Traffic & Transportation

Enhancing Autonomous Vehicle Navigation by Detecting Lane and Objects based on LaneNet and CustomYOLOv5

Jayamani SIDDAIYAN[1], Kumar PONNUSAMY[2]

Original Scientific Paper
Submitted: 1 Apr 2024
Accepted: 9 Dec 2024

[1] jayamanis111@gmail.com, K.S. Rangasamy College of Technology
[2] kumar@ksrct.ac.in, K.S. Rangasamy College of Technology

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
Lane and object detection are the major concerns of an autonomous vehicle's ability to move continuously without creating any traffic congestion or collisions. Highly populated rural and urban roads are still facing many challenges to enabling the intelligent transport system with an end-to-end customer connectivity. The proposed work is to identify the drivable space by combining lane line detection by using the LaneNet with sliding window and front road object detection and using the customised YOLOv5. The appropriate pre-processing methods are carried out to reduce the computational complexity and speed up the process. Followed by pre-processing, the reference line is assumed at the far-end distance from the host vehicle to identify the driving space. The lane line borders and objects bounding box coordinate intersecting points on the reference line are picked up to calculate the drivable space. Finally, the proposed system is validated on various public and own datasets. Lane line detection and object detection accuracy of 97% and 98%, respectively are achieved by the LaneNet with sliding windows and custom YOLOv5.

KEYWORDS
drivable space detection; intelligent vehicle; LaneNet; YOLO.

1. INTRODUCTION

In today's fast-paced society, everyone pursues their own lifestyle and sticks to a schedule to fulfil expectations and comfort. A minor accident or violation of the traffic laws on any type of road causes congestion and slows down the day's progress. There are 1.3 million individual fatalities every year in roadway accidents, according to a report by the World Health Organisation. Disciplined roads have made significant progress in identifying lane lines for autonomous vehicles and driver support systems, but not in urban and rural environments. Operating an autonomous intelligent vehicle in a residential area is still challenging due to undesired roadside occupation, poor maintenance and lane avoidance behaviour. The computer vision system is promising to detect any type of object and track in front of the host vehicle. According to the overview of well-structured lane recognition and results reported in [1, 2], a system fails if there is less lane care and poor lane conduct. However, the features of an image boosted method were carried out in [3] and [4] to obtain a better output. On the other side of the road, where there is less visibility of the borders, it is difficult to distinguish light variance due to the shadow cast by trees and buildings [5]. Improvements were obtained in various algorithms based on the feature, model and machine learning algorithms to detect the lane markings [6–9]. Recent deep learning algorithms [10] are directed towards confidence in the implementation of intelligent systems. Numerous sensors are crucial to an autonomous vehicle's ability to gather data about its surroundings. The most frequently used sensors, such as LiDAR (Light Detection and Ranging), GNSS (Global Navigation Satellite System), GPS (Global Positioning System) and many others [11] are efficient and perform the required work. Computer vision and deep learning methods have become more popular in every area of Intelligent Transportation Systems (ITS) in various ways, including reserve vehicle finding, traffic

control, sign recognition and plate number recognition. The detection of computational behavior and results in several fields, including picture organization [12], segmentation [13], tracking and object identification, object totaling, detection of moving objects, overtaking vehicles, object classification [14], and lane change detection [15] motivates further research.

In the earlier path-finding methods the detection of the driving path is made regardless of the front object. The system has to detect the lane line in addition to the host vehicle and assume the entire width of a lane as a driving path. In an object detection method, most of the system is localised to all the objects that have been trained during the training phase. The customised YOLOv5 object detection method is developed to detect possible objects engaging with the road environment. However, the proposed method, the front object's road occupancy and the width of the road are considered to identify the best driving space for the host vehicle.

This paper is structured as follows: The second section presents a review of the related literature. The third section presents supporting techniques to pre-process the input. In the fourth part, the proposed methodology is presented. Part five presents the experiment setup and evaluation metrics used. In section six, the test results are discussed and compared with other relevant techniques. Finally, in the section seven, the conclusions and further directions for the development of this solution are given.

2. RELATED WORK

Drivable space identification is one of the critical factors for achieving vehicle movement in urban and rural environment. The majority of literature uses lane line and object detection techniques separately to investigate the problems. Detecting objects in front of the host vehicle is a valuable feature for supporting accident-free mobilisation. In the drivable space identification case, objects and lane line detection is the most important element, and a single algorithm cannot produce optimal outcomes in every setting. Typically, an SVM (Support Vector Machine) based technique is used in conjunction with supervised or semi-supervised online learning to identify roads [16]. For instance, [17] describes an effective method of using self-supervised online learning with SVM for road detection. The correlation feature set and the raw feature set are evaluated and make use of boosting the SVM and random forest classifiers. Neural networks are also recently used to identify roads [18]. The aforementioned examples are concentrated on the organised and disciplined highways, which is not typical for all connected roads. On gravel roads, lane detection is challenging because of the road's texture, lighting variations and fuzzy borders. The overall performance of detection also degrades due to the weather and ruts. The state of features of dirty roads as described by [19, 20] are correct, but unsubstantiated information are obtained accurately [21]. In general, two types of deep learning algorithms, such as regression-based models and region-based models for object detection are employed. The object had previously been located in two steps, which took more processing time. On the other hand, simple regression uses a single shot of the full image to provide bounding boxes and class probabilities [22]. In comparison to the double-stage object detector, the single shot model performs faster. It has been suggested that an improved version of You Only Look Only (YOLO) be developed to detect electrical components [23], licence plates [24] and many other elements. YOLO-UA for traffic flow monitoring [25] and YOLO-CA for accident detection are two examples. The YOLO-P [26] model that finds the crop being harvested from the palm tree plantation is employed. In recent advancement, methods are light-weighted and integrate the various techniques [38] proposed by the CNN-PP model for achieving better performance in qualitative and quantitative terms. Study [39] approaches the camera guided by the neural network to find the wrong way of driving with a precision of 0.99. The author [40] aimed to prevent the accidents on highway by considering the vehicle information and road configuration through bird eye view and reached a 50% lower prediction error than the base model. The Multi-Scale Feature Aggregation Module is introduced by [41] for extracting the strong feature and outperforming the USFDNet with SFENet a 2.1% enhanced accuracy. The impact of lane detection on unclear and occluded condition of road was studied by [42], exploring the deep learning through semantic segmentation simulated with a standard data set.

The coordinate attention module of the MCS-YOLO [43] is integrated into the backbone in order to collect the cross-channel and spatial coordinate information of the feature map. To increase the sensitivity of dense small item recognition, a multiscale small object detection structure was created in order to allow the CNN to concentrate on contextual spatial information by implementing the Swin Transformer structure. In the unique instance of small items [44], the YOLO v4 neck module has been updated and its learning capability to detect small objects is improved by paying attention to the observed Soft CIOU and CIOU loss function. A 90% accuracy was attained when counting and tracking objects in a pre-recorded video using the TensorFlow API

with the YOLO model [45]. In the MEB-YOLO model [46], by merging the MixUp and Mosaic data augmentation, the vehicles might be identified in intricate traffic road scenarios. The process of incorporating several stages begins with strengthening the detection dataset and then employing the CSPDarknet to remove information about irreverent traits. Finally, the model's neck network is constructed using the BiFPN structure, which allows for more complex feature fusion to be achieved without raising the processing costs. By triggering the effective features with richness of features extraction on road scene identification, the CMS-YOLO [47] proposes the cross stage partial DWNeck algorithm and achieves an accuracy of 90.3%. The above one-shot trained model can detect the various objects present in an image or video frame. The take away from the literature is that over the past year the technology evolved in three different categories of traditional methods, the machine learning method and the deep learning method. Each method has its own set of trade-offs, and the choice of method often depends on the precise requirements of the application. In each category, the accuracy of the model, feature consideration and incorporating different condition has improved. In recent years, the availability of data is sufficient but difficult to ensure the computation resource. The model is struggling in complex environment because of the fixing of features, various lightning condition, over fitting the training model and implementing the system in an unaware condition.

