OBJECT RECOGNITION FOR AUTONOMOUS DRIVING SYSTEM

Nagachandra kambhathalli Parswanatha (RedId - 821728226) and
Sharath Chandra Kumar Periketi(RedId -822062066)

Abstract— **The goal of this project is to provide object detection and information on environment model on traffic activity which helps autonomous vehicles or surveillance systems. Computer vision is an essential component for autonomous scars. Accurate detection of vehicles, street buildings, pedestrians, and road signs could assist self-driving cars the drive as safely as humans. However, object detection has been a challenging task for years since images of objects in the real-world environment are affected by illumination, rotation, scale, and occlusion. A unified object detection model, You Only Look Once (YOLO), is used which could directly regress from the input image to object class scores and positions. In this project, we applied YOLO to two different datasets to test its general applicability. We fully analyzed its performance from various aspects on KITTI data set which is specialized for autonomous driving. We proposed a novel technique called memory map, which considers inter-frame information, to strengthen YOLO's detection ability in the driving scene. We broadened the model's applicability scope by applying it to a new orientation estimation task. For this project objective is to provide a information of quality and environmental model on traffic activity and to signal potentially anomalous situation and also apply various machine learning models for object detection such as SVM and CNN and compare and contrast the results with YOLO .**

I. INTRODUCTION

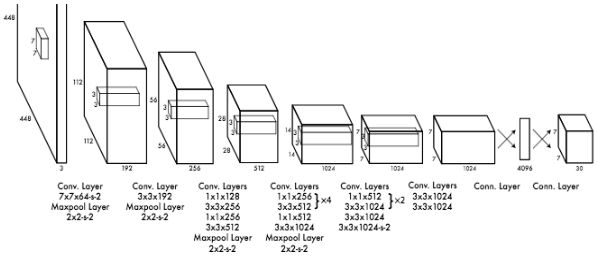
A safe and robust autonomous driving system relies on accurate perception of the environment. To be more specific, an autonomous vehicle needs to accurately detect cars, pedestrians, cyclists, road signs, and other objects in real time in order to make the right control decisions that ensure safety. In order to attain these goals, smart systems need to be developed for monitoring and understanding our surroundings so Object detection is a crucial task for autonomous driving. Different autonomous vehicle solutions may have different combinations of perception sensors, but image based object detection is almost irreplaceable. For autonomous driving some basic requirements for image object detectors include the following: a) Accuracy. More specifically, the detector ideally should achieve 100% recall with high precision on objects of interest. b) Speed. The detector should have real-time or faster inference speed to reduce the latency of the vehicle control loop. c) Small model size .

Often there are multiple targets present in the same image for the object detection system to detect. YOLO (You Only Look Once) is a fast and accurate object detection system consisting of one single convolutional network.

In this project first we analyse whether the Yolo version is an appropriate model for general object detection, for that we trained the model on data sets such as KITTI . We perform a comprehensive analysis of YOLO over Road - object detection and orientation estimation benchmark. According to the comprehensive analysis results the YOLO have a limitations in on realtime autonomous driving system. To implement this process for realtime autonomous driving system we design a novel YOLO method by combining the memory map and on-max suppression (NMS) model to make YOLO suitable for real-time object detection in video streaming and improve the YOLO potentiality by applying YOLO on orientation estimation, the YOLO can estimate the orientations of the detected objects.The KITTI Vision Benchmark Suite is specialized for autonomous driving.. KITTI also extract benchmarks for 2D/ 3D object detection, object tracking and pose estimation.

In this project work we consider self-driving application, where we need to improve object detections which could assist self-driving cars to drive as safely as humans. In this application we extend the YOLO model with memory map component to the end of neural network. When we validate the effect of gridding size, the size of last layer will be changed to multiple sizes. In orientation estimation experiment, we will revise the cost function, so that the network could learn to distinguish objects' orientations.

