DRIVING COACH

CSC 528

BY HARSHA MARELLA

ABSTRACT

This paper looks into the prospects of new growing field in computer vision for self-driving cars also known as Automated Driving Systems (ADS).in this paper we will concentrate on typically one of the sensors used in ADS which is the camera. A car camera can be used to process lot of information in our daily life and in cars we use them for parking through rare camera but in this project, we will focus on lane detection and object detection mainly obstruction such as other cars and environments objects like people, traffic lights etc.

INTRODUCTION

Recent years, self-driving cars has attached widespread attention of researchers. ADS can effectively assist the driver and reduce the incidence of traffic accidents. Lane detection and tracking are a key technology for autonomous vehicle and ADS applications such as Lane Departure Warning and Lane Change Assist. The task of lane extraction is to separate the traffic lanes from the background within an image

The general goal of my research is to develop an intelligent, camera-assisted car that is able to interpret its surroundings automatically, robustly, and in real time. Even in the specific case of a highway's well-structured environment, this is a difficult problem. Traffic volume, driver behavior, lighting, and road conditions are difficult to predict. So, I implemented a method to test out the possibilities of detecting lanes and vehicles. It segments the road surface from the image using color classification, and then recognizes and tracks lane markings, I am using a pre trained COCO-YOLO model for the object detection purposes.

METHODOLOGY

For this paper the methodology is split in two phases the lane detection and the other is object/vehicle detection

LANE DETECTION PIPELINE

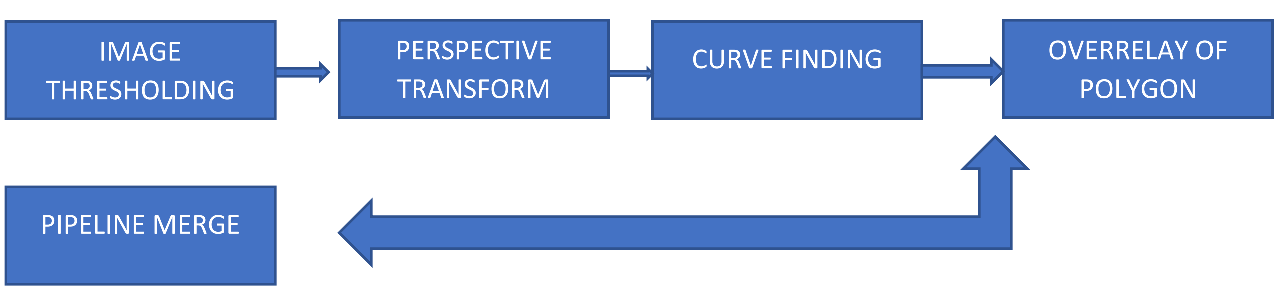
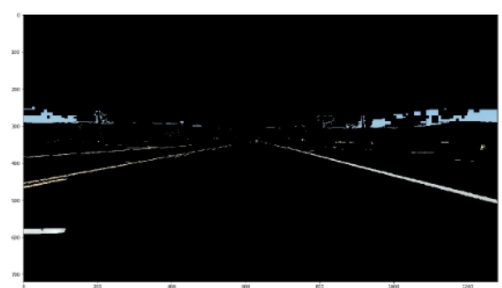


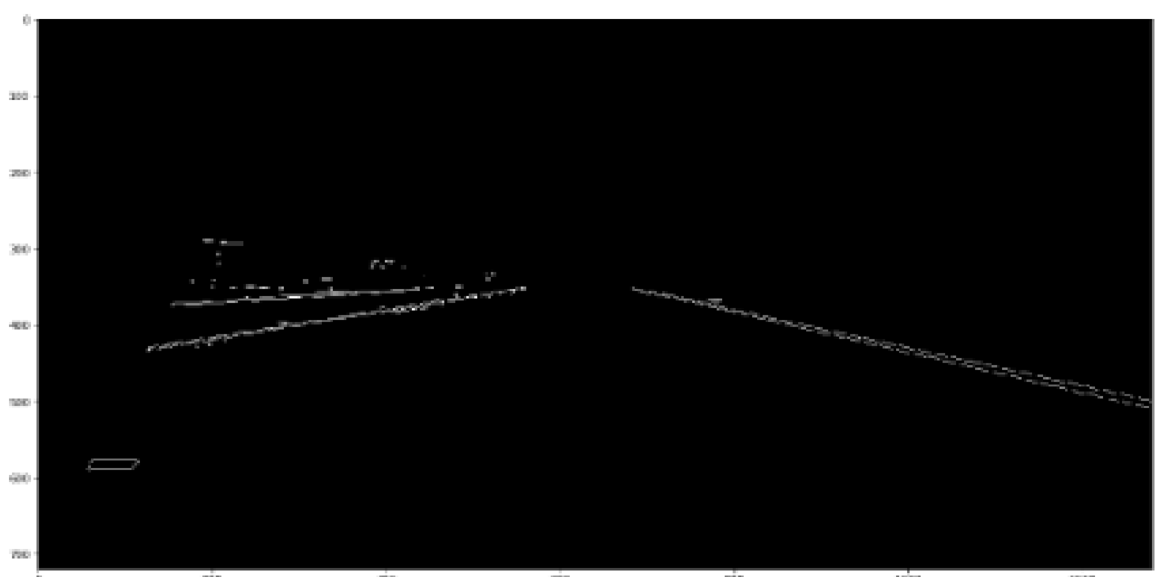
Image Thresholding

In this phase A sample video is broken down to frames and then one random frame is used for detecting lanes this is done in multiple steps. First step is to threshold with HSL (HUE, SATURATION, LIGHTNESS) filter this filter is useful because we know that roads especially lanes are very distinct in color that the tarmacs color which is black so we have a very high contrast ratio between the road and road lanes
Next Sobel operator for canny edges is applied to the image to give further distinction of the lane markings final checking the magnitude gradient we are able to separate the ROI (REGION OF INTREST) form the image that is the lane marking.