Other modern deep learning models like the Faster R-CNN, the SSD and RetinaNet are generally more accurate and faster than the traditional lane and object detection methods such as the Haar cascades and HOG+SVM, which frequently rely on hand-crafted features and image processing techniques. These methods can struggle with complex road scenarios and varying lighting conditions. With its deep learning methodology, LaneNet can adjust to these complexities and provide more reliable and precise lane detection. LaneNet has been specifically designed for this role and has demonstrated performance under a variety of scenarios. YOLOv5, with its end-to-end deep learning architecture, significantly outperforms these methods in both speed and accuracy. YOLOv5 achieves a balance of high accuracy and low latency, making it particularly well-suited for the fast-paced requirements of autonomous driving.

It can be concluded from the literature survey that various lane line and object detection methods outperform individually. These detection methods do not fully fill the needs of driver assistance in traffic-congested areas on rural and urban road environments. By adding the essence of an object that occupies the space on the road in the forward-facing host vehicle with lane line detection, it will help the driver to avoid traffic congestion at crucial stages. By considering performance and simpler models, the LanNet and YOLOv5 are taken into account to find the drivable space. LaneNet helps to detect the boundaries and YOLO provides the bounding box coordinates by applying simple mathematical logic to find the sufficient gap for the host vehicle movement.

3. PRE-PROCESSING METHODS

3.1 Camera calibration

A vision system is one of the major components in an autonomous vehicle and driver assistance system that gathers data in front of the host vehicle equipped with the sensors. For some reason, some images collected in a real-time scenario are distorted as shown in *Figure 1*. Tangential distortion occurs when the images are not maintained in parallel to the image plane with the camera lens; in other cases, warped edges belong to the category of radial distortion. The actual driving environment reflects these distortions. The following is the mathematical model definition of the radial and tangential distortion correction.

Distortion parameters:

$$D = (k_1, k_2, p_1, p_2, k_3) \tag{1}$$

Correction on radial manner:

$$x_c = x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2}$$

$$y_c = y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3}$$

Correction on tangential manner:

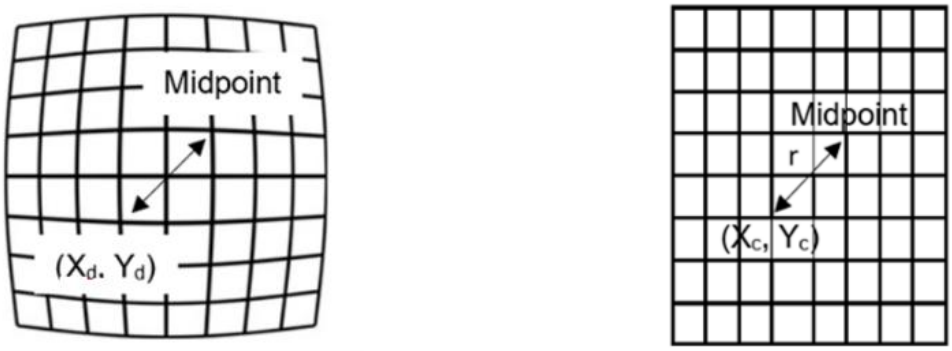
$$x_c = x_d + [(2p_1 x_d y_d + p_2(r^2 + 2x_d^2)] \tag{4}$$

$$y_c = y_d + [p_1(r^2 + 2x_d^2) + 2p_2 x_d y_d] \tag{5}$$

where $r^2 = x^2 + y^2(x_c, y_c)$ is the position location in the image pixel coordinate at point d.

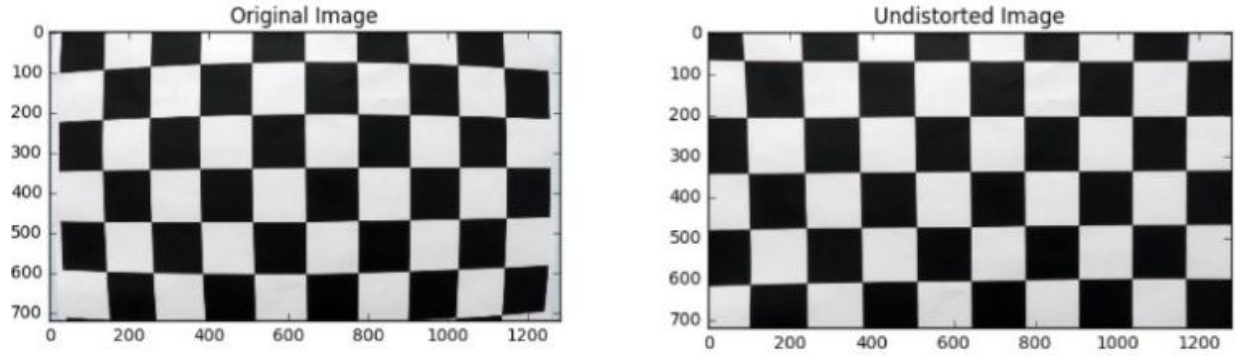
Suffice d and c represent the distorted and corrected coordinate, respectively; r represents the distance between the centre of the image and the correct coordinate point; k,p represent the radial and tangential distortion parameters, respectively.

Before installing the camera, these types of known distortions are addressed. One of the greatest ways to examine camera distortion and calibration is with a checkerboard.



(a) *(b)*

Figure 1 – Camera distortion correlation parameters.
a) Distorted; b) Corrected



(a) *(b)*

Figure 2 – Chess board image for camera calibration
a) Before calibration; b) After calibration

The regimented regular forms made of a checkerboard pattern of black and white aid in adjusting the camera to detect scenes with distorted images. The images, as seen in *Figures 2,* show how the distorted images are transformed into undistorted images.

3.2 Region of interest

An image is a complete description of a scene that contains relevant and unwanted information for a specific task. When considering the entire image, it results in a delay in performance because of processing the unwanted information. The work relates to the road environment and the most of the information lies at the bottom of an image. Selecting such an informative portion of an image accelerates the process. The Region of Interest (ROI) aids in locating an image so as to choose an informative area. The ROI is the region enclosed by a circle, rectangle or another shape surrounding the interesting area. The ROI is calculated by using a rectangular pair and a trapezoid mask, which may be stated as follows:

1) Using a trapezoid mask to create a rectangle couple.
2) Merging an image as well as the mask.
3) Choosing only the region that interacts with the mask and the image.

As a result, an informative region of an image is segmented and considered for further processing.

3.3 Inverse perspective mapping

In a real-world scenario, the detected object appears to be dynamically varying, being larger when it is near the camera and smaller if it keeps its distance from the camera. The disappearing point is harmful to the activation of lane borders where two parallel lines meet at the far range of vision. Inverse Perspective Mapping (IPM) restores the link between the parallel line and the world coordinates seen from [27, 28]. The IPM is used to transform the forward camera's vision into a bird's-eye perspective. The road apparent point (X_w Y_w Z_w) that projects to construct the homograph relations of the four corners of the image plane (u, v) is necessary to produce a top-down view. *Figure 3* shown the projection principle and the equation as shown in (6).

