Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020

# ADVANCED LANE AND VEHICLE DETECTION FOR SELF-DRIVING CARS

*Shushma S Jagtap[1], Janani S[2], Kaaviya N[3], Madhumitha K[4], Meenakshi L[5]*
*[1]Assistant Professor, [2]U G Scholar*
*Department of Electronics and Communication Engineering,*
*Rajalakshmi Engineering College, Chennai, India*

***Abstract****—This paper produces an efficient working model of lane and vehicle detection system using computer vision and machine learning. Lane and vehicle detection are used in autonomous vehicles to track the road lanes and avoid collision respectively. The input is a live image and it is processed into a 3D format to keep track of the lane and to detect vehicles by analysing the distance of the object detected with respect to the host system. This method detects curved lanes and forward objects . The proposed model detects lanes using CNN and vehicles using YOLO. This is achieved with the help of object detection and recognition using ML models. To enhance the accuracy and real-time performance of lane line and object detection, a lane line and object detection algorithm by Perspective Transform, color masking and gradient threshold can be used.*

*Keywords — Curved Lane Detection, Object Detection, Collision prevention, Machine Learning, Computer Vision, Advanced Lane Detection, YOLO.*

## I. INTRODUCTION

Autonomous road vehicles are increasingly becoming more important and there are several techniques and sensors that are being applied for vehicle control. For Lane Detection, we should teach the computer about the road. When the computer is aware of the lane, the vehicle will avoid the risk of running into other lanes and prevent accidents. Despite best practices, traffic laws, and numerous safety campaigns, collisions happen every single day. With the help of computers we can overcome these problems. Computer-vision techniques and Machine Learning together provide the capabilities of detection of the lane and prevents collision[1].

The lane detection is done through Perspective Transform, Colour Masking and gradient threshold. A Perspective Transform maps the points in the given image to different, desired, image points[1]. To identify the lane lines, colour masking and gradient threshold is used. During colour transformation from RGB to GrayScale, some of the information is lost. To overcome this, HSL Transform is used. Transformed image is used to Identify the lane line. Histogram will be applied on the first frame and this same pipeline will go on for every frame in a video[8]. Polynomial Coefficient is used to compute the radius of curvature and deviation of a car from mid of lane.

For object detection Aggregate channel features (ACF) is used. For pedestrian and vehicle detection ACF is applied[4]. And also compute the vehicle parameters - position velocity, time to collision. If we take the midpoint on the lower edge as the positional reference to the vehicle we can compute all these parameters[2].

For collision avoidance, we are primarily concerned with vehicles immediately near the camera in the same lane as the ego vehicle. We can ignore some vehicles traveling across the road in the opposite

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020

direction. Therefore, we could report the velocity of a vehicle, time to a collision, and the radius of curvature of the road at a point. This runs real-time on a desktop or laptop without a GPU. Parameters to which the processing speed is sensitive are: the time after which it triggers YOLO, the frame rate of the incoming video and the pixel-resolution of the video. Reducing any of these parameters increases the processing speed. Experimental results are presented with real scene video[8] sequences.

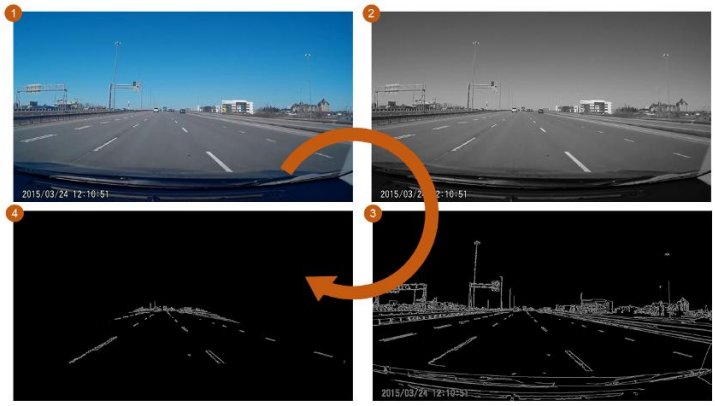
The proposed CNN-based algorithm for moving object detection consists of two phases:object recognition and tracking. Tensor flow-based detection of entities algorithm was used.The Tensor Flow-centered object or entity detection API is a platform open to all. TensorFlow-based model checks the image and returns the image location of the object(x,y,w,h)[5].

