An Accurate, Efficient and Effective Method for Vehicle Spotting and Lane Identification with Distance Estimation Using YOLOv3 and Keras

Aseem Patil1, Ajay Talele2, Vaishnavi Mate3

1,2,3 Department of Electronics Engineering, Vishwakarma Institute of Technology, Pune, India

Abstract: This paper provides an algorithm for lane detection and an approach to autonomous vehicles in real time. Efficient and sturdy during day and night, the fast adaptive lane detection algorithm. In order to improve the precision of detection, updated global lane models are suggested. An adaptive adaptation system to respond to lane width changes and road tilt in different situations is also introduced. The developed fuzzy detection procedure, Contour Size Similarity (CSS), makes a logical comparison of vehicles projection sizes with the estimated sizes. We will also use local lane identification map restraints to improve lane identification accuracy. In addition, it is difficult to estimate the distance to the car from wide angle images / videos taken in real time. In our method, we use a FPV camera mounted in the front of the car. The lane marker will try to estimate the distances between vehicles passing by and the cross divider. In this paper we propose a technique to identify the lanes and vehicle using the famous CNN based architecture, YOLO (You Only Look Once).

Keywords: YOLOv3, Radial Distortion, Model Comp, identification, Image Classification, Image Pyramid, Hough Transformation

I. INTRODUCTION

An adaptive lane-learning algorithm which can automatically discover the features of the lane in various scenarios is recommended to improve the effectiveness of the road detection in complex scenarios. First, the YOLO v3 (You only Look Once, v3)-based, two-stage learning system is built. To make it more suitable for lane detection, the structure parameters of YOLO v3 algorithm have been updated. In order to enhance training performance, an automated image generation method for the first-stage network training is suggested, in a simple scenario that provides etiquette data. In the next model testing, the unexplained paths are also shielded to prevent interference. In the course of the second phase model training the images processed by the above method will be added as label data.[1] In the next model testing, the unexplained paths are often screened to prevent interference. In the course of the second phase model learning the pictures processed by the above method will indeed be presented as mark information. The lane detection algorithm has to be highly accurate and flexible to meet the needs of intelligent driving. The quick and accurate identification of a road under complex road conditions is always a big challenge because of the variety of driving environments. Lane detection has therefore attracted the greatest attention of lane detection researchers. A scientist constructed the lane with Canny operators by detecting the edge of the lane and finding the best line by turning Hough. The path detection is separated into two sections. The very first element to detect a lane which was then monitored by the Kalman filter was the active threshold Sobel operator and the Hough transform method. However, the Hough Transformation-based lane detection method is lacking in that the false rate is relatively high when there are many lines in the image.
In order to improve lane detection accuracy, the detected lane can be extended to the bottom of the region of interest (ROI) by means of a highway curve model based on a generated top-view photograph. He suggested an algorithm for path detection centered on the disappearance point estimate. First, the position of a lane vanishing point in the existing framework was determined, but then the disappearance point area was taken as the ROI underneath the vertical coordinate. Once the edge point of the ROI was found, the Hough transformed pixels at the edge points, and then the longest line in the pre-set angle range was taken as a path. Such approaches are effective on the roads with smaller changes in colour, but detection precision is not good if the luminosity varies in the sun, shadow, and rainy weather conditions. [2] The CNNs (Convolutional Neural Networks) have shown significant advantages with the rapid development of profound learning and other related technologies in the extraction of images, and have therefore been widely used in the identification and classification of artifacts. Some researchers have researched lane detection using label images and have developed CNNs in different depths. The YOLO algorithm only categorizes and locates the object in just one step and directly takes its place and category to the output layer. At YOLO, object detection is considered a problem of regression that improves operational speed dramatically by fulfilling real-time requirements. The SSD combines the YOLO end-to-end and the Faster RCNN anchor box system and increases detection capability for small objects by increasing input image size. The SSD detector is also known as a single shot multibox.

