SCHOOL OF ELECTRICAL ENGINEERING

ECE3502-IOT DOMAIN ANALYST

WINTER SEMESTER 2022 -23

J Component Report

Project Title : Lane and Object Detection Using Computer Vision and YOLO.

Slot: E1

Group Members:

1. Yash Trivedi – 20BEE0370
2. Yash Khetan-20BEE0381
3. Anamay Tiwari-20BEE0273
4. Aditya Raj-20BEE0209

Faculty In charge:

Dr. Washima Tasnin

Assistant Professor Sr. Grade1

SELECT

Abstract

This project's objective is to create and refine a lane recognition algorithm for use in applications involving driverless vehicles. The market for self-driving cars is expanding quickly. Numerous businesses are seeking to find every solution to this issue so that autonomous vehicles can operate securely on public roads. Due to the several facets, it depends on, including robotics, path planning, navigation, computer vision, mechanics, etc., it is an extremely complex topic.

The project's emphasis is on the computer vision component, a key component. An automated vehicle needs to be capable of perceiving and detecting every minute aspect in its environment if it is to navigate over uncertain terrain.

The system uses a camera installed on the car to gather the front image before using a few processes to identify the lanes. The proposed system finds lane pixels, fits them to determine the lane boundary, warps the discovered lane boundaries back onto the original image, and generates an output that shows the lane boundaries visually and provides a numerical estimate of the lane curvature and the position of the vehicle.

The suggested lane detecting technology is applicable to straight and curving roads, painted and unpainted, under a variety of weather conditions. This strategy was put to the test, and the findings indicate that the suggested method was reliable and quick enough for real-time needs. Finally, a critical analysis of the methodologies and their potential for future application were covered.

Introduction

Intelligent vehicles work in tandem with smart infrastructure in intelligent transportation systems to improve traffic flow and create a safer environment. Though enhancing safety by fully or partially automating driving responsibilities is a more compelling argument in favor of creating intelligent vehicles.

Road detection was one of these duties, and it played a significant part in driving assistance systems, which give information such as lane structure and vehicle position in relation to the lane. Detection of road or lane is indispensable for the environmental perception of advanced driver assistance systems. The objective of lane detection is to locate and track the lane boundaries in road images so that the vehicle can be maintained to run along the host lane. However, the demand for safety is the most compelling argument for adding autonomous capability to automobiles. Vehicle collisions continue to be the world's top source of accident fatalities and injuries, taking thousands of lives and harming millions of others every year. On the nation's roadways, these transportation-related fatalities and injuries predominate.

Most of the Deep CNN models created include the use of a monocular camera to input into the FCN network [7]. There are several models relying on Progressive Probabilistic Hough Transform (PPHT) to address the problem of straight and curved lane detection and tracking under different lighting conditions [8]. Our Proposed Work makes an efficient use of techniques like Color Transforms, Gradients to create a Thresholder Binary Images which makes further analysis based on Temporal-Spatial Model and Particle Filtering to make it more robust [12].

We use above-mentioned methods because the lines seem different in practice and because driving situations are frequently complex, detecting traffic lines can be difficult. Traffic lines have a variety and complexity of structures. A traffic line curve frequently has curves with different radii of curvature. Additionally, the lines can be solid or dotted. We need to figure out how to connect all the disconnected pieces on a semantic level for the discontinuous dotted lines.

Additionally, the continuous merging and detaching of traffic lines makes it extremely challenging to detect traffic lines accurately. Aside from the traffic line's outward looks, the complex external environment should also be carefully taken into account. This includes the weather (sunny/rainy/snowy), lighting (day/night), and other unforeseen variables (shadow/occlusion). Currently, most methods remain riddled with assumptions and limitations [18], still not good enough for safe and reliable driving in the real world [18].

Cameras are most popular due to their rich content features and affordable cost. Various objects need to be recognized in the procedure of environmental perception, such as lane markings, pedestrians, and adjacent vehicles. Similarly, different sensing modalities are used to achieve autonomous driving, for example, visual and thermal cameras, radars, Light Detection and Ranging (LiDAR) sensors, and ultrasonic sensors. These sensors are equipped alone or combined with others for self-driving vehicles to perceive the driving environment [20]. Among the many targets that need to be perceived, precise lane marking detection is of paramount importance for autonomous vehicles maintaining the advanced driving assistance systems (ADAS), which directly affects the behaviors of driving. Furthermore, the accurate identification of lane markings will be helpful for other transportation applications, such as the localization of vehicles, mobile mapping technology, and optimization of routes.

The incidence of curve accidents and the seriousness of accidents remain high. When the car is turning, there will be a blind zone of sight which is accompanied by increased centrifugal force. The turning radius will decrease and the lateral sliding will occur easily, which can cause collision accidents. In our proposed work, one of our key focus will be towards detecting and recognizing the road ahead before the advent of curved road conditions and warn the driver in advance [25].

Moreover, by choosing a region of interest (ROI), we can reduce computation and calculate an adaptive threshold to detect the lane boundary [1]. One of the most crucial components of ADAS and autonomous vehicles is lane-mark detection, which is also a requirement for lane departure warning. A geometric computer vision-based approach can be adopted as they are cheaper in computational cost for lane detection [2]. Also, drivers may have a certain blind spot of vision that leads to traffic accidents. To reduce the occurrence of traffic accidents, a real-time communication between vehicles and infrastructure can provide a wider view for driver and avoid the drivers' blind spots by studying the information interaction between the vehicle and the infrastructure. In order to achieve this, LiDAR sensors prove to be very helpful as they generate a 3D point cloud of surrounding objects and it can also work well in various non-normal scenarios, such as rain and darkness [24].

Also, for future improvement on Lane Detection Aspect in hilly areas or tough terrains we are further planning to implement a novel multitask attention method for lane marking detection, which combines deep and traditional methods [11]. We are also working on certain embedded techniques inspired from other works based on Confidence Area Detection and Field Inference [13]. Therefore, lane marking detection is the fundamental feature to develop the autonomous driving system. Several methods have been proposed to recognize the lane markings from camera images [16].

LITERATURE SURVEY

S. No.	**Paper Title**	**Journal Name**	**Work done**	**Technique Used**	**Limitations**	**Student Name**
1.	**Improved Lane Detection With Multilevel Features in Branch Convolutional Neural Networks**	IEEE Access (Volume: 7) 02 December 2019	A deep learning approach is adopted to achieve robust lane detection inspired by LaneNet model, which uses semantic segmentation concepts	Convolutional Neural Networks (CNN) Semantic Segmentation Based on LaneNet model with one encoder and two decoders	A powerful 16-Layer VGG-16 krnel encoder is adopted which increases system loading In TuSimple dataset, the pixels which are close to vanishing point and image boundary are not accurate enough	Yash Trivedi
2.	**Robust Lane Detection for Complicated Road Environment Based on Normal Map**	IEEE Access (Volume: 6) 06 September 2018	To address the issues of illumination and interference, such as vehicles and shadows, a robust method for road segmentation and lane detection based on a normal map is proposed.	A Combination of Hough transform and vanishing point Image preprocessing, feature extraction, lane model fitting	The generation of normal maps is very time consuming and needs to be accelerated to save computing resources. Low visibility of lanes is a challenge for this algorithm.	Yash Trivedi

3.	**Vehicle Lane Change Prediction on Highways Using Efficient Environment Representation and Deep Learning**	IEEE Access (Volume: 9) 23 August 2021	A novel method of lane-change and lane-keeping detection and prediction of surrounding vehicles based on Convolutional Neural Network classification approach is introduced.	Convolutional Neural Network (CNN) Digital image processing Deep Learning	It does not predict maneuvers at intersections from a static and extrinsic point of view, such as infrastructure cameras.	Yash Trivedi
4.	**A Cooperative Lane Change Model for Connected and Automated Vehicles**	IEEE Access (Volume: 8) 16 March 2020	Focus is laid on cooperative trajectory planning of lane changes for connected and automated vehicles. It considers the traffic scene with multiple mandatory lane change demands	The Gaussian mixture model, Continuous Hidden Markov Model, Deep Belief Network and Long Short-term Memory neural network are utilized	The model only focuses on the mandatory lane changes. As for the traffic scene with discretionary lane changes, the model should be optimized. Simulation experiments of more complex traffic scenes should be considered.	Yash Trivedi

