IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 August 2022, accepted 6 September 2022, date of publication 20 September 2022,
date of current version 29 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208128

SURVEY

Pedestrian Lane Detection for Assistive Navigation of Vision-Impaired People: Survey and Experimental Evaluation

YUNJIA LEI[1], SON LAM PHUNG[1], (Senior Member, IEEE), ABDESSELAM BOUZERDOUM[1,2], (Senior Member, IEEE), HOANG THANH LE[1,3], AND KHOA LUU[4]

[1]School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
[2]Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Ar Rayyan, Qatar
[3]Faculty of Information Technology, Nha Trang University, Nha Trang 650000, Vietnam
[4]Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, AR 72701, USA

Corresponding author: Son Lam Phung (phung@uow.edu.au)

This work was supported by the Discovery Project DP190100607 titled "*Assistive Micro-navigation for Vision Impaired People*" from the Australian Research Council, and a matching Ph.D. scholarship from the University of Wollongong. The work of Son Lam Phung and Abdesselam Bouzerdoum is also supported in part by grants from the Queensland Department of Transport and Main Roads (under the FrontierSI TMR Spatial Labs), the New South Wales (NSW) Space Research Network (under the Pilot Research Project scheme), the NSW Defence Innovation Network (under the PhD Project scheme), and the NSW Government (under the Tech Voucher scheme).

ABSTRACT Pedestrian lane detection is a crucial task in assistive navigation for vision-impaired people. It can provide information on walkable regions, help blind people stay on the pedestrian lane, and assist with obstacle detection. An accurate and real-time lane detection algorithm can improve travel safety and efficiency for the visually impaired. Despite its importance, pedestrian lane detection in unstructured scenes for assistive navigation has not attracted sufficient attention in the research community. This paper aims to provide a comprehensive review and an experimental evaluation of methods that can be applied for pedestrian lane detection, thereby laying a foundation for future research in this area. Our study covers traditional and deep learning methods for pedestrian lane detection, general road detection, and general semantic segmentation. We also perform an experimental evaluation of the representative methods on a large benchmark dataset that is specifically created for pedestrian lane detection. We hope this paper can serve as an informative guide for researchers in assistive technologies, and facilitate urgently-needed research for vision-impaired people.

INDEX TERMS Pedestrian lane detection, assistive navigation, vision impairment, semantic segmentation, deep networks.

I. INTRODUCTION

According to the World Health Organization (WHO), there are about 253 million visually impaired people worldwide, of whom 217 million have moderate or severe vision impairments, and 36 million are blind [1]. Various studies have shown that visual impairment causes a significant reduction in mobility [2], and a higher risk of falling or collision [3], [4]. Due to their reduced capability in scene perception, blind people have tremendous difficulties navigating unfamiliar environments.

Traditional walking aids for the visually impaired include white canes and guide dogs. White canes are simple to use, but they have short detection ranges [5], [6]. Guide dogs can evade obstacles and memorize routes, but they require extra training and care, and are effective primarily in familiar environments [5], [7]. Hence, there is a growing need to develop assistive navigation systems that extend beyond the

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

traditional capabilities. In these assistive systems, pedestrian lane detection is a core component that enables vision-impaired users to walk within the pedestrian lane and maintain their balance. An accurate, reliable, and real-time algorithm for pedestrian lane detection can significantly improve the mobility and safety of visually impaired people.

Despite its importance for assistive navigation, pedestrian lane detection has not attracted much interest in the research community. This survey paper aims to set a foundation for assistive navigation research by reviewing and assessing the applicable methods, including general road detection and semantic segmentation methods. The design principles and performances of these methods on a benchmark pedestrian lane detection dataset are informative resources when developing new methods.

Note that methods designed for vehicle road detection are not optimized for pedestrian lane detection, albeit the two tasks have some similarities. First, compared to pedestrian lanes, vehicle roads (especially in urban environments) are usually more structured. For example, vehicle roads often have clear boundaries and asphalt surfaces, whereas pedestrian lanes usually have arbitrary shapes and various surface textures (e.g., bricks, concrete, tiles, grass, sand, and soil). Second, vehicle road detection methods mainly deal with outdoor situations, whereas pedestrian lane detection methods need to consider also indoor scenes. Hence, although road detection methods can be used to detect pedestrian lanes, they are not the complete solution for pedestrian lane detection.