Fig 1: YOLO model Architecture in object detection



II. DATA SET

The dataset that is been used in this project is gathered and maintained by KITTI which is an vision Benchmark Suite is specialized for autonomous driving. In this thesis, we used the object detection and orientation estimation benchmark The dataset provided by KITTI is in the form of labeled images Single image size is 800kb to 900kb.

The number of Images in training and testing dataset of KITTI is 7481 images were divided into 70% (5237 images) for training and 30% (2244 images) for testing.

A. *Features of Dataset*

A total of five attributes are present in the datasets which are mostly recordable and practically applicable in real life.
Following are the basic datasets that were provided by the KITTI vision Benchmark Suite:

1) Training Dataset

The training dataset contains the following attributes:

- Class Type: Describes the type of the object like ‘Car’,’Van’,’Truck’,’Pedestrian’,’Person Sitting’,’Cyclist’,’Tram’,’Misc’ or ‘DontCare’
- Truncation: Float from 0 (non-truncated) to 1 (truncated), where truncated refers to the object leaving image boundaries.
- Occlusion: Integers (0,1,2,3) indicates occlusion state
  - 1= partly occluded.
  - 2 = largely occluded.
  - 3 = unknown
- Orientation: Observation angle of object , ranging [-π,   -π]
- bounding box(bbox): 2D bounding box of object in the image: contains left, top, right, bottom coordinates in pixel.

2) Testing Dataset

The testing dataset contains the following attributes:

- Class Type: Describes the type of the object like ‘Car’,’Van’,’Truck’,’Pedestrian’,’Person Sitting’,’Cyclist’,’Tram’,’Misc’ or ‘DontCare’
- Truncation: Float from 0 (non-truncated) to 1 (truncated), where truncated refers to the object leaving image boundaries.
- Occlusion: Integers (0,1,2,3) indicates occlusion state
  - 1= partly occluded.
  - 2 = largely occluded.
  - 3 = unknown
- Orientation: Observation angle of object , ranging [-π,   -π]
- bounding box(bbox): 2D bounding box of object in the image: contains left, top, right, bottom coordinates in pixel.

ALGORITHMS:

We tried to exploit the power of various machine learning algorithms and deep learning approaches on the available dataset. Therefore, we implemented the basic models and techniques the techniques that were implemented in our project are as follows:

1. Yolo
2. SVM
3. CNN

III. EXPERIMENTS

A. *Yolo:*

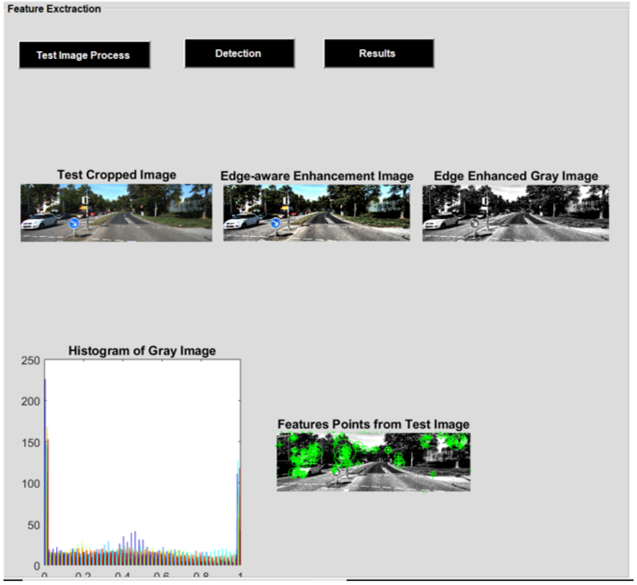
YOLO (You Only Look Once), is a network for object detection. The object detection task consists in determining the location on the image where certain objects are present, as well as classifying those objects. Previous methods for this, like R-CNN and its variations, used a pipeline to perform this task in multiple steps. This can be slow to run and also hard to optimize, because each individual component must be trained separately. YOLO, does it all with a single neural network.

Step 1:- Execute the final.m file from the code folder which provides a dash board to select an image.