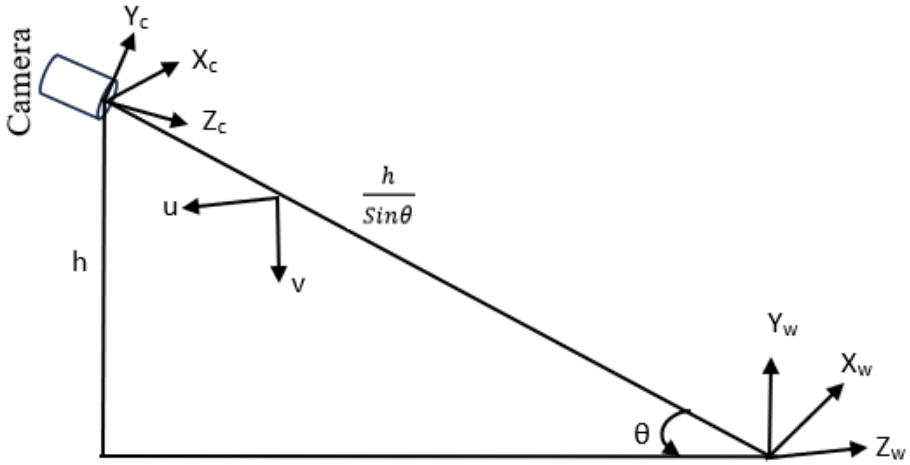


Figure 3 – Invers perspective projection view

$$(u, v, 1)^T = K\ T\ R\ (X_W, Y_W, Z_W, 1)^T \tag{6}$$

R – Rotation matrix

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

T – Translation Matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{-h}{\sin\theta} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Considering the above *Equations 7 and 8* with the camera matrix (K) and the road plane (Y_w=0), the equation becomes as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{-h}{\sin\theta} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

By using *Equation 9*, it is possible to map each pixel on the image plane to a top-down view, which helps to detect the lane line.

4. METHODOLOGY

To achieve the driving space for the host vehicle in any type of road and weather conditions, the algorithm has to look for the front object (vehicle) occupancy and lane width of the road. The proposed system executes

in two ways to achieve the driving space by detecting the lane and obstacle. A video stream or image input is used to begin the procedure. The input data is pre-processed in order to prepare it for further analysis. Pre-processing may involve operations like noise reduction, scaling, normalisation and other techniques to enhance the quality of the incoming data. The ROI is used further to locate the core area of the research. A fixed reference (imaginary) line has been established at a height of ¾ from the bottom of the image – the position where objects on the road and the road boundary should be detected and the driving space for the further vehicle movement calculated.

The driving free space is identified by collecting the intersect coordinate points of the lane border as well as the vehicles ahead bounding the box coordinate point. *Figure 4* shows the system's overall overview. The overall section is separated into two concurrent processing cases: case 1 involves detecting the lane line in a road setting, and case 2 involves detecting road objects that obstruct free forward motion.

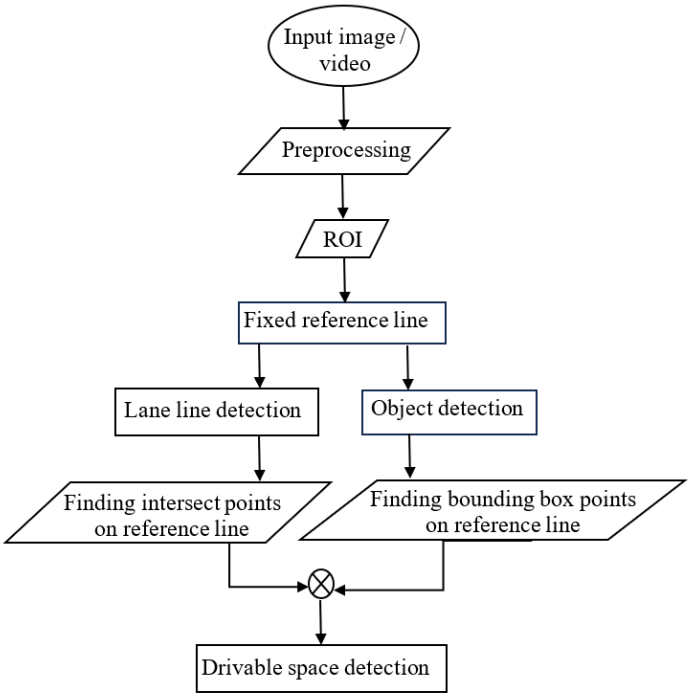


Figure 4 – Flow of drivable space detection

Case 1: Lane line detection: For the perfect operation of an intelligent system on the road, the details of the feature have to be supplied to take the decision. Here the features are the lane border, recognised by adopting the pre-processing techniques and LaneNet lane detection, and finally, the sliding window method is used to obtain the line.

Case 2: Object detection: Different types of objects ahead, such as vehicles, pedestrians and traffic signs, are considered as an obstacle and they are detected by using the customised pre-trained YOLOv5 model. By detecting the coordinates of the bounding box points, they are recorded and used further for driving space identification.

In both cases, the reference line intersect points on the lane line boundary and bounding box are taken to identify the lane space availability on the sides of the vehicle ahead. The detailed space occupancy and space availability with reference to the host vehicle is described in further session. Finally, positive driving space is decided based on which side space of the vehicle ahead gives more.

4.1 Lane line detection

The driverless vehicle fared best on highways, where lane markings are brighter and more pronounced. However, in practice, the connectivity of the roads between urban and rural areas is not very informative because they are poorly maintained. In this case, the autonomous vehicle operation is not successful in detecting the lane line. Finding the lane line with the use of a lane line predictor and lane localisation is one of the simplest ways to locate the road area. In order to successfully detect lanes, it is typically necessary to first employ pertinent methods to extract the pixel features of the lane line. Next, a proper pixel-fitting algorithm is

required to complete the process. The network termed LaneNet and depicted [29] in *Figure 5* combines the benefits of binary lane segmentation and identifies the pixels that are part of lanes. In order to capture the coordinates of pixels, the sliding window is scanned from the bottom up. The centre of the next sliding window is determined by the mean value of the pixels when the number of pixels exceeds a predetermined threshold. After threshold filtering, the hybrid incompatible operator and graph, along with the sliding window, fit the lane.

The pixels from the preceding frames are used to fit the curve, while the pixels from the recorded position information are used in a quadratic polynomial [30]. The LaneNet model is used in this case, with segmentation to detect the lane line in any situation with the sliding window search, when it is blocked by objects or when there are no visible lane line marks. The proposed method's outcome is shown in *Figure 6*.