5.	**A Deep Learning Method for Lane Changing Situation Assessment and Decision Making**	IEEE Access (Volume: 7) 11 September 2019	A deep learning model to simulate the situation assessment and decision making process during lane changing events is proposed which takes drivers historical experience and vehicle-to- vehicle memory effect into consideration.	Deep Neural Network (DNN) Recurrent Neural Network (RNN) Kalman Filtering	Model is based on limited dataset, which is obviously not suitable for all scenarios. No analysis on the lane change maneuvers in some specific situations, such as mandatory lane changing and emergency lane changing events.	Yash Trivedi
6.	**A Method for Predicting Diverse Lane-Changing Trajectories of Surrounding Vehicles Based on Early Detection of Lane Change**	IEEE Access (Volume: 10) 07 February 2022	A behavior-based method of predicting diverse lane-changing trajectories of surrounding vehicles with lane-changing behavior recognition and diverse lane-changing trajectory prediction is proposed.	A lane changing behavior recognition model based on the Continuous Hidden Markov Model Long Short-Term Memory Neural Network (LSTM) Generative Adversarial Network	It does not consider the bidirectional interaction between the lane changing vehicle and the surrounding vehicle, which may affect the accuracy of prediction This model does not account on the lane changing intention which may be inferred from the relative speed and distance of surrounding	Aditya Raj

7.	**Multi-Lane Detection Based on Deep Convolutional Neural Network**	IEEE Access (Volume: 7) 16 October 2019	Classification of lane images and achieving end-to-end detection of multi-lane images using Deep CNN	deep convolutional neural network based on FCN network, Hough Transform, Least Square Method	It can't present all lanes, such as the lane marking in front of the turning car. Variable Accuracy of the model based on Environmental Conditions	Aditya Raj
8.	**A Lane Tracking Method Based on Progressive Probabilistic Hough Transform**	IEEE Access (Volume: 8) 04 May2020	addressing the problem of straight and curved lane detection and tracking under different lighting conditions using PPHT	Progressive Probabilistic Hough Transform (PPHT), Kalman filter	PPHTs require much processing time and more processing power	Aditya Raj
9.	**Graph-Embedded Lane Detection**	IEEE Transactions on Image Processing (Volume: 30) 10 February 2021	a learning-based low-level lane feature extraction algorithm, and a graph-embedded lane inference algorithm	graph representation,semantic segmentation, deep learning, Hough Transformation	the ROI (Region of Interest) location cannot be correctly estimated in more diverse and complex road environments	Aditya Raj

10.	**Multiple Lane Detection via Combining Complementary Structural Constraints**	IEEE Transactions on Intelligent Transportation Systems (Volume: 22, Issue: 12, December 2021)	a practical and robust method for multiple lane detection by including length, parallel, distribution, pair, and uniform width constraints	Hough Transform, Dynamic Programming	Missed Detection Ratio is quite higher as compared to other Deep Learning Based Methods	Aditya Raj
11.	**Multitask Attention Network for Lane Detection and Fitting**	IEEE Transactions on Neural Networks and Learning Systems (Volume: 33, Issue: 3, March 2022)	a novel multitask attention method for lane marking detection, which combines deep and traditional methods	Inverse perspective mapping (IPM), Lane fitting, Lane Segmentation	Variation in IPM and Fitting leads to higher rates of False Positive.	Anamay Tiwari

12.	**Multi-Lane Detection and Tracking Using Temporal-Spatial Model and Particle Filtering**	IEEE Transactions on Intelligent Transportation Systems (Volume: 23, Issue: 3, March 2022)	Extension of the previous lane detection and tracking processes from the spatial domain to the temporal-spatial domain using more robust and general multiple lane model	weak lane model, temporal-spatial model, particle filter, lane tracking	road conditions with wide range text signs and the road edge strip that is very similar to a lane boundary results in false detection cases And affects the efficiency of the model	Anamay Tiwari
13.	**Lane Marking Regression From Confidence Area Detection to Field Inference**	**IEEE Transactions on Intelligent Vehicles (Volume: 6, Issue: 1, March 2021)**	a novel method, named as Lane marking Regression Network (LRN), which can simultaneously consider confidence area detection and field inference for producing more precise lane markings in an unified encoder-decoder framework	Lane marking Regression Network (LRN), Confidence Area, Field Inference	Dataset is collected by cellphones mounted on driving vehicles which results in low pixel quality images, Thereby reducing the efficiency of the overall model	Anamay Tiwari

14.	**A Novel Strategy for Road Lane Detection and Tracking Based on a Vehicle's Forward Monocular Camera**	**IEEE Transactions on Intelligent Transportation Systems** Year: 2018	novel strategy for lane detection and tracking, which fits as a functional requirement to deploy DAS features like Lane Departure Warning and Lane Keeping Assist.	Digital image processing Canny Edge Algorithm Hough Transform	The tracking suffers a position deviation in relation to the real lane marking due to opposite vehicles headlight incidence. The visual representation of the road far-field tracking is disabled if there are no valid coordinates in the set of points.	Anamay Tiwari
15.	**Line-CNN: End-to-End Traffic Line Detection With Line Proposal Unit**	**IEEE Transactions on Intelligent Transportation Systems** Year : 2019	this paper focuses on the task of traffic line detection an end-to-end system called Line-CNN (L-CNN), in which the key component is a novel line proposal unit (LPU). The LPU utilizes line proposals as references to locate accurate traffic curves, which forces the system to learn the global feature	Line-CNN (L-CNN) Line Proposal Unit (LPU) Loss Function for Learning Line-CNN	the backbone model with smaller depth tends to have a sharp performance drop for probably two reasons: (i) its small complexity lacks powerful feature representation ability (ii) the perceptive field is not large enough to capture the long "line" objects. Additionally, the improvement of the deeper model	Anamay Tiwari

16.	**Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks**	**IEEE Transactions on Vehicular Technology** Year : 2019	lane detection by using multiple frames of a continuous driving scene, and propose a hybrid deep architecture by combining the convolutional neural network (CNN) and the recurrent neural network (RNN).	convolutional neural network (CNN) recurrent neural network (RNN). semantic segmentation	The proposed models take a sequence of images as input, it may cost more running time. if too many previous frames are used, the outcome may be not good as lane situations in far former frames are sometimes significantly different from the current frame.	Yash Khetan
17.	**Bridging the Gap of Lane Detection Performance Between Different Datasets: Unified Viewpoint Transformation**	**IEEE Transactions on Intelligent Transportation Systems** Year : 2020	unified viewpoint transformation (UVT) method that transforms the camera viewpoints of different datasets into a common virtual world coordinate system, such that the mismatched lane position distributions can be effectively	Lane Segmentation Network Lane Position Bias Unified Viewpoint Transformation (UVT)	The lane line segmentation results might include some incorrectly detected regions, which affects the evaluation score.	Yash Khetan

18.	**Super: A Novel Lane Detection System**	**IEEE Transactions on Intelligent Vehicles** Year : 2021	1) a hierarchical semantic segmentation network as the scene feature extractor. 2) a physics enhanced multi-lane parameter optimization module for lane inference	CNN Hierarchical semantic segmentation Multi-level classifier design	but for accurate estimation of unparalleled lanes, such as lane merge and split conditions, additional operations/strategies e.g., extra local correction or integration with map prior, are required	Yash Khetan
19.	**Real-Time Road Curb and Lane Detection for Autonomous Driving Using LiDAR Point Clouds**	**IEEE Access** Year : 2021	A real-time lane marking detection method by using LiDAR point clouds is proposed along with a road curb detection method based on segment point density.	A constrained RANSAC algorithm is applied to select the regions of interest and filter the background data,	LiDAR sensors have the ability to offer three-dimensional (3D) information of objects. However, the 3D information increases the processing time and complexity due to the tremendous amount of data.	Yash Khetan

20.	**A Survey of Autonomous Driving: Common Practices and Emerging Technologies**	**IEEE Access** Year : 2020	This paper discusses unsolved problems and surveys the technical aspect of automated driving. Studies regarding present challenges, high-level system architectures, emerging methodologies and core functions were thoroughly reviewed	Deep learning model Architecture Models Different Sensors also explained to use in ADS.	It's only survey that show results of which type of technology can be help to make best ADS No disadvantage of survey for information only.	Yash Khetan