## II. METHODOLOGY

### A. Detecting Lane

There are a few characteristics of lanes are: Lanes lines are parallel; they are white or yellow. For the most part, they are continuous and have a standard width throughout the road. For a windshield-mounted camera, they will typically start evenly spaced about the image frame. From frame to frame, the identified curves will be continuous. In a front view, the apparent lane width reduces as we move up towards the horizon. This is not great for computing distances. It's for this reason that the satellites are sent into a low earth orbit and take bird's-eye view images for making topographical computation. As a first step, convert the dash-cam front view into a top view.

The lane lines appearing to intersect at the horizon are called the vanishing point. In the top view, the points near the vanishing point or the horizon are further apart than they are in the front view. Map a set of points from the source image in the front view to a set of images on the top view. This step can be automated using the vanishing point as a reference since all the slanting edges of the rhombus will intersect onto it. The edges of surfaces inside the images appear to form a line which intersects at the vanishing point. Apply a canny filter to a grey-scaled image to obtain a point cloud which represents the edges. Canny computes the intensity gradients at each pixel. It then uses thresholds to filter out some of the noises. These thresholds seem to work out best if we benchmark it to the gray-scale image median. Now there might be signage in the top half of the image which might add up as noise to our subsequent step hence we add a rhombus mask to filter a region of interest.



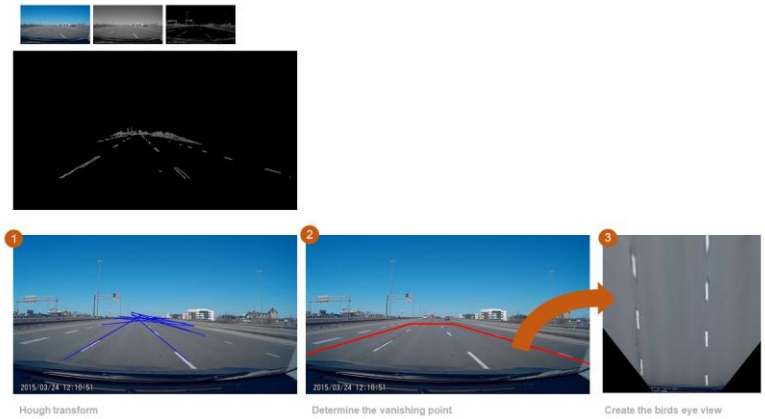
The edge points contain multiple lines embedded within noise. There are gaps in some of these

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020

apparent lines. the slopes of these apparent lines have noise. Hough Transform, which is used to identify lines and shapes in images, uses a voting algorithm along with constraints to decide candidate lines given a set of points. Bench-mark these controls to the dimension of the image, which seems to give a reasonable prediction for the lines for different frame sizes. Sometimes, the driven vehicle's (ego vehicle) bonnet and even parts of the dashboard might come inside the front view. Crop them out in the first stage itself before any processing.

### B. Perspective transform

Now obtain the lines from edge points. The point which minimizes the sum of perpendicular distance to all these lines is our Vanishing point. Use mathematical constructs to ease it out. Subsequently use this vanishing point to create a set of source points to map to the destination points. Set the top view size as 360 X 360px seems to be good enough for the rest of the process even if the destination image is 720 px high.

