IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 8, 2021, accepted August 16, 2021, date of publication August 20, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106377

A Deep Learning-Based Benchmarking Framework for Lane Segmentation in the Complex and Dynamic Road Scenes

RETAJ YOUSRI[1], MUSTAFA A. ELATTAR[2], AND M. SAEED DARWEESH[1], (Senior Member, IEEE)

[1]Wireless Intelligent Networks Center (WINC), School of Engineering and Applied Sciences, Nile University, Giza 12677, Egypt
[2]Center for Informatics Science (CIS), School of Information Technology and Computer Science, Nile University, Giza 12677, Egypt

Corresponding author: M. Saeed Darweesh (mdarweesh@nu.edu.eg)

This work was supported by Nile University.

ABSTRACT Automatic lane detection is a classical task in autonomous vehicles that traditional computer vision techniques can perform. However, such techniques lack reliability for achieving high accuracy while maintaining adequate time complexity in the context of real-time detection in complex and dynamic road scenes. Deep neural networks have proved their ability to achieve competing accuracy and time complexity while training them on manually labeled data. Yet, the unavailability of segmentation masks for host lanes in harsh road environments hinders fully supervised methods' operability on such a problem. This work proposes integrating traditional computer vision techniques and deep learning methods to develop a reliable benchmarking framework for lane detection tasks in complex and dynamic road scenes. Firstly, an automatic segmentation algorithm based on a sequence of traditional computer vision techniques has been experimented. This algorithm precisely segments the semantic region of the host lane in the complex urban images of nuScenes dataset used in this framework; hence corresponding weak labels are generated. After that, the developed data is qualitatively evaluated to be used in training and benchmarking five state-of-the-art FCN-based architectures: SegNet, Modified SegNet, U-Net, ResUNet, and ResUNet++. The performance evaluation of the trained models is done visually and quantitatively by considering lane detection a binary semantic segmentation task. The output results show robust performance, especially ResUNet++, which outperforms all the other models while testing them in different complex road scenes with dynamic scenarios and various lighting conditions.

INDEX TERMS Advanced driving assistance systems (ADAS), lane detection, lane segmentation, computer vision, dynamic scenes, fully convolutional networks (FCN), semantic segmentation.

I. INTRODUCTION

There has been an increasing interest in autonomous driving research because of its great impact on traffic management and the economy. Autonomous vehicles mimic human driving by making decisions and performing intelligent operations like a lane change, collision avoidance, object detection, and lane departure warning [1], [2]. The accuracy of these intelligent decisions and operations has the potential to alleviate human driver's burden and reduce traffic accidents that are almost entirely caused by human's improper decisions and actions [1]. Different artificial intelligence (AI) techniques enable autonomous vehicles to manage actions and take decisions based on various input data. Such data can be acquired by vehicle's camera, radio detection and ranging (RADAR), light detection and ranging (LIDAR), global positioning system (GPS), or communication system [1], [3], [4]. Different actuators can then perform physical output actions based on the taken intelligent decisions.

Automatic lane detection is considered one of the most challenging perception tasks found in autonomous vehicles nowadays. Many factors may result in poor road perception and make robust lane detection hard to achieve, especially in dynamic and harsh road environments. Some of these

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Zhao.

factors can be the vague nature of the lane patterns, the limited visibility of these lines at night, the variance in lane shapes and colors, the deterioration of the lane patterns over time, or the illusionary road shadows. These challenges make many current novelties focus on improving the accuracy and reliability of lane detection systems. This is because lane detection is considered only a sub-task from bigger ones like lane changing, lane departure warning, and lane keeping [2], [5].

Recently, cameras have become more reliable and capable of capturing any situation of the road environment in any direction. Different computer vision algorithms can be used to perform intelligent perception tasks based on the captured frames of road scenes. Lane lines have some unique features, like being parallel and distinguishable by their colors or edges. Fortunately, with the rapid growth of computer vision-based techniques, there have been various methods that can utilize these features for lane detection and segmentation. Some of these methods can be traditional and based on geometrical analysis, while others can be based on trainable deep neural networks (DNNs) [6]. Both methods have advantages and disadvantages which give a push towards such a topic in the research field.

The advantages of traditional computer vision techniques are various and can considerably be used for accurate lane detection. Yet, their computational time is high in complex scenes and cannot cope with the limitations of real-time applications. On the other hand, deep learning approaches have shown robustness in the prediction timing and can be reliably used for real-time lane detection. However, if the deep learning models are not well-trained, false predictions are likely to occur. The inefficient training can occur due to limited data availability, imprecise training labels, or poor information among the training data.

The contributions of this work can be listed as follows:

- Proposing a sequence of traditional computer vision techniques for automatic and precise lane segmentation in complex and dynamic road scenes.
- Developing a weak supervision framework that utilizes the proposed sequence to build up labels for a subset of nuScenes dataset [7] which is being used for the first time in the lane detection context.
- Extending nuScenecs dataset by generating lane labels for selected challenging road frames that contain different illumination conditions, lane shapes, and dynamic scenarios.
- Benchmarking the performance of five state-of-the-art deep learning segmentation models trained supervisely on our developed dataset to detect road lanes.
- Employing ResUNet++ to be trained for the first time on the lane detection task where it predominately outperforms the other tested models.
- Introducing a robust lane detection using an ensemble-based approach while testing the models by investigating the ensemble prediction of our top three trained models in shadowy scenes and obscuring road scenarios.

The remaining sections of this paper are organized as follows: Section II conducts the related work in using traditional computer vision techniques and deep learning methods for lane segmentation and detection. Section III introduces the proposed framework, including the segmentation for labels generation and the deep learning approaches. Section IV presents the data setup, experiments, results, analysis, comparisons, and limitations. Finally, the conclusion is given in section V.

II. RELATED WORK

As lane detection is an essential task in the advanced driving assistance systems (ADAS), several previous works have been developed to detect road lanes efficiently. In this section, a brief overview of the most efficient developed methods will be conducted. Firstly, in traditional methods, preprocessing is crucial to correct image distortion, remove pixel noise, and enhance overall information among the image. Gaussian filter [8]–[10] and Median filter [11] can be used for the smoothening operation that is usually done before edge detection. Image distortion removal was done in [12] to drive a corrected image with uniform dimensions. Moreover, the region of interest (ROI) is determined using several techniques to segment a part of the road containing the needed lane detection information. Selecting ROI can be done by conventionally choosing the lower two-thirds of the image area as done in [12] or the bottom side as in [13], or a subset from the frame as in [14] and [15]. However, these methods are inefficient with urban road scenes. Thus, some developed works selected ROI based on the vanishing point (VP) estimation [16]–[18].

After defining the region containing lanes, some techniques can be done to enhance lane features information. As lane lines are usually parallel, this information can be enhanced using the bird's-eye view that can be obtained from different perspective mapping [9], [19]–[21]. For lane color variations, [22]–[25] considered the usage of color spaces to segment different bright lane colors from the roads using specific channels like the lightness channel in HLS (Hue Lightness Saturation) color space. The next step is to fit lane lines where different line fitting models have been developed. For straight lanes, Hough transform (HT) is widely used as in [9], [26]–[28]. There have been other parametric fitting models such as hyperbola [29] and parabola [30] that can efficiently cover straight lines. For more flexibility and complex shapes (*e.g.* curved) coverage, semi-parametric models such as Catmull–Rom [31], [32], B-Snake [33], and Cubic Spline curves [34], [35] were developed. Random sample consensus (RANSAC) is considered a widely used lane line fitting algorithm that was adopted in many previous novels [9], [12], [24], [36]–[39].

For the deep learning-based methods, Huval *et al.* [40] trained convolutional neural network (CNN) architecture to detect lane lines for real-time usage. A unique approach was introduced in [41] where a dual-view convolutional neutral network (DVCNN) framework was proposed for

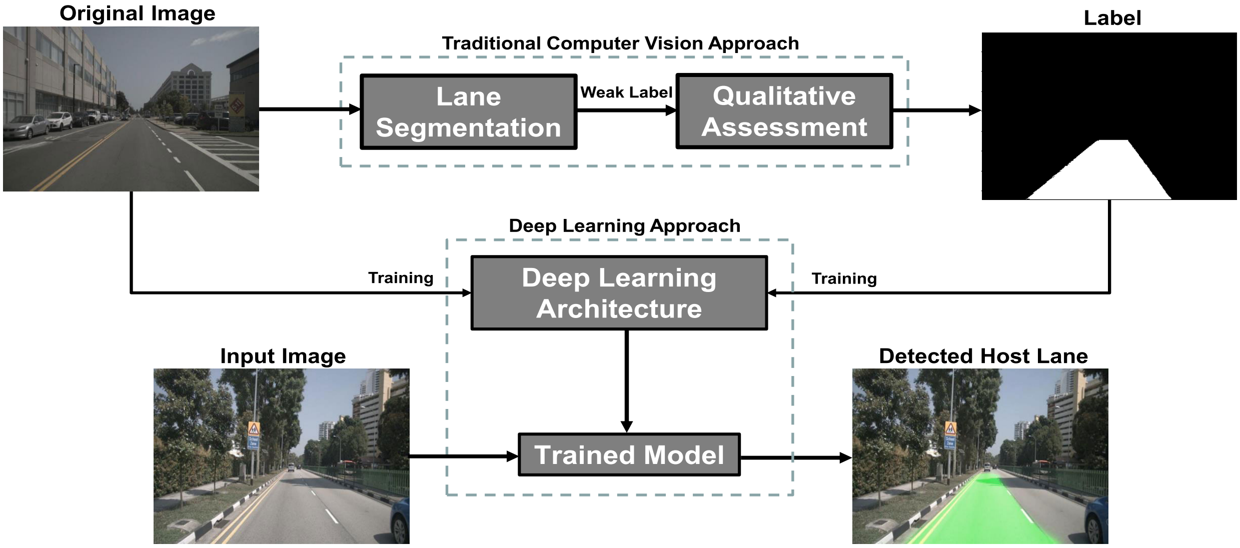


FIGURE 1. Overall framework.

robust lane detection. This novel was based on utilizing front-view and top-view images from which the false detections and non-club-shaped structures were removed, respectively. A weighted hat-like filter was then applied to find lane candidates, which were then processed by a CNN [41]. An efficient CNN framework based on point clouds was designed by [42] where the cloud points were preprocessed to produce reflectivity information that can be then fed into CNN. In [43] a hybrid framework was developed based on CNN and recurrent neural network (RNN) to detect lanes. In that study, CNN was adopted to detect the geometric lane attributes with respect to the region of interest, while RNN was utilized to visually infer the presence of lane structure relying on its internal memory [43]. An architecture based on SegNet was developed in [44] called LaneNet where the lane detection problem was introduced as an instance segmentation problem. Convolutional long short-term memory (ConvLSTM) has been widely used in computer vision and video analysis because of its feedback mechanism on temporal dynamics and the abstraction power on image representation. By relying on ConvLSTM in a hybrid architecture, Zou *et al.* [45] used multiple frames of a continuous driving scene to detect lane lines from the information of many frames rather than a single one. In [46], a robust multiple lane detection algorithm was proposed where a fully convolutional network (FCN) was used for lane boundary feature extraction, then Hough transform, and the least square method combined with the perspective transform (PT) was used to determine the lane lines accurately.