This paper presents a comprehensive review and an experimental evaluation of methods that can be applied for pedestrian lane detection. The survey aims to serve as an informative guide for researchers in assistive technologies and facilitate urgently-needed research for vision-impaired people. The main contributions of this paper can be highlighted as follows:

1) We provide a comprehensive review and analysis of the traditional and deep learning methods that can be applied for pedestrian lane detection. The traditional methods include color-based approaches, border-based approaches, and combined approaches. The state-of-the-art (SOTA) deep learning methods include deep networks for road detection and general semantic segmentation. A summary of the representative methods is presented in Table 1. A timeline of the representative methods is shown in Fig. 1.
2) We conduct an extensive performance evaluation of the representative methods on a large Pedestrian Lane and Vanishing Point detection (PLVP3) dataset[1]. To date, it is the largest pedestrian lane dataset in the literature. This evaluation provides baseline performances on pedestrian lane detection and allows practitioners to focus on the promising directions.
3) We discuss the technical challenges and future research directions in pedestrian lane detection to bridge the gaps towards a practical assistive navigation system.

The remainder of this paper is organized as follows. Section II reviews the traditional detection methods that are based on hand-crafted features. Section III reviews road segmentation methods that are based on deep neural networks. Section IV presents experimental evaluations of the major methods on the PLVP3 dataset. Section V discusses the technical challenges and future directions for pedestrian lane detection. Section VI gives the concluding remarks.

II. TRADITIONAL METHODS

This section reviews representative feature-based methods, which are categorized into three different groups: (i) color-based approaches (Section II-A); (ii) border-based approaches (Section II-B); and (iii) combined approaches using both road color and border features (Section II-C).

A. COLOR-BASED APPROACHES

Color-based approaches classify image pixels by comparing each pixel to a reference color model. The reference color model can be constructed using different color spaces [8], [9], [10], [11], [20].

In [8], Crisman and Thorpe proposed a road detection method called SCARF. This method constructs color models as multiple Gaussian distributions in the red-green-blue (RGB) color space for both road and off-road classes. First, regions corresponding to road and the background in the previous frame are selected to construct color models for the current frame. Next, each region is clustered into four homogeneous color groups. Then, four Gaussian distributions are generated for each class from the color groups. Finally, two color models are constructed to segment road and background regions. The road location in the first frame needs to be defined manually or by another algorithm (e.g., UNSCARF [20]). Because color models are represented by multiple Gaussian distributions, this method can cope with variations in road colors and textures. However, it relies heavily on the continuity of adjacent frames, which may produce errors if there are sudden changes between two frames.

In [11], Ceryen *et al.* proposed a road detection algorithm that uses color histograms to represent road models in the normalized red and green color space. Compared to the standard RGB color space, the normalized space can cope better with illumination variations. This method assumes that the center-bottom part of an input image is a homogeneous road region. Accordingly, the sample region is defined as a rectangle at the center-bottom part of the input image. For each frame, one color distribution is generated from the pixels in the sample region. The final road model is represented by four color distributions generated over time from different frames. Since this method considers the frame continuity with multiple color distributions, it improves the detection stability

[1] https://documents.uow.edu.au/ phung/plvp3.html

TABLE 1. A summary of representative methods for pedestrian lane detection.