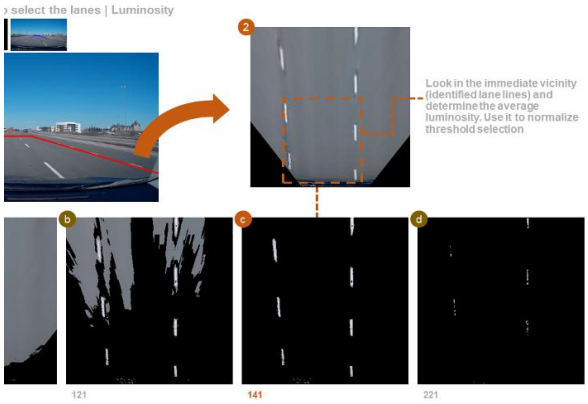



Shift the top edge of the source points towards the vanishing points. This will increase the road surface on which to base the lane line curves. However, as the vanishing point gets closer, the noise increases as greater image space (of the top view) is squeezed into a smaller pixel area in the front view. The two black triangular tails in the top view is the final source point below the front view image. This lets us use the complete lane area (right up to the ego vehicle) in the top view as it is unrolled out of the front image using perspective transformation. It consequently leaves a black triangular artifact.

### C. Creating a mask out of the perspective image

The perspective image (3 channel RGB) is not usable yet. Turn it to a mask (masked matrix) from which the lane information is extracted. The varying illumination conditions are: Dawn, noon, night, shaded, within a forest on a highway and so on. Some of these changes are gradual and some as a building / over-bridge shadow are quite sudden. It is possible to isolate the white lanes in the RGB color space (255,255,255) but yellow is a bit tricky.

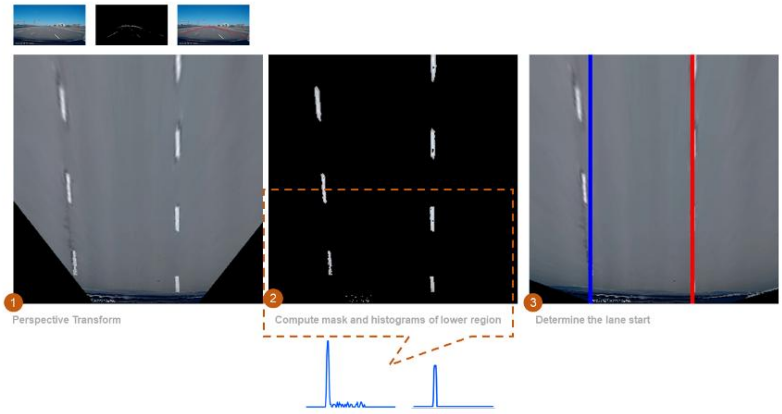
Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020




Start by putting a low/upper threshold for the white and yellow mask. If everything was static lighting and background, this would be good enough. However, as the background and lighting conditions change, it is necessary to update out thresholds every few seconds. The most sensitive factor is the lower lightness bound (L in HLS) picks an incorrect number and everything else goes for a toss. It is better to have a normalizing step before applying the thresholds. Use the road surface immediately in front of the vehicle for the same by computing the average lightness (L) over this area and using this average to moderate the thresholds to compute the masks.

### D. Detecting the lane start and width

Now the masks are obtained from the top view. Start extracting the lane information out of it. As a first step determine the start of the left and right lanes. Peaks on the histogram of the masked matrix (sum of columns). Sometimes the lane might curve left/ right so it's more prudent to use a lower portion near the car to compute the histogram [1].




On most roads, the lanes are of a standard width of 3.5 to 3.75 m. Using this ratio, all positions in the top view can be translated into real-world positions. Therefore, it becomes easy to report the velocity of a vehicle, time to a collision, and the radius of curvature of the road at a point.