Step 2:- Select the Test image process button select the dataset folder from the provided directory (for easy execution we have a Dataset folder which has a sample of 100s of images from KITTI).

<u>CASE1:- Occlusion = 0</u>

After the selection the dash board displays the output

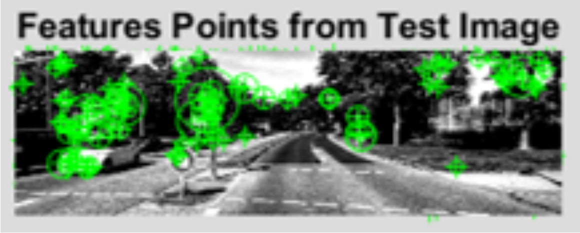


1) Test Cropped Image (the input image)

2) Edge-aware Enhancement Image

3) Edge Enhances Gray Image

4) Feature point test image(Output image)



Following are the parameters which were used which might affect the results:
Occlusion=0, Orientation=1, bbox=4
Truncation = 1.

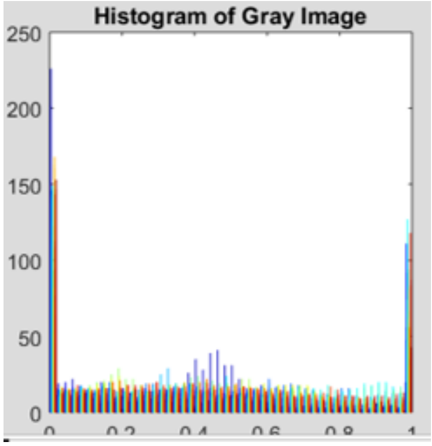
Evaluation Metrics
Yolo method 30 frames per second

Achieved a precision of: - 85.5%

recall = 62.8%

detection speed of 0.03s per images

For a more appealing visual representation we have a histogram of Gray scale image which is also generated in output of our dashboard.



The output dashboard automatically takes all the images in the dataset folder setup and display the all the above mentioned outputs.
To compare the performance under different situations we are using case1 and case2 with varying occlusion values.

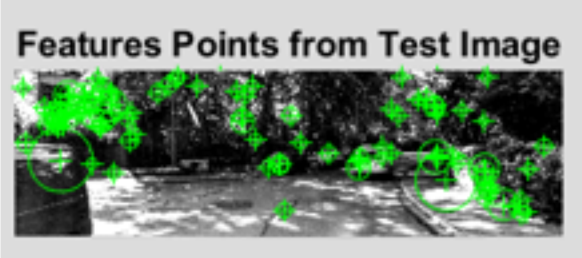
<u>CASE 2: Occlusion = 1</u>

1) Test Cropped Image (the input image)

2) Edge-aware Enhancement Image

3) Edge Enhances Gray Image

4) Feature point test image(Output image)



To compare the performance under different situations we are using case1 and case2 with varying occlusion values and observed there was a slight drop in precision and recall percentage.

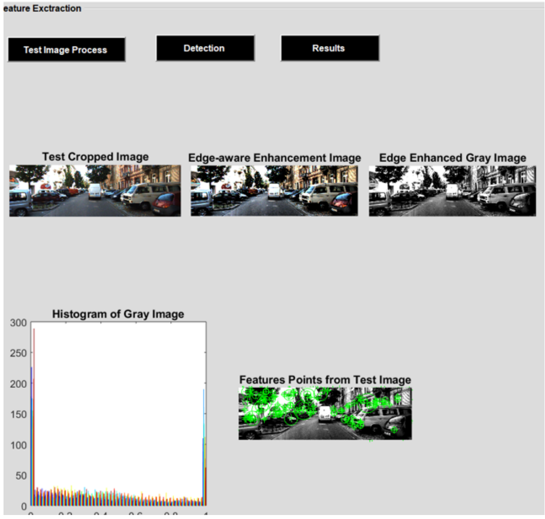
B. Support Vector Machine (SVM):

Support vector machines is a supervised machine learning model that analyze data used for classification**(SVM Classifier for Object Detection.m in our code folder)** and regression analysis. SVM offers very high accuracy when compared with other classifier such as logistic regression and decision trees
SVM is used in a variety of different applications such as object detection, face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition.