Overall Architecture

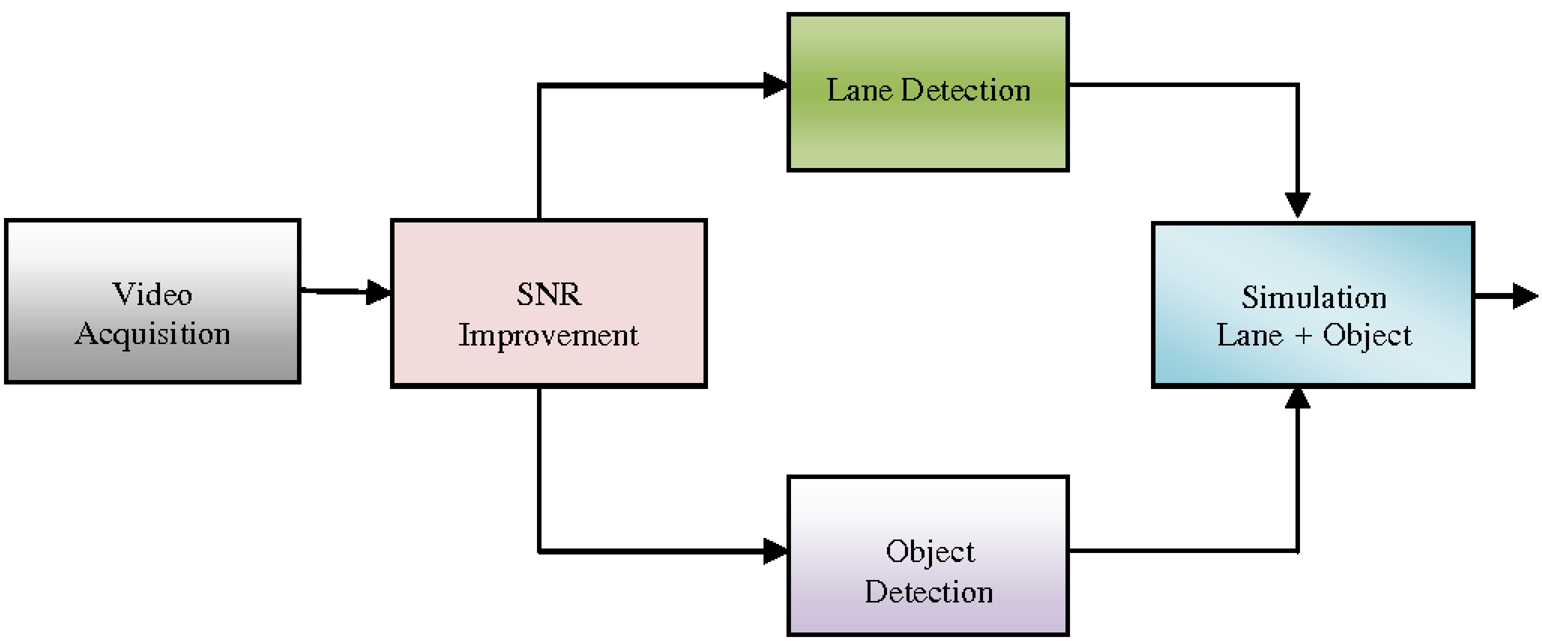
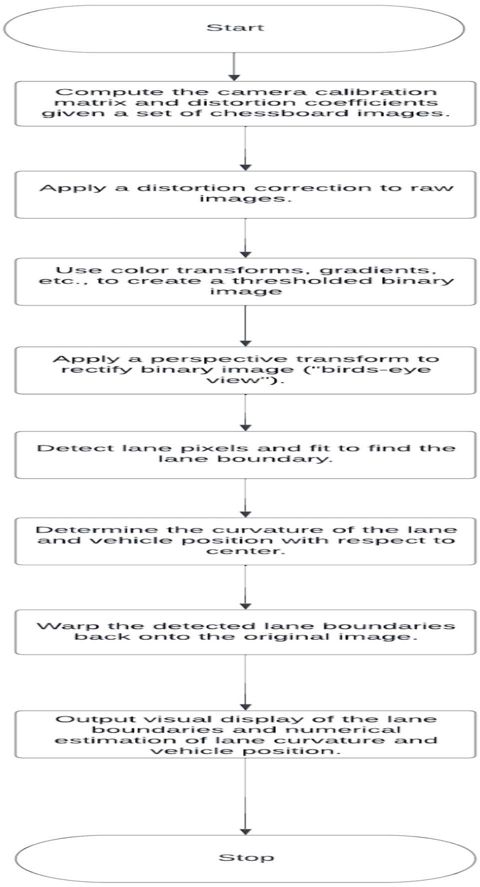


Figure 1 General Block diagram of proposed Research

Proposed Methodology



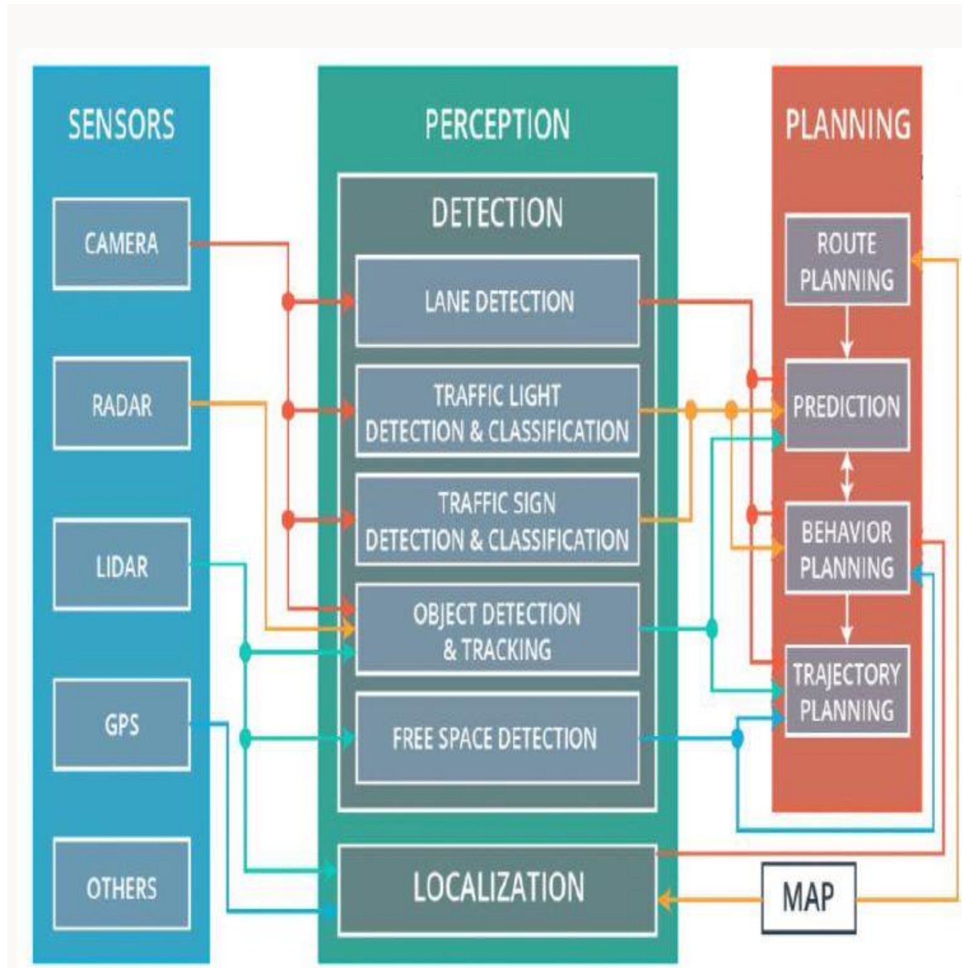


Fig: Algorithm we applied

Fig: Working Flow of Modules

Discussion

In this Project we have first implemented existing module for detecting Lane for vehicle and we successfully able to detect lane using lane marking and after that we have developed improved version of algorithm in which it will detect the near by objects and show the percentage of its occurrence of specific object.

We have Connected TensorFlow Object Detection API with this project where it will Detect the object and will display on user screen.

We have then used YOLO (You only look once) algorithm for object detection which is a popular object detection model known for its speed and accuracy

Details Work flow given below in this Paper.

Detailed Explanation of Methodology

Camera Calibration

Using the camera calibration matrices in 'calibrate_camera.py', We undistort the input image. Below is the example image above, undistorted:

Fig: Undistorted Image

The code to perform camera calibration is in 'calibrate_camera.py'.