By surveying many novelties related to the lane detection task based on traditional computer vision and deep learning approaches, we realized that both have advantages and limitations as discussed earlier. Thus, this paper aims to efficiently utilize the advantages of each approach without intervening in its limitations to detect lanes accurately. The major usage of the traditional computer vision techniques in this work is to develop an algorithm to automatically generate uncertain annotation knowledge for lane segmentation in challenging complex scenes without considering the time complexity. On the other hand, this work also provides benchmarking state-of-the-art architectures on the developed data, giving intuition that achieving accurate and robust lane detection is possible. Visual and quantitative experiments will be done to demonstrate the effectiveness of this framework.

III. PROPOSED METHOD

The whole framework for achieving high-performance lane detection in complex scenes is illustrated in Fig. 1. The framework consists of two main approaches: the traditional computer vision approach and the deep learning approach. The traditional computer vision approach includes a proposed sequence of advanced, optimized, and adaptive techniques to perform automatic lane segmentation in images of challenging scenes. Consequently, weak labels are generated to be then evaluated using a qualitative assessment, which will be presented in Section IV before being used in the deep learning approach. The deep learning approach utilizes the developed data to supervise train different state-of-the-art deep learning architectures on the lane detection task.

A. LANE SEGMENTATION

In order to supervise train deep neural networks, images with their corresponding labels are essential for making the networks capable of distinguishing the different classes of the recognition task. In this work, images of challenging

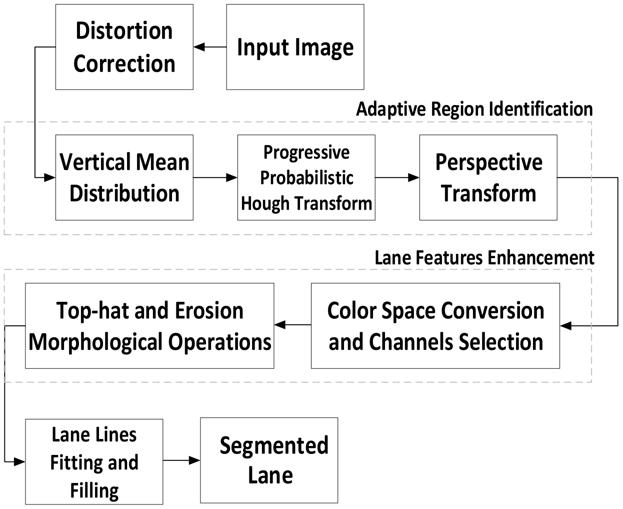

Distortion Correction
Input Image
Adaptive Region Identification
Vertical Mean Distribution
Progressive Probabilistic Hough Transform
Perspective Transform
Lane Features Enhancement
Top-hat and Erosion Morphological Operations
Color Space Conversion and Channels Selection
Lane Lines Fitting and Filling
Segmented Lane

FIGURE 2. The proposed sequence for segmenting the lanes in images with complex and dynamic road scenes.

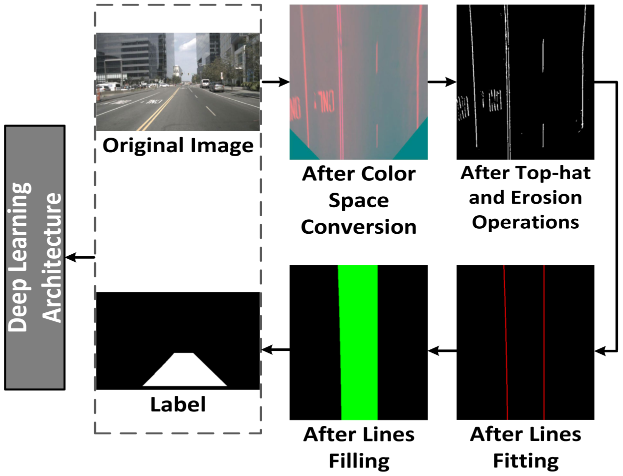

Original Image
After Color Space Conversion
After Top-hat and Erosion Operations
Deep Learning Architecture
Label
After Lines Filling
After Lines Fitting

FIGURE 3. Brief representation for the lane segmentation steps based on enhanced lane features. The original image and the generated label are inputs for training the deep learning architecture.

road scenes and dynamic scenarios are going to be precisely labeled. The labels contain only two classes: lane and non-lane. A sequence based on traditional computer vision techniques is experimented with to determine the pixels that lane lines occupy and bound accurately. Fig. 2 illustrates our proposed sequence for segmenting the semantic region of the host lane in challenging images. The sequence includes two major stages: adaptive region identification and lane features enhancement. These two stages are considered the most critical while dealing with challenging images of diverse road scenarios. Fig. 3 briefly represents the steps of lane segmentation in the context of enhanced lane features.

1) DISTORTION CORRECTION

The road scene images represent 2D mapping for the 3D real world. There are two types of image distortions that are likely to occur: radial distortion and tangential distortion. When radial distortion occurs, the lines on an image appear either less or more curved than how they actually are while in tangential distortion, the objects appear at deceptive distances [12], [47]. k_1, k_2, k_3 are the radial distortion coefficients, while p_1 and p_2 are the tangential distortion coefficients. The radial correction formulas are given as follows [12], [47]:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{1}$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2}$$

while the tangential correction formulas are given as [12]:

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)] \tag{3}$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x] \tag{4}$$

where r is the distance between a point on a corrected image (undistorted) and the center of that image, (x, y) are the coordinates of a point on the distorted image, while $(x_{corrected}, y_{corrected})$ is where that point will appear on the undistorted image. According to the previous equations, it is obvious that the distortion coefficients must be known first to eliminate these types of distortion and restore the straightness within an image. These coefficients can be derived using the camera calibration process where the above formulas are utilized as mathematical models. In this work, the checkerboard-based calibration technique is adopted to drive the needed distortion coefficients [12]. The camera matrix, which represents the intrinsic parameters matrix K, also obtained by camera calibration, is given as [12], [16], [48]:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where f_x and f_y represent camera focal lengths, while c_x and c_y represent the optical centers. After obtaining the distortion coefficients and the camera matrix, transformation matrix is derived to map the undistorted (corrected) images.

2) ADAPTIVE REGION IDENTIFICATION

Images captured from road scenes come with many details (*e.g.* sky and buildings), yet some of them can be useless or lead to inaccurate lane segmentation. Consequently, different adaptive and optimized methods are adopted in this approach to overcome this limitation. An adaptive region of interest (AROI) based on the vertical mean distribution (VMD) method is chosen for road segmentation. For identifying the lane region, we utilize the progressive probabilistic Hough transform (PPHT) to estimate the vanishing point. Based on the estimated vanishing point, it is possible to generate warped images showing the lane region without interference from undesired information.

a: VERTICAL MEAN DISTRIBUTION

In order to minimize the undesired effects of such off-lane information, filtering out must be done by masking parts of the images. Hence, we use in this work an adaptive algorithm based on a horizon line to segment the road [49]. Identifying the horizon line position is done using the VMD method proposed in [50]. The reason behind using VMD is that road scenes are generally divided horizontally into two main regions: sky/buildings region and road region. The intensities of the pixels throughout these two regions vary unevenly where the pixels of the sky region usually possess higher intensities than road pixels [50]. This variation shows a sudden change in pixel intensities across the line (rows) dividing the two regions. The VMD method relies on this feature and determined by using this equation [49]:

$$VMD(R) = \frac{1}{W}\sum_{C=1}^{W} I_G(R, C) \tag{6}$$

Among them, W is the width of the image (number of columns), while R and C stand for the row and column numbers, respectively. $I_G(R, C)$ is the gray pixel intensity at row R and column C. This equation is applied on every row of an image to finally plot the row numbers versus their corresponding average pixels values which represents the vertical mean distribution of an image. In this work, we use the images in size of 1280 × 720. The best horizon line is found to be identified at the local minimum occurs from row 300 to row 400 (counting from above to below) as shown in Fig. 4. It can be noticed that no big jumps of intensities difference are found in the desired region due to the urban nature of the images.

b: PROGRESSIVE PROBABILISTIC HOUGH TRANSFORM

After segmenting the road, we need to identify the region containing only lanes with no undesired road information (*e.g.* pavements, trees, and parked cars). This helps in extracting the most beneficial lane features out of an image. Perspective transform has proven its efficiency in identifying the lane region in many previous studies. However, to map an image to another perspective, vanishing point estimation is crucial. In this work, the progressive probabilistic Hough transform along with an optimized procedure are used for estimating the VP of each image. PPHT algorithm which was proposed in [51] is an optimization of HT that can detect different line orientations efficiently [49]. This algorithm utilizes only a small random subset of the available edge points that are sufficient to detect lines. As a result, PPHT is applied here to the edges obtained by Canny technique to deal with the arbitrary lane shapes found in the used images.

According to [52], straight lines can be parameterized by (ρ, θ) and the points on a specific straight line in the image space can be mapped into a single point in the parameter (Hough) space by applying Hough transform as shown in Fig. 5. ρ is the perpendicular distance from the origin to the line, while θ is the angle between ρ and the horizontal axis. Hence, the mapping relation between image space (X, Y) and polar parameter space (ρ, θ) is giving by:

$$\rho = x\ cos\ \theta + y\ sin\ \theta \tag{7}$$

where (x, y) is a point on a straight line in an image.