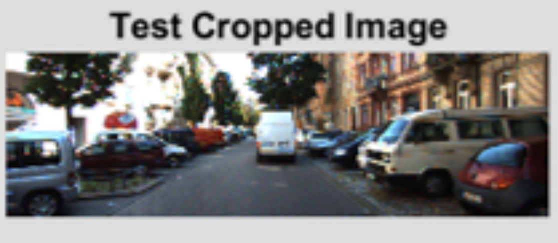
Step 1:- Execute the final.m file from the SVM Model the code folder which provides a dash board to select an image.

Step 2:- Select the Test image process button select the image from datase.

After the selection the SVM algorithm runs and the dash board displays the appropriate output

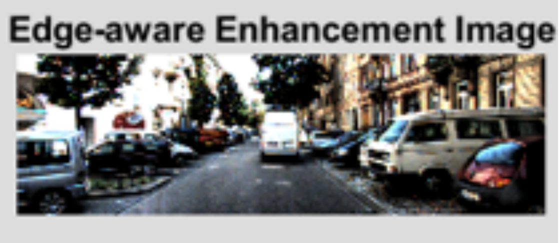


1) Test Cropped Image (input image)

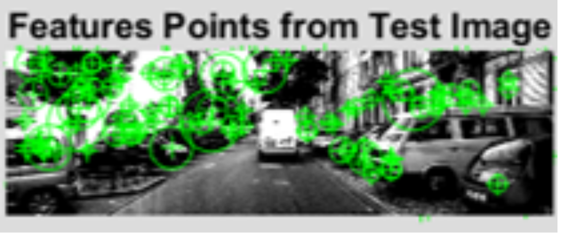


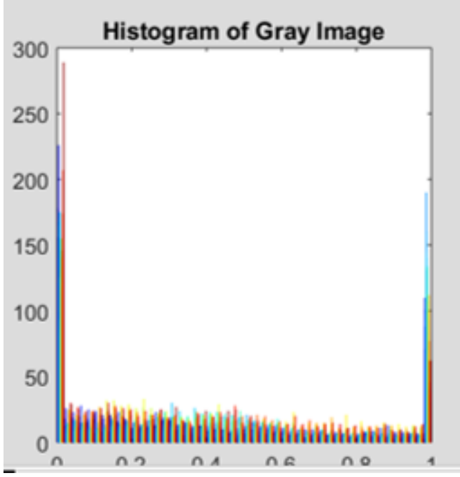
2) Edge-aware Enhancement Image

3) Edge Enhanced Gray Image



4) Feature point test image output





Following are the parameters which were used which might affect the results:
Occlusion=0, Orientation=1, bbox=4
Truncation = 1.
Evaluation Metrics
detection speed is not as good as compared with Yolo Method (0.03s per images)

By using SVM Achieved a precision of: - 72.05%

recall = 50.7%

C. Convolutional neural network (CNN):

The reason why we choose one of the machine learning model as CNN is that they are a category that have proven to be very effective in areas such as image and object recognition and classification. Convolutional neural networks are used primarily to classify images cluster them by similarity, and perform object recognition within scenes.
They are algorithms that can identify faces, individuals, street signs, tumors, platypuses and many other aspects of visual data.

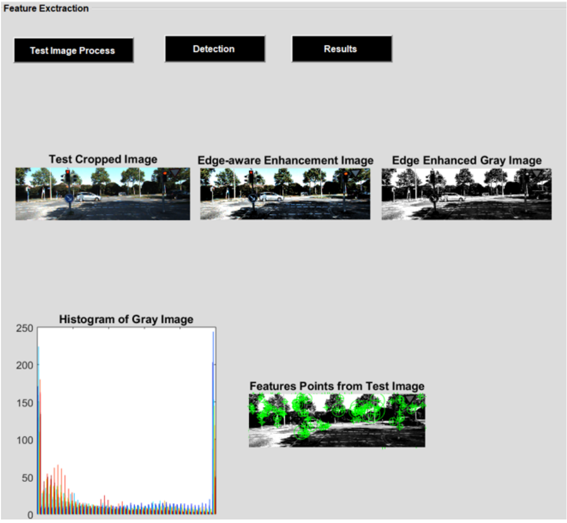
In this network we organize this process by following sets