Thresholded binary image

The next step is to create a thresholded binary image, taking the undistorted image as input. The goal is to identify pixels that are likely to be part of the lane lines. In particular, We perform the following:

- Apply the following filters with thresholding, to create separate "binary images" corresponding to each individual filter
  - Absolute horizontal Sobel operator on the image
  - Sobel operator in both horizontal and vertical directions and calculate its magnitude
  - Sobel operator to calculate the direction of the gradient
  - Convert the image from RGB space to HLS space, and threshold the S channel
- Combine the above binary images to create the final binary image

Here is the example image, transformed into a binary image by combining the above thresholded binary filters:

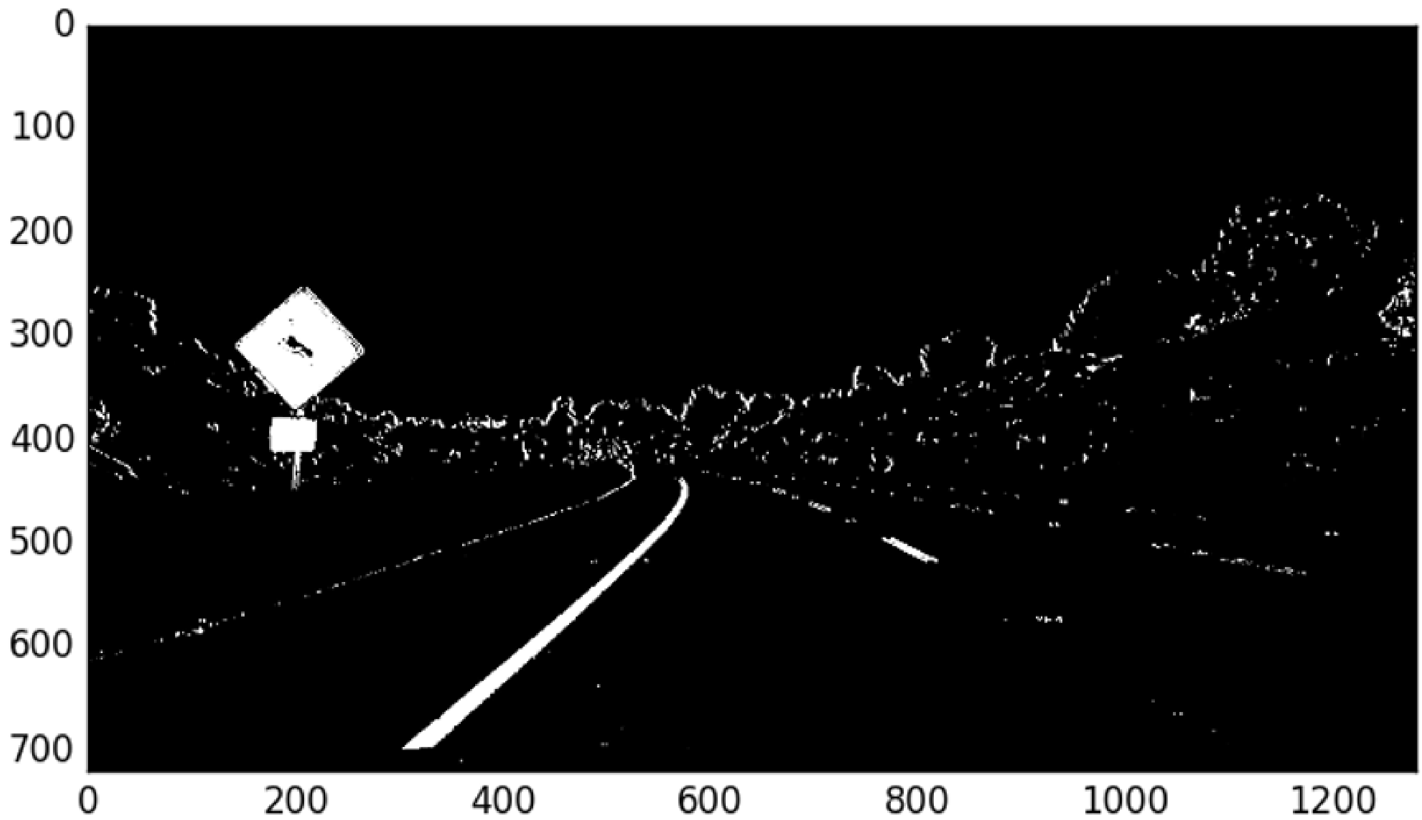


Fig: Thresholded Binary Image

The code to generate the thresholded binary image is in 'combined_thresh.py', in particular the function combined_thresh(). For all images in 'test_images/*.jpg', the thresholded binary version of that image is saved in 'output_images/binary_*.png'.

Perspective transform

Given the thresholded binary image, the next step is to perform a perspective transform. The goal is to transform the image such that we get a "bird's eye view" of the lane, which enables us to fit a curved line to the lane lines (e.g., polynomial fit). Another thing this accomplishes is to "crop" an area of the original image that is most likely to have the lane line pixels.

To accomplish the perspective transform, We use OpenCV's get PerspectiveTransform() and warpPerspective() functions. We hard-code the source and destination points for the perspective transform. The source and destination points were visually determined by manual inspection, although an important enhancement would be to algorithmically determine these points.

Here is the example image, after applying perspective transform:

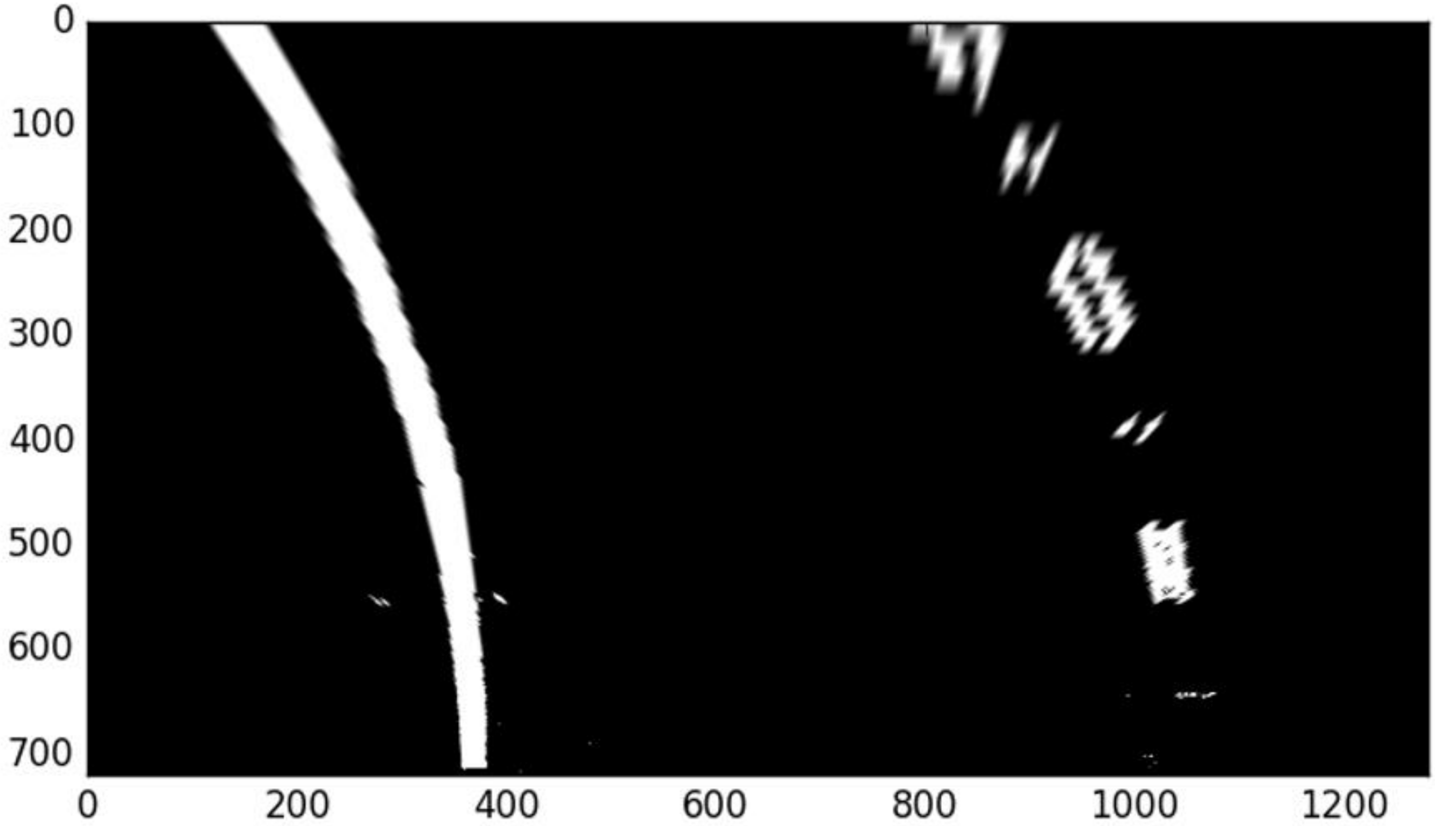


Fig: Image After perspective transform

The code to perform perspective transform is in 'perspective_transform.py', in particular the function perspective_transform(). For all images in 'test_images/*.jpg', the warped version of that image (i.e., post-perspective-transform) is saved in 'output_images/warped_*.png'.