Original Image Figure



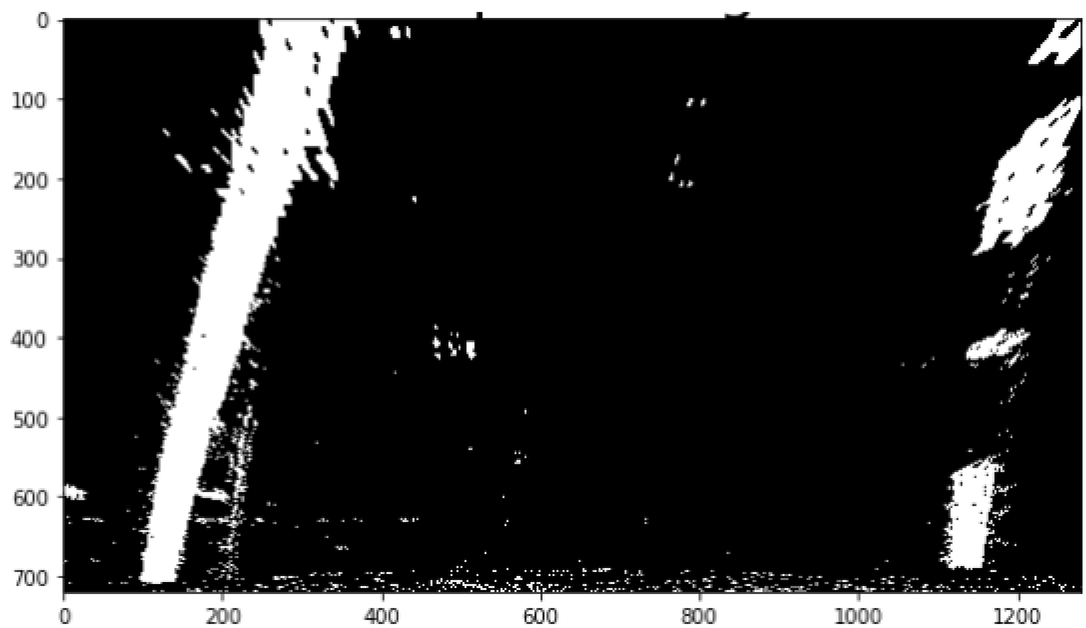
HSL Thresholding



Region of Interest (ROI)

Perspective Transform

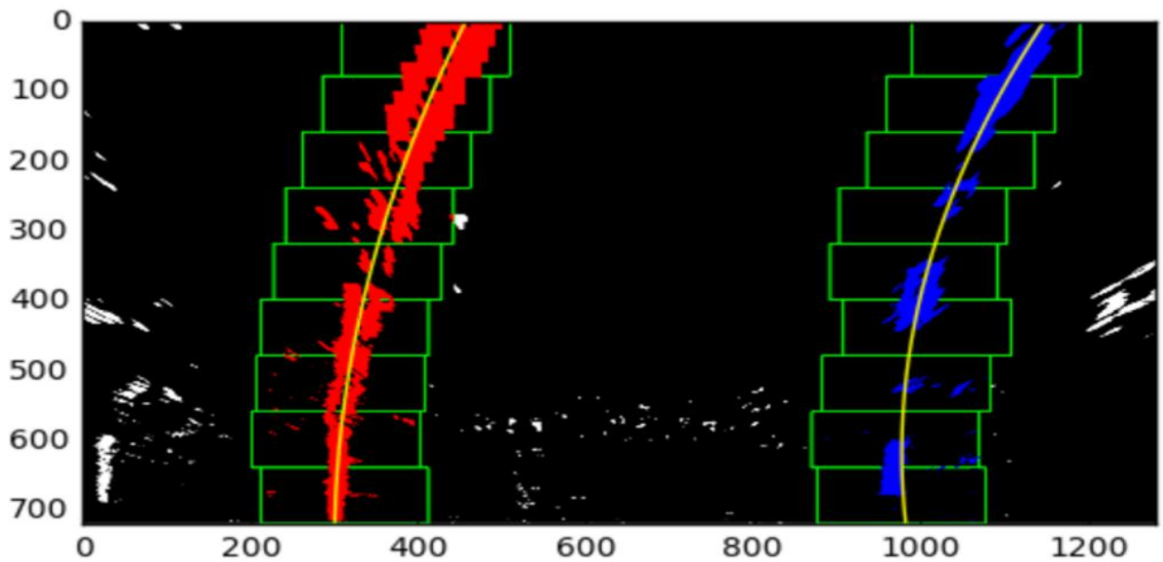
In this phase The ROI image is used and it is warped in a way which create a Birdseye view perspective of the ROI image why this transformation is necessary is because it helps us to track the lanes