- TrainingSet
  A collection of input-output patterns that are used to train the network
- TestingSet
  A collection of input-output patterns that are used to assess network performance
- LearningRate-η
  A scalar parameter, analogous to step size in numerical integration, used to set the rate of adjustments

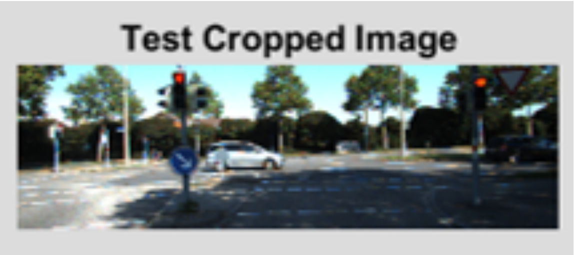
Step 1:- Execute the final.m file from the CNN Model the code folder which provides a dash board to select an image.

Step 2:- Select the Test image process button select the image from database.

After the selection the CNN algorithm runs and the dash board displays the appropriate output



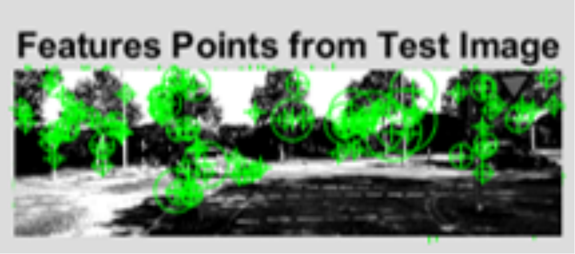
1) Test Cropped Image (input image)

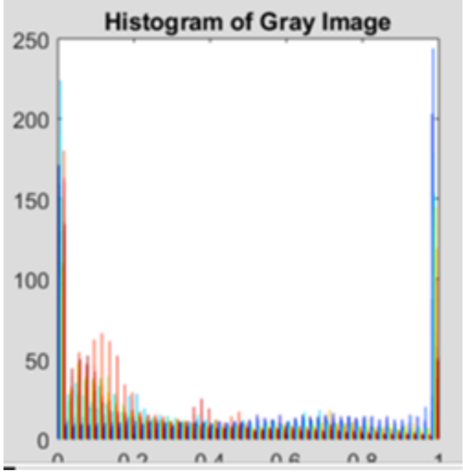


2) Edge-aware Enhancement Image

3) Edge Enhanced Gray Image
Pre-smoothing the greyscale images- further improves the edge for our purposes.

4) Feature point test image output





Following are the parameters which were used which might affect the results:
Occlusion=0, Orientation=1, bbox=4
Truncation = 1.

Evaluation Metrics

detection speed is not as good as compared to Yolo Method (0.03s per images)