For an input image, the (ρ, θ) plane is divided into N_p x N_θ 2D matrix (rectangular cells) and represented by an accumulator array to hold places (bins) for all ρ and θ possible values [53]. PPHT algorithm works as follows:

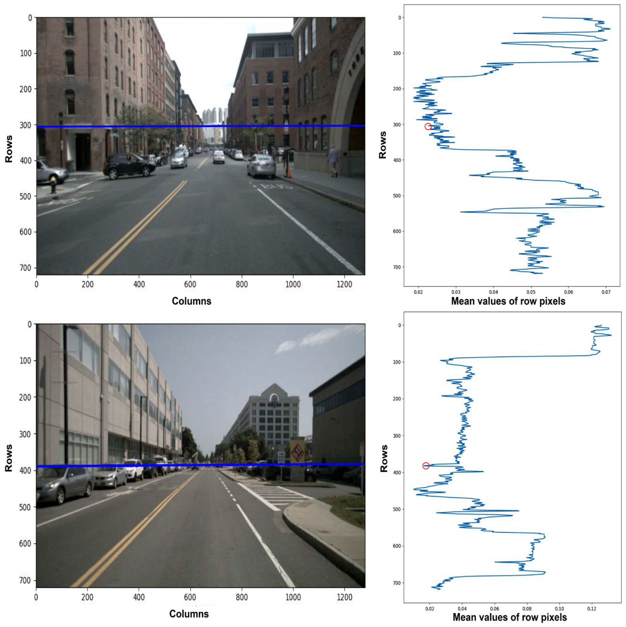
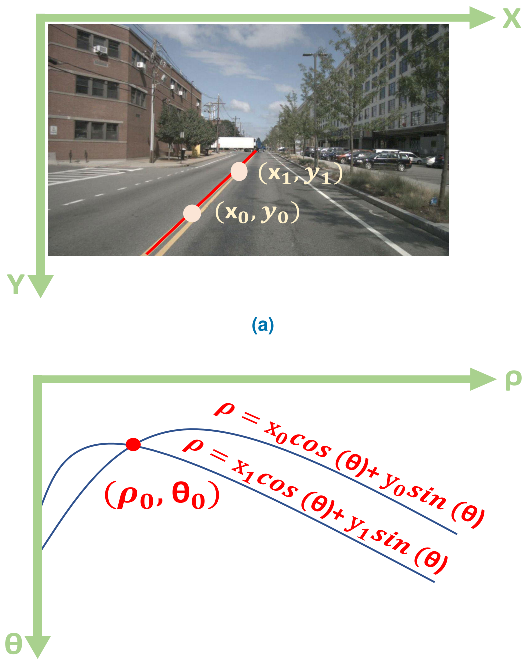


FIGURE 4. Detected horizon lines using VMD method.


X
Y
(x_1, y_1)
(x_0, y_0)
ρ
θ
$\rho = x_0 cos\ (\theta) + y_0 sin\ (\theta)$
$\rho = x_1 cos\ (\theta) + y_1 sin\ (\theta)$
(ρ_0, θ_0)

(a)

(b)

FIGURE 5. Mapping from (a) image space (X, Y) to (b) polar parameter space (ρ, θ).

1) Selecting randomly a point from an input image, and then deriving all the possible pairs of (ρ, θ) by substituting in *equ*.(7) with all the possible θ values to get the corresponding ρ.
2) Removing the selected pixel point from the input image, then updating the accumulator.
3) Scanning all over the updated accumulator to get the highest peak (bin that contains pair of (ρ, θ) with the most voting points) and comparing it with a pre-defined threshold (Th) value. If greater than (Th), proceed in the steps, otherwise return to *Step*1.
4) Choosing the longest segment found along the corridor of the peak in the accumulator that either is continuous or exhibits a gap not exceeding a given threshold. After that, removing all the points of the longest segment from the input image pixels.
5) Eliminating the points of the selected segment from the accumulator to no longer be a part of any voting process. Then, taking the selected segment as one of the output lines if it is longer than a predefined minimum length. Return to *Step*1.

After PPHT is applied to detect the lines, the vanishing point can be estimated. However, it is hard to get a unique intersection point when more than two lines exist. As a result, an optimization procedure should be employed as done before in [16]. If each output line i, after applying PPHT, can be represented by a point on it p_i and unit normal to it n_i, then the total squared distance from the VP to all the lines can be defined as a cost function given by [16]:

$$I = \frac{1}{2} \sum (n_i^T \, (V_p - p_i) \,)^2 \tag{8}$$

And it is required to find the minimum cost function to define the vanishing point. Accordingly, differentiation with respect to V_p is done to obtain the following expression which identifies the vanishing point [16]:

$$V_p = (\sum \, n_i n_i^T \,)^{-1} \, (\sum \, n_i n_i^T \, p_i \,) \tag{9}$$

c: PERSPECTIVE TRANSFORM

From just one image, we can mimic various images taken for the same scene at different angles and positions using perspective transform [54]. The road scene frames are usually captured using camera attached to the top of the vehicle resulting in images with many off-lane information. Consequently, using PT is useful in the context of lane segmentation where the original images can be transformed into warped images as if they are acquired from above the lanes as shown in Fig. 6. In order to get the target perspective of a warped image, it is needed to transform a trapezoid patch of the frontal road view into a rectangular image of the road from above. The trapezoid patch can be easily defined from the top, bottom, and side edges that all meet in the vanishing point [54]. By utilizing the vanishing point which we have estimated earlier in Equation (9), the needed edges can be known. Fig. 6b, shows the identified lane region which obviously visualizes the desired lane features.

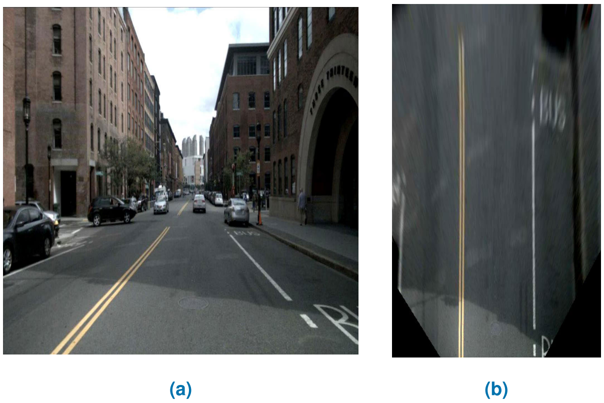

(a) (b)

FIGURE 6. Applying perspective transform: (a) The normal scene view; (b) Result of perspective transform.

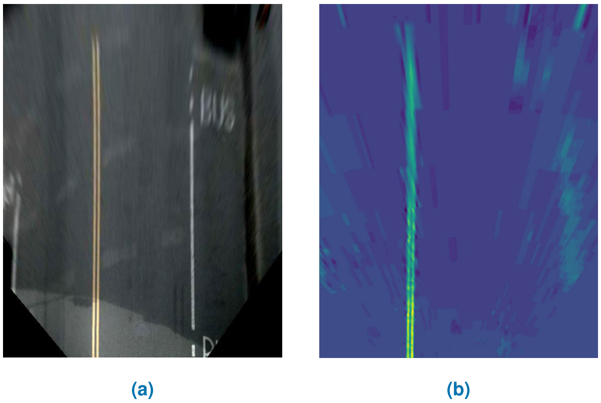

(a) (b)

FIGURE 7. Lab color space for yellow lanes detection: (a) Warped image contains yellow lane; (b) 'B' channel in Lab color space.

3) LANE FEATURES ENHANCEMENT

By knowing that the lane lines are parallel, straight, and have different colors, some image processing techniques can be done to preserve and enhance such features. Color space conversion and morphological operations are employed here to urge an accurate lane segmentation.

a: COLOR SPACE CONVERSION AND CHANNELS SELECTION

After applying the PT for better lane recognition, features enhancement is an important step for an efficient lane segmentation. To show more information about the lanes, the images are converted into the HLS (Hue Lightness Saturation) color space and Lab color space. The (L) channel in both spaces stands for the Lightness and utilized to track the bright regions of an image where the lanes are considered from [25], [55]. On the other hand, the (B) channel in Lab color space is used to visualize and track the yellow lanes as illustrated in Fig. 7 [25], [55]. Consequently, different lane colors can be differentiated by utilizing the effect of both channels in the two color spaces.

b: TOP-HAT AND EROSION MORPHOLOGICAL OPERATIONS

The morphological top-hat operation is typically used in this approach to isolate the brighter areas in the images from their darker surroundings. Lane lines are represented by bright pixels in the images. Hence, top-hat operation boosts accurate lane segmentation against different lightning changes by helping in de-noising and enhancing the contrast [56]. The bright edges can be detected easily using the top-hat operation without any interference from the other non-bright edges. This undesired interference is likely to occur using other detection techniques like Canny edge detection. Fig. 8 shows a visual comparison between the top-hat operation and the Canny technique in detecting the edges within the warped image (found in Fig. 6b). It is clear from the figure that the top-hat operation efficiently isolates the lane lines, which enhances the lane information for the upcoming stage. Erosion morphological operation is then utilized to eliminate noises coming from regions smaller than a defined structuring element.

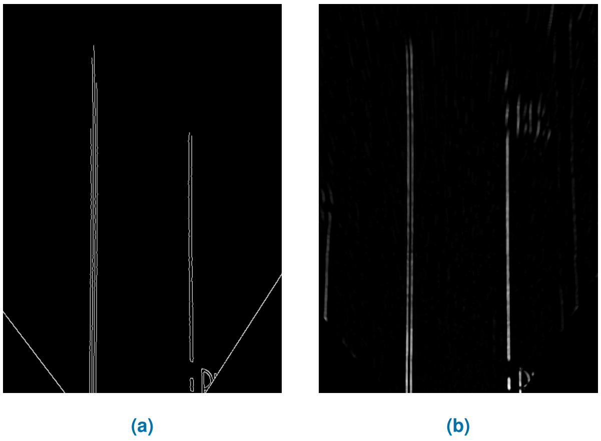


FIGURE 8. Edge detection: (a) Result of applying Canny edge technique; (b) Result of applying top-hat morphological operation.

c: LANE LINES FITTING AND FILLING

After applying the perspective transform to identify the lane region and enhancing lane features, line fitting is needed to finalize the segmentation stage. Afterward, we can easily generate the desired ground truth labels. In this framework, the objective is to deal with different lane colors and orientations. Thus, fitting the straight, dashed, and curved lane lines are essential. The histogram of each image is computed along its width (columns) to get the peaks at which lane lines are present. Prominently, there are two peaks in each image around its center, giving intuition about where to start the line fitting [57]. For more flexible fitting while dealing with arbitrary shapes, a sliding window search is used to iterate upon different lines shapes starting from the found starting point. Based on the previous, it is needed to fit a polynomial line on the detected lanes to segment it. Accordingly, we use second-order polynomial fit, which can be described as follows [57]:

$$f(y) = ay^2 + by + c \tag{10}$$

The fitted parallel lines are then drawn, and the area between them is filled to segment the whole region bounded by the lane. Eventually, the inverse perspective transform is applied to unwrap the images to the normal view, and then single channel conversion is done to produce the required ground truth labels.