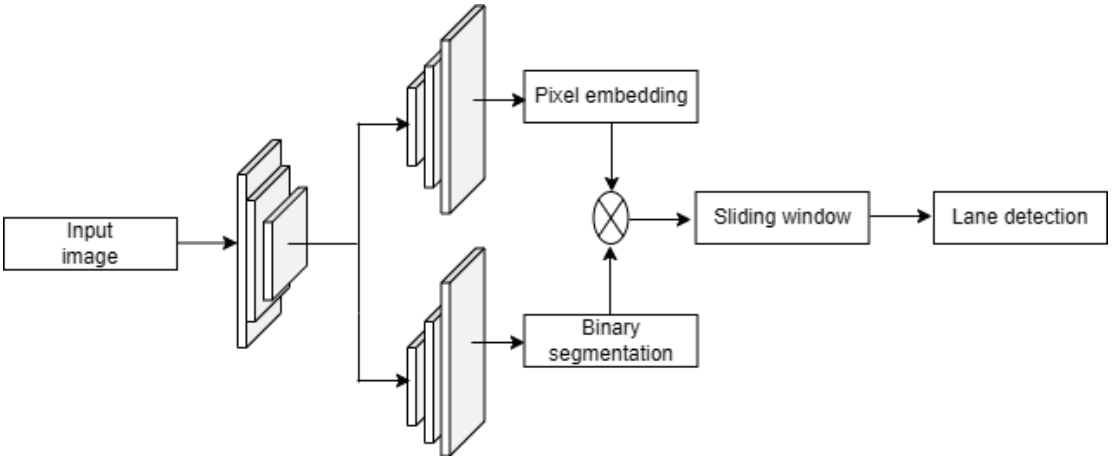


Figure 5 – LaneNet with sliding window

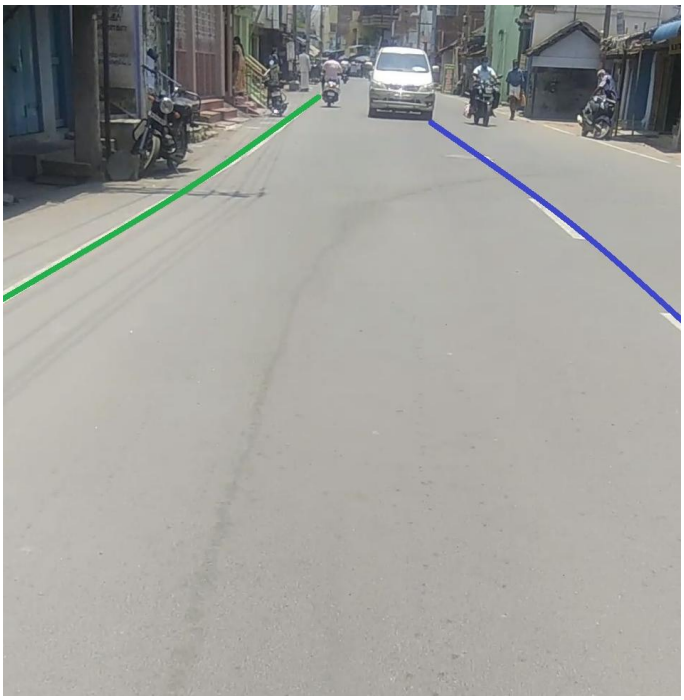


Figure 6 – Lane line detection

4.2 Object detection

The characteristics are manually retrieved by using a traditional method, which takes more time. The basic feature extraction models for road object recognition Haar [31] and HOG [32] are discussed. Recent advancements in vision sensors have enhanced data collection over time, on the same time rapid hardware development and parallel computing have been enabled the deep learning to outperform the traditional algorithms.

The object detection algorithm used in the deep learning approach contains two stages in addition to a single stage [33] to ensure that the real-time performance requirements for timeliness and accuracy are met. Double-stage object detection models with high complexity and calculation time are included in the Region-Based Fully Convolutional Networks (RFCN) [34] and Mask-regions with convolutional neural networks (Mask R-CNN) [35]. When compared to the wo-stage detection models, one-stage networks like YOLO and (Single Shot MultiBox Detector) SSD offer faster calculation time. The YOLO algorithm [36] is significantly quicker than the existing approaches followed by YOLOv3 in 2018, which drastically improved finding the speed and precision. The YOLO sequence of algorithms will have evolved to YOLOv5 by 2020. Additionally, the sequences of models can be identified based on the network's breadth and the size of the feature map. More specifically, all four previous models share the identical input, backbone, neck and prediction components of the network.

The one stage YOLO architecture has a convolutional [37] layer and an input image, which are the two main components of the regression-based whole YOLO structure. In a single step, the convolution layer processed an image and returned the spot and grouping of the detected objects. The input size for the three-layer YOLO architecture is 640 x 640 x 3. The Conv2D, Batch Normal and Leaky subcomponents make up the Convolution-BatchNorm-Leaky ReLU (CBL). Thirty-two convolution kernels are used in the RELU to slice the input images, which produces a final product with the dimensions 320 x 320 x 32. The feature extraction, gradient information elimination and optimisation processes are handled by the Cross-Stage-Partial (CSP) bottleneck module. Different feature scales are acquired via the Spatial Pyramid Pooling (SPP). Algorithmic tasks include classifying and identifying one or more objects in an image. The customised YOLOv5 pre-trained model is used to detect the listed objects considered as roadblocks such as vehicles, pedestrians, motorbikes, bicycles and animals. The specifics of the process and the results are specified in *Figure 7*.

Figure 7 – YOLOv5 – Object detection

Step 1: The input image to the neural network needs to have a particular shape, like a blob. Using the blob and the image functions, the frame is translated into a blob for a neural network in order to be understood from a video sequence. The process also changes the pixel values between 0 and 1, and resizes an image to the required size (640 x 640). The YOLO network is then used for blob forwarding. The YOLOv5 method generates bounding boxes to aid in detection and prediction.

Step 2: A vector is used to represent the class, five network components (the centre x, centre y, breadth, height and sureness box) and each output bounding box. The output of the YOLOv5 algorithm consists of many boxes. Although the majority of the boxes are superfluous, they must be screened and removed. In the first step, a box will be removed if there is a low likelihood that an object will be detected. Only those boxes

are maintained that have probabilities greater than the specified threshold. The remaining boxes will do non-max suppression. Fewer boxes will overlap as a result.

Step 3: Combining lane detection techniques, the first layer of the work has to detect the lane and second layer detects the objects. Overlapping of these two layers helps to investigate further in order to find the driving space for the host vehicle. Interference from immovable objects and losing track are the main causes of inaccurate lane identification. Therefore, the obstacle detection in the lane detecting approach would be advantageous.

4.3 Drivable space localisation

In a complex road environment, a system is required to guide the driver to choose the driving space for avoiding the accident and traffic congestion. For obtaining the non-occupancy space, a horizontal reference line is fixed to detect the inter set points on lane line borders and front object coordinates. The graphic representation of how to estimate the coordinates of an object is shown in *Figure 8*. Each object's centre point (x,y), object width (w) and object height (h) are obtained from YOLOv5. The schematic in *Figure 9* provides a perspective on how to estimate the driveable area for an eco-vehicle.

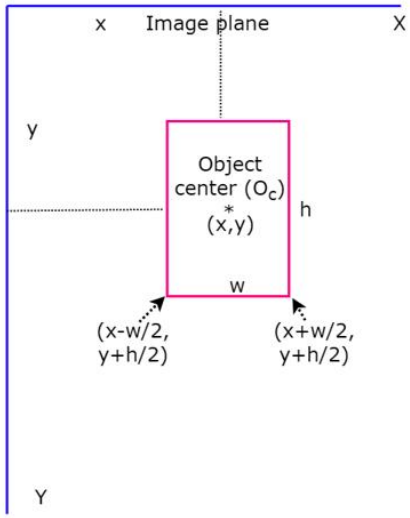


Figure 8 – Coordinate finding

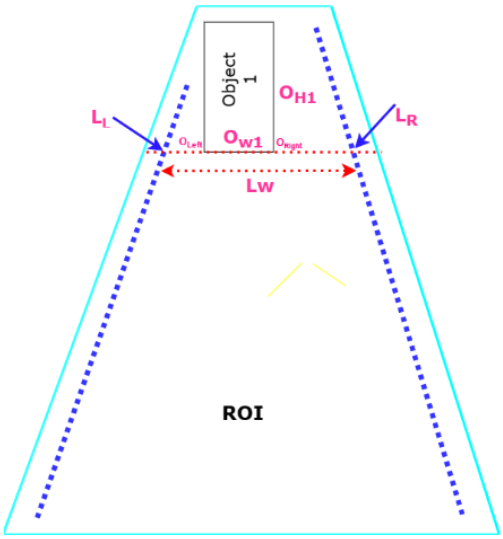


Figure 9 – Drivable space detection

The endpoints of the boundary lane are selected on this reference line, noted as lane left edge (L_L) and lane right edge (L_R), and the width of the lane is determined (L_w).