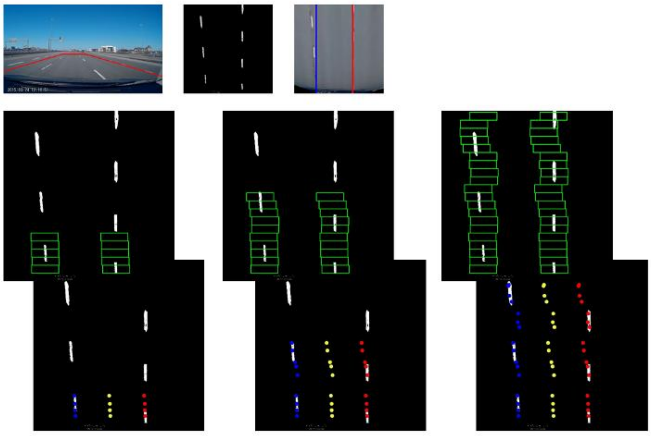
### E. Sweeping windows

To extract the pixels containing the lane lines in this step use a sliding window mechanism. The first

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020

step is to make a horizontal pass and obtain the pixels bound by each of the left and the right rectangles. Subsequently, determine the midpoint of these pixels to determine the horizontal position of the window in the next step. Keep repeating the steps to extract the pixels for the next row until the entire image is covered.

Set the window height and width parameter. The height is governed by the number of windows wanted to slide across the frame. In general, a higher number of windows allows them to fit in curves better and so does window width. Setting it too high will end up wasting computing resources. Normally, it is found that 25–35 windows per frame is optimal. Increasing the window width to high will start picking noise from the pavement or foliage along the side of the road.



Within a window itself sometimes there might be too little and on other occasions there may be too many selected pixels. To identify the x position for the next window, reject the information in the window as unusable and put in a guesstimate of the next window x position. There are three alternatives to select from: If the adjacent row has been filled, offset its position by the lane width and continue. Otherwise, if a sufficient number of rows have been prefilled, use the general curve to estimate the next position. If that doesn't work out, use the positions obtained in the previous frame to continue. If nothing works as a last recourse, continue vertically.

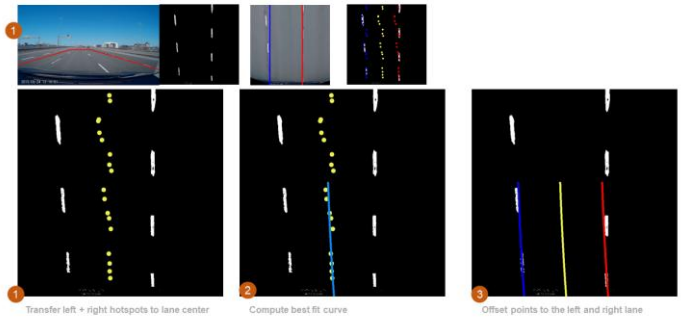
Sometimes there might be a gap in between the highlighted pixels of a row. It creates noise. It is better to reject such a frame and use information from the previous frame to correct it. Whenever an entire frame is rejected, it is prudent to recalibrate the thresholds used to create the mask. At the end of this stage, set the left and right pixels.

### F. Best fit lane center

The left and right lane coordinates are obtained, and now it is necessary to fit a curve to these[3]. Taking all the points from the bottom few windows and tasking an optimizer to produce an estimate for the next point or curve may end up giving a high weightage to a sunlit spot and produce a wayward curve. One way to moderate this is to use the centroid (of the points) in each window and use the centroids to estimate the curve. This reduces the noise in the detection.

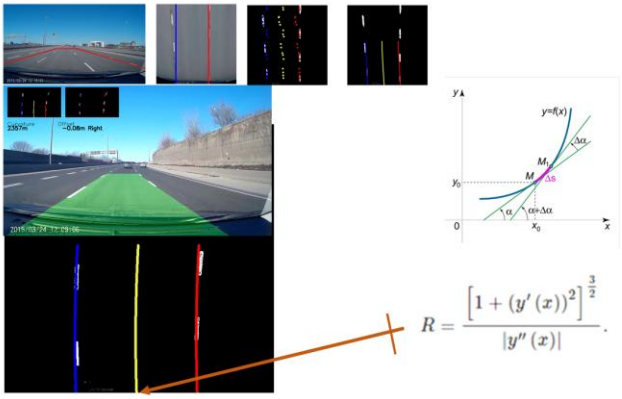
Once the curve is detected, compare it with the curve from the previous frame to check if they are close to each other. In case of a rejection, set up a counter once it clocks out a threshold to accept the solution even if it has exceeded the maximum acceptable deviation. This catches systemic errors and recovers from them. Instead of taking the curve estimates directly, use a moving average to remove some noise.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020