B. DEEP LEARNING ARCHITECTURES

By proceeding in this framework, lane detection will be recognized as a semantic segmentation task of two classes: lane or non-lane. FCNs, introduced in [58], take advantage of the existing CNNs as being powerful visual models capable of learning hierarchies of features. Different FCN-based architectures can be supervised trained to produce a pixel-to-pixel semantic segmentation map by identifying each output pixel as a lane, or non-lane pixel [58]–[61]. In FCN, the fully connected layers were replaced with convolutional ones to form a fully convolutional network that outputs spatial maps. Inspired by the success of the two basic FCN-based architectures: SegNet [62], and U-Net [63], they will be used in this approach along with other improved architecture. A brief description of these two architectures will be presented below:

1) SEGNET

Based on the encoder-decoder architecture where at the encoder network, convolution and max pooling operations are performed. Each encoder performs convolution with a filter bank to produce a set of feature maps followed by batch normalization. Element-wise rectified linear unit (ReLU) is then applied, followed by max-pooling. Each encoding layer has a corresponding decoding layer in the decoder network. The decoder upsamples its input feature maps using the indices of the max-pooling to develop sparse feature maps. These maps are convolved then with a decoder filter bank. In the end, the final decoder outputs high dimensional feature representation, which is fed to a trainable Softmax classifier that classifies each pixel giving the final segmentation [62].

2) U-NET

This architecture consists of two main paths: a contracting path and an expansive path. The contracting path represents a repeated application of two unpadded convolutions followed by the rectified linear unit (ReLU) activation function and max-pooling operation for downsampling. The number of feature channels is doubled at each downsampling step. Each step consists of feature map upsampling and then up-convolution in the expansive path, concatenating with the corresponding cropped feature map from the contracting path. This halves the number of feature channels. At the same

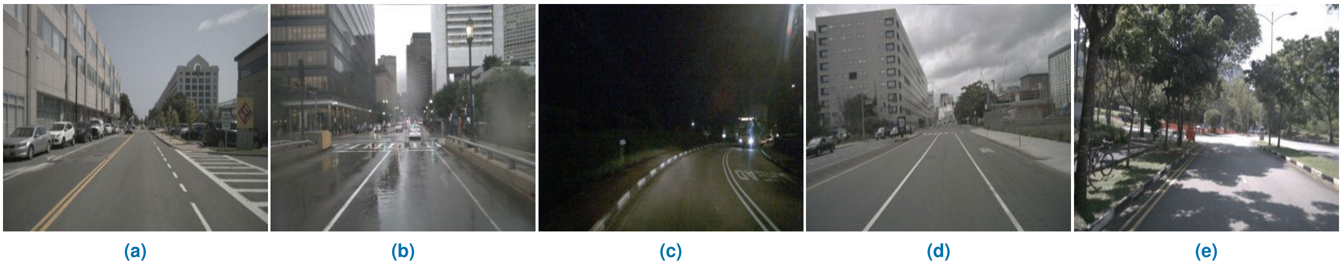


FIGURE 9. A sample from the selected frames of nuScenes dataset: Frame (a) is at daylight; (b) is rainy; (c) is at night, (d) is cloudy, and (e) is shadowy. Frames (a) and (e) contain yellow straight lane lines; (c) contains curved white lane lines; while (b) and (d) contain white straight lane lines.

decoding step, two convolutions followed by ReLU are done. At the final layer, a convolution operation maps the output of the network [63].

Since we aim in this work to benchmark different state-of-the-art architectures on the developed data, a comparative approach is supposed to be done. Accordingly, both ResUNet [64] and ResUNet++ [65], beside U-Net and SegNet, are implemented. ResUNet stands for deep Residual U-Net, where it uses the encoder-decoder backbone of U-Net combined with residual connections, atrous convolutions, spatial pyramid pooling (SPP), and multi-tasking inference [64]. ResUNet++ significantly outperforms U-Net and ResUNet according to [65]. It contains one stem block followed by three encoder blocks, Atrous Spatial Pyramid Pooling (ASPP), and three decoder blocks. There are other networks that have focused on multi-scale feature extraction modules which can be used in our study, such as SPP and Inception blocks. However, in a couple of chosen architectures for our study, we utilize the SPP block, which is comparably similar to the inception block in its effect [64], [65]. Other layer types such as self-attention, squeeze, and excitation modules have not been experimented here.

Moreover, simple modifications are done to SegNet to make it computationally less complex. Instead of adding batch normalization after each convolutional layer, it is added only in the input layer of the encoder part. Also, by knowing that Softmax fits more in multi-class classification and our problem is based on only two classes, it is replaced in a modified version with ReLU. Finally, we add some dropouts to avoid overfitting. At this point, we have five architectures: SegNet, Modified SegNet, U-Net, ResUNet, and ResUNet++ to be trained on the lane detection task in complex road scenes.

IV. EXPERIMENTS AND RESULTS

Early in this section, the data setup and the training strategy will be presented. After training the architectures based on the developed data, certain evaluation criteria will be used to measure the performance of the models on testing them in various challenging conditions. An ensemble-based approach will be conducted in this section as well. Finally, the results will be discussed and compared to other related work.

A. DATA SETUP

By focusing on dealing with the complex and dynamic road scenes, choosing an adequate dataset to apply our framework on is important. Specifically, we are concerned with different illumination conditions and lane shapes common in the real driving environment. The steps for developing the extended data, including the data selection and the qualitative assessment, will be presented in the upcoming part.

1) DATA SELECTION

NuScenes [7] is the only chosen dataset to be used in this framework. It is considered the first dataset to carry the full autonomous vehicle sensor suite (6 cameras, 5 radars, 1 LIDAR, GPS, and Inertial Measurement Unit (IMU)) [7]. However, only the front camera frames are utilized in this work. The driving scenes data were collected in Boston and Singapore where dense traffic and highly challenging driving situations are found [7]. The reason behind choosing nuScenes in this approach is the availability of various unlabeled frames for harsh road scenes. This is needed to train reliable models capable of detecting road lanes under various conditions. NuScenes dataset contains 1.4 million RGB (Red Green Blue) images for various road scenes, especially the challenging urban ones. However, some frames either does not contain any lane lines or contain pedestrian crossing road marking and turning spots which are not useful in our work.

Consequently, around 26, 000 sequential frames from different scenes were randomly downloaded, then converted into videos to pick and build up useful training data efficiently. To discard the frames containing no useful information, some parts of the videos were cropped. Once again, the videos were trimmed down to balance the different lane categories and road conditions contained within the training data to reach 9, 121 frames finally. These frames provide all the essential information and conditions needed for training the implemented FCN-based architectures as they contain different:

- Lane colors (yellow and white).
- Lightning conditions (shadowy, daylight, night, cloudy, and rainy).
- Lane orientations (straight and curved).

Fig. 9 shows a sample from the selected images, which contains a diversity of the needed complex road scenes.

TABLE 1. Content distribution among the selected frames from nuScenes dataset for the training stage.

Content List		
Frame Content	Frames Number	% from the whole labeled data
Daylight	670	7.3
Night	1,931	21.2
Shadows	2,033	22.3
Clouds	3,287	36.0
Rain	1,200	13.2
Curved Lanes	2,451	26.9
Straight Lanes	6,670	73.1
Yellow Lanes	2,980	32.7
White Lanes	9,067	99.4

The distribution of the various lighting conditions and the morphological information of the lanes among the selected frames is illustrated in Table 1. The upper part of the table shows the distribution of different lighting conditions, while the lower part shows the distribution of different lane colors and orientations. A road with a yellow lane line usually contains a parallel white one; thus, we can notice from Table 1 that the frames containing white lanes are predominant.

2) QUALITATIVE ASSESSMENT

Afterwards, the selected images are passed through our proposed automatic segmentation algorithm, presented earlier, to generate their corresponding labels. At this point, no reference tells about the reliability of these automatically generated labels, and thus they can be considered weak labels. As the precise segmentation of the host lane in the images would significantly improve the training efficiency of the deep networks, a qualitative assessment is needed to ensure the validity of the generated labels. During this qualitative assessment, two independent raters were asked to visually evaluate 200 generated ground truth labels selected randomly with their corresponding original frames. For each label, the rater should give a score according to the four categories described in Table 2. This visual assessment was done two times by each rater where the data were shuffled in the second assessment to ensure unbiased decisions. Results of the qualitative assessment are illustrated in Fig. 10. The chart shows the reliability of the generated labels based on the four visual assessments of the two raters, a total of 800 evaluations, as the labels with insufficient lane segmentation are a minority. For a better realization of the qualitative assessment, we considered the inter-raters variability and the intra-rater variability. The inter-raters variability represents the number of disagreements between the two raters. Whereas the intra-rater variability represents the number of disagreements between the two visual assessments of each rater. Table 3 shows the number of disagreements corresponding to variability kinds. The ratio in the fourth column of the table represents the number of labels evaluated with disagreements to the total number of labels (200) selected for evaluation.

Accordingly, we can now consider the generated labels reliable to develop a framework for training different state-of-the-art FCN-based architectures on the lane detection task. A sample from the segmentation results showing the generated ground truth labels is found in Fig. 11.

