

Real-Time Lane Detection and Object Recognition in Self-Driving Car using YOLO neural network and Computer Vision

Nikita Mesvaniya, Meghna Dhruva, Ishita Khared, Krishna Adhia

Abstract: *The Darwinism of Artificial Intelligence and robotics has grown up incredibly. Recently, there are a lot of progress have been undertaken in the context of Autonomous vehicle. Robo-car or self driving car consist many module like localization and mapping, scene understanding, movement planning, and driver state. In movement planning lane perception and recognition of the object plays vital role. This proposed state-of-art recognizes the road track in the video's frame and perform lane detection using canny edge detector and Hough transform algorithm. In this paper, Object recognition is possible with help of YOLO (you only look once) which is one of the real time CNN methods aims to detect object inside the image as part of road track. The result manifests the road lane detection guidance and object recognition along with prediction probability and bounding box.*

Keywords: Convolution neural network, Hough Transform, Lane Detection, Object Recognition, YOLO.

I. INTRODUCTION

Now-a-days, Development in robotics has brought to a higher level by deploying Artificial intelligence into robot. Now robot can accomplish most of task effectively instead of human in terms of Military, Medical, Education and Delivery, etc. Self-driving car has been an important research field because of its valuable application helps to resolve many problems. One of the major problem is Accident, it is occurred due to 4 D's (Drunk, Drugged, Distracted and Drowsy Driving) [1, 5]. Many industries take this research as challenge like Waymo/Google self-driving car, Uber, Tesla Autopilot, and nuTonomy. For Autonomous robo-car research need a lot of real- world driving data with included field computer vision, Machine learning, Big data and Understanding. There are many modules that are included in Autonomous Vehicle like Movement Planning, Behaviour analysis, localization and mapping etc [8]. In our state-of-the-art, we detect the Lane that is used to detect the road track from video's frame [1, 3]. Lane detection is performed by applying famous Image processing algorithm canny edge detection and Hough Transform [1]. After detecting Lane, we also implements Conv Nets for implementing real-world object detection using YOLO neural network [2, 4].

Revised Manuscript Received on May 20, 2020.

Nikita Mesvaniya, Department of Computer Engineering, Darshan Institute of Engineering and Technology, Rajkot, Gujarat, India.

Meghna Dhruva, Department of Computer Engineering, BVM Engineering College, VVNagar, Gujarat, India. E-mail: niketamesvaniya3011@gmail.com

Ishita Khared, Department of Computer Engineering, Darshan Institute of Engineering and Technology, Rajkot, Gujarat, India.

Krishna Adhia, Department of Computer Engineering, Darshan Institute of Engineering and Technology, Rajkot, Gujarat, India.

Images are feeding into YOLO neural network and result will be generated with detecting object along with bounding box and predicted probabilities [5].

The "Real time lane detection and object recognition in self driving car using YOLO [9] neural network and computer vision" several module implementations.

- Lane detection from the video frame using Computer vision Algorithm.
- Object Detection from the images using YOLO convolution neural network [4-6].

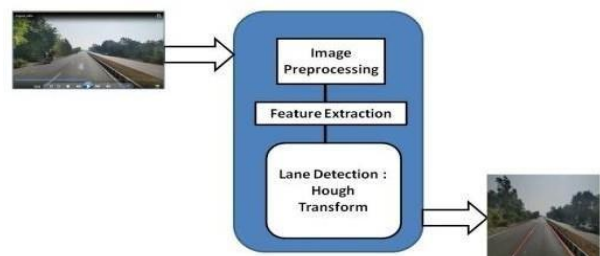


Fig. 1 Lane Detection System flow

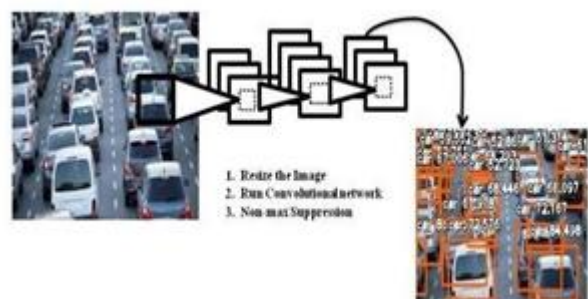


Fig. 2 Object Recognition System flow

II. METHODOLOGY & EXPERIMENTATION

A. Lane Detection

1. Image Preprocessing

▪ Convert Image to Gray-scale image

An image is made up of three channel color. Image would have red, green and blue channels and each pixel in image is combination of three intensity values. In feature extraction, edge detection is measured by detecting sharp changes in intensity in adjacent pixels. The intensity of each pixel is represented by a numerical value from 0 to 255. And, the intensity value of zero does not indicate intensity and 255 represents the maximum intensity. The conclusion is that by converting a RGB image (three color channel) to a Gray-scale image (one channel) leads to the faster image processing [4].