II. IMPLEMENTATION OF THE SYSTEM

2.1 Camera Calibration and Radial Distortion

The camera output is a recording, basically a sequence of pictures. Due to the nature of the photo lenses, images taken using camera models for pinholes tend to result in radial distortion which, depending on the distance of the object from the optical axis, leads to incompatible magnification. We need to correct the radial distortion in order to correctly detect the lane lines in the picture.

Scientists in computer vision have developed a way to correct this radial distortion. The cameras to be calibrated are used for photography of checkerboard patterns, which are of the same size in all white and black boxes. [7] When the camera is distorted, the images captured incorrectly display the checkerboard measurements. On taking a picture, 3D objects become a 2D image in the real world. It creates distortion, i.e. you can see that the scale of an object is increased or that the distance is changed. The distortion coefficients of each cameras are their own. In this case, we calculate the coefficients and apply them to our picture frame. The checkerboard corners are recognized to rectify the consequences of distortion and the deviations from the expected measurements of the checkerboard are used for calculation of distortion coefficients.

![Original Image](image1.png) ![Undistorted Image](image2.png)

*Fig 1. Converting the image taken from the camera and un-distorting the image (example)*
If corners are identified, image points along with object points are collected as img_points; on the assumption that the chessboard is fixed at z=0 (the objects points are thus equal for each image). The undistorted image takes an image and deletes variables before the undistorted image is returned. The image on the right shows the corners drawn on the top of the distorted image and the middle image displays the resulting undistorted image after camera calibration. [4] In the diagram below, the image on the left side is the distorted image.

Fig 2. Collecting the image from the live feed and distorting the image frame from the output video

2.2 YOLO architecture configuration
YOLO divides the image into grids S by S and for each grid cell calculates the probabilities of the B bounding box and C grade. Five predictions exist for each bounding box: w, h, x, y and object trust. Width and height of the box are the values of w and h relative to the entire image. The (x y) value is the center of the box co-ordinate relative to the grid cell boundaries. The trust of the object is the reliability of the actual object in the box, defined as:

$$Confidence = \Pr(\text{object}) \times IOU_{pred}^{truth}$$

Fig 3. Using one frame from a random video and predicting the object using YOLOv3
In equation (1), the probability of an object falling in the current grid cell is expressed by \( \text{Pr}(\text{object}) \). \( \text{IOU}_{\text{truth}, \text{pred}} \) is the intersection of the bounding-box and the real box with the union (IOU). Then the majority of boundary boxes with low object trust are excluded under the threshold. Eventually, a procedure for eliminating redundant bounding boxes is used. Normally the general detection model of a network model like YOLOv3 is built. Therefore, the number and differences between classes detected in such a network may be high, such as individuals, bikes, automobiles, houses, etc. There are three permanent and replicated 5 by 5 by 1024 high levels for the YOLOv3 system. Normally, many groups with broad difference, including people and apples, will experience a repeated convolution in high layers. The number of vehicle type identified was only six, with very small differences in specifications between vehicle types. Finally, the YOLOv3 Vehicle network was designed with the application of multi-layer functional fusion and the removal of repeated convolutional stratification in high layers. In addition, we built another Model Comp network for comparison to check the efficacy of eliminating the repeated convolutional layers in high layers. In contrast to YOLOv3, just 5x 5x 1024 convolution layer was taken from Model Comp.

### 2.3 Conversion of YOLO architecture

YOLOv3 network is supplied with input images in order to predict 3D tensors (the last function map) that are 3 scales, as shown in the diagram in the center above. The three dimensions are intended for the identification of objects of different sizes. In this example, we take the scale 13x13. [6] The picture is divided in 13x13 grid cells for this distance, each grid cell being a 1x1x255 voxel within a 3D tensor. For this scale There is 255 (3x(4 + 1 + 80)) here. The figures in the 3D tensor are shown on the right side of the diagram, including binding box coordinates, object score and trust. In second place, this grid cell predicts a bounded object's box, if the center of the bird's bottom truth box falls within a certain grid cell (i.e., the red cell on a bird's picture). For these grid cells the corresponding scoring value is "1" and for others "0." For every grid cell, 3 previous boxes of different sizes are numbered. It is a rule that only the box is chosen which most overlaps the bordering truth box.