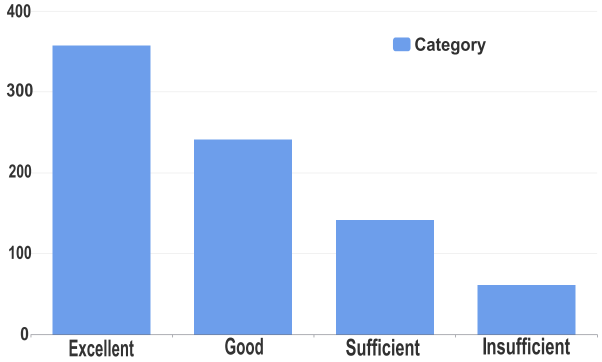


FIGURE 10. Categorical distribution among our generated labels according to the results of the qualitative assessment.

To enrich the number of available data and to achieve a more balanced distribution among the available information, augmentation is done by using flipping and rotation operations to increase the data up to 13, 521 available frames. The augmentation is done on the frames by considering only the lighting conditions. The content distribution before and after augmentation is illustrated in Fig. 12.

B. TRAINING STRATEGY

Based on the extended part of nuScenes dataset, the five state-of-the-art architectures can be efficiently trained. We randomly split the training data for the training stage, where the ratio of the training set to the validation set is chosen to be 0.9 : 0.1. The training is performed twice for each architecture: one time for 50 epochs and the other for 100 epochs. In the experiments, the images are sampled to a resolution of 256×128. In order to improve the performance of the FCN-based architectures, it is crucial to obtain the optimal parameters during the training stage. Thus, defining loss function(s) suitable for the semantic segmentation task is carefully done to approach the needed optimal parameters. In this work, a hybrid loss function using the Binary Cross-Entropy and the Dice Loss is utilized [66]. They are

TABLE 2. Scoring criteria defined for the qualitative assessment.

Category	Score	Description
Excellent	4	Accurate following for both lane markings and road curvature/orientation.
Good	3	Slight deviation outer lane markings but accurate following for road curvature/orientation.
Sufficient	2	Inaccurate following for the road curvature/orientation but satisfactory following for lane markings.
Insufficient	1	Inaccurate following for both the lane markings or the road curvature/orientation.

TABLE 3. Inter- and intra-rater variability during the qualitative assessment.

Variability Kind	Rater Number	Number of Disagreements	Ratio
Intra-rater	1	4	0.020
Intra-rater	2	7	0.035
Inter-raters	Both	11	0.055

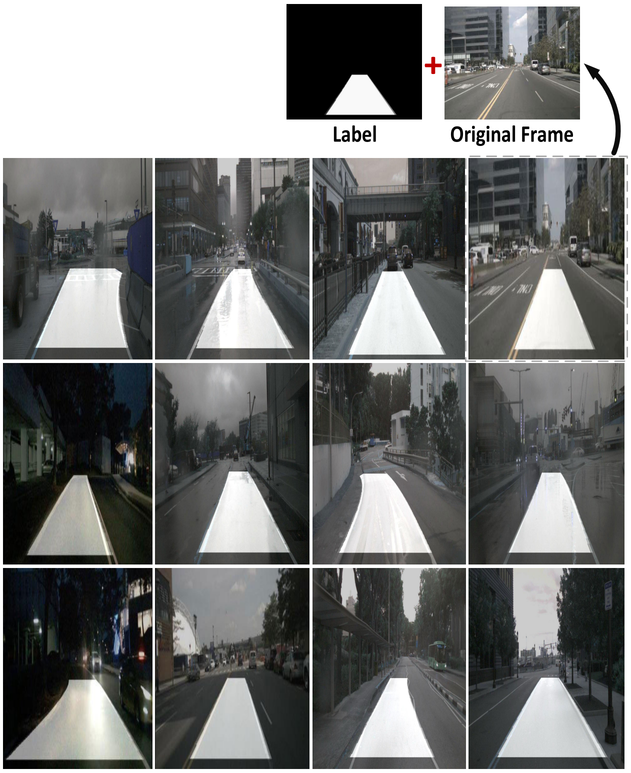


FIGURE 11. Visualizing a sample from the generated labels after lane segmentation.

mathematically given as the following:

$$L_{BCE}\,(y,\hat{y}) = -\,(y\,log\,(\hat{y})\, +\, (1-y)\,log\,(1-\hat{y})) \qquad (11)$$

$$DL\,(y,\hat{y}) = 1\, -\, \frac{2y\hat{y}\, +\, 1}{y\, +\, \hat{y}\, +\, 1} \qquad (12)$$

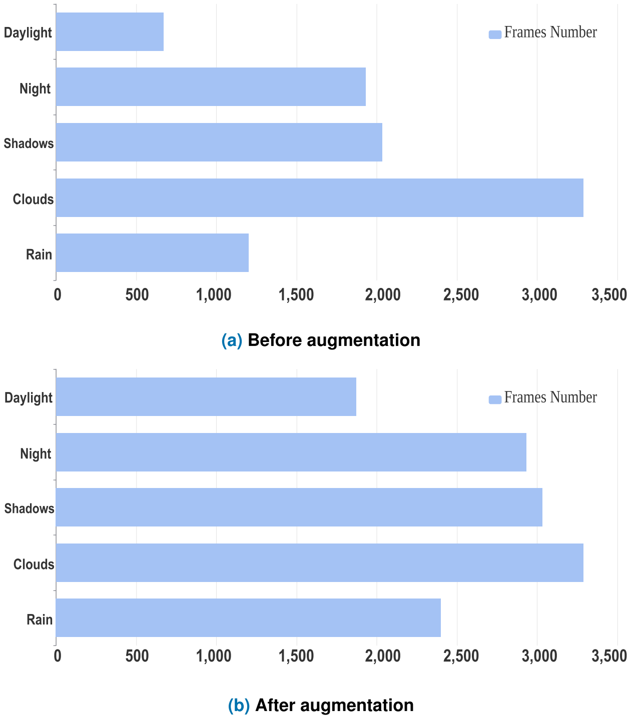
where y is the predicted value by the prediction model and $\hat{y}$ is the actual value.

C. PERFORMANCE EVALUATION

Evaluating and comparing the performance of the five trained models is one of the objectives of this work to benchmark different state-of-the-art architectures. All the implemented architectures are meant to perform semantic segmentation based on pixel-wise classification of two classes. Consequently, metrics that were employed for evaluating the models are:

- Pixel Accuracy
- Insertion-Over-Union
- Dice Coefficient

The pixel accuracy is recognized as the percent of pixels classified correctly when a binary semantic segmentation is applied. However, pixel accuracy is not the best metric to rely on for evaluating semantic segmentation models. The reason behind this is the class imbalance nature of the images. For the lane detection example, the classes within an image are extremely imbalanced, where the lane information class makes up only a small portion of the image. Thus, misclassifications are likely to occur, and still, this metric can give out high accuracy. On the other hand, insertion-over-union (IoU), also known as Jaccard Index, and dice coefficient can represent the performance more efficiently as they depend on the degree of overlap between the predicted segmentation mask and the reference segmentation mask [67]. Both metrics can be formulated as follows:



(a) Before augmentation

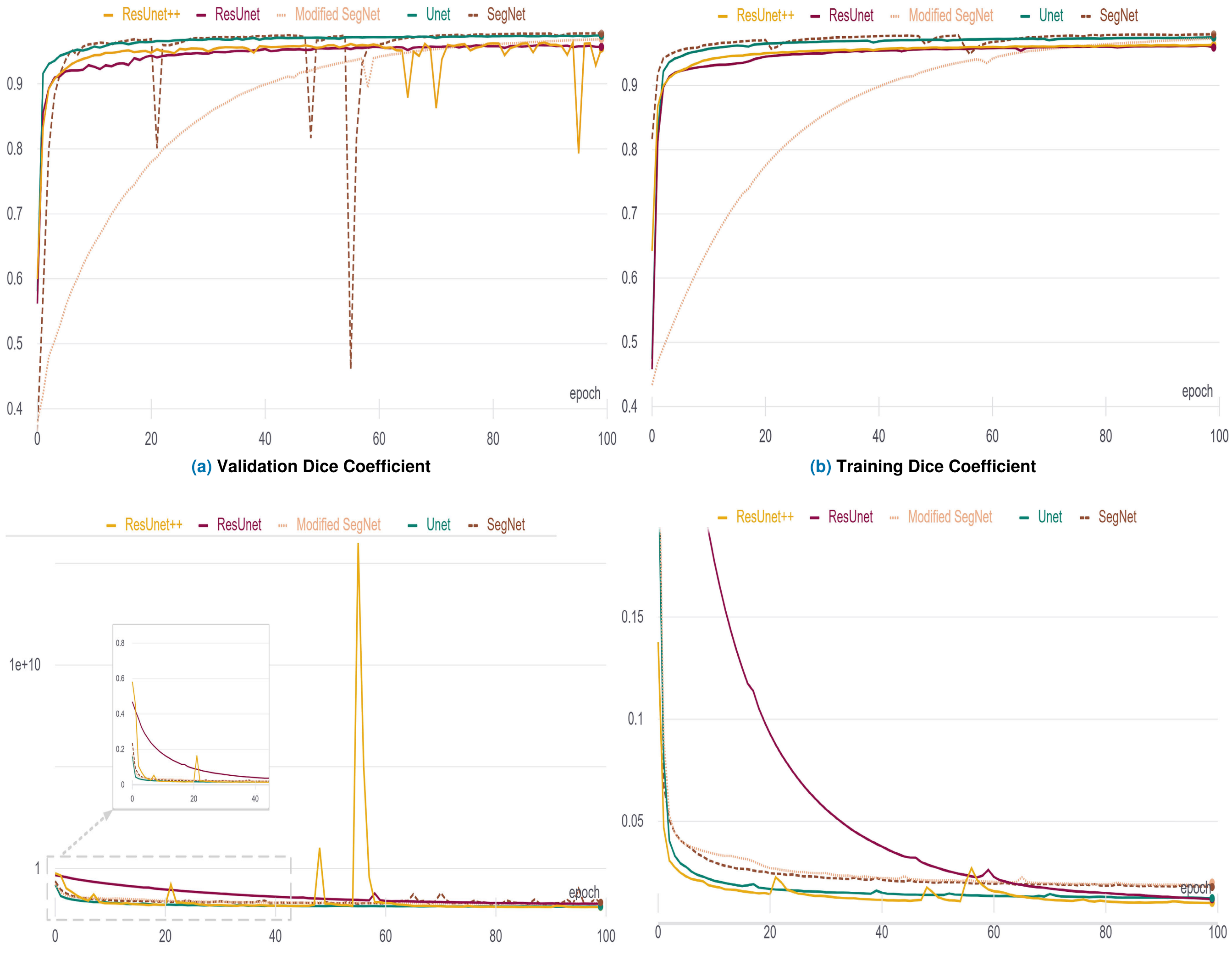
(b) After augmentation

FIGURE 12. The content distribution before and after the augmentation.