$$D_{path} = L_w - O_w \tag{10}$$

$$D_{Rpath} = L_R - O_{right} \tag{11}$$

$$D_{Lpath} = O_{left} - L_L \tag{12}$$

L_w – Lane width
O_w – Object width
L_L – Left lane boundary and reference line meeting point
L_R – Right lane boundary and reference line meeting point

According to the *Equation 11 and 12,* where the difference is exceed the threshold (L_w / 2 = threshold) positive driveable area is designated as a favorable driveable region for the host vehicle to approach in advance. In case the front object closes to the host vehicle or crosses the reference line, the 'Y' coordinate point of an object has to be considered to decide the driving space by drawing a new reference line to collect the corresponding LL, LR points.

5. EXPERIMENTS SETUP AND EVALUATION METRIC

5.1 Experiment setting

The experimental setup is the most important part and a supporting system for obtaining a good and accurate result. The summary of the dataset and details of the simulation parameters are shown in *Table 1* and *Table 2*, respectively.

5.2 Data set

BDD100K: The 100K video frames in the BDD100K dataset are divided into 70000 for training, 10000 for validation and 20000 for testing. The original validation set was tested, but the training set was unmodified because the testing partition's ground-truth labels are not accessible to the general public.

TuSimple: The TuSimple dataset, which is comprised of 6K road images taken from US highways, is now the most often used for lane recognition. The image has a resolution of 1280 x 720. 3K records for training, 358 records for justification and 2K records for testing make up the dataset. The images in the dataset are captured on the highway in primarily favourable weather and lighting conditions.

CULane: The CULane dataset, which is 20 times larger than the TuSimple dataset, contained a total of 133,235 frames. The training, validation and test set is 88K, 9K and 34K respectively. There are nine distinct scenarios included regular, throng, curve, dazzle, night, no line and arrow in the city.

TRLane: The Mixture of Rural Roads in China with the TuSimple dataset known as TRLane. The lanes and backdrops are distinguished in annotation, instances of lanes are not. The dataset includes 1.1K rural road image arrangements, each of which consists of 20 successive frames, in addition to a portion of the TuSimple dataset.

Table 1 – Dataset Summarisation

Dataset	Training	Testing	Size	Scene
Caltech	NA	1224	640 x 480	Urban streets
CULane	88880	34680	1640 x 540	Urban streets
KITTI	170	46	1248 x 384	Urban streets
Tu Simple	3432	2678	1280 x 720	Highways
TRLane	4323	3452	1280 x 720	Urban, rural streets
BDD K100	7845	2645	1280 x 720	Mixed-road type

5.3 Simulation environment

Choosing the working environment helps the model to work without any interruption. The proposed method and parameters in *Table 2* are majorly considered to execute the results.

Table 2 – Parameters of the simulation system

Parameters	Values
Environment	Major district road, urban and rural
Objects	Car, Bike, Person
Weather conditions	Dry
Camera	HD-RGB resolution 1280 x 720
Road surface	Dry, Damaged
YOLOv5	Input 640 x 640
Software	Windows 11, 64bit, Python 3.7, Opencv 4.1
Dataset	Caltech, CULanes, TuSimple. BDK100, Own Data

5.4 Evaluation metrics

The final segmentation result for the lane task requires an index to estimate the effectiveness of the test. The F-Measure assessment index was used to assess the model's capacity to forecast the lane line, since it can

take into account both the precision and recall measurements from the experiment. Using the F-Measure index in the formulas for evaluating image prediction outcomes is shown in (13) and (14)

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{13}$$

$$Recall\ = \frac{True\ Positive}{True\ Positive+False\ Negative} \tag{14}$$

F1 Scores: F scores are a statistician's tool for evaluating a binary model's precision. Both recall and precision are considered simultaneously. The F1 score, which ranges from 0 to 1, can be understood as a periodic mean of precision and recall. It is defined as follows:

$$F_1 = 2 \times \frac{precision \times recall}{precision+recall} \tag{15}$$

The average number of valid points per image serves as the basis for calculating accuracy:

$$\text{Accuracy} = \sum_{im} \frac{C_{im}}{S_{im}} \tag{16}$$

where C_{im} is the number of true points and S_{im} is the number of ground-truth points. When there is less than a predetermined amount of difference among the projected point and the ground truth, the point is considered correct.

6. RESULTS AND DISCUSSION

The recommended method is applied on the Google Cloud platform with an OpenCV library. Real-time data is collected from a rural and urban environment in and around the Salem region, Tamil Nadu, India with varying traffic conditions and congested roads. The combined proposed method of driving space identification on real time data results are shown in *Figure 10*. The green line detects one end of the continuous solid line boundary and the blue line indicates the centre dashed line of the two-way road type. The pink-coloured dashed line is the driving path area identified from the host vehicle view by considering the intersecting points of the front object and the lane boundary point inter set with the reference line. The customised YOLOv5 model is made to detect on-road objects like cars, bikes and people. The reference line is the portion where the road object occupancy is estimated. The object that are inside the lane area are considered, and the bounding box coordinate points at the bottom are taken for calculation. In the horizontal scanning, the interest points of the object coordinate and lane boundaries on imaginary line are considered to draw the driving path. The gap on both sides of the front object was found with reference to the lane boundary; the path with the greater gap on one side was chosen. The training losses for various methods for lane detection as carried out are shown in *Figure 11*. The most common and frequent object occurrences on road precision and recall (PR) curves obtained during the model training on each object detection algorithm are shown in *Figure 12*. *Tables 3* and *Table 4* give the comparative result of Accuracy, Recall, Precision, F1 score and Speed with the planned method of lane line and object finding separately. The model is validated through the various dataset and their results are represented in *Figure 13*.

Case1: Figure 13a The proposed method is validated with an image picked from the CULane dataset. In the parental lane, objects are detected for driving space but the gap is no more than the threshold value.

Case2: Figure 13b and c The driving space identification on sides of the front object from the BDK100, TuSimple datasets are validated. Even when the system recognized the driving space, it is difficult in making assumptions about which side to take grand for moving forward.

Case3: Figure 13d In rural and urban environments, two-way passing is carried out on the single-lane road. The lane line and road objects are detected with good accuracy and the driving space is identified. In the other case, if the object is moving towards the host vehicle, which indicates crossed the reference line, the new (x,y) coordinate point of nearest object is compared with the LR and LL and the driving space is assigned accordingly.