Using open CV library we can easily convert RGB image to Gray-Scale image.

▪ **Image Smoothing**

In gray scale image, each of pixels for a gray scale image is depicted by a single number that describe the brightness value of the pixel. For image smoothing, we apply Gaussian blur in gray scale image. Image is represented as two dimensional arrays. Mathematics behind Image Smoothing.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where x and y is distance from the derivation in the plane of horizon and vertical axis. σ is standard deviation over Gaussian distribution.



Fig. 3 Gray scale Image



Fig. 4 Blur Image

2. Feature Extraction

▪ **Edge Detection**

It is a technique of an image processing to discover the edges of the objects within the images. The main prominence of edge detection is in Image segmentation and feature extraction.

We can perform edge detection by using several methods as.

- Sobel, Robert, prewit method
- Laplacian of Gaussian
- Canny Edge detector

In feature extraction, we apply the canny edge detector to find the edges present inside the image. Some of the criteria of canny edge detector are: Low error rate, good Localization, Minimal response for canny edge detection, the first step is to perform filtering in original image using Gaussian Filter which we have performed in image pre-processing step. For finding intensity gradient of the image, we can use following formula.

$$\text{Edge Gradient (G)} = \sqrt{(s_x^2 + s_y^2)} \quad (2)$$

$$\text{Direction } (\theta) = \tan^{-1} \frac{s_x}{s_y} \quad (3)$$

Gradient is measure by changing brightness over adjacent pixel. The next step after calculating the intensity gradient is to perform non-max suppression that takes out pixels that are not considered to be a part of an image. Result is generated in binary image .Now, the final step is to perform canny edge detection using OpenCV library in python language.



Fig. 5 Canny Edge detection on original Image

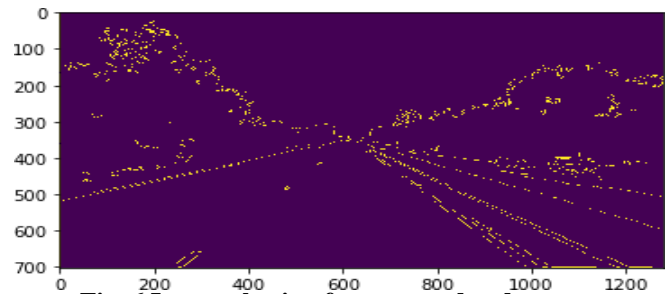


Fig. 6 Image plotting for canny edge detector

3. Lane Detection: Hough Transform

The Hough transform [10] image analysis is a feature extraction approach utilized in computer vision and digital Image processing. The older version of Hough transform was originate to identify the lines in the image, but later redesigned to identify the arbitrary shape. The following Fig. 7 represent region of masking [1] in canny image.



Fig. 7 Region masking in canny edge

Hough Transform Algorithm for fitting a Linear Line

1. Quantize the parameter space $p[c_{min}..c_{max}, m_{min}...m_{max}]$.
2. FOR EACH edge point(x,y) DO FOR (m=mmin , m<=mmax, m++) C=(-x) m+y P[c,m]+1
3. Locate the local maxima in the parameter space.

B. Object Recognition

Movement planning in self-driving car is depends on three things: object localization, objects tracking and mapping of an object. For safe navigation in self-driving car, it is vital to detect information about object around the autonomous vehicle Detecting object in video and images can be feasible due to rise of various deep learning algorithm and machine learning algorithm. To detect, locate and recognize the object some kind of specialized algorithm have been developed, some of are RCNNs, SSD, RetinaNet, YOLO etc[5].

Object classification is labeling the object and finding position of the object along with label is known as object localization.

Exploring Dataset

For Object detection, huge number of standard dataset is available for research community. In this proposed research, we are applying object detection on COCO (Common Object in Context) object detection dataset. COCO is large scale object detection, captioning and analysis dataset which has some of the advantageous features over 80 categories of different object, 330k images, etc [7].

YOLO (“You Only Look Once”) Convolutional neural network

YOLO is a convolution neural network (CNN) algorithm for object detection in the real world [2].