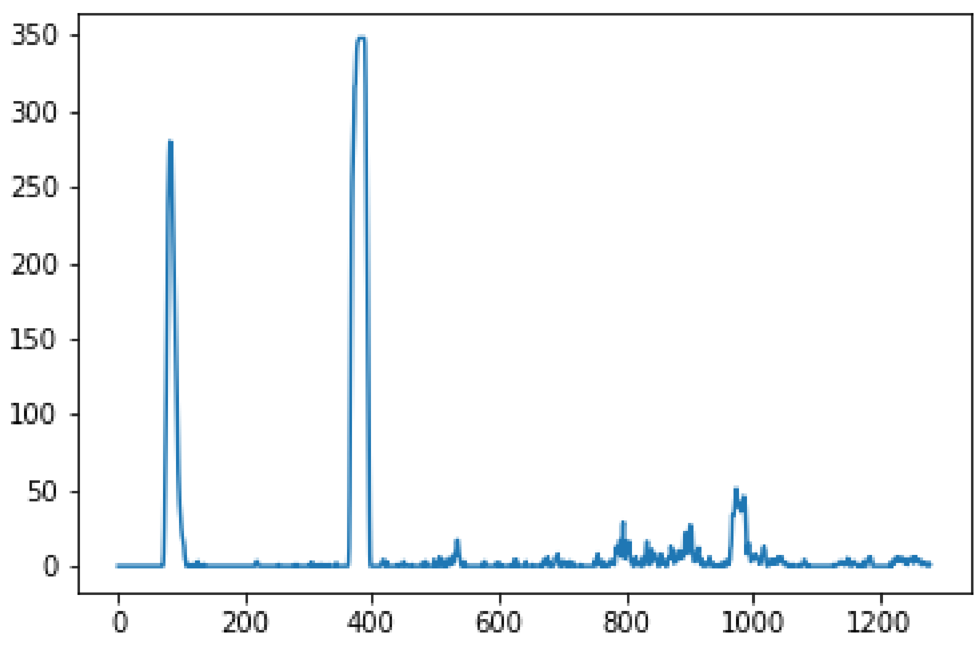
Warped image Bird's Eye Image

Curve Finding

In this phase a sliding window technique is used to keep track of the lanes lines as the car moves forward this is helpful especially when the car is at the bank of the turn as its is hard for the before methods to detect turn how this layer works is that it creates a sliding window on the lane line and updates the position of the lane every sec for the computer to understand that the camera has captures the image of the lane in case if there is no lane line detected the algorithm automatically searches for the lane line again by doing this we ensure that the lane tracking continues.



Sliding window (in green color), placed around the on line center(yellow Line),finds the line and follows till the top the red and blue color represent the lane pixels



Lane Histogram Peaks suggest the Lane detected

Overlay of Polygon

This is for demonstration purpose a polygon tracker is fitted using tracking points from the ROI data for lane detection purposes.

Pipeline Merge

Finally, all the methods are combined in a single function and applied to our dataset(video)

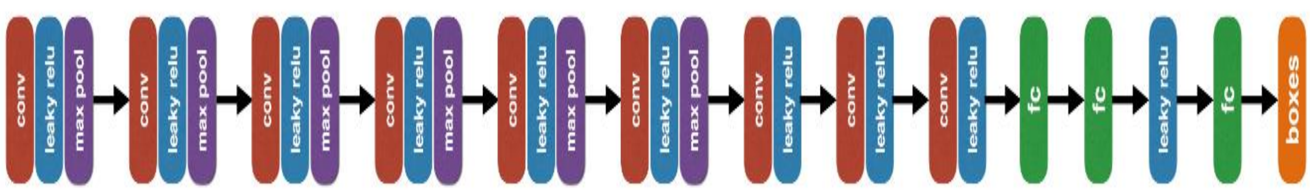
OBJECT DETECTION PIPELINE

Detecting vehicle in a video stream is a object detection problem. An object detection problem can be approached either as a classification problem or a regression problem. If the problem is of classification then the image is divided into small patches each of which will be run through a classifier to determine whether the patch belongs to a object is it does then a bounding box is applied which determines the probability of the object present.in regression approach the whole image is run through a convolutional neural network to directly generate one ore more bounding boxes depending on what dataset the CNN is trained on.

classification	**regression**
Classification on portions of the image to determine objects, generate bounding boxes for regions that have positive classification results.	Regression on the whole image to generate bounding boxes
1. sliding window + HOG 2. sliding window + CNN 3. region proposals + CNN	generate bounding box coordinates directly from CNN
RCNN, Fast-RCNN, Faster-RCNN	SSD, YOLO

In this project for the ease of detection as classifier methods are slower than the regression models used for object detection because of an extra step of classification involved I have used YOLO(you only look once) neural network which was pre trained on COCO dataset provided by Microsoft which has more than 80 classes