$$Dice\ Coefficient = \frac{2\sum P_{pred}\,P_{true}}{\sum P_{pred}\, +\, \sum P_{true}} \qquad (13)$$

$$IoU = \frac{\sum P_{pred}\,P_{true}}{(\sum P_{pred}+\sum P_{true})-(\sum P_{pred}\,P_{true})} \qquad (14)$$

According to the formulas, the dice coefficient represents double the overlap area between the predicted and the reference segmentation masks divided by the total number of pixels in both masks. On the other hand, IoU represents the overlap area divided by the union area between the predicted mask and the ground truth label (reference mask). Based on these metrics, Table 4 illustrates the performance of the models while training at 50 and 100 epochs. Fig. 13 shows how loss and dice coefficient values of the training and validation sets change over the pre-defined epochs. According to Table 4, we can conclude that ResUNet++ outperforms all



(a) Validation Dice Coefficient

(b) Training Dice Coefficient

(c) Validation Loss

(d) Training Loss

FIGURE 13. Curves of loss and dice coefficient while training and validating the models at 100 epochs.

the other architectures and gives dice coefficient value up to 0.978. The training results also show that the modifications to SegNet yielded better performance relative to the original SegNet architecture.

D. TESTING RESULTS AND DISCUSSIONS

For the testing stage, 100 images for each of the testing road conditions shown in Table 5 are selected from the nuScenes dataset other than that were used in training and validating the models. Using our automatic segmentation algorithm, labels for these images are generated to form a testing set. Based on the best weights of the models and the developed testing set, benchmarking the five state-of-the-art architectures is done on the lane detection task. In this subsection, there are two main objectives will be conducted. Firstly, performance evaluation for the models will be done quantitatively based on different lighting conditions and lane morphologies. Secondly, a robustness verification will be done based on a visual basis.

TABLE 4. Validation results of the training stage.

Epochs	Models	Dice Coeff.	IoU	Pixel Acc.
50	SegNet	95.7 %	91.9 %	96.2 %
	Modified SegNet	95.4 %	91.2 %	96.2 %
	U-Net	97.2 %	94.6 %	96.3 %
	ResUNet	92.2 %	85.6 %	96.3 %
	ResUNet++	**97.6** %	**95.4** %	**96.4** %
100	SegNet	95.4 %	91.4 %	96.2 %
	Modified SegNet	95.7 %	92.0 %	96.3 %
	U-Net	97.4 %	95.0 %	96.4 %
	ResUNet	96.9 %	94.0 %	96.3 %
	ResUNet++	**97.8** %	**95.7** %	**96.4** %

Based on the pre-defined evaluation metrics, Table 5 illustrates the average testing results of every category separately. As discussed earlier, the dice coefficient gives a better intuition while dealing with semantic segmentation. Thus, we will be focusing on this metric while analyzing the testing results. The main objective of this work is to deal with complex

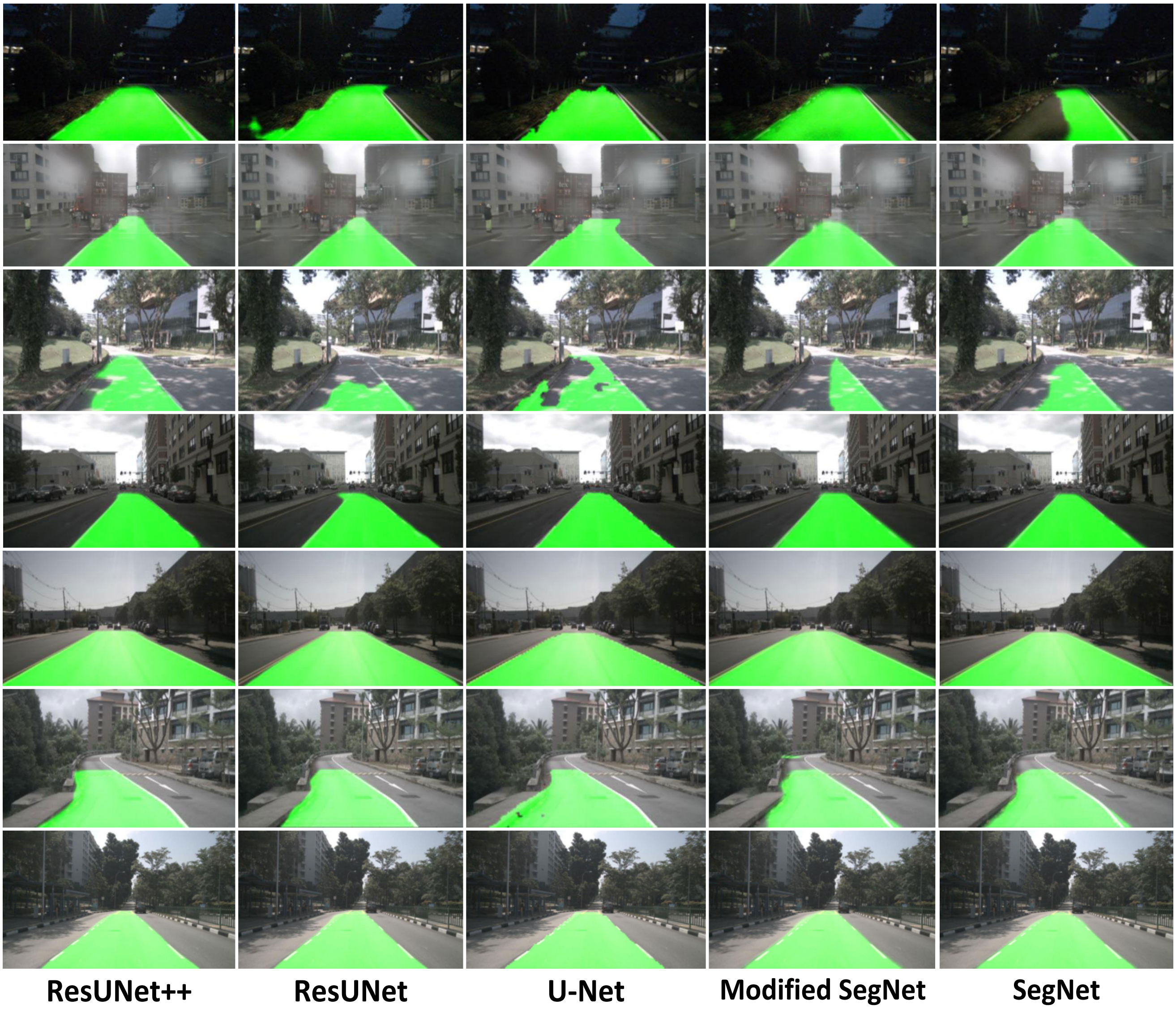


FIGURE 14. Samples from the detection results based on the extended nuScenes dataset. Each row represents a specific testing road condition. The conditions are arranged from above to below as follows: night, rainy, shadowy, cloudy, daylight/yellow lane line, curved lane, and dashed lane. The rows from left to right represent the output results of ResUNet++, ResUNet, U-Net, Modified SegNet, and SegNet.

TABLE 5. The testing results of the trained models.

Models	Daylight			Cloudy			Shadowy			Night		
	Acc.	Dice	IoU	Acc.	Dice	IoU	Acc.	Dice	IoU	Acc.	Dice	IoU
ResUNet++	**0.991**	**0.983**	**0.967**	**0.992**	**0.989**	**0.978**	**0.940**	**0.669**	**0.503**	**0.980**	**0.922**	**0.855**
ResUNet	0.990	0.981	0.962	0.990	0.978	0.959	0.928	0.508	0.340	0.971	0.888	0.798
U-Net	0.990	0.979	0.959	0.991	0.988	0.976	0.931	0.615	0.444	0.979	0.891	0.803
Modified SegNet	0.986	0.983	0.966	0.988	0.985	0.970	0.912	0.472	0.308	0.934	0.846	0.733
SegNet	0.982	0.972	0.942	0.988	0.979	0.959	0.928	0.501	0.334	0.920	0.842	0.727
	Rainy			Yellow Lanes			Curved Lanes			Dashed Lanes		
	Acc.	Dice	IoU	Acc.	Dice	IoU	Acc.	Dice	IoU	Acc.	Dice	IoU
ResUNet++	**0.987**	**0.972**	**0.946**	**0.978**	**0.984**	**0.969**	0.962	0.950	0.905	**0.974**	**0.980**	**0.961**
ResUNet	0.986	0.971	0.944	0.977	0.976	0.953	0.957	0.961	0.925	0.973	0.979	0.959
U-Net	0.980	0.969	0.940	0.980	0.969	0.940	**0.973**	**0.963**	**0.929**	0.977	0.978	0.957
Modified SegNet	0.976	0.964	0.931	0.969	0.955	0.914	0.948	0.931	0.871	0.971	0.965	0.932
SegNet	0.974	0.963	0.929	0.964	0.949	0.903	0.941	0.933	0.874	0.968	0.950	0.905

scenes which are likely to give a better realization of different and wide-scale road environments and conditions. Accordingly, an accurate lane detection model is supposed to be able to cope with challenging driving conditions like the changes in illumination and the different lane shapes. Because the used dataset is ultimately made up of urban roads, all the categories can be considered challenging. For the scenes recorded in daylight, ResUNet++ can give out dice coefficient up to 98.3%, while in a more challenging context, it can give out values up to 98.9% and 97.2% for cloudy and rainy conditions, respectively. In the case of shadowy scenes, the models cannot detect lanes efficiently, as shown in the table, where dice coefficient values do not exceed 66.9%. In night scenarios, the performance is considered satisfactory rather than robust as ResUNet++ can give values up to 92.2% dice score. However, both SegNet and modified SegNet are not reliable to be used in night scenes as they give a relatively unsatisfactory prediction.