To convert the architecture, the following steps are considered:
1. Create an image pyramid, insert every level of a pyramid into a specific network for its size.
2. The prediction is done at the end of the function map because each level has its own network or operation.
3. This system is not capable of handling many scales. The forecast is rendered at different depths on feature charts. SSD follows this approach.
4. The prediction is made by using the features we have learned so far and further features cannot be used in deeper layers. Like but other features, up sampling of the function map and fusion with the current function map is used.
5. This technique enables the network to capture both low- and high-level information on the object.
6. The converted architecture is shown in figure 5. This is the architecture that will be used for the vehicle and lane detection in the live feed.

![Converted Architecture for YOLO_v3 to be used in the model](image)

The main reason was that the feature fusion approach was implemented by both Model Comp and YOLOv3, so that more local feature knowledge would be collected, thereby accelerating training convergence. While the YOLOv3 Vehicle model's average loss fluctuated during testing, the minimum of three models was first and the lowest. Model Comp also has an average loss, but it was less than YOLOv3. The YOLOv3 Vehicle model network could therefore accelerate the vehicle data set convergence and better adapt the vehicle detection mission.

### III. RESULTS

The following table shows the testing results on different road types that were used to evaluate the correct and incorrect percentage and analyze the accuracy of the system. Using the YOLOv3 algorithm (You Only Look Once), we could help in identifying and detecting vehicles and lanes respectively with an accuracy of an average of 94.06%.

| Road Type        | Traffic       | Avg. Detection Rate Per Minute |
|------------------|---------------|-------------------------------|
|                  |               | Correct | Incorrect | Missed  |
| Isolated Highway | Light + Moderate | 98.71% | 1.29%     | 0.00%   |
| Metro Highway    | Light + Moderate | 96.53% | 3.47%     | 0.00%   |
| City             | Variable      | 86.94% | 11.96%    | 0.94%   |
From figure 6 (Fig. 6) we can relate that the number of losses were minimum for the test cases using YOLOv3 as compared to the other CNN architecture readings taken from surveys.

Fig 6. Train loss and no. of iterations in the model built

Fig 7. The output of the system
REFERENCES

I. Dai J., Li Y., He K., Sun J. R-FCN: IEEE International Conference of Advances on neural INI Systems 2016; Barcelona, Spain: object detection by regionally based completely convolutionary networks; December 5-8, 2016; pp. 379-387.

II. Konoplich G.V., Putin E.O., Filchenkov A.A. Implementation of in-depth training in UAV image detection of vehicles; Proceedings of the International Soft computing and measurement (SCM) Conference 2016 XIX IEEE; St. Petersburg, Russia. May 25-27, 2016; pp. 4-6.

III. Cai Z., Fan Q., Feris R.S., Vasconcelos N. A deep convolutionary multiple-scale neural network for fast object detection; European Computer Vision Conference 2016; Amsterdam, The Netherlands. Proceedings from 8th to 16th October 2016; pp.

IV. Tang T., Zhou S., Deng Z., Zou H., Lei L. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. Sensors. 2017; 17:336. doi: 10.3390/s17020336.

V. Sang J., Guo P., Xiang Z., Luo H., Chen X. Vehicle detection based on faster-RCNN. J. Chongqing Univ. (Nat. Sci. Ed.) 2017; 40:32–36.

VI. Rane M., Patil A., Barse B. (2020) Real Object Detection Using TensorFlow. In: Kumar A., Mozar S. (eds) ICCCE 2019. Lecture Notes in Electrical Engineering, vol 570. Springer, Singapore

VII. A. Talele, A. Patil, and B. Barse, “Detection of Real Time Objects Using TensorFlow and OpenCV”, ajet, Apr. 2019.