- For a full image detection, YOLO applies single neural network on image and image is divided by YOLO into S* S regions.
- For each region in image, YOLO predicts the bounding box along with predicated probabilities in detected object.
- YOLO “only look once” at the image means it require only one forward propagation that passes through the neural network for making a predication. Non-max suppression is applied, which ensure that algorithm only detect each object once. And then result is recognized with bounding box and contained predication of probabilities with detected object [6].
- The architecture of YOLO is simple. It is just a convolution neural network. The neural network only uses standard layer types: convolution with a 3 * 3 kernels and max pooling with a 2*2 kernel.

Table 1. A Table of neural network in YOLO

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)
Convolution	1×1	1	(13, 13, 125)

Performance (Test Set) Evaluation

For evaluating the object detector, modern object detection is count on a metric called mean Average Precision (mAP).we compute Average precision separately for each object class detected in the image based on their sorted confidence afterwards average over all APs for each separated object. Following figure depict the each detected object with its class name and confidence from figure (11) and figure (12).

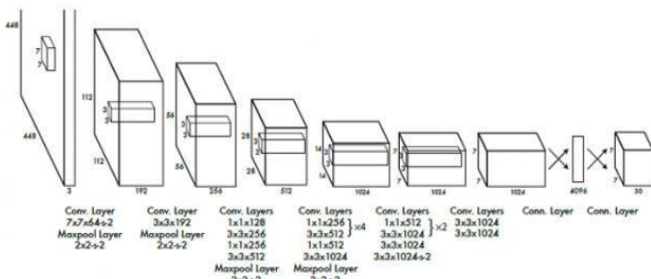


Fig. 9 YOLO Architecture[6]

This detection algorithm has number of benefits over other detection algorithm.

1. Extremely Fast (can process 45 FPS).
2. Only one Image scan.

III. EXPERIMENTAL RESULTS & DISCUSSION



Fig. 10 Real-Time Lane Detection



Plotted Plant : 50.40



Umbrella : 59.60



Truck : 94.16



Car : 78.59



Car : 78.81



Car : 79.84



Car : 90.55



**Person : 54.64
Person : 67.37**



**Person : 84.58
Person : 96.82
Person : 97.48**



Car : 98.24



Motorcycle : 99.75



Motorcycle : 99.88

Fig. 11 Detected object from figure(12,13) with its class label and confidence



Fig. 12 Object Detection with YOLO



Fig. 13 Object detection with YOLO

Precision and Recall Curve

Following figure depict detected object’s average precision along with precision/Recall curve with mAP: 74.81%.

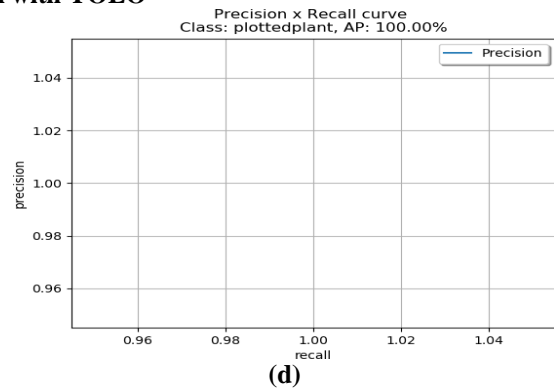
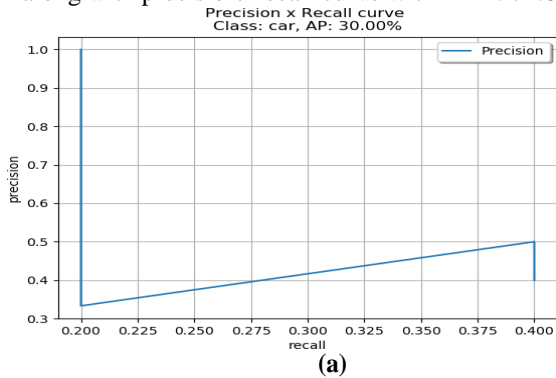
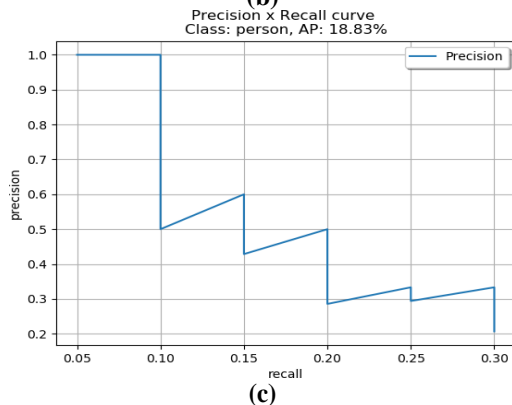
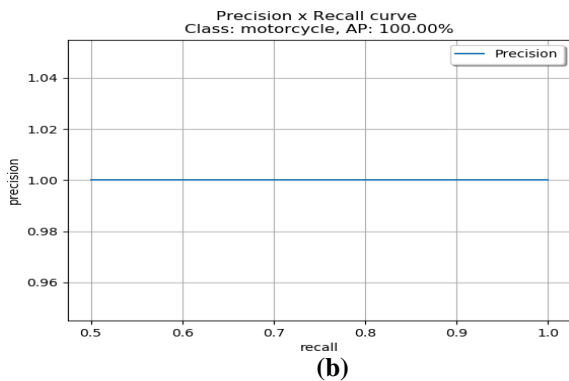