For the remaining three lane categories: dashed, curved, and yellow, the selected scenes are different from the used ones in the previously discussed lighting conditions. Also, these selected scenes do not include any shadows to avoid misclassifications. From Table 5, it is clear that ResUNet++ outperforms all the other models except for the curved lanes as U-Net gives better quantitative results with this lane orientation. Quantitatively, the results show robust performance as the average dice coefficient values on testing the models on all the lane categories and road conditions are 93.1%, 90.5%, 91.9%, 88.8%, and 88.6% for ResUNet++, ResUNet, U-Net, Modified SegNet and SegNet, respectively. From a visual aspect, according to Fig. 14, it is obvious that the quantitative results discussed earlier give out a realistic representation of the models' performance. However, the output images shown in Fig. 14 are samples from the testing phase; thus, the unpalatable realization may occur.

From the visual results and after benchmarking the state-of-the-art architectures quantitatively, we can conclude that relying on deep learning for the lane detection task is promising. Consequently, it is necessary to overcome the appearing limitation of inaccurate lane detection in the case of shadowy scenes. As each model separately performs a partially reliable prediction in shadowy scenes, we can generate ensemble predictions using the top three models: ResUNet++, ResUNet, and U-Net by averaging their output. Fig. 15 shows that we have sufficiently overcome the limitation of lane detection in shadowy scenes by merging the predictions of the most outstanding models. Another way to qualitatively test the robustness of our top models' ensemble segmentation is to obscure the scene by including some distorting and distracting elements and investigate their effect on the detection. These distorting objects are likely to exist in different dynamic road scenarios. They can be in form of a preceding vehicle in the host lane, parked vehicles, sidelong trees, or pedestrians. As shown in Fig. 16, the enhanced ensemble segmentation still shows robustness in different scenes with distorting elements.

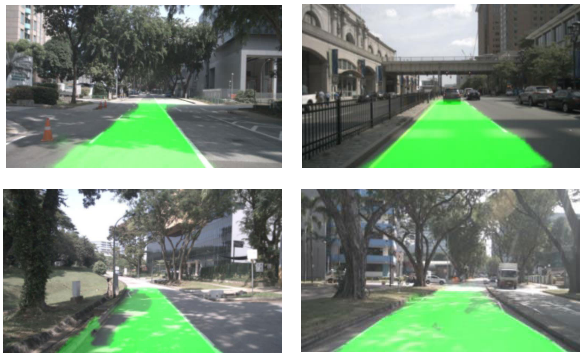


FIGURE 15. Sample from the ensemble prediction in the shadowy scenes.

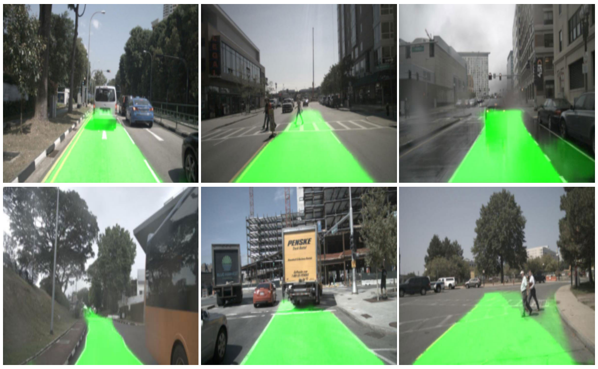


FIGURE 16. Sample results on testing the ensemble segmentation in different scenes with distorting objects.

E. COMPARISON WITH OTHER RELATED WORK

A comparative analysis will be carried out in this subsection concerning the lane detection task. In this paper, we have regarded the following: (i) Dealing with the host lane (ego-lane) detection as a semantic segmentation task; (ii) Using nuScenes dataset to be employed for the first time for the lane detection task; and (iii) Focusing on the complex and dynamic road scenes. Accordingly, it is challenging to find other related work covering all the mentioned concerns to compare our work with. However, we can compare our results with other related work by focusing on just the common concerns for lane detection. Wang *et al.* [15] proposed a framework that utilizes range and camera images along with OpenStreetMap for ego-lane detection in challenging scenarios with dynamic features. Chen and Chen [35] introduced RBNet to simultaneously detect road lanes, while a deep learning methodology for lane segmentation using up-convolutional networks was presented in [68]. In terms of dice coefficient (which is equivalent to F1-measure in the binary segmentation context), our method shows maximum dice coefficient of 98.9% while the maximum F1-measure (MaxF) was 93.56%, 90.54%, and 89.88% in [15], [35] and [68], respectively. In [16], authors considered the host lane detection based on the normal map in harsh road conditions. The average accuracy of their method reached nearly 94.51% under various scenarios, while the performance of

ResUNet++ in our approach reaches average accuracy of 96.55%. Cao *et al.* [12] proposed an automatic host lane detection algorithm in challenging road scenes based on traditional computer vision techniques. The authors used the accurate recognition rate that reached 99.15% as the main evaluation metric; however, the accurate recognition rate is 100% with our deep learning approach. The work in [12], [16], [69] and [49] have considered testing their proposed methodologies on different challenging road scenes, i.e., night, shadowy, urban, etc. Unfortunately, their output segmentation or detection considered only the lane lines while we segmented the host lane, which makes the quantitative comparison technically difficult.

V. CONCLUSION

This paper has developed a benchmarking framework for lane detection in complex road scenes with harsh environments and dynamic scenarios. This framework combines traditional computer vision and deep learning to employ the advantages of each of them. A proposed sequence of adaptive and optimized traditional computer vision techniques has been experimented to generate ground truth labels of the host lane. After developing the labels based on nuScenes dataset, which contains the needed challenging scenes, a visual qualitative assessment was done to validate their reliability before being utilized in the deep learning approach. By proceeding in this framework, lane detection is recognized as a semantic segmentation task of two classes: lane or non-lane. Hence, five state-of-the-art deep learning architectures were supervisely trained on this task relying on the developed data. SegNet, Modified SegNet, U-Net, ResUNet, and ResUNet++ are compared on the lane detection task based on quantitative evaluation and visual examination. On testing, the models show high performance, especially ResUNet++, under various challenging conditions. The ensemble segmentation proves its reliability to strengthen the lane detection in harsh scenarios like shadowy scenes and obscured road perception. The overall experimental results give a promising intuition about the reliability and robustness of semantic segmentation for the lane detection task.

REFERENCES

[1] I. Yaqoob, L. U. Khan, S. M. A. Kazmi, M. Imran, N. Guizani, and S. C. Hong, "Autonomous driving cars in smart cities: Recent advances, requirements, and challenges," *IEEE Netw.*, vol. 34, no. 1, pp. 174–181, Jan./Feb. 2020.

[2] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognit.*, vol. 73, pp. 216–234, Jan. 2018.

[3] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2584–2601, Oct. 2017.

[4] F.-Y. Wang and S.-M. Tang, "Concepts and frameworks of artificial transportation systems," *Complex Syst. Complex. Sci.*, vol. 1, no. 2, pp. 52–59, 2004.

[5] Y. Xing, C. Lv, H. Wang, H. Wang, Y. Ai, D. Cao, E. Velenis, and F. Y. Wang, "Driver lane change intention inference for intelligent vehicles: Framework, survey, and challenges," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4377–4390, May 2019.

[6] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Veh. Commun.*, vol. 20, Dec. 2019, Art. no. 100184.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[8] P.-Y. Hsiao and C.-W. Yeh, "A portable real-time lane departure warning system based on embedded calculating technique," in *Proc. IEEE 63rd Veh. Technol. Conf.*, May 2006, pp. 2982–2986.

[9] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 7–12.

[10] M. B. de Paula and C. R. Jung, "Automatic detection and classification of road lane markings using onboard vehicular cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3160–3169, Dec. 2015.

[11] Y. Li, A. Iqbal, and N. R. Gans, "Multiple lane boundary detection using a combination of low-level image features," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1682–1687.

[12] J. Cao, C. Song, S. Song, F. Xiao, and S. Peng, "Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments," *Sensors*, vol. 19, no. 14, p. 3166, Jul. 2019.

[13] V. Gaikwad and S. Lokhande, "Lane departure identification for advanced driver assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 910–918, Apr. 2015.

[14] A. Bosaghzadeh and S. S. Routeh, "A novel PCA perspective mapping for robust lane detection in urban streets," in *Proc. Artif. Intell. Signal Process. Conf. (AISP)*, Oct. 2017, pp. 145–150.

[15] X. Wang, Y. Qian, C. Wang, and M. Yang, "Map-enhanced ego-lane detection in the missing feature scenarios," *IEEE Access*, vol. 8, pp. 107958–107968, 2020.

[16] C. Yuan, H. Chen, J. Liu, D. Zhu, and Y. Xu, "Robust lane detection for complicated road environment based on normal map," *IEEE Access*, vol. 6, pp. 49679–49689, 2018.

[17] J. Son, H. Yoo, S. Kim, and K. Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Syst. Appl.*, vol. 42, pp. 1816–1824, Oct. 2014.

[18] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 309–318, Dec. 2004.

[19] R. Jiang, M. Terauchi, R. Klette, S. Wang, and T. Vaudrey, "Low-level image processing for lane detection and tracking," in *Proc. Int. Conf. Arts Technol.*, 2009, pp. 190–197.

[20] B.-S. Shin, J. Tao, and R. Klette, "A superparticle filter for lane detection," *Pattern Recognit.*, vol. 48, no. 11, pp. 3333–3345, 2015.

[21] S. Xu, P. Ye, S. Han, H. Sun, and Q. Jia, "Road lane modeling based on RANSAC algorithm and hyperbolic model," in *Proc. 3rd Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2016, pp. 97–101.

[22] T.-Y. Sun, S.-J. Tsai, and V. Chan, "HSI color model based lane-marking detection," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2006, pp. 1168–1172.

[23] T.-T. Tran, C.-S. Bae, Y.-N. Kim, H.-M. Cho, and S.-B. Cho, "An adaptive method for lane marking detection based on HSI color model," in *Proc. Int. Conf. Intell. Comput.*, 2010, pp. 304–311.

[24] X. Du and K. K. Tan, "Comprehensive and practical vision system for self-driving vehicle lane-level localization," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2075–2088, May 2016.