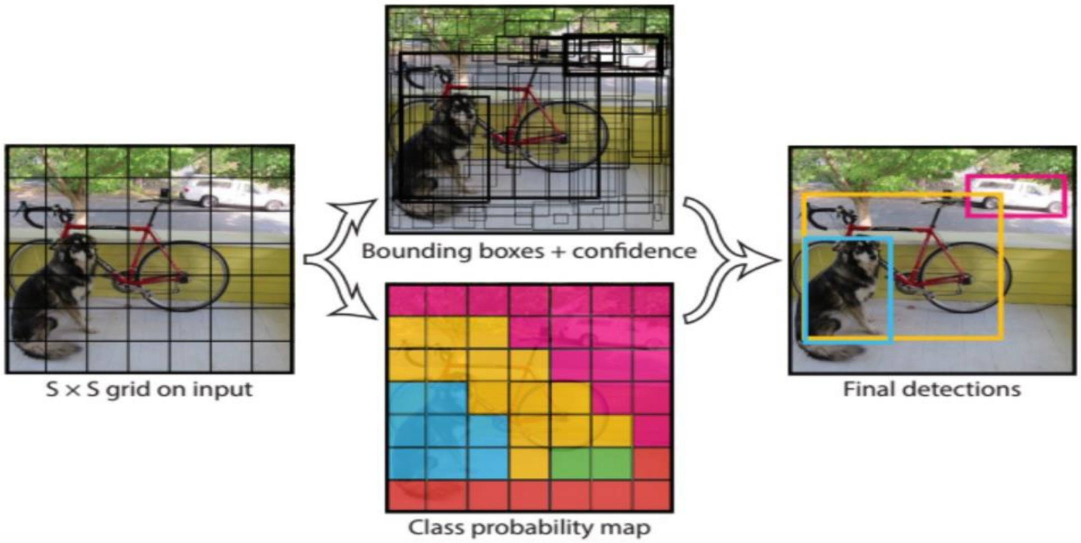
Architecture of Yolo



YOLO ARCHITECTURE

Yolo consist of 9 convolutional layers. And 3 full connected layer each layer consists of Convolution, leaky relu and max pooling operation. The first 9 convolution layers can be understood as feature extractors and the last three fully connected layers are used as regression head which predict bounding boxes.

How Yolo Works on images



YOLO WORK FIgure Explains the steps of YOLO CNN work. (Left) original image. (Top Center) Bounding box and Confidence creation. (Bottom Center) Probability and confidence detector (Right)End result.

As we can see from the figure the original image is passed to the CNN then The CNN creates the bounding boxes based on the confidence level of the pretrained data once this is finished a normalize suppression is used to create probability of object placement and finally based on the probability a final detection is presented.

DATASET COLLECTION

Udacity Nanodegree(self-driving) Has all the dataset for this project to be done I have used some of the resources from their dataset to solve some of the problems in lane detection(clear and accurate images easier to find ROI in this images but are not enough for all condition namely like light, weather, road condition).

COCO-Trained YOLO I have used pretrained yolo for object detection the COCO dataset. COCO data set is a large data set made in collaboration with Microsoft and partners it has 330000 images,80 clases,5 caption per category.

Some customs testing video taken from google pixel 3 with bad placement to judge the performance.

RESULTS

Good lane detection (Udacity example)

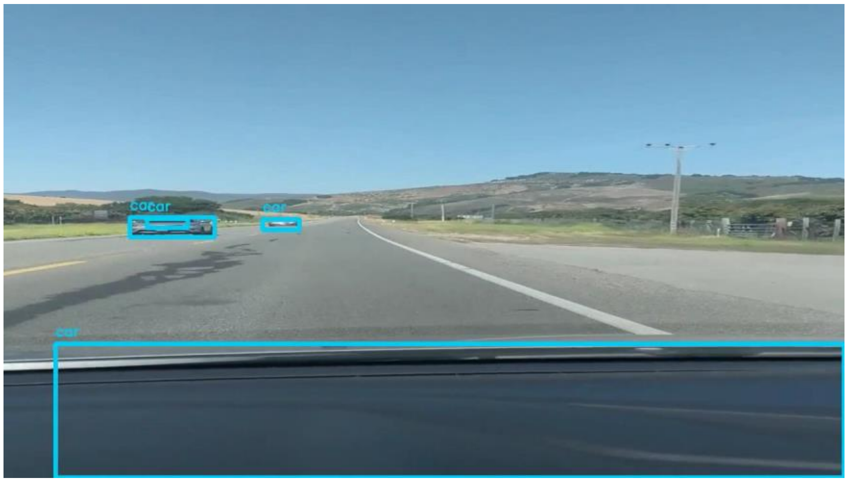
Lane detection

Bad lane detection

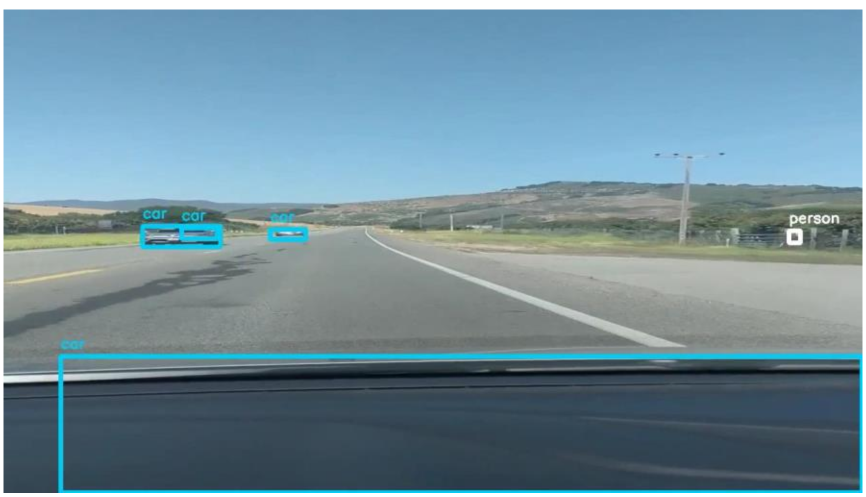
Bad lane detection due to car dash as shadow and bad camera detection