Category	Method based on	Authors	Year	Technique
Traditional: Color-based approach	Road models	Crisman *et al.* [8]	1993	RGB color space; Multiple Gaussian distributions
		Sotelo *et al.* [9]	2004	HSI color space; Intensity and chromatic distances
		Ramstrom *et al.* [10]	2005	UV, normalized R and G, and intensity channels; GMMs
	Sample lane regions	Tan *et al.* [11]	2006	Normalized R and G color space; Color histograms
		Alvarez *et al.* [12]	2011	Illuminant-invariant feature space; Normalized histogram
Traditional: Border-based approach	Lane markers	Le *et al.* [7]	2012	Patches of interest
	Edge features	Yu *et al.* [13]	1997	Hough Transform; Canny edge detectors
		Viosin *et al.* [14]	2005	Hough Transform; Sobel filters
		Chen *et al.* [15]	2011	Gradient-enhanced images
		Yoo *et al.* [16]	2013	Gradient direction features
	Vanishing points	Rasmussen *et al.* [17]	2004	Color and texture features
		Kong *et al.* [18]	2010	Orientation consistency ratio features
		Le *et al.* [19]	2014	Obstacle detection; Thresholding
Traditional: Combined approach	Lane templates	Crisman *et al.* [20]	1991	Clustering; Edge features; Lane templates
	Vanishing points	He *et al.* [21]	2004	Sample regions; Lane width assumption
		Miksik *et al.* [22]	2011	Trapezoidal sample regions; History of road models
		Chang *et al.* [23]	2012	Color branch and border branch; Kalman filter
		Phung *et al.* [24]	2016	Sample regions; Local orientations; Lane templates
DL: Lane detection approach	Segmentation networks	Bianco *et al.* [25]	2020	ERFNet
		Cao *et al.* [26]	2021	Depthwise separable convolutions; Atrous spatial pyramid pooling modules
	Uncertainty maps	Nguyen *et al.* [27]	2020	SegNet; Hierarchical Gaussian process classifier
		Le *et al.* [28]	2022	Bayersian Gabor layers; Variational Bayesian inference
	Boundary and color information	Yadav *et al.* [29]	2015	SegNet; Color-line models in CRFs
		Zhang *et al.* [30]	2018	Multi-task; Geometric constrains
		Almedia *et al.* [31]	2020	ENet and LaneNet; Multi-task; Lane boundaries
	NAS	Ang *et al.* [32]	2021	Network-level searching space; Gradient descent algorithm
DL: Generic semantic segmentation	Fully convolutional structure	Long *et al.* [33]	2015	FCN: Fully convolutional network
	Encoder-decoder structure	Ronneberger *et al.* [34]	2015	U-Net: E-D structure; Feature map concatenation
		Badrinarayanan *et al.* [35]	2017	SegNet: E-D structure; Pooling indices
	Multiscale and pyramid structure	Lin *et al.* [36]	2017	FPN: Feature pyramid; Lateral connections
		Zhao *et al.* [37]	2017	PSPNet: Pyramid pooling module
	Dilated convolution	Chen *et al.* [38]	2018	DeepLabv3+: Atrous separable convolutionl
	Attention mechanism	Fu *et al.* [39]	2019	Dual attention network: Self-attention mechanism
		Tao *et al.* [40]	2020	Hierarchical multiscale attention

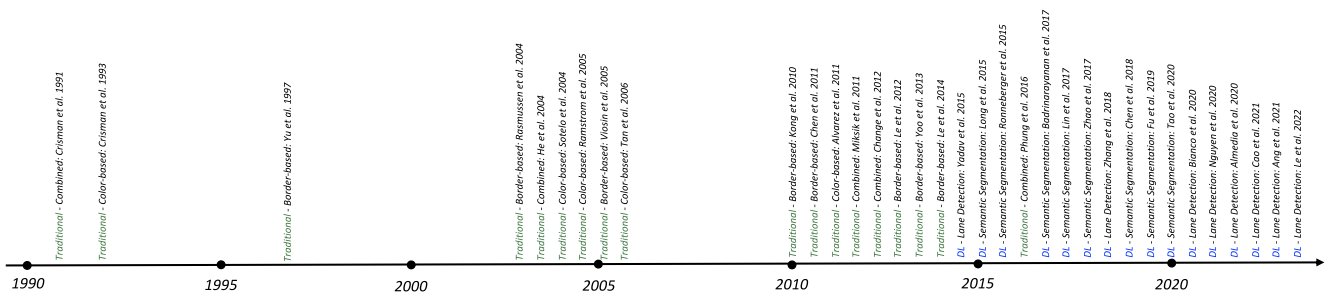

FIGURE 1. A timeline of representative methods that can be used for pedestrian lane detection.

over frames. However, it is not effective when sample regions contain multiple road colors.