Polynomial fit

Given the warped binary image from the previous step, we now fit a 2nd order polynomial to both left and right lane lines. In particular, we perform the following:

- Calculate a histogram of the bottom half of the image
- Partition the image into 9 horizontal slices
- Starting from the bottom slice, enclose a 200 pixel wide window around the left peak and right peak of the histogram (split the histogram in half vertically)
- Go up the horizontal window slices to find pixels that are likely to be part of the left and right lanes, recentering the sliding windows opportunistically
- Given 2 groups of pixels (left and right lane line candidate pixels), fit a 2nd order polynomial to each group, which represents the estimated left and right lane lines

The code to perform the above is in the line_fit() function of 'line_fit.py'.
Since our goal is to find lane lines from a video stream, we can take advantage of the temporal correlation between video frames.

Given the polynomial fit calculated from the previous video frame, one performance enhancement We implemented is to search +/- 100 pixels horizontally from the previously predicted lane lines. Then we simply perform a 2nd order polynomial fit to those pixels found from our quick search. In case we don't find enough pixels, we can return an error (e.g. return None), and the function's caller would ignore the current frame (i.e. keep the lane lines the same) and be sure to perform a full search on the next frame. Overall, this will improve the speed of the lane detector, useful if we were to use this detector in a production self-driving car. The code to perform an abbreviated search is in the tune_fit() function of 'line_fit.py'.

Another enhancement to exploit the temporal correlation is to smooth-out the polynomial fit parameters. The benefit to doing so would be to make the detector more robust to noisy input. We used a simple moving average of the polynomial coefficients (3 values per lane line) for the most recent 5 video frames. The code to perform this smoothing is in the function add_fit() of the class Line in the file 'Line.py'. The Line class was used as a helper for this smoothing function specifically, and Line instances are global objects in 'line_fit.py'.
Below is an illustration of the output of the polynomial fit, for our original example image. For all images in 'test_images/*.jpg', the polynomial-fit-annotated version of that image is saved in 'output_images/polyfit_*.png'.

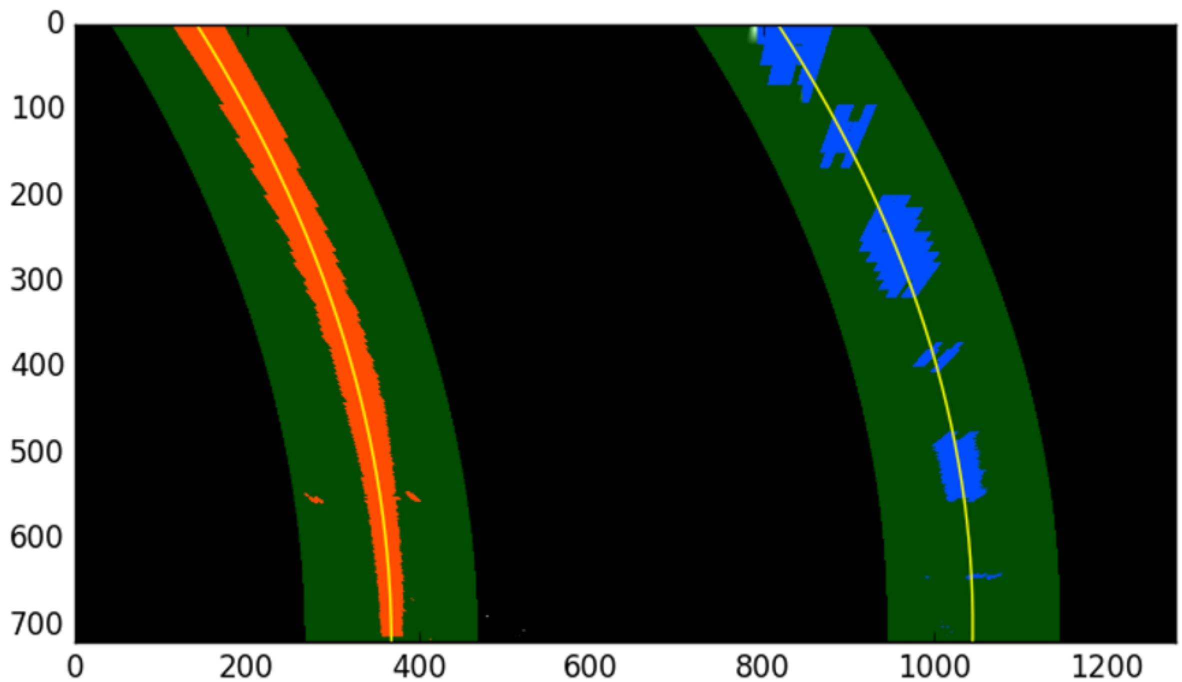


Fig: Image After polynomial Fit

Radius of curvature

Given the polynomial fit for the left and right lane lines, I calculated the radius of curvature for each line according to formulas presented here. I also converted the distance units from pixels to meters, assuming 30 meters per 720 pixels in the vertical direction, and 3.7 meters per 700 pixels in the horizontal direction.

Finally, I averaged the radius of curvature for the left and right lane lines, and reported this value in the final video's annotation.

The code to calculate the radius of curvature is in the function calc_curve() in 'line_fit.py'.

Vehicle offset from lane center

Given the polynomial fit for the left and right lane lines, I calculated the vehicle's offset from the lane center. The vehicle's offset from the center is annotated in the final video. I made the same assumptions as before when converting from pixels to meters.

To calculate the vehicle's offset from the center of the lane line, I assumed the vehicle's center is the center of the image. I calculated the lane's center as the mean x value of the bottom x value of the left lane line, and bottom x value of the right lane line. The offset is simply the vehicle's center x value (i.e. center x value of the image) minus the lane's center x value.

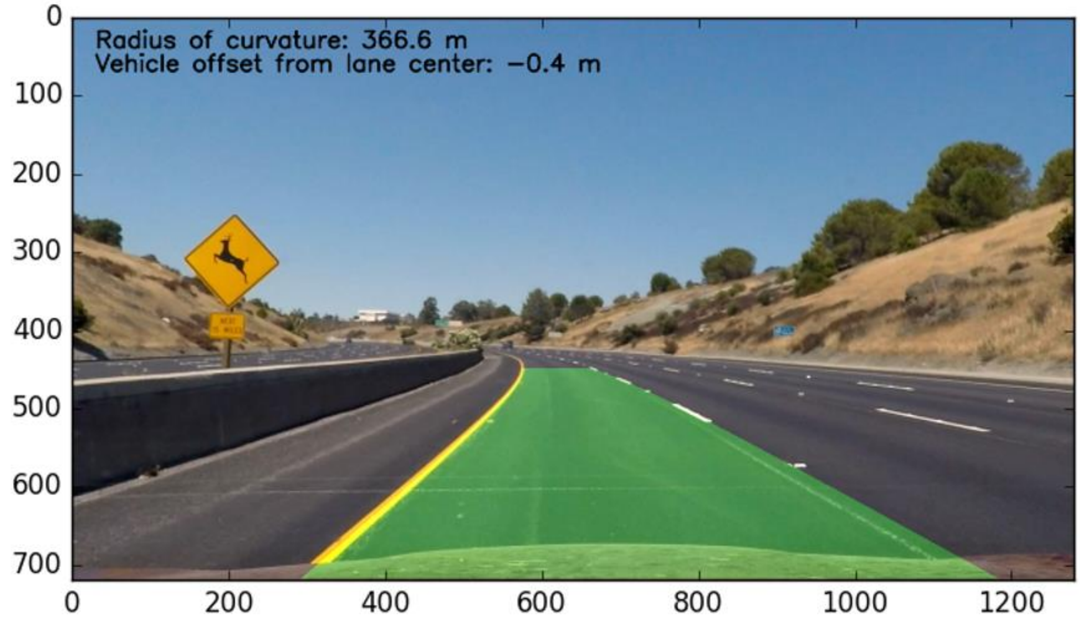
The code to calculate the vehicle's lane offset is in the function calc_vehicle_offset() in 'line_fit.py'.