Object detection (Vehicle Detection)

Video 1 from Nashville

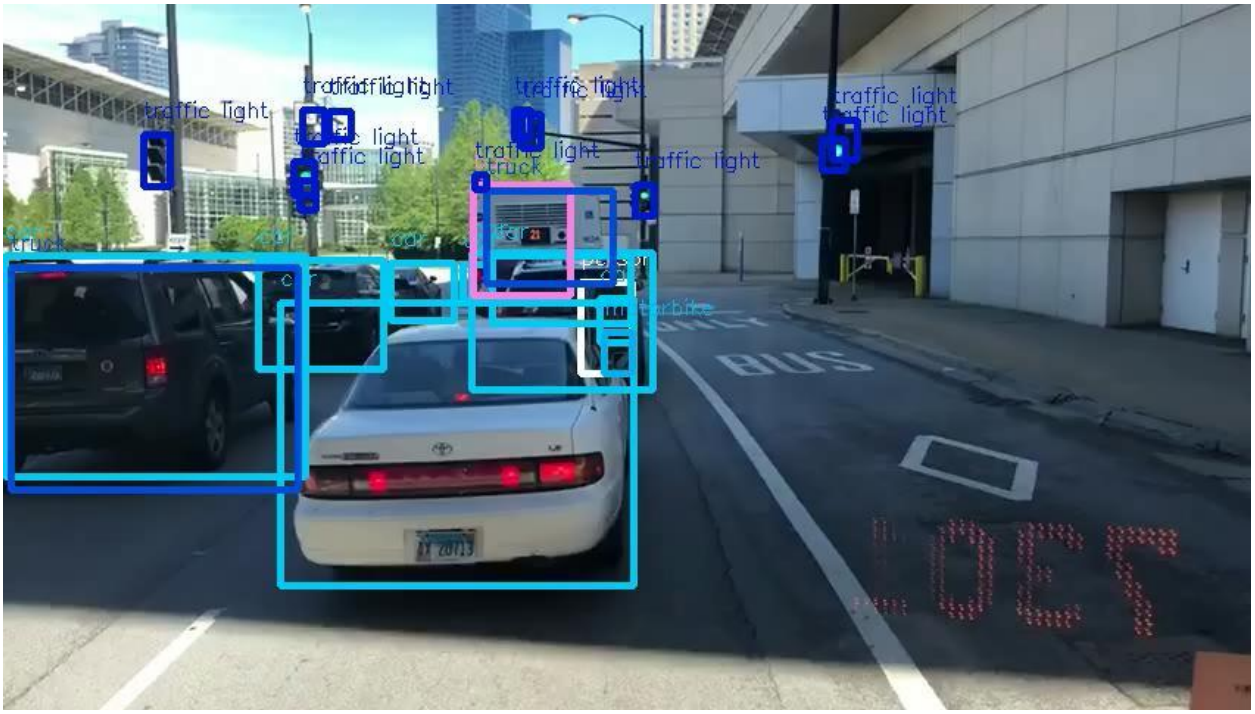


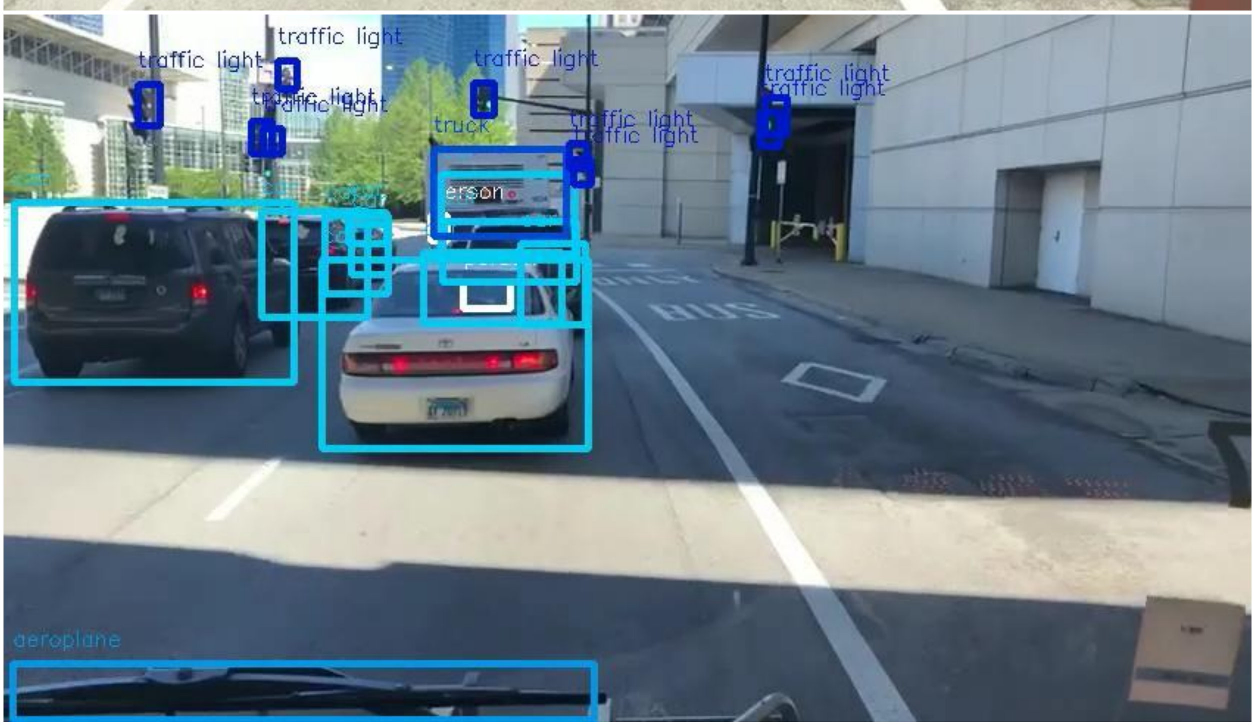
Vehicle Detection 1