**Radius of curvature**

It is pertinent to note that both the left and right lane lines can estimate two separate curves[3]. However, using the center-line estimate is better on two counts. First, it pools information from two lanes; this helps produce an estimate even if one of the lane-line is missing or erroneous. Second, the lane-lines are parallel treating them as two separate entities lose this information.



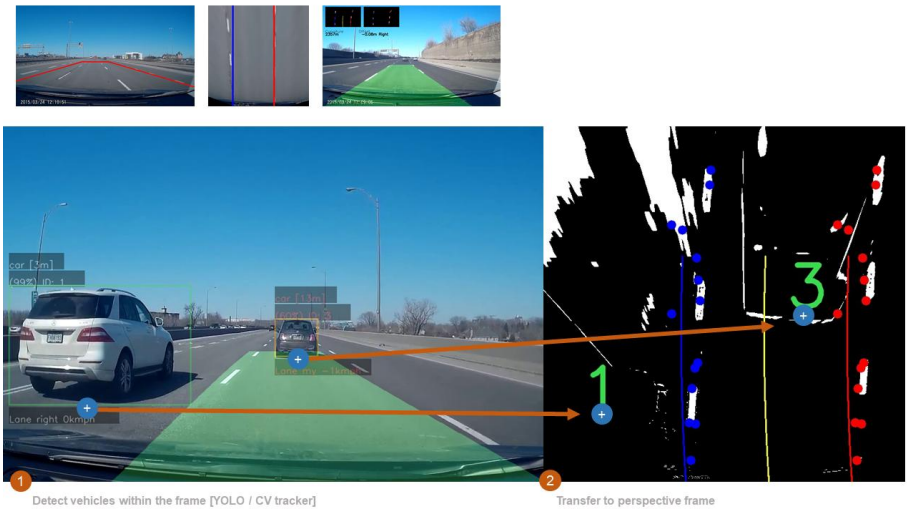

**G. Detecting vehicles**

YOLO is pretty efficient in balancing both accuracy and computation cost. Object trackers are accurate and fast and can do a real-time stream with limited computational resources. We can task YOLO to produce the object map once a while and have an object tracker follow it around for most of the time. Other than switching from one coordinate system to the other this is pretty straightforward to implement.

In this, we have to draw bounding boxes around the vehicles detected in a video stream. This can be done in two approaches i.e. CNN (Convolution Neural Network)/Deep Learning based approach and the HOG (Histogram of oriented Gradients) approach. Here we use CNN approach implemented using YOLO architecture.

YOLO reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. A single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes[6].YOLO trains on full images and optimizes detection performance. For detecting just the cars, here we are using tiny-YOLO v1. YOLO is pretty efficient in balancing both accuracy and computation cost.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020