In [9], Sotelo *et al.* introduced a road model with the hue-saturation-intensity (HSI) color space. This method divides pixels into chromatic pixels and achromatic pixels. Chromatic pixels are classified using both intensity and chromatic information, whereas achromatic pixels (extreme intensity or low saturation) are classified using intensity only. To classify a pixel, a threshold value is defined as a function of two parameters: (i) the distance between the pixel and the previously-predicted road center, and (ii) the maximum threshold value of the previous frame. The initial

road-center model is selected from seven predefined models with an assumption of the road width. To cope with the extreme illumination variations, this method reconsiders non-road pixels within predicted road edges based on their intensity and chromatic features. The main limitation of this method is that it builds color models from a small number of randomly-selected pixels near the road center, which may not efficiently represent the entire road surface.

In [10], Ramstrom and Christensen employed the Gaussian Mixture Model (GMM) to represent road and non-road classes with three different color-based feature vectors: UV (of the HUV color space), normalized red and green, and intensity channels. For each feature vector, two GMMs are generated. The GMMs for the road are constructed from pixels within road regions. Here, road regions are determined from a road shape model (centerline of the road) and a road-width parameter. The road shape model is continually updated based on each new GMM. The initial road model is selected based on assumptions of road shape and road width. In summary, this method uses a simple color model and a simple road shape model, and it therefore does not cope well with complex road shapes, e.g. S-shaped lanes or intersections.

In [12], Alvarez and Ĺopez presented road models from illumination-invariant images to address the lighting variations. These images are computed from input RGB images with a camera parameter, called illumination-invariant angle. This method models the road as a normalized histogram from a set of seeds in the center-bottom part of the input image, see Fig. 2. Since the road histogram is built from a few seeds within the sample regions, this method could be ineffective for road regions with non-homogeneous surface textures.

FIGURE 2. Example of using the normalized histogram as a road color model. Figure is from [12].

The above color-based methods can cope with the variations in road shapes and illumination conditions, but their generalization capability relies heavily on the assumptions of road surfaces and road locations. For example, methods proposed in [8], [9], and [10] assume that there are no sudden changes in image sequences, so they build color models from previous predictions. These methods also require further assumptions of road widths for the initial frame. A few methods build color models from the center-bottom part of the input image with the assumption of potential road locations [11], [12].

B. BORDER-BASED APPROACHES

Border-based approaches detect the regions of interest using either lane markers [7], edge features [13], [14], [15], [16], or vanishing points [17], [18], [19]. In [7], Le *et al.* employed lane markers to detect pedestrian lanes at traffic junctions. This method extracts patches of interest on lane marker edges using normalized cross-correlation template matching. The lane markers are then detected using the random sample consensus (RANSAC). The lane regions are segmented from the pair of lane markers according to geometric constraints. This method works well with clear lane markers, but often fails when the markers are occluded or when lane surfaces are under strong shadows.

To detect unmarked lanes, several studies have employed either Hough Transform (HT) [13], [14] or image gradients [15], [16]. In [14], Voisin *et al.* extracted the edge points using Sobel filters. The HT is then used to detect two lane boundaries within the regions of interest predicted by a Kalman filter. This method considers lane borders as two straight lines. In [13], Yu and Jain used the HT to detect lane boundaries from edge images generated by Canny detectors. Unlike [14], this method handles curved borders by applying the HT to multiple image resolutions. In [16], Yoo *et al.* generated gradient-enhanced images from RGB images to detect lane regions. This method also uses Canny edge detectors and the HT to extract lane edges. In [15], Chen *et al.* used gradient direction features to enhance lane features in input images (Fig. 3). This method assumes that the two lane boundaries are approximately parallel.

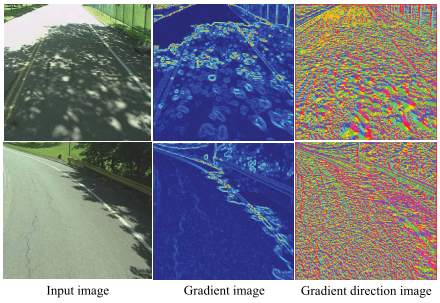

FIGURE 3. Example of gradient direction features. Figure is adapted from [15].

The above methods can accurately detect simple roads with clear boundaries and structured scenes, but they are not effective in coping with occlusions, degraded lane edges, or atypical road shapes. This is because road models used to match lanes are simplified, and the performance of these methods depends highly on the clear road features.