Figure 10 – Result of the drivable space identification

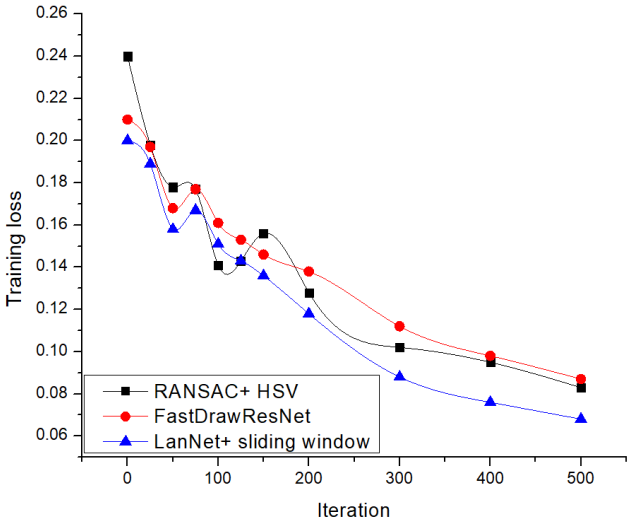
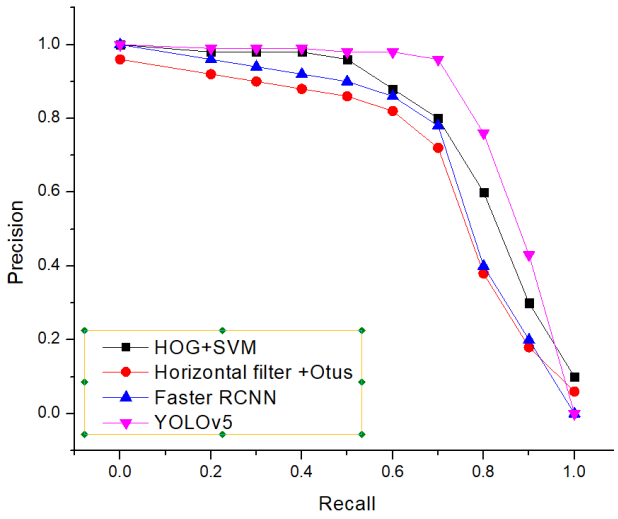
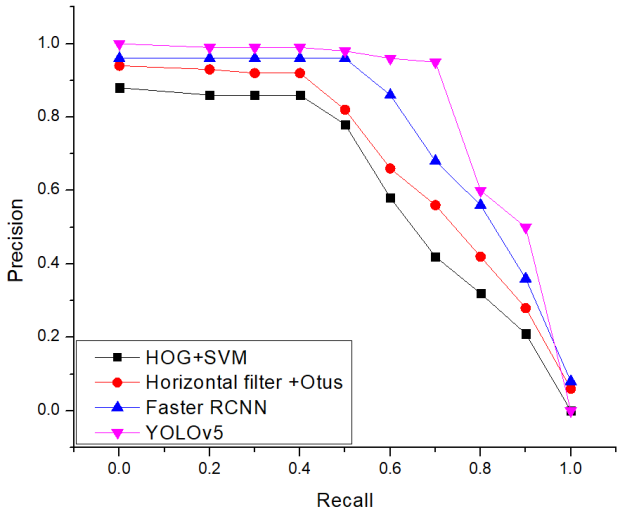


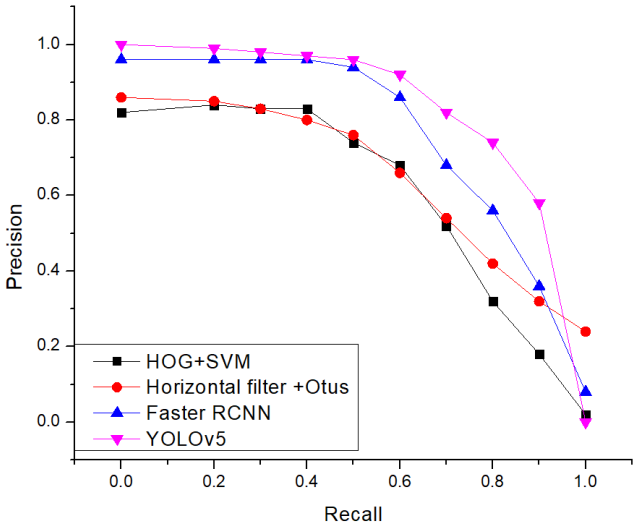
Figure 11 – Different lane line detection training loss curve



(a)



(b)



(c)

Figure 12 – PR curves
a) Car; b) Bike; c) Person

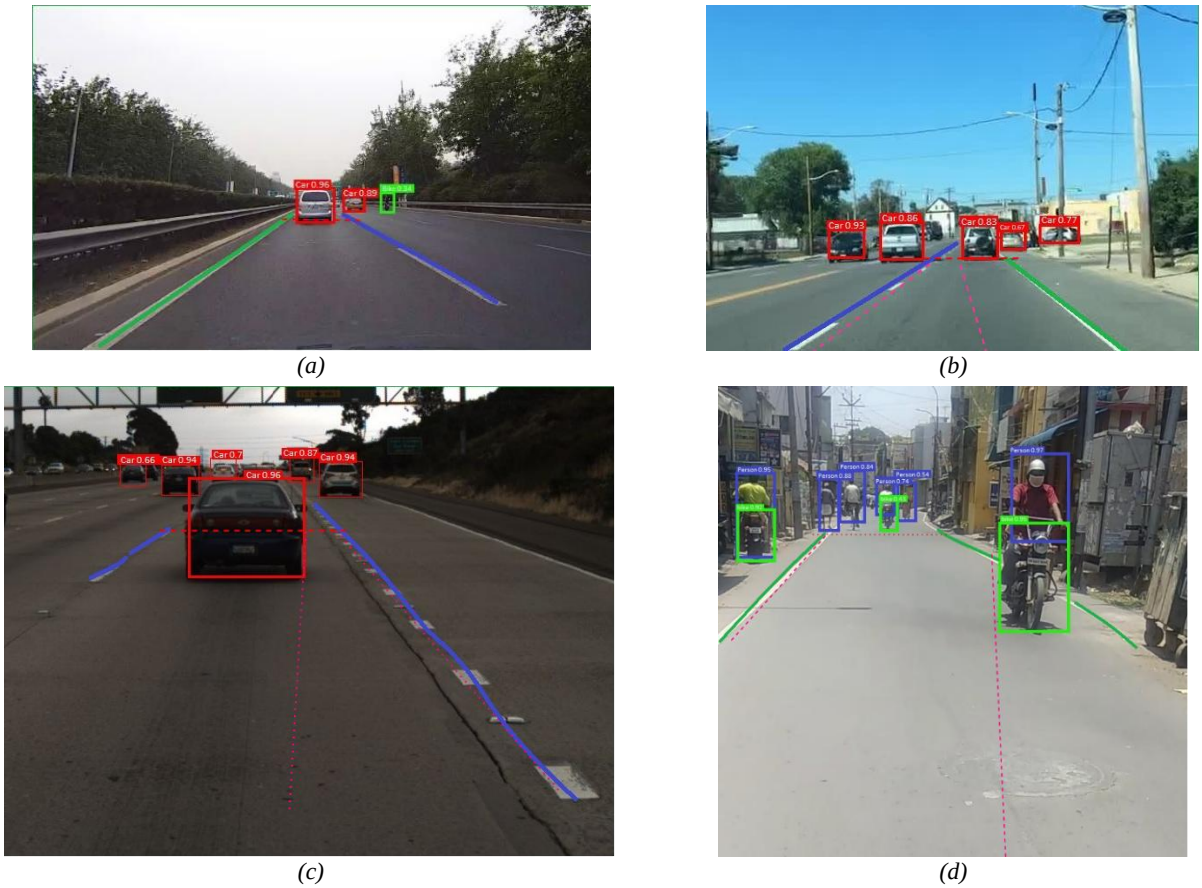
Table 3 – Comparison of proposed model with other Lane detection methods