## III. LITERATURE SURVEY

In this section, Advanced Lane and vehicle detection algorithms are discussed. The paper "Accurate Land detection for self-driving cars: An approach based on colour filter adjustment and k-means clustering filter" was proposed by D.Liu, Y.Wang, T.Chen and E. Matson in the year 2020. Robust to process noise appearing in the input image, improves the accuracy in lane detection are merits and It requires a large computational resource are demerits. The paper "Detection of objects for Autonomous Cars using Lane Detection method" was proposed by Dr.P.P.Priya, K.Pavan Kumar, MD.Akram Husain, S.CH.S.S.Teja, K.Sampath in the year 2020. It uses Convolution neural network which detect every small object are merits and The CNN used in this project can't store the data for a long time. So, if same picture is taken in side angle, it shows as new objects [5] are demerits. "Real-Time Lane and Object Detection for Driver Alertness Systems" is the paper was proposed by D.G.Ganage, N.S.Nikam, S.A.Wagh in the year 2019.This system is recognizing the object like pedestrian and car and also gives the notification like honking are advantages and the specificity is less compared to the sensitivity and accuracy. It is one of the performance parameters are disadvantages [4]. "Improving the Lane Reference Detection for Autonomous Road vehicle control" is the paper was proposed by Felipe Jimenez, Miguel Clavijo, Jose Engenio Naranjo, Oscar Gomez in the year 2016. It does not require the sensor to have vision of the reference element behind the vehicle and It does not imply high computational complexity are advantages and disadvantages of this paper. "You Only Look Once: Unified, Real-Time Object Detection is the paper was proposed by Joseph Redmon,SantoshDivvala,Ross Girshick,Ali Farhadi in the year 2016. Pros and Cons of this paper are Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly.

These complex pipelines are slow and hard to optimize because each individual component must be trained separately [6]. "Lane Detection and Following approach in self driving miniature vehicles" is the paper was proposed by Ibtissam karouach, Simeon Ivanov in the year 2006. It increases the performance by extracting the road from the image. It decreased the processing time are pros and It does not handle different light and weather conditions are cons. This tells about the advanced lane, curve and object detection.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020

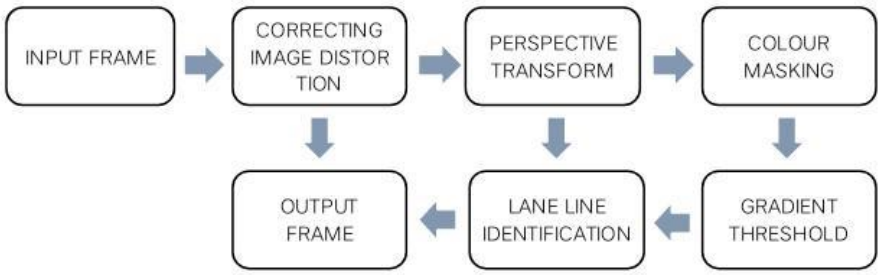
## IV. EXISTING SYSTEM

1) Parameters to which the processing speed is sensitive are: the time after which it triggers YOLO, the frame rate of the incoming video[8] and the pixel-resolution of the video. Reducing any of these parameters increases the processing speed.
2) Hough Transform works at least for clear straight roads. However, it is fairly obvious that this method would break instantly on curved lanes or sharp turns.
3) Another shortcoming is that lanes with dotted markings or with no clear markings at all are also ignored by the lane detector since there are no continuous straight lines that satisfy the Hough transform threshold.
4) Some systems still need tweaking before a mobile device can run the algorithm in real-time. Some code can be refactored to be shared between object detection and lane detection. One of the directions to pursue is to use the top view (scaled to 446 X 446) directly for object detection and tracking. This helps us skip the part to rescale between different coordinate systems multiple times.

## V. NOVELTY

1) Convolutional Neural Network (**CNN**) is used for processing and detecting pixel data. CNNs are great for extracting semantics from raw pixels but perform poorly on capturing the spatial relationships (e.g. rotational and translational relationships) of pixels in a frame. These spatial relationships, however, are important for the task of lane detection, where there are strong shape priors but weak appearance coherences.
2) For the implementation Image processing and deep learning using python is used. Computer Vision and Machine Learning is deployed for faster processing.
3) Proposed method involves Curve lane detection which is not satisfied by Hough transform.
4) This technology not only works in highways, but also works in streets and having small roads too.
5) It is also found that working at 360px at 10fps with Yolo triggered every 2 seconds gives better than real-time frame rates on a laptop.