To overcome this problem, several methods have employed vanishing points to determine lane boundaries [17], [18], [19]. In [17], Rasmussen selected lane borders from the edges pointing to the vanishing point. The edges are ranked based on an objective function that measures texture and color differences between lanes and the background. This method is only effective when lane regions do not significantly differ from non-lane regions in terms of color and texture. In [18], Kong *et al.* used vanishing points to detect lane edges, which are then ranked using an orientation consistency

ratio feature. Because this method only relies on edges for lane-border detection, it is sensitive to background edges. In [19], Le *et al.* determined lane regions from vanishing points and road obstacles (pedestrians). The optimal boundary pair is selected by a threshold determined from the training set. However, this method may fail to detect complex road shapes because it models lane boundaries as straight lines.

In summary, the vanishing-point-based methods can cope with degraded lane edges, occlusions, and texture variations of lane surfaces. The main limitation of these methods is that they can detect only simple road shapes such as straight lines or curves with one arc. Complex road shapes may lead to poor performances.

C. COMBINED APPROACHES

To address the aforementioned limitations, several methods combine color and border features by: (i) matching color segmentation results with lane templates [20], (ii) combining color segmentation with border detection [23], or (iii) using extracted sample regions for color model construction [21], [22], [24], [41], [42]. In [20], Crisman and Thorpe first clustered image pixels into homogeneous color regions using a modified ISODATA algorithm. Next, the edges of each group are extracted by removing small regions (noise). Finally, lane regions with the highest probability are selected by matching with lane templates. Edges of the selected regions are defined as lane borders. Although this method can handle road surface variations, it is unable to handle complex road scenes due to the use of predefined lane templates.

In [23], Chang *et al.* estimated road regions using color features and boarder features separately. The final results are produced by combining information from the two branches. An illustration of this method is shown in Fig. 4. In the color feature branch, the input image is first segmented into homogeneous color regions. A rectangle at the center-bottom part of the first frame is then selected. The largest color area within the sample region is used to construct color models in the RGB color space. From the second frame, the sample regions are selected from the rectangle at the center-bottom part of the previously predicted road regions. In the boundary branch, the road regions within road borders are determined by the vanishing points and a boundary score. In the final stage, the results from the two branches are combined using a Kalman filter. The output segmentation map is also used as feedback for updating the road models in the two branches. This method uses a floating window for sample region extraction, which is more efficient in coping with road location changes. However, because the sample region of the first frame is extracted from the center-bottom part of the image, this method often fails if the road center of the first frame is far from the image center. Furthermore, it is sensitive to sudden changes between frames.

In [22], Miksik *et al.* constructed road models as GMMs from sample regions in the RGB color space. Firstly, the sample region is initialized as a trapezoid at the center-bottom part of the input image. Then, it is refined by the vanishing

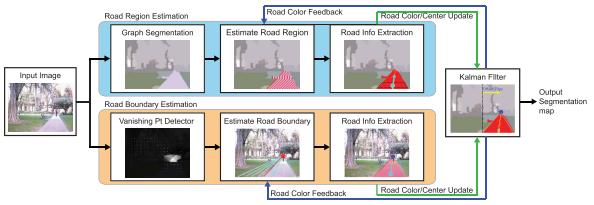

FIGURE 4. Illustration of the road recognition method that combines a color branch for road region estimation with a boundary branch for road border estimation. Figure is from [23].

point. To construct road models, input images are clustered into homogeneous color regions. One Gaussian distribution is generated for each homogeneous pixel group in the sample region. One GMM represents the road colors in one video frame. A fixed number of GMMs from different frames are stored for lane detection. This method can cope with various road shapes and road surface textures. However, because the sample regions are refined by the lines connecting the vanishing points and two predefined points at the image bottom, this method cannot cope well with road regions far from the image center.

In [21], He *et al.* generated color models as Gaussian distributions from the pixels within estimated road boundaries. First, edge images are generated by applying edge detectors on projection images of lanes. Next, pseudo road boundaries are determined from edge images using vanishing points and eight curvature models. The pseudo road boundaries are much narrower than the real boundaries, which ensures that all pixels within these boundaries belong to the road class. Finally, color models constructed from these pixels are used to segment the real lane areas. Due to the assumption of the predefined curvature models, this model can detect only a few types of road structures.