Annotate original image with lane area

Given all the above, we can annotate the original image with the lane area, and information about the lane curvature and vehicle offset. Below are the steps to do so:

- Create a blank image, and draw our polyfit lines (estimated left and right lane lines)
- Fill the area between the lines (with green color)
- Use the inverse warp matrix calculated from the perspective transform, to "unwarp" the above such that it is aligned with the original image's perspective
- Overlay the above annotation on the original image
- Add text to the original image to display lane curvature and vehicle offset

The code to perform the above is in the function final_viz() in 'line_fit.py'.
Below is the final annotated version of our original image. For all images in 'test_images/*.jpg', the final annotated version of that image is saved in 'output_images/annotated_*.png'.



Advantages of methods Used:

1. Camera calibration yields parameters which describe the relationship between 2D images and 3D environment
2. They allow a correct 3D reconstruction from the given images. This information is indispensable for lane and obstacle detection with the goal of driving an autonomous, unsupervised vehicle. The impacts of uncertainty in the calibration parameters on lane recognition and obstacle detection are quantified
3. OpenCV has multiple functions to utilise different colourspaces. One more thing to note though is that OpenCV by default reads an image in BGR which can be converted to RGB
4. In Perspective Transform, lines form when planes in the 3D space cut through the origin and intersect the Z=1 plane. In computer graphics and computer visions, the Homogenous coordinates in the projective space offer a few advantages compared to the Cartesian coordinates system in the Euclidean space.