[25] M. Haloi and D. B. Jayagopi, "A robust lane detection and departure warning system," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 126–131.

[26] Y. Kortli, M. Marzougui, B. Bouallegue, J. S. C. Bose, P. Rodrigues, and M. Atri, "A novel illumination-invariant lane detection system," in *Proc. 2nd Int. Conf. Anti-Cyber Crimes (ICACC)*, Mar. 2017, pp. 166–171.

[27] A. Mammeri, A. Boukerche, and G. Lu, "Lane detection and tracking system based on the MSER algorithm, Hough transform and Kalman filter," in *Proc. 17th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst. (MSWiM)*, 2014, pp. 259–266.

[28] A. Borkar, M. Hayes, and M. T. Smith, "Polar randomized Hough transform for lane detection using loose constraints of parallel lines," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 1037–1040.

[29] Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 570–579, Dec. 2008.

[30] W. Lu, E. Seignez, F. S. A. Rodriguez, and R. Reynaud, "Lane marking based vehicle localization using particle filter and multi-kernel estimation," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 601–606.

[31] Z. Nan, P. Wei, L. Xu, and N. Zheng, "Efficient lane boundary detection with spatial-temporal knowledge filtering," *Sensors*, vol. 16, no. 8, p. 1276, 2016.

[32] K. Zhao, M. Meuter, C. Nunn, D. Muller, S. Muller-Schneiders, and J. Pauli, "A novel multi-lane detection and tracking system," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 1084–1089.

[33] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-snake," *Image Vis. Comput.*, vol. 22, no. 4, pp. 269–280, Apr. 2004.

[34] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, Mar. 2008.

[35] Z. Chen and Z. Chen, "RBNet: A deep neural network for unified road and road boundary detection," in *Proc. Int. Conf. Neural Inf. Process.*, 2017, pp. 677–687.

[36] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and Kalman filter," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 3261–3264.

[37] J. Deng and Y. Han, "A real-time system of lane detection and tracking based on optimized RANSAC B-spline fitting," in *Proc. Res. Adapt. Convergent Syst. (RACS)*, 2013, pp. 157–164.

[38] Z. Lu, Y. Xu, X. Shan, L. Liu, X. Wang, and J. Shen, "A lane detection method based on a ridge detector and regional G-RANSAC," *Sensors*, vol. 19, no. 18, p. 4028, Sep. 2019.

[39] P. V. Ingale and K. Bhaga, "Comparative study of lane detection techniques," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 4, no. 5, pp. 381–390, 2016.

[40] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng, "An empirical evaluation of deep learning on highway driving," 2015, *arXiv:1504.01716*. [Online]. Available: http://arxiv.org/abs/1504.01716

[41] B. He, R. Ai, Y. Yan, and X. Lang, "Accurate and robust lane detection based on dual-view convolutional neutral network," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 1041–1046.

[42] B. He, R. Ai, Y. Yan, and X. Lang, "Lane marking detection based on convolution neural network from point clouds," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2475–2480.

[43] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 690–703, Mar. 2017.

[44] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 286–291.

[45] Q. Zou, H. Jiang, Q. Dai, and Y. Yue, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Mar. 2019.

[46] F. Chao, S. Yu-Pei, and J. Ya-Jie, "Multi-lane detection based on deep convolutional neural network," *IEEE Access*, vol. 7, pp. 150833–150841, 2019.

[47] Z. Tang, Y.-S. Lin, K.-H. Lee, J.-N. Hwang, and J.-H. Chuang, "ESTHER: Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans," *IEEE Access*, vol. 7, pp. 10754–10766, 2019.

[48] F. Zhou, Y. Cui, H. Gao, and Y. Wang, "Line-based camera calibration with lens distortion correction from a single image," *Opt. Lasers Eng.*, vol. 51, no. 12, pp. 1332–1343, Dec. 2013.

[49] M. Marzougui, A. Alasiry, Y. Kortli, and J. Baili, "A lane tracking method based on progressive probabilistic Hough transform," *IEEE Access*, vol. 8, pp. 84893–84905, 2020.

[50] W. Lu, Y. Zheng, Y. Ma, and T. Liu, "An integrated approach to recognition of lane marking and road boundary," in *Proc. 1st Int. Workshop Knowl. Discovery Data Mining (WKDD)*, Jan. 2008, pp. 649–653.

[51] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic Hough transform," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 119–137, Apr. 2012.

[52] R. O. Duda and R. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[53] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A probabilistic Hough transform," *Pattern Recognit.*, vol. 24, no. 4, pp. 303–316, 1991.

[54] K. Wang, B. Fang, J. Qian, S. Yang, X. Zhou, and J. Zhou, "Perspective transformation data augmentation for object detection," *IEEE Access*, vol. 8, pp. 4935–4943, 2020.

[55] R. Muthalagu, A. Bolimera, and V. Kalaichelvi, "Lane detection technique based on perspective transformation and histogram analysis for self-driving cars," *Comput. Electr. Eng.*, vol. 85, Jul. 2020, Art. no. 106653.

[56] L. Chen, Q. Li, Q. Mao, and Q. Zou, "Block-constraint line scanning method for lane detection," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 89–94.

[57] S. Samantaray, R. Deotale, and C. L. Chowdhary, "Lane detection using sliding window for intelligent ground vehicle challenge," in *Proc. Innov. Data Commun. Technol. Appl.*, 2021, pp. 871–881.

[58] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[59] A. Adachi and T. Gonsalves, "White lane detection using semantic segmentation," in *Proc. 2nd High Perform. Comput. Cluster Technol. Conf. ZZZ (HPCCT)*, 2018, pp. 24–26.

[60] G. M. Gad, A. M. Annaby, N. K. Negied, and M. S. Darweesh, "Real-time lane instance segmentation using SegNet and image processing," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Oct. 2020, pp. 253–258.

[61] Q. Wang, F. Chen, and X. Liang, "Lane detection: A semantic segmentation approach," in *Proc. Int. Conf. Comput. Intell. Syst. Netw. Remote Control (CISNRC)*, 2020.

[62] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[63] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*, 2015, pp. 234–241.

[64] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet—A: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS J. Photogramm. Remote Sens.*, vol. 162, pp. 94–114, Apr. 2020.

[65] D. Jha, P. H. Smedsrud, M. A. Riegler, D. Johansen, T. D. Lange, P. Halvorsen, and H. D. Johansen, "ResUNet++: An advanced architecture for medical image segmentation," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2019, pp. 225–2255.

[66] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, Oct. 2020, pp. 1–7.

[67] K. Xia, H. Yin, P. Qian, Y. Jiang, and S. Wang, "Liver semantic segmentation algorithm based on improved deep adversarial networks in combination of weighted loss function on abdominal CT images," *IEEE Access*, vol. 7, pp. 96349–96358, 2019.

[68] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4885–4891.

[69] W. Liu, F. Yan, K. Tang, J. Zhang, and T. Deng, "Lane detection in complex scenes based on end-to-end neural network," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 4300–4305.

RETAJ YOUSRI was born in Cairo, Egypt, in 1997. She received the B.Sc. degree in biomedical engineering from Helwan University, Cairo, in 2020. She is currently pursuing the M.Sc. degree in micro-electronics system design with the School of Engineering and Applied Sciences, Nile University, Giza, Egypt. She is currently working as a Research Assistant at the Wireless Intelligent Networks Center (WINC), Nile University. Her research interests include signal processing, image processing, computer vision, and neuroscience.

MUSTAFA A. ELATTAR was born in Cairo, Egypt, in 1986. He received the bachelor's degree in systems and biomedical engineering from Cairo University, Cairo, in 2008, the master's degree in communication and information technology in 2010, after conducting image analysis research for cardiac imaging, and the Ph.D. degree in biomedical engineering and physics, medicine from the Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands, in 2016, after developing preoperative planning framework for transcatheter aortic valve implantation. He joined the Medical Imaging and Image Processing Research Group, Nile University, Giza, Egypt, as a Research Assistant. In 2015, he joined the Netherlands Cancer Institute (NKI) as a Postdoctoral Fellow, where he conducted research for image-guided radiotherapy. In August 2017, he joined Nile University as an Assistant Professor. Also, he is currently leading the Medical Imaging and Image Processing Research Group focusing on incorporating deep neural networks in 2-D and 3-D medical image analysis contexts. He has more than 25 journal articles and conference publications. On the professional side, he worked in Research and Development Division, Diagnosoft Inc., 3mensio B.V., PieMedical N.V., and Myocardial Solutions Inc. In August 2018, he has founded and is currently leading Intixel Company S.A.E., which focuses on serving medical imaging solution firms with its turnkey artificial intelligence solutions that achieve their business needs.

M. SAEED DARWEESH (Senior Member, IEEE) received the master's and Ph.D. degrees (Hons.) in electronics and electrical communications engineering from the Faculty of Engineering, Cairo University, Giza, Egypt, in 2013 and 2017, respectively. He is currently a full-time Assistant Professor with the ECE Program, School of Engineering and Applied Sciences, Nile University, Egypt. He is an Expert with the Phi Science Institute. Also, he is a former Adjunct Assistant Professor at American University in Cairo (AUC), Zewail City (ZC) of Science and Technology, and Institute of Aviation Engineering and Technology (IAET). He is a PI of different research projects funded by the Information Technology Industry Development Agency (ITIDA), a Co-PI, and a Research Associate in seven research projects funded by different agencies like the Science and Technology Development Fund (STDF), National Telecom Regulatory Authority (NTRA), and the Academy of Scientific Research and Technology (ASRT). He has a solid technical background with a keen interest in machine learning and artificial intelligence. He has H-index 4 reported by Scopus with total citations of 64, with more than 28 publications distributed between high-impact journals, conferences, and book chapters. He has a research stay at the American University in Cairo (AUC), Zewail City (ZC) of Science and Technology, Faculty of Engineering, and National Institute of Laser Enhanced Sciences (NILES), Cairo University. He worked in many telecom operators and suppliers like (Orange, Alcatel.Lucent, and Geniprocess) and has a strong computer network and security background. His research interests focus on narrow-band IoT, autonomous driving vehicle to vehicle (V2V) systems, wireless communications, biomedical engineering (EEG seizure detection, sleepiness detection using EEG, and breast cancer classification), and data compression.