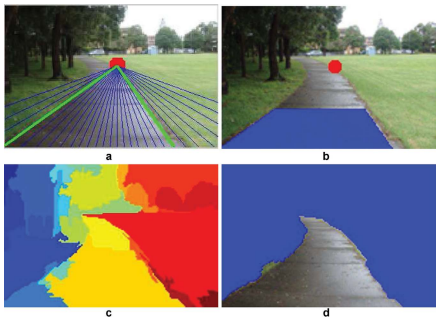

FIGURE 5. Illustration of the pedestrian lane detection method: (a) the imaginary rays (blue lines) and the detected borders (green lines) from the vanishing point; (b) lane sample region; (c) color homogeneous sub-regions; (d) segmented walking lane. Figure is from [24].

In [24], Phung *et al.* represented road color models in the RGB color space and an illumination-invariant space. An illustration of this method is shown in Fig. 5. The color models are generated from the sample regions that are selected as the lower half of the area within lane boundaries (a trapezoidal region). The lane boundaries are selected

from imaginary rays pointing to the vanishing points by a lane score that considers the texture and geometric features of the lane. The vanishing points are detected by applying color tensors and a Canny edge detector. Before applying color models to detect road regions, the input images are first segmented into homogeneous color sub-regions. This method combines geometric and color features of the lane, so it can handle various road shapes. However, because the detection is based on homogeneous color regions, it may fail if the scene background has similar colors to road surfaces.

In summary, the combined methods reduce the need for prior knowledge, and thereby increase the model's generalization ability in terms of various deformation and occlusion conditions.

III. DEEP LEARNING METHODS

A. ROAD DETECTION METHODS

In recent years, deep convolutional neural networks (CNNs) have been applied successfully to semantic segmentation. They have achieved SOTA performances because of their ability to learn from large-scale image datasets and extract salient visual features automatically. Many lane segmentation methods have been proposed based on deep CNNs, and they have achieved promising results.

Several lane detection methods have employed semantic segmentation networks directly for road detection. In [25], Bianco *et al.* proposed a network trained on two separate datasets: a lane detection dataset and an obstacle detection dataset. ERFNet, proposed in [43], is employed in this method to perform both weak labeling generation and the final segmentation. Because ERFNet has a good trade-off between accuracy and inference speed, it is suitable for real-time pedestrian lane detection. However, this method cannot cope with extreme illumination or weather conditions due to the lack of training examples. In [26], Cao *et al.* proposed a lightweight segmentation network for blind people. This network utilizes depthwise separable convolutions to increase computation efficiency, and densely connected atrous spatial pyramid pooling modules to enhance multiscale and contextual information. However, this method is proposed only for blind roads and crosswalks, which typically have structured shapes and fewer variations in appearance. As a result, this method might not generalize well for other types of pedestrian lanes.

Other methods have been proposed to generate the output segmentation map together with an uncertainty map for enhanced safety of blind users. In [27], Nguyen *et al.* proposed a DL-HGP network, which combines the SegNet encoder-decoder network (proposed in [35]) with a hierarchical Gaussian process classifier. The HGP classifier produces a segmented lane map and a calibrated uncertainty map, which provides extra safety for visually impaired users, see Fig. 6. However, this network cannot reach real-time prediction due to the computation requirements of the HGP classifier. In [28], Le *et al.* proposed a Bayesian Gabor Network (BGN) that generates a segmentation map with two calibrated uncertainty maps. This network contains 13 Bayesian Gabor layers, where each Gabor parameter is represented as a learnable Gaussian distribution. By using the Gabor layers instead of the standard CNN layers, this method achieves high prediction accuracy and real-time segmentation with a small network size. However, the parametric form of the Gabor filters has a reduced representation power for complex lane textures.