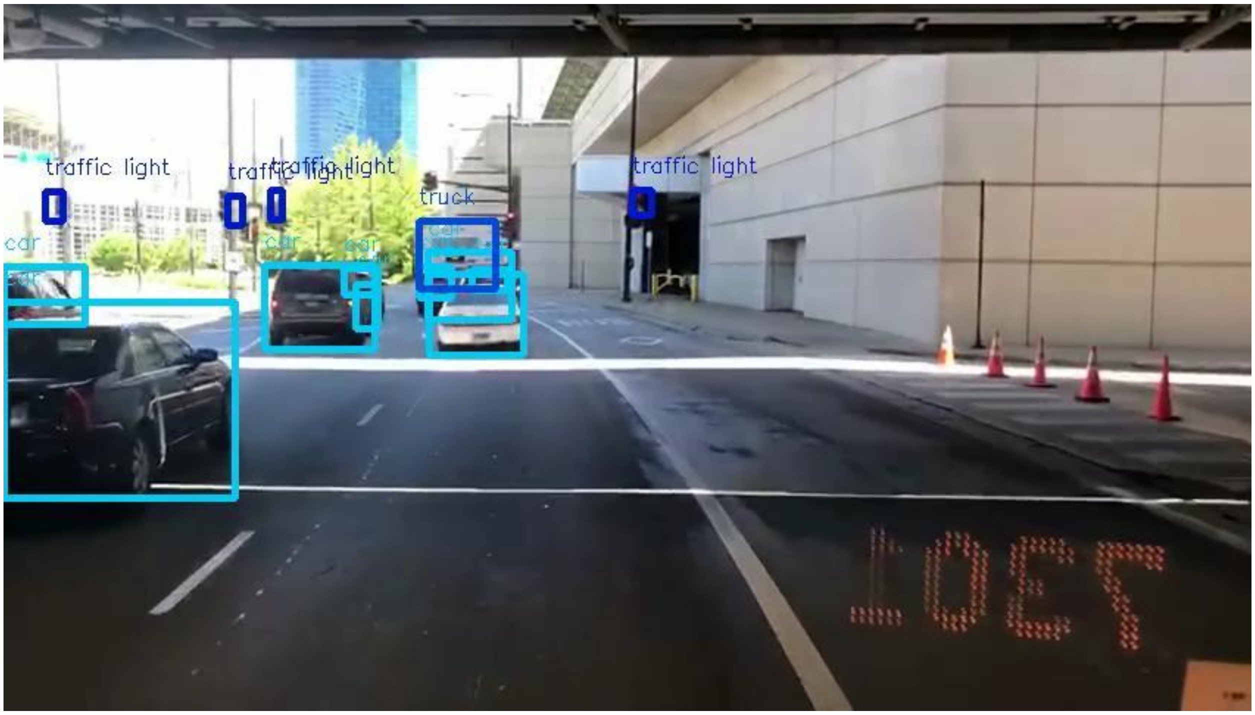


Vehicle Detection 2

More Detection with Lot of Clutter





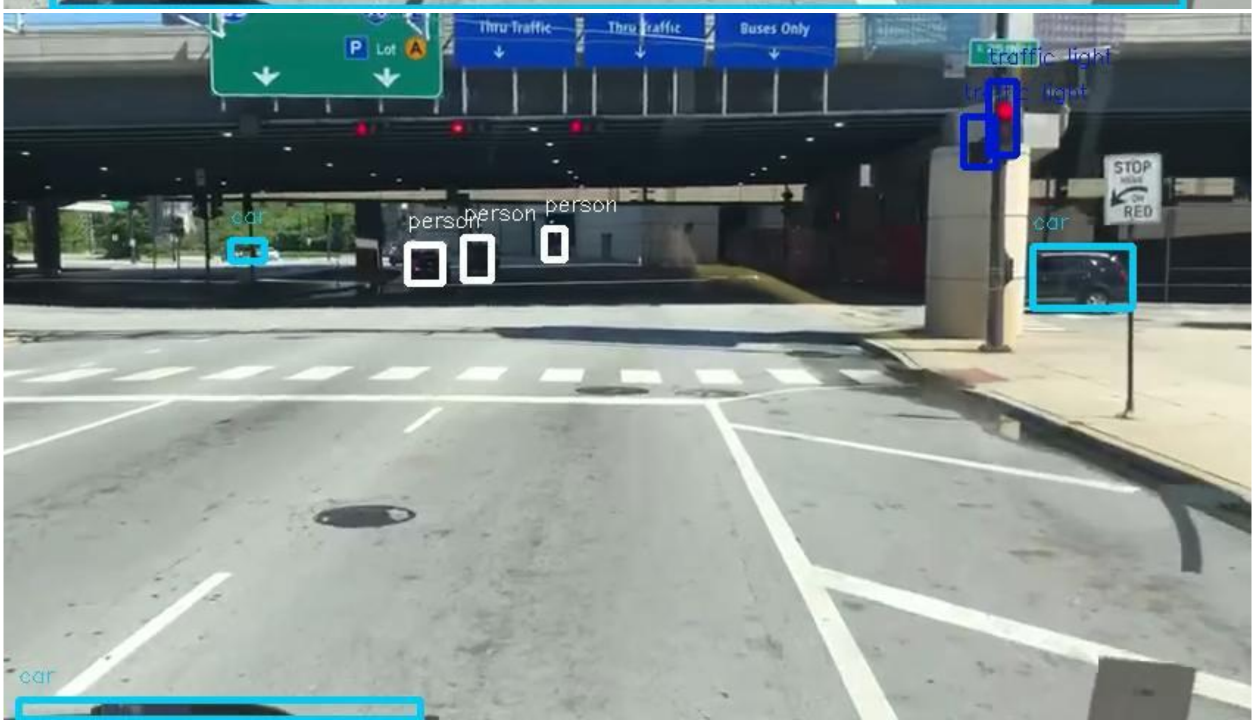

traffic light
traffic light