Method	Algorithm	Average accuracy (%)	Recall	Precision	F1
Traditional method	Hough transform	95.70	0.9194	0.9409	0.9302
Traditional method	RANSAC + HSV	86.21	0.9028	0.9213	0.9021
Deep learning	FastDraw ResNet	95.00	0.9525	0.9622	0.9537
Deep learning (Proposed)	LaneNet + Sliding window	**97.03**	**0.9679**	**0.9767**	**0.9723**

Table 4 – Comparison of different Object detection methods

Model	Average accuracy (%)	Recall
HOG + SVM [42]	57.0	0.4892
Horizontal filter +Otus [39]	75.00	0.6833
Faster R-CNN [43]	81.60	0.8138
YOLOv5x [44]	**98.57**	**0.9812**

7. CONCLUSION



(a) *(b)* *(c)* *(d)*

Figure 13 – Model performance on various datasets
a) CULane; b) BDK100; c) TUSimple; d) Real Data

The proposed work is to identify the drivable space for the host vehicles, continuous movement and collision avoidance in the driver assistance and traffic congestion avoidance system. The proposed methods considered the front object's road occupancy and lane width to identify the best driving path for the host

vehicle. The novel method utilises the LaneNet with a sliding window for lane detection and a customised YOLOv5 pre-trained model for objects like car, bike and person detection and localisation with an accuracy of 97% and 98%, respectively. The combined proposed method is examined on different public datasets and real-time data. The model produced a promising output with an accuracy of 87 % to 95% on various dataset scenes, and this variation can be eliminated by obtaining the image data with a good quality acquisition system. In the future, the host vehicle's dimensions can be integrated into the path projection. This will enhance the model's capability to identify the drivable space more accurately by considering the physical size of the vehicle. Expanding the range of detectable road objects, optimising the pre-processing and parallelly enhancing the quality of the image data acquisition systems will increase to identify the best suitable driving space besides the front vehicle, which will help the driver make the right decision for further mobilisation. This method can also be used for finding better solutions when it is integrated with vehicle-to-vehicle communication to reduce the traffic congestion, vehicle to infrastructure to address the maintenance of the road, vehicle to people to ensure the safe pedestrian cross walk etc. and the performance can be improved on a daily basis by collecting new challenging data that help to upgrade the system to adapt to the current environment.

CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest with respect to the publication of this work.

DATA AVAILABILITY

The data used to support the findings of this study are available upon request from the corresponding author.

REFERENCE

[1] McCall JC, Trivedi MM. Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*. 2006;7(1):20–37. DOI: 10.1109/tits.2006.869595.

[2] Zheng T, et al. CLRNet: Cross layer refinement network for lane detection. ArXiv (Cornell University). 2022. Available from: https://arxiv.org/abs/2203.10350.

[3] Narayan A, Tuci E, Labrosse F, Alkilabi MHM. A dynamic colour perception system for autonomous robot navigation on unmarked roads. *Neurocomputing*. 2017;275:2251–63. DOI: 10.1016/j.neucom.2017.11.008.

[4] Yang F, et al. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*. 2019;21(4):1525–35. DOI: 10.1109/tits.2019.2910595.

[5] Zürn J, Burgard W, Valada A. Self-supervised visual terrain classification from unsupervised acoustic feature learning. ArXiv (Cornell University). 2019. Available from: https://arxiv.org/abs/1912.03227.

[6] Lee C, Moon JH. Robust lane detection and tracking for real-time applications. *IEEE Transactions on Intelligent Transportation Systems*. 2018;19(12):4043–8. DOI: 10.1109/tits.2018.2791572.

[7] Tian Y, et al. Lane marking detection via deep convolutional neural network. *Neurocomputing*. 2017;280:46–55. DOI: 10.1016/j.neucom.2017.09.098.

[8] Yu Y, Xu M, Gu J. Vision-based traffic accident detection using sparse spatio-temporal features and weighted extreme learning machine. *IET Intelligent Transport Systems*. 2019;13(9):1417–28. DOI: 10.1049/iet-its.2018.5409.

[9] Qiu Z, Zhao J, Sun S. MFIALane: Multiscale feature information aggregator network for lane detection. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(12):24263–75. DOI: 10.1109/tits.2022.3195742.

[10] Ahangar MN, Ahmed QZ, Khan FA, Hafeez M. A survey of autonomous vehicles: Enabling communication technologies and challenges. *Sensors*. 2021;21(3):706. DOI: 10.3390/s21030706.

[11] Huang Y, Chen Y. Survey of state-of-art autonomous driving technologies with deep learning. 20th International Conference on Software Quality. 2020;221–8. DOI: 10.1109/qrs-c51114.2020.00045.

[12] Garcia-Garcia A, et al. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*. 2018;70:41–65. DOI: 10.1016/j.asoc.2018.05.018.

[13] Minaee S, et al. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021;1. DOI: 10.1109/tpami.2021.3059968.

[14] Haris M, Hou J. Obstacle detection and safely navigate the autonomous vehicle from unexpected obstacles on the driving lane. *Sensors*. 2020;20(17):4719. DOI: 10.3390/s20174719.

[15] Haris M, Glowacz A. Lane line detection based on object feature distillation. *Electronics*. 2021;10(9):1102. DOI: 10.3390/electronics10091102.

[16] Huang DY, et al. Vehicle detection and inter-vehicle distance estimation using single-lens video camera on urban/suburb roads. *Journal of Visual Communication and Image Representation*. 2017;46:250–9. DOI: 10.1016/j.jvcir.2017.04.006.

[17] Yuan Y, Jiang Z, Wang Q. Video-based road detection via onli.ne structural learning. *Neurocomputing*. 2015;168:336–47. DOI: 10.1016/j.neucom.2015.05.092.

[18] Conrad P, Foedisch M. Performance evaluation of color based road detection using neural nets and support vector machines. 32nd Applied Imagery Pattern Recognition Workshop, 2003 Proceedings. 2004;157–60. DOI: 10.1109/aipr.2003.1284265.

[19] Bertozzi M, et al. Artificial vision in road vehicles. *Proceedings of the IEEE*. 2002;90(7):1258–71. DOI: 10.1109/jproc.2002.801444.

[20] Shang E, et al. Robust unstructured road detection: The importance of contextual information. *International Journal of Advanced Robotic Systems*. 2013;10(3):179. DOI: 10.5772/55560.

[21] Yan C, et al. A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. *IEEE Signal Processing Letters*. 2014;21(5):573–6. DOI: 10.1109/lsp.2014.2310494.

[22] Le TT, Lin CY, Piedad EJ. Deep learning for noninvasive classification of clustered horticultural crops – A case for banana fruit tiers. *Postharvest Biology and Technology*. 2019;156:110922. DOI: 10.1016/j.postharvbio.2019.05.023.

[23] Huang R, et al. A rapid recognition method for electronic components based on the improved YOLO-V3 network. *Electronics*. 2019;8(8):825. DOI: 10.3390/electronics8080825.

[24] Min W, et al. New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification. *IET Image Processing*. 2019;13(7):1041–9. DOI: 10.1049/iet-ipr.2018.6449.

[25] Cao CY, et al. Investigation of a promoted you only look once algorithm and its application in traffic flow monitoring. *Applied Sciences*. 2019;9(17):3619. DOI: 10.3390/app9173619.

[26] Junos MH, Khairuddin ASM, Thannirmalai S, Dahari M. An optimized YOLO-based object detection model for crop harvesting system. *IET Image Processing*. 2021;15(9):2112–25. DOI: 10.1049/ipr2.12181.