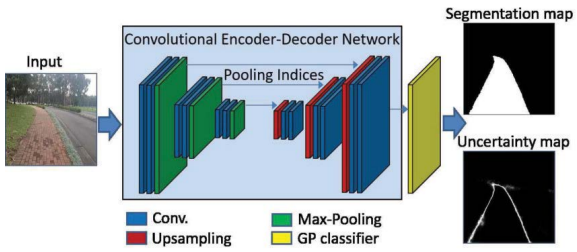

FIGURE 6. Network structure of the DL-HGP method. Figure is from [27].

Several methods have been proposed to improve segmentation results using road boundary features. In [31], Almeida *et al.* combined the results from two separately-trained models: ENet [44] for lane segmentation, and LaneNet [45] for lane boundary detection. A post-processing step is applied to create the final segmentation masks from the detected lane boundaries by the LaneNet. This method then computes a weighted sum of the output segmentation masks from each model. Higher weights are given to regions whose shapes are more similar to a typical road (usually a trapezoid). However, the performance of this method may be severely affected if one model produces imprecise predictions. In [29], Yadav *et al.* proposed a conditional random field (CRF) framework, in which the segmentation masks produced by SegNet is used as prior knowledge to create two color lines models, one for road and the other for the background. The above methods apply boundary features to refine segmentation results, but they only use these features in the last few stages and therefore may not fully utilise the road features.

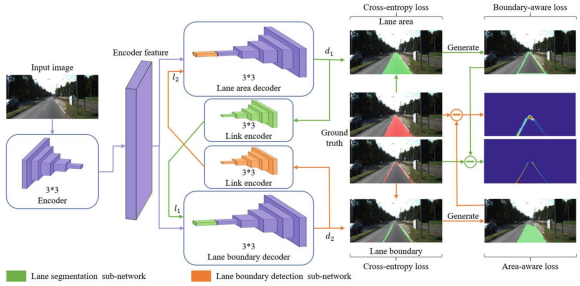

FIGURE 7. Architecture of the multi-task network that uses geometric priors for road detection. Figure is from [30].

To overcome this problem, in [30], Zhang *et al.* proposed a multi-task network that follows inherent geometric priors

between two tasks: lane region segmentation and road boundary detection (Fig. 7). In their method, lane boundaries are considered as outer contours of lane regions, and lane regions are considered as the interior of lane boundaries. The network contains a shared encoder for initial feature extraction, and two link encoders for geometric relationship extraction. The link encoders take modified preliminary predictions as inputs to generate refined features for decoders. This network is trained on a loss function that combines the cross-entropy loss, boundary-aware loss, and area-aware loss. This method can cope with complex lane types because it utilizes geometric priors between lane regions and lane boundaries. However, the complex network structure requires extra computational resources and increases the inference time.

The lane detection methods mentioned above are all based on custom deep neural networks. However, designing a high-performing deep network requires significant experience and many trial-and-error experiments. To address this issue, in [32], Ang *et al.* developed a neural architecture search (NAS) algorithm to find the optimal DNN structure for pedestrian lane detection. This method searches a network-level space using the gradient descent algorithm, and the search is performed directly on the target dataset without relying on secondary datasets. As a result, the network found by this NAS method is compact and has fast inference. However, this method requires significant time to find an optimal network structure before the network is further trained.

B. GENERIC SEMANTIC SEGMENTATION NETWORKS

Pedestrian lane segmentation can be considered a subset of semantic segmentation, where there are two classes: pedestrian lane and the background. Hence, applying semantic generic segmentation networks to lane detection is a plausible direction. Therefore, this section reviews representative deep networks for semantic segmentation. The list includes: fully convolutional networks, encoder-decoder networks, multi-scale and pyramid feature networks, dilated convolution networks, and attention-based networks.

1) FULLY CONVOLUTIONAL NETWORKS

Long *et al.* proposed the fully convolutional network (FCN) for semantic segmentation [33]. The FCN, shown in Fig. 8, replaces the fully connected layers in the classification networks (e.g., VGG-16 [46], AlexNet [47], and GoogLeNet [48]) with convolutional layers. It upsamples the coarse outputs to produce pixel-wise dense predictions. Moreover, the FCN uses skip connections to refine the spatial precision. The FCN is a milestone work in image segmentation as it was among the first to apply deep learning for semantic segmentation in an end-to-end manner. Subsequently, many semantic segmentation models have been developed based on FCNs.