Results & Analysis

Existing Approach

Algorithm: Advanced Lane Detection Using Computer Vision

Input: Video from car dash camera

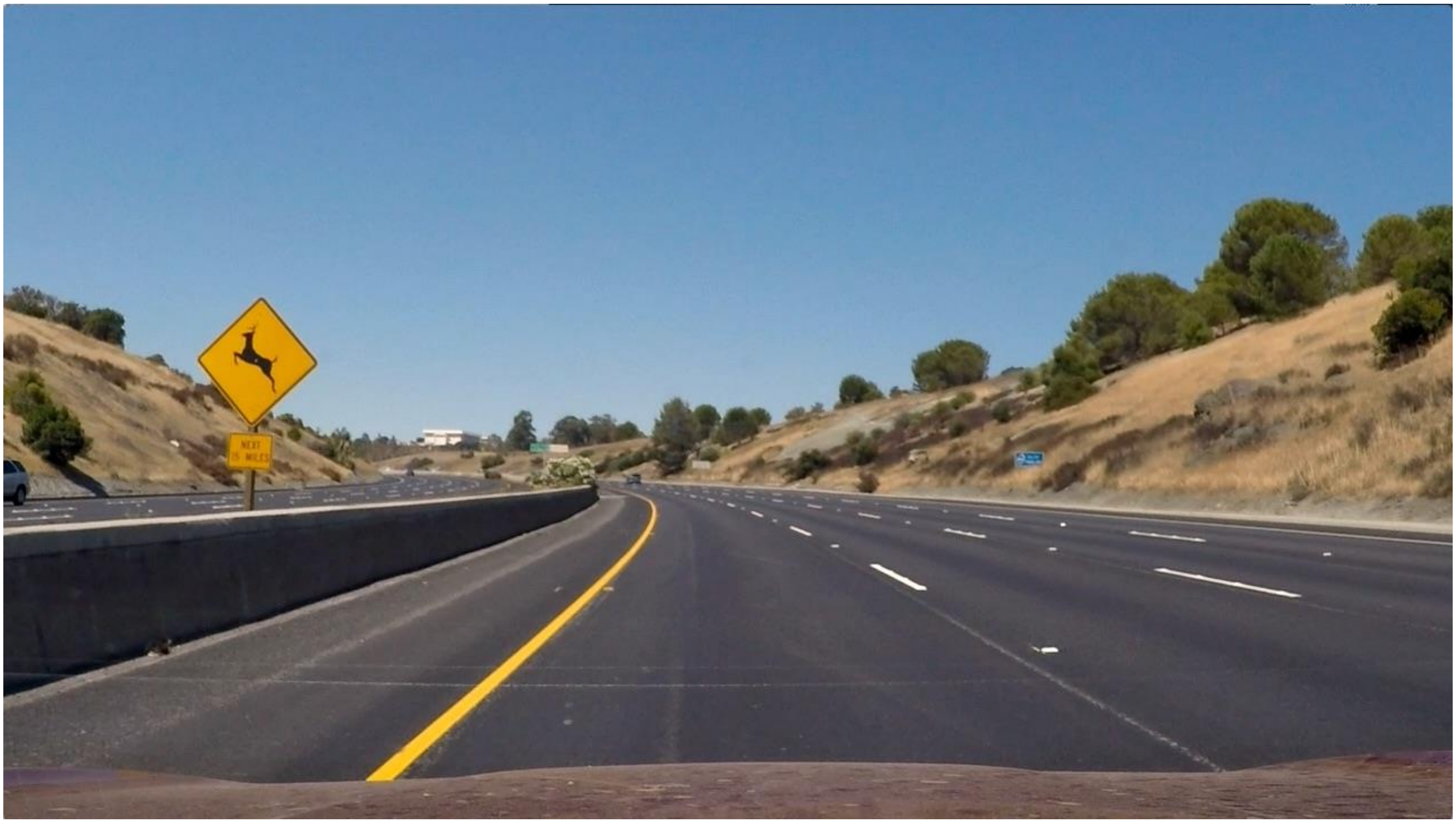


Fig showing single video frame

Output: Video showing Detected Lanes using the algorithm

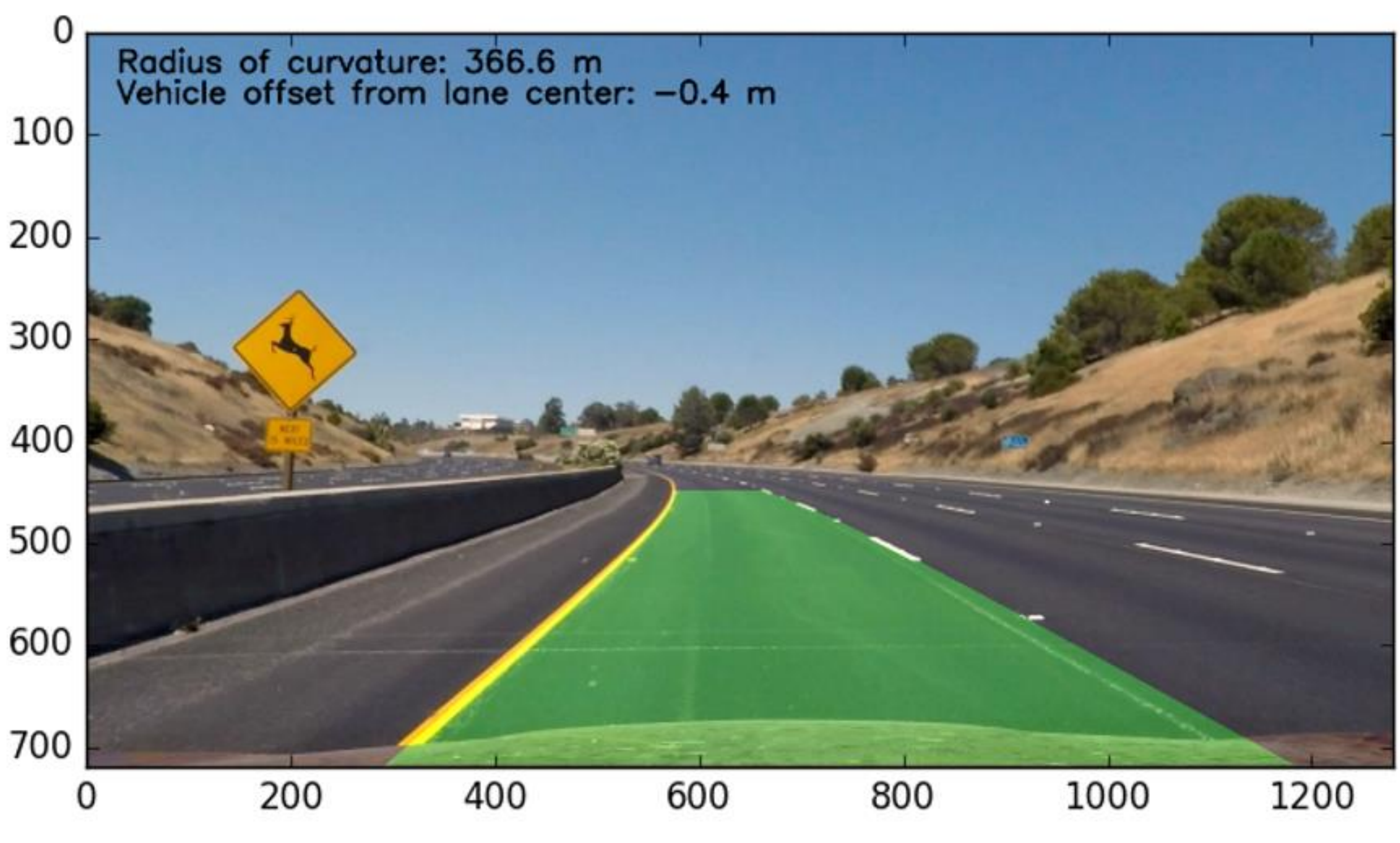


Fig showing Detected Lane Path on a Video Frame

Improved Approach (YOLO)

Algorithm Used: Advanced Lane and Object Detection using TensorFlow
Input: Video from car dash camera

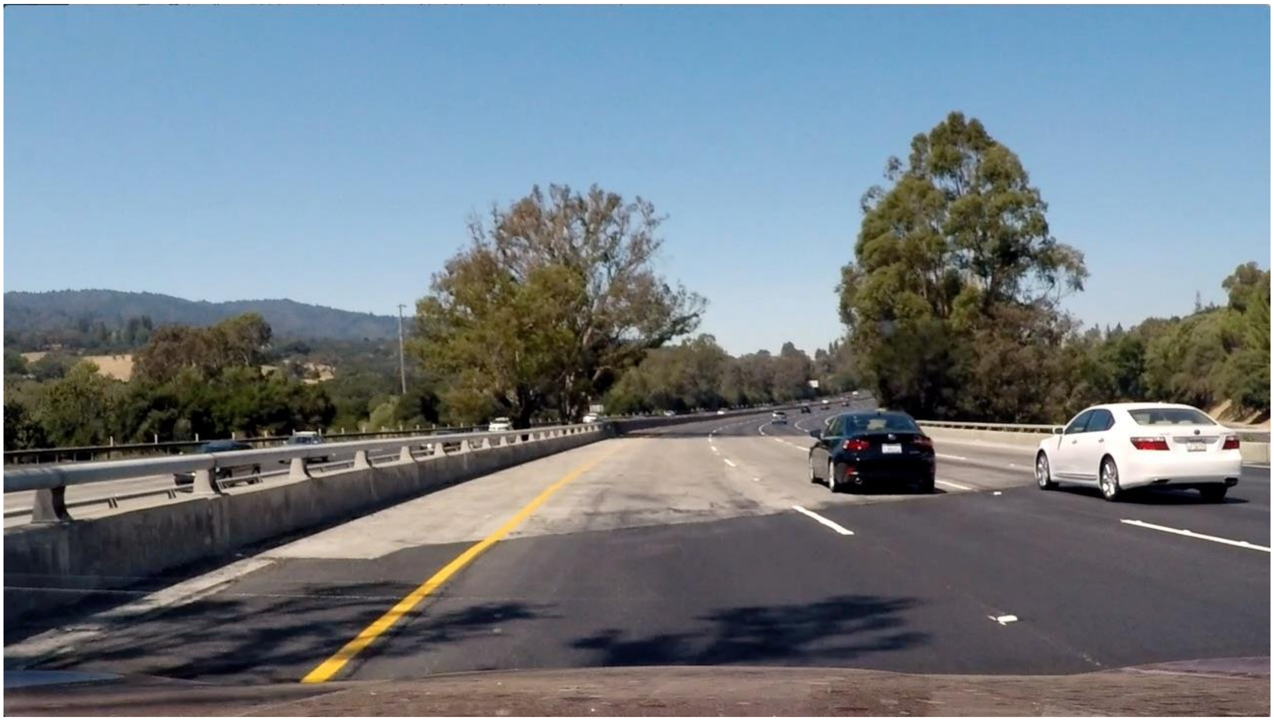


Fig showing single video frame

Output: Video showing Detected Lanes using the algorithm

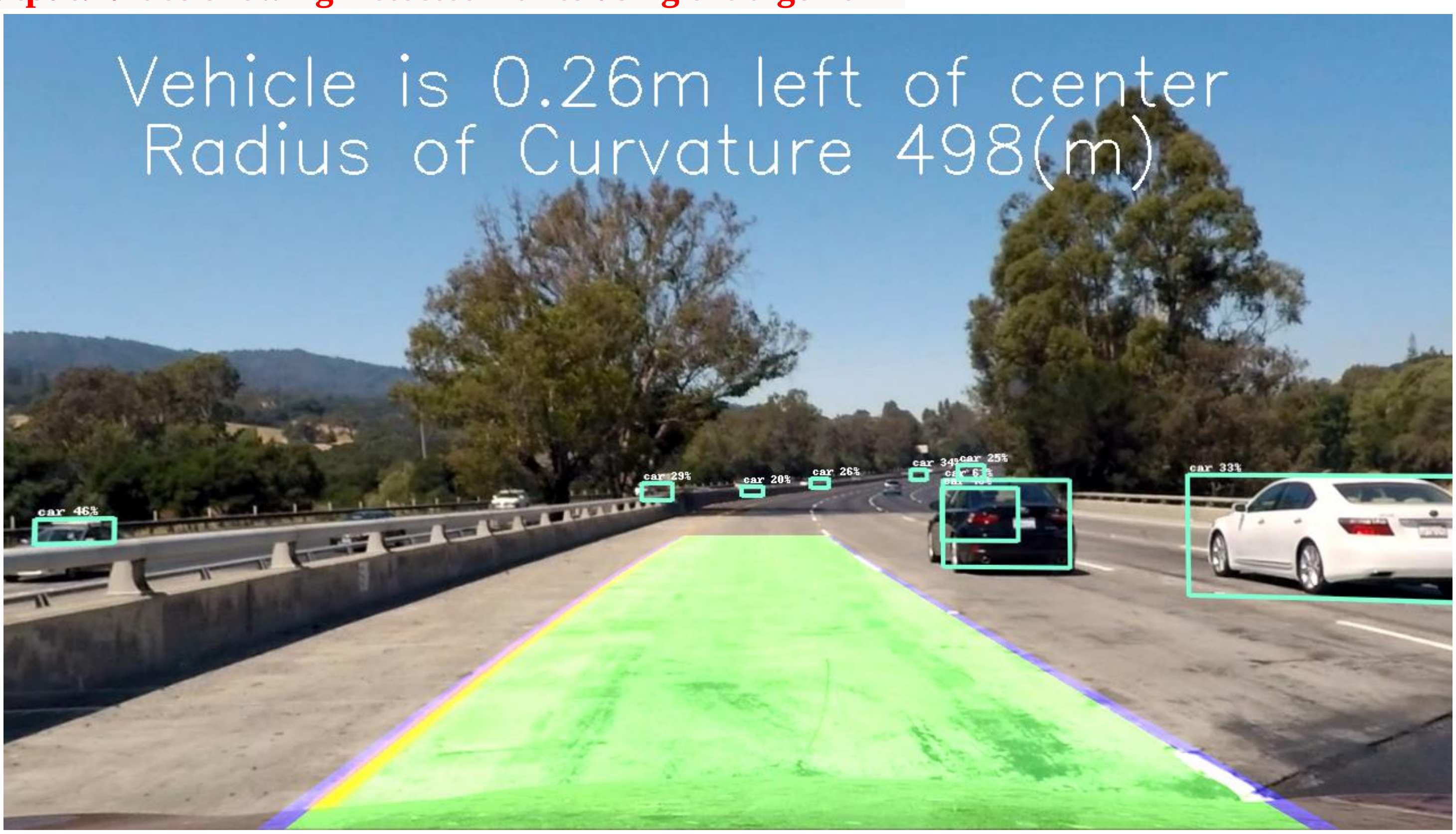


Fig showing detected Lane path + Objects in a video frame

Novelty Approach Explanation:

In this project We have used OpenCV and the TensorFlow Object Detection API to identify the lane lines and other vehicles on the highway. We have also attempted to measure the radius of curvature of the road and calculate how far the car has deviated from the center of the lane.
We have given a pre-defined object dataset to the trained model for detecting the obstacles in the surrounding environment with Lane detection.