traffic light
traffic light
truck
car
car
car

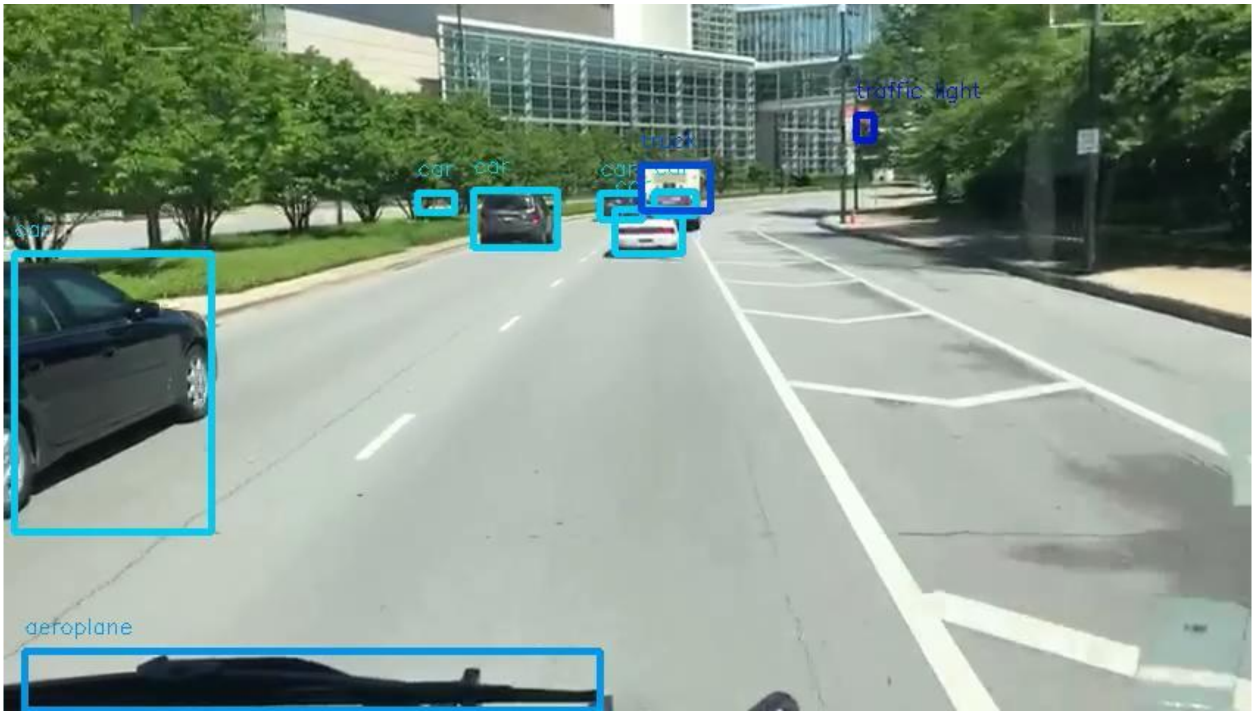
traffic light
traffic light
traffic light
car
car
car
person
fire hydrant
diningtable