Fig. 14 (a) precision/recall curve for car. (b) precision/recall curve for motorcycle. (c) precision/recall curve for person (d) precision/recall curve for plotted plant.



IV. CONCLUSION & FUTURE WORK

“Real-time lane detection and object recognition using YOLO neural networks and computer vision” proceed towards the successful implementation of lane detection from the video frames and provide proper guidance on video frame by detecting the lane on road track. We have approached well known canny edge detector and Hough transform for implementing our state-of-the-art. Previously, object recognition was implemented using support vector machine, various deep learning methods. We have used YOLO V3 for recognizing the object.

This research is helpful to provide suggestion to the driver and object detection is helpful in self driving car to avoid collision. Self driving car itself has vast number of implementation area. This research is just implementation of several modules of, we can further extended this research into traffic analysis, real time collision avoidance, steering guidance, etc.

REFERENCES

1. Aziz, M., Prihatmanto, A., & Hindersah, H. (2017). Implementation of Lane Detection Algorithm for Self-Driving Car on Toll Road Cipularang using Python Language. 4th International Conference on Electric Vehicular Technology (ICEVT), 144-148. doi:10.1109/ICEVT.2017.8323550
2. Čorović, A., Ilić, V., Đurić, S., Marijan, M., & Pavković, B. (2018). The Real-Time Detection of Traffic Participants Using YOLO Algorithm. 26th Telecommunications Forum (TELFOR), 1-5. doi:10.1109/TELFOR.2018.8611986
3. Kim, J.-G., Yoo, J.-H., & Koo, J.-C. (2018). Road and Lane Detection using Stereo Camera. International Conference on Big Data and Smart Computing (BigComp), 649-652. doi:10.1109/BigComp.2018.00117
4. Liu, C., Tao, Y., Liang, J., Li, K., & Chen, Y. (2018). Object Detection Based on YOLO Network. 4th Information Technology and Mechatronics Engineering Conference (ITOEC 2018), 799-803. doi:10.1109/ITOEC.2018.8740604
5. Nugraha, A., Su, S.-F., & Fahmizal. (2017). Convolutional, Towards Self-driving Car Using Convolutional Neural Network and Road Lane Detector. 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), 65-69. doi:10.1109/ICACOMIT.2017.8253388
6. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Conference on Computer Vision and Pattern Recognition (CVPR), 1-10. doi:10.1109/CVPR.2016.91
7. Shanmugamani, R. (2018). Deep Learning for Computer Vision. PACKT.
8. Tran, L.-A., & Le, M.-H. (2019). Robust U-Net-based Road Lane Markings Detection for Autonomous Driving. International Conference on System Science and Engineering (ICSSE), 62-66. doi:10.1109/ICSSE.2019.8823532
9. Zhang, R., Yang, Y., Wang, W., Zeng, L., Chen, J., & McGrath, S. (2018). An Algorithm for Obstacle Detection based on YOLO and Light Filed Camera. Twelfth International Conference on Sensing Technology (ICST), 223-226. doi:10.1109/ICSensT.2018.8603600
10. Zhao, X., Liu, P., Zhang, M., & Zhao, X. (2010). A Novel Line Detection Algorithm in Images Based on Improved Hough Transform and Wavelet Lifting Transform. International Conference on Information Theory and Information Security, 767-771. doi:10.1109/ICITIS.2010.568968.