Conclusion and Future Work:

Thus, here we have created a model which measures the radius of curvature, vehicle position from the center of the lane markings, the percentage of distance between the obstacle and the vehicle. This model can detect Lane Paths and Objects present in the surrounding environment based on the trained model. For Future, we would like to implement this as a fully automated module for Self-Driving Cars in terms of detecting obstacles in front of car and further controllability of cars using Deep Learning Algorithms.

References

[1]. Yang, W. J., Cheng, Y. T., & Chung, P. C. (2019). Improved Lane Detection With Multilevel Features in Branch Convolutional Neural Networks. *IEEE Access*, *7*, 173148–173156. https://doi.org/10.1109/access.2019.2957053

[2]. Yuan, C., Chen, H., Liu, J., Zhu, D., & Xu, Y. (2018). Robust Lane Detection for Complicated Road Environment Based on Normal Map. *IEEE Access*, *6*, 49679–49689. https://doi.org/10.1109/access.2018.2868976

[3]. Izquierdo, R., Quintanar, A., Lorenzo, J., Garcia-Daza, I., Parra, I., Fernandez-Llorca, D., & Sotelo, M. A. (2021). Vehicle Lane Change Prediction on Highways Using Efficient Environment Representation and Deep Learning. *IEEE Access*, *9*, 119454–119465. https://doi.org/10.1109/access.2021.3106692

[4]. Li, T., Wu, J., Chan, C. Y., Liu, M., Zhu, C., Lu, W., & Hu, K. (2020). A Cooperative Lane Change Model for Connected and Automated Vehicles. *IEEE Access*, *8*, 54940–54951. https://doi.org/10.1109/access.2020.2981169

[5]. Liu, X., Liang, J., & Xu, B. (2019). A Deep Learning Method for Lane Changing Situation Assessment and Decision Making. *IEEE Access*, *7*, 133749–133759. https://doi.org/10.1109/access.2019.2940853

[6]. Ren, Y. Y., Zhao, L., Zheng, X. L., & Li, X. S. (2022). A Method for Predicting Diverse Lane-Changing Trajectories of Surrounding Vehicles Based on Early Detection of Lane Change. *IEEE Access*, *10*, 17451–17472. https://doi.org/10.1109/access.2022.3149269

[7]. Chao, F., Yu-Pei, S., & Ya-Jie, J. (2019). Multi-Lane Detection Based on Deep Convolutional Neural Network. *IEEE Access*, *7*, 150833–150841. https://doi.org/10.1109/access.2019.2947574

[8]. Marzougui, M., Alasiry, A., Kortli, Y., & Baili, J. (2020). A Lane Tracking Method Based on Progressive Probabilistic Hough Transform. *IEEE Access*, *8*, 84893–84905. https://doi.org/10.1109/access.2020.2991930

[9]. Lu, P., Xu, S., & Peng, H. (2021). Graph-Embedded Lane Detection. *IEEE Transactions on Image Processing*, *30*, 2977–2988. https://doi.org/10.1109/tip.2021.3057287

[10]. Luo, S., Zhang, X., Hu, J., & Xu, J. (2021). Multiple Lane Detection via Combining Complementary Structural Constraints. *IEEE Transactions on Intelligent Transportation Systems*, *22*(12), 7597–7606. https://doi.org/10.1109/tits.2020.3005396

[11]. Wang, Q., Han, T., Qin, Z., Gao, J., & Li, X. (2022). Multitask Attention Network for Lane Detection and Fitting. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(3), 1066–1078. https://doi.org/10.1109/tnnls.2020.3039675

[12]. Chen, S., Huang, L., Chen, H., & Bai, J. (2022). Multi-Lane Detection and Tracking Using Temporal-Spatial Model and Particle Filtering. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 2227–2245. https://doi.org/10.1109/tits.2020.3035614

[13]. Lv, H., Liu, C., Zhao, X., Xu, C., Cui, Z., & Yang, J. (2021). Lane Marking Regression From Confidence Area Detection to Field Inference. *IEEE Transactions on Intelligent Vehicles*, *6*(1), 47–56. https://doi.org/10.1109/tiv.2020.3009366

[14]. Andrade, D. C., Bueno, F., Franco, F. R., Silva, R. A., Neme, J. H. Z., Margraf, E., Omoto, W. T., Farinelli, F. A., Tusset, A. M., Okida, S., Santos, M. M. D., Ventura, A., Carvalho, S., & Amaral, R. D. S. (2019). A Novel Strategy for Road Lane Detection and Tracking Based on a Vehicle's Forward Monocular Camera. *IEEE Transactions on Intelligent Transportation Systems*, *20*(4), 1497–1507. https://doi.org/10.1109/tits.2018.2856361

[15]. Li, X., Li, J., Hu, X., & Yang, J. (2020). Line-CNN: End-to-End Traffic Line Detection With Line Proposal Unit. *IEEE Transactions on Intelligent Transportation Systems*, *21*(1), 248–258. https://doi.org/10.1109/tits.2019.2890870

[16]. Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., & Wang, Q. (2020). Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks. *IEEE Transactions on Vehicular Technology*, *69*(1), 41–54. https://doi.org/10.1109/tvt.2019.2949603

[17]. Wen, T., Yang, D., Jiang, K., Yu, C., Lin, J., Wijaya, B., & Jiao, X. (2021). Bridging the Gap of Lane Detection Performance Between Different Datasets: Unified Viewpoint Transformation. *IEEE Transactions on Intelligent Transportation Systems*, *22*(10), 6198–6207. https://doi.org/10.1109/tits.2020.2989349

[18]. Lu, P., Cui, C., Xu, S., Peng, H., & Wang, F. (2021). SUPER: A Novel Lane Detection System. *IEEE Transactions on Intelligent Vehicles*, *6*(3), 583–593. https://doi.org/10.1109/tiv.2021.3071593

[19]. Huang, J., Choudhury, P. K., Yin, S., & Zhu, L. (2021). Real-Time Road Curb and Lane Detection for Autonomous Driving Using LiDAR Point Clouds. *IEEE Access*, *9*, 144940–144951. https://doi.org/10.1109/access.2021.3120741

[20]. Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A Survey of Autonomous Driving: *Common Practices and Emerging Technologies*. *IEEE Access*, *8*, 58443–58469. https://doi.org/10.1109/access.2020.2983149

[21]. Feng, D., Haase-Schutz, C., Rosenbaum, L., Hertlein, H., Glaser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2021). Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3), 1341–1360. https://doi.org/10.1109/tits.2020.2972974

[22]. Munir, F., Azam, S., Jeon, M., Lee, B. G., & Pedrycz, W. (2022). LDNet: End-to-End Lane Marking Detection Approach Using a Dynamic Vision Sensor. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 9318–9334. https://doi.org/10.1109/tits.2021.3102479

[23]. Dewangan, D. K., & Sahu, S. P. (2021). Driving Behavior Analysis of Intelligent Vehicle System for Lane Detection Using Vision-Sensor. *IEEE Sensors Journal*, *21*(5), 6367–6375. https://doi.org/10.1109/jsen.2020.3037340

[24]. Lin, C., Guo, Y., Li, W., Liu, H., & Wu, D. (2021). An Automatic Lane Marking Detection Method With Low-Density Roadside LiDAR Data. *IEEE Sensors Journal*, *21*(8), 10029–10038. https://doi.org/10.1109/jsen.2021.3057999

[25]. Wang, H., Wang, Y., Zhao, X., Wang, G., Huang, H., & Zhang, J. (2019). Lane Detection of Curving Road for Structural Highway With Straight-Curve Model on Vision. *IEEE Transactions on Vehicular Technology*, *68*(6), 5321–5330. https://doi.org/10.1109/tvt.2019.2913187