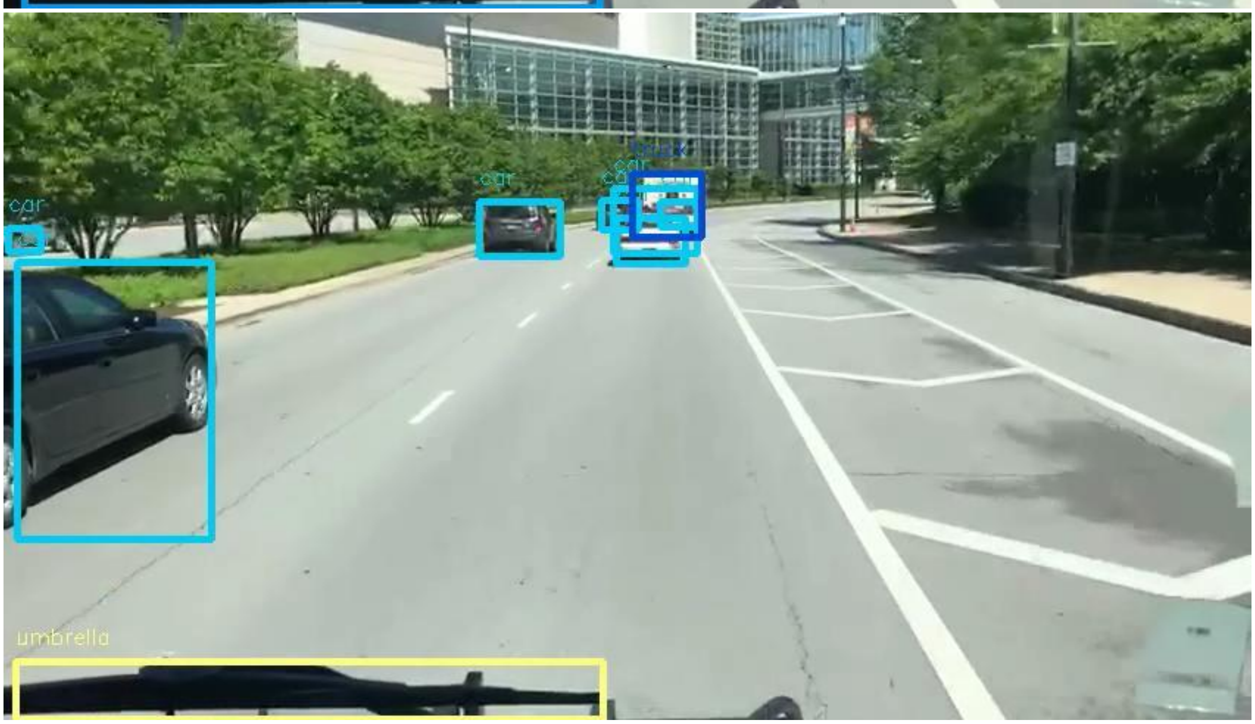

traffic light
traffic light
traffic light
person
car
car
car
car
bench
Lot


traffic light
traffic light
traffic light
car
car
car
car
car
Lot


traffic light
person
car
bench


Thru Traffic
Buses Only
P Lot A
traffic light
STOP HERE ON RED
person
car





Evaluation

One Method to evaluate any CNN is to evaluate confidences, Accuracy<precision and recall and since this is done on large dataset of images and our dataset is primarily video (Realtime detection of objects we calculate frames per second and average time to render this gives us the metric of what hardware is suitable for actual Detection and works in my case I was processing all on CPU.

HARDWARE	FRAMES PER SECOND	VIDEO LENGTH	TIME TAKEN
CPU (i5 5^{th} gen)	0.595	3.16 minute’s (Chicago)	3 hrs.

GPU	8.01	3.16 minute's (Chicago	15 mins.

As we see GPU are better suited for Detection and tracking purposes.

CONCLUSION

There are many concluding points in this project first Lane detection For this to be successfully compiled we need is a very high definition camera which has to placed at the center of the car and should be overlooking the road, the lane marking have to be clear for the algorithm to understand that there are lane line present in the capture frame of the camera and the weather should be good with less occlusion happening. The biggest problem of all I face was placement of the polygon as it has to match the warped image coordinate points.

For object detection I would like to use a classifier class CNN where I will have more control on the object detection and what my neural network detects.

FUTURE WORK

To apply Lane, net another CNN which is trained for lane findings and accurately predict them segmenting highways in number of lanes object collision system and perfecting and combining all the pipelines into one single video.

REFERENCES

[1] Betke, M., Haritaoglu, E. and Davis, L. S. (2000) "Real-Time Multiple Vehicle Detection and Tracking from a Moving Vehicle," *Machine Vision and Applications: An International Journal.*

[2] Wu, B.-F., Lin, C.-T. and Chen, C.-J. (2010) "Real-Time Lane and Vehicle Detection Based on a Single Camera Model," *International Journal of Computers & Applications*, 32(2), pp. 1–1.

[3] Ponsa, D., Serrat, J. and López Antonio (2011) "On-Board Image-Based Vehicle Detection and Tracking," *Transactions of the Institute of Measurement and Control*, 33(7), pp. 783–805.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779-788.
doi: 10.1109/CVPR.2016.91

[5] Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham

[6] Guo, J., Wang, J., Guo, X., Yu, C. and Sun, X. (2014) "Preceding Vehicle Detection and Tracking Adaptive to Illumination Variation in Night Traffic Scenes Based on Relevance Analysis," *Sensors*, 14(8), pp. 15325–15347. doi: 10.3390/s140815325.