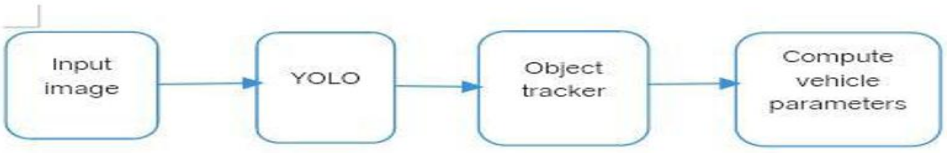
## VI. Block Diagram




***Lane Detection***

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020

***Object Detection***




***Object Detection***

## VII. CONCLUSION

In this paper, a reliable and sophisticated lane boundaries detection and collision prevention solution based on computer-vision algorithms is developed and presented. The proposed technique needs only raw RGB images from a single CCD camera mounted behind the front windshield of the vehicle. Firstly we need to take the input of the road image. By image processing we get the undistorted image. This should be done by image calibration. Camera calibration for all the images followed by perspective transform, and colour transform. Use color transforms, gradients, etc., to create a thresholded binary image. Apply a perspective transform to rectify binary image ("birds-eye view"). Detect lane pixels and fit to find the lane boundary. Determine the curvature of the lane and vehicle position with respect to center. Warp the detected lane boundaries back onto the original image. The final output shows the concept of Perspective transform, Colour masking and gradient threshold. The performance of this system is tested and evaluated using many stationary images and several images and several real-time videos.

## VIII. FUTURE SCOPE

Lane and vehicle detection is an inevitable module in the advanced driver assistance systems. Vision based approach is a simple method for detecting lanes. Even though a lot of progress has been attained in the lane detection , there is still scope for enhancement due to the wide range of variability in the lane environments and object distance. We could also tweak in a custom head only doing vehicle + pedestrian + traffic light detection instead of 80 classes.

The future of the transportation economy will be built upon micro transactions which has an inherent need to differentiate between the careful and the negligent. This is where the active assist algorithms can contribute. Starting from nudging behavior while driving to determining the premium of insurance based on ride score history. The end product of the auto industry is changing. A change which needs a new skill-set and relationships. These features can also be integrated in smartphones for further development in the automobile industry.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal ( ISSN : 2581 - 611X )
Volume 3- Issue 2, Paper 7, August 2020

## REFERENCES

[1] Wael Farag, "A Comprehensive Real-Time Road-Lanes Tracking Technique for Autonomous Driving", International Journal of Computing and Digital Systems, (2020).

[2] Ammu M Kumar and Philomina Simon, "Review of Lane Detection and Tracking Algorithms in Advanced Driver Assistance System", International Journal of Computer Science and Information Technology, (2015).

[3] H. Tan, Y.Zhou, Y.Zhu, D.Yao and K. Li, "A novel curve lane detection based on improved river flow and RANSA", in 17th Int. IEEE Conf. Intelligent Transportation Systems, (2014).

[4] D. G. Ganage, N. S. Nikam, S. A. Wagh, "Real-time Lane and Object Detection for Driver Alertness Systems", International Journal of Scientific and Technology Research Volume 8, Issue 8, [2019].

[5] Dr. P.P. Priya, K. Pavan Kumar, MD. Akram Hussain, S .CH. S.S. Teja, K. Sampath, "Detection Of Objects For Autonomous Cars Using Lane Detection Method", International Journal of Scientific & Technology Research Volume 9, Issue 03, [2020].

[6] J.Redmon, S.Divvala, R.Girshick, & A.Farhadi, "You only look once: Unified, real-time object detection", IEEE conference on computer vision and pattern recognition; [2016].

[7] Chin-Pan Huang, Chaur-Heh Hsieh, Jin-Long Li, "A Robust Lane Detection Method Using Adaptive Road Mask", The Proceedings of the International Conference on Digital Information Processing, Data Mining, and Wireless Communications, Dubai, UAE, (2015).

[8] Mehmet Miman, Osman Onur Akırmak, Hakki Can Korkmaz, "Lane Departure System Design using IR Camera for Night-time Road Conditions", (2015) .