[27] Huang Y, Li Y, Hu X, Ci W. Lane detection based on inverse perspective transformation and Kalman filter. *KSII Transactions on Internet and Information Systems*. 2018;12(2). DOI: 10.3837/tiis.2018.02.006.

[28] Bonin-Font F, Burguera A, Ortiz A, Oliver G. Concurrent visual navigation and localisation using inverse perspective transformation. *Electronics Letters*. 2012;48(5):264. DOI: 10.1049/el.2011.3577.

[29] Neven D, et al. Towards end-to-end lane detection: An instance segmentation approach. 2022 IEEE Intelligent Vehicles Symposium (IV). 2018;286–91. DOI: 10.1109/ivs.2018.8500547.

[30] Ma N, Pang G, Shi X, Zhai Y. An all-weather lane detection system based on simulation interaction platform. *IEEE Access*. 2018;8:46121–30. DOI: 10.1109/access.2018.2885568.

[31] Park KY, Hwang SY. An improved Haar-like feature for efficient object detection. *Pattern Recognition Letters*. 2014;42:148–53. DOI: 10.1016/j.patrec.2014.02.015.

[32] Arróspide J, Salgado L, Camplani M. Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. *Journal of Visual Communication and Image Representation*. 2013;24(7):1182–90. DOI: 10.1016/j.jvcir.2013.08.001.

[33] Aziz L, Salam MdSBH, Sheikh UU, Ayub S. Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review. *IEEE Access*. 2020;8:170461–95. DOI: 10.1109/access.2020.3021508.

[34] Park JJ, Pan Y, Yi G, Loia V. Advances in computer science and ubiquitous computing. *Lecture notes in electrical engineering*. 2016. DOI: 10.1007/978-981-10-3023-9.

[35] Xu B, et al. Automated cattle counting using Mask R-CNN in quadcopter vision system. *Computers and Electronics in Agriculture*. 2020;171:105300. DOI: 10.1016/j.compag.2020.105300.

[36] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016;779–88. DOI: 10.1109/cvpr.2016.91.

[37] Zhou F, Zhao H, Nie Z. Safety helmet detection based on YOLOv5. 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA). 2021;6–11. DOI: 10.1109/icpeca51329.2021.9362711.

[38] Lee DH, Liu JL. End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal Image and Video Processing*. 2022;17(1):199–205. DOI: 10.1007/s11760-022-02222-2.

[39] Haghighat A, Sharma A. A computer vision-based deep learning model to detect wrong-way driving using pan–tilt–zoom traffic cameras. *Computer-Aided Civil and Infrastructure Engineering*. 2022;38(1):119–32. DOI: 10.1111/mice.12819.

[40] Haghighat A, Sharma A. A computer vision-based deep learning model to detect wrong-way driving using pan–tilt–zoom traffic cameras. *Computer-Aided Civil and Infrastructure Engineering*. 2022;38(1):119–32. DOI: 10.1111/mice.12819.

[41] Li Z, et al. Lane line detection network based on strong feature extraction from USFDNet. 2024 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA). 2024. DOI: 10.1109/icpeca60615.2024.10470951.

[42] Kumar S, Pandey A, Varshney S. Exploring the impact of deep learning models on lane detection through semantic segmentation. *SN Computer Science*. 2024;5(1). DOI: 10.1007/s42979-023-02328-5.

[43] Cao Y, Li C, Peng Y, Ru H. MCS-YOLO: A multiscale object detection method for autonomous driving road environment recognition. *IEEE Access*. 2023;11:22342–54. DOI: 10.1109/access.2023.3252021.

[44] Ji SJ, Ling QH, Han F. An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information. *Computers & Electrical Engineering*. 2022;105:108490. DOI: 10.1016/j.compeleceng.2022.108490.

[45] Majumder M, Wilmot C. Automated vehicle counting from pre-recorded video using you only look once (YOLO) object detection model. *Journal of Imaging*. 2023;9(7):131. DOI: 10.3390/jimaging9070131

[46] Song Y, et al. MEB-YOLO: An efficient vehicle detection method in complex traffic road scenes. *Computers, Materials & Continua/Computers, Materials & Continua*. 2023;75(3):5761–84. DOI: 10.32604/cmc.2023.038910.

[47] Lv Z, et al. Road scene multi-object detection algorithm based on CMS-YOLO. *IEEE Access*. 2023;11:121190–201. DOI: 10.1109/access.2023.3327735.

ஜெயமணி சித்தையன், குமார் பொன்னுசா

LaneNet மற்றும் CustomYOLOv5 அடிப்படையில் லேன் மற்றும் பொருட்களைக் கண்டறிவதன் மூலம் தன்னியக்க வாகன வழிசெலுத்தலை மேம்படுத்துதல்

Abstract

தன்னாட்சி வாகனம், போக்குவரத்து நெரிசல் அல்லது மோதல்களை உருவாக்காமல் தொடர்ந்து நகரும் திறனின் முக்கிய கவலைகள், லேன் மற்றும் ஆப்ஜெக்ட் கண்டறிதல் ஆகும். நுண்ணறிவுப் போக்குவரத்து அமைப்பை அதிக மக்கள்தொகை கொண்ட கிராமப்புற மற்றும் நகர்ப்புற சாலைகள், எண்ட் டூ எண்ட் இணைப்பு செயல்படுத்துவதற்கு இன்னும் பல சவால்களை எதிர்கொண்டுள்ளன. லேன்நெட்டைப் பயன்படுத்தி லேன் லைன் கண்டறிதலை ஸ்லைடிங் விண்டோவுடன் இணைத்து, தனிப்பயனாக்கப்பட்ட YOLOv5 ஐப் பயன்படுத்தி முன் சாலைப் பொருளைக் கண்டறிவதன் மூலம் இயக்கக்கூடிய இடத்தை அடையாளம் காண்பதே முன்மொழியப்பட்ட வேலை ஆகும். கணக்கீட்டு சிக்கலைக் குறைப்பதற்கும், செயல்முறையை விரைவுபடுத்துவதற்கும் பொருத்தமான முன் செயலாக்க முறைகள் மேற்கொள்ளப்படுகின்றன. முன் செயலாக்கத்தைத் தொடர்ந்து, ஓட்டுநர் பாதை இடத்தை அடையாளம் காண ஹோஸ்ட் வாகனத்திலிருந்து வெகு தொலைவில் குறிப்புக் கோடு அனுமானிக்கப்பட்டது. லேன் லைன் பார்டர்கள் மற்றும் ஆப்ஜெக்ட்ஸ் எல்லைக்குட்பட்ட பாக்ஸ் ஆயத்தொலைவு புள்ளிகள் குறிப்புக் கோட்டில் உள்ள வெட்டும் புள்ளிகள், ஓட்டுநர் பாதை இடத்தைக் கணக்கிட எடுக்கப்படுகின்றன. இறுதியாக, முன்மொழியப்பட்ட அமைப்பு பல்வேறு பொது மற்றும் சொந்த தரவுத்தொகுப்பில் சரிபார்க்கப்படுகிறது. லேன் லைன் கண்டறிதல் மற்றும் பொருள் கண்டறிதல் துல்லியம் முறையே 97% மற்றும் 98% ஆகியவை லேன்நெட் மூலம் நெகிழ் சாளரங்கள் மற்றும் தனிப்பயன் YOLOv5 மூலம் அடையப்படுகின்றன.

Keyword

ஓட்டுநர் பாதை கண்டறிதல், நுண்ணறிவு வாகனம், LaneNet, YOLO