By using CNN Achieved a precision of: - 76.2%

recall = 60.7%

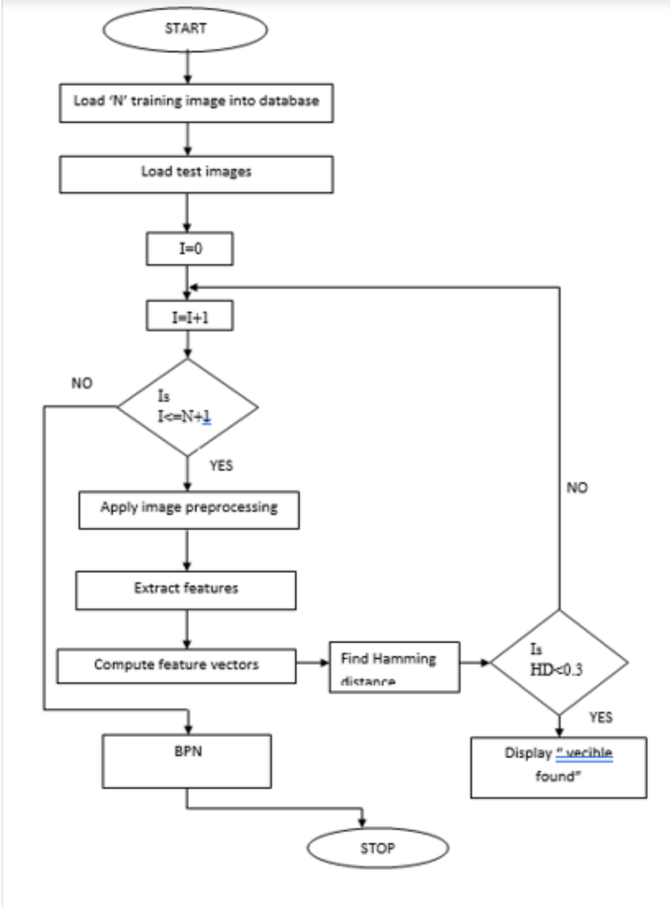
IV. TASK DESCRIPTION

The object detection system You Only Look Once (YOLO) on the other hand, proved it is possible to unify the region proposal and class score prediction in one single convolutional neural network. YOLO has been improved upon it first original introduction in a second version, YOLOv2. YOLOv2 is similar to the original version, but some adjustments in the model makes it both more accurate and faster.

The KITTI Vision Benchmark Suite is specialized for autonomous driving. . KITTI also extract benchmarks for 2D/3D object detection, object tracking and pose estimation. In this thesis, we used the object detection and orientation estimation benchmark. This benchmark includes 7481 labelled images from 80 classes. Single image size is 800kb to 900kb and total size of 7481 images is 6.2GB. Among these classes, only the frequent objects (car, truck, pedestrian, cyclist, tram, and sitting people) are labelled independently, all the other classes are labelled as 'Misc' or 'don't Care' class. The 7481 images were divided into 70% (5237 images) for training and 30% (2244 images) for testing.

Here first, we analyse whether the YOLO and Yolo version is an appropriate model for general object detection, for that we trained the model on dataset and KITTI . We perform a comprehensive analysis of YOLO over Road - object detection and orientation estimation benchmark. According to the comprehensive analysis results the YOLO have a limitations in on real time autonomous driving system. To implement this process for real time autonomous driving system we design a novel YOLO method by combining the memory map and on-max suppression (NMS) model to make YOLO suitable for real-time object detection in video streaming and improve the YOLO potentiality by applying YOLO on orientation estimation, the YOLO can estimate the orientations of the detected objects.

Below is a flow chart description of the project flow which pictorially represents the Tasks performed and its description



IV. MAJOR CHALLENGES AND SOLUTION

One of the major open challenges in self-driving cars is the ability to detect objects and pedestrians to safely navigate in the world.

For a safety critical application such as autonomous driving, the error rates of the current state of-the-art are still too high to enable safe operation.

The main objective is to provide a information of poor quality and environmental model on traffic activity and to signal potentially anomalous situations, e.g., accident detection or dangerous driving In this We perform a comprehensive analysis of YOLO over Road - object detection and orientation estimation benchmark. According to the comprehensive analysis results the YOLO have a limitations in on real-time autonomous driving system. In this we use an important dataset called KITTI. which is specialized for autonomous driving. By driving the car equipped with multiple sensors around in mid-size city, rural areas and on highways, they collected rich data, including images and optical flow from camera, points cloud from laser scanner and odometry information from a GPS.

