection Based on YOLO Network. 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC).*

[2] Adarsh, P., Rathi, P., & Kumar, M. (2020). *YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. 2020 6th International Conference on Advanced Computing*

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[4] Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., &Haq, A. ul. (2020). *Object Detection through Modified YOLO Neural Network. Scientific Programming, 2020, 1–10.*

[5] Protschky, V., Feit, S., &Linnhoff-Popien, C. (2014). *Extensive Traffic Light Prediction under Real-*

World Conditions. 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall).

[6] Protschky, V., Wiesner, K., &Feit, S. (2014). *Adaptive traffic light prediction via Kalman filtering. 2014 IEEE Intelligent Vehicles Symposium*