IV. MAJOR RESULTS

By performing a comprehensive analysis of YOLO over KITTI dataset, we found that YOLO can achieve 85% precision with 62% recall at 30 frames per second. The results are encouraging and suggest that YOLO is an excellent model for detecting objects required for autonomous driving systems. However, YOLO processes the images individually despite the fact that there is continuous information in video stream in real-time driving situation. YOLO needs further modifications to better fit real-time driving system. To fill this gap, the most important contribution of this research lies in proposing a novel technique called memory map to make YOLO suitable for real-time video streaming. The memory map mechanism, which accumulates class-confidence throughout temporal frames, 73 helps increase recall while reduce precision. By comparing with state-of-the-art methods, our revised model with memory map got a little lower detection precision. However, our methods won the first place in detection speed. Our detection speed is 0.03 seconds per image, which is 10 times faster than the best methods. Our modified model is the only one that achieve rea ltime, 30 frames per second, which can be used in driving situation. In the orientation estimation section, we found that YOLO works well in predicting an object's orientation. In an auto-driving system, predicting an object's direction is crucial for the system to be able to make correct decisions.

V. CONCLUSION

In an auto-driving system, predicting an object's direction is crucial for the system to be able to make correct decisions. Through our experiments, we proved that YOLO has great By performing a comprehensive analysis of YOLO over KITTI dataset, we found that YOLO can achieve 85% precision with 62% recall at 30 frames per second. The results are encouraging and suggest that YOLO is an excellent model for detecting objects required for autonomous driving systems performance in image level object detection and orientation estimation. **Compared to SVM and CNN Our novel memory map improvement makes YOLO much suitable in driving situation, that makes it one of the best choice for autonomous driving system**.

VI. FUTURE WORK:-

- In YOLO network, the input images are divided into unique-sized grids to predict objects with various sizes. Thus, the unique-sized grid may affect the bounding box accuracy of smaller or larger objects.

- In the future, we will try to divide the input images into multiple sizes and select the best results from overlapping grids. From our experiments, we observed an increase in recall as an impact of using memory map.

- In the future, we can add memory map to training process to learn the optimum weights. KITTI dataset has multiple ground truth data, 2D and 3D.

- In our experiments, Currently, our system can only run on a single GPU. In the future, we can revise the system to make it run on multiple GPUs to get higher detection speed. That will enable the network to work better in high speed driving situations.

How this research encourages long-term study in computer vision or other fields?

This project will help in future for majorly in Automation driving Automating the driving process majorly helpful for **disabled and blind people**.
Manual errors can be removed and can save a huge amount of loss from accidents.It also make sure that pedestrians can also be saved from accidents.As Manual driving may lead to accidents and Automated driving is booming topic in present technology.While we are working on Yolo , we got a better understanding of Computer vision and its applications.

YOLO is always coming up with updated versions with better solutions and more accurate detections which caught our attention towards it and we are looking further to learn about it and in the meanwhile explore more of computer vision.

CONTRIBUTION OF TEAM MEMBERS

- **Nagachandra Kambathalli Parswanatha -** Implemented **You Only Look Once (YOLO)** and C**onvolutional neural network(CNN)** which involved coding of the .m files in yolo and CNN directory He conducted experiments on YOLO and CNN and noted his insights in the final project report

- **Sharath Chandra Kumar Periketi -** Implemented **Support Vector Machine(SVM)** which involved coding of .m files in SVM directory and he conducted experiments on CNN and noted her insights in the final project report.

VII. REFERENCES

http://guanghan.info/blog/en/my-works/train-yolo/

A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012,
pp. 3354–3361

https://www.computer.org/csdl/proceedings/bigmm/2017/6549/00/07966765.pdf

http://www.cvlibs.net/datasets/kitti/eval_road.php

E. Ohn-Bar and M. M. Trivedi, “Learning to detect vehicles by clustering appearance patterns,” IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 5, pp. 2511–2521, 2015.

https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/
http://iopscience.iop.org/article/10.1088/1742-6596/1004/1/012029/pdf

https://www.researchgate.net/publication/324754316_Understanding_of_Object_Detection_Based_on_CNN_Family_and_YOLO

https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf

https://grail.cs.washington.edu/wp-content/uploads/2016/09/redmon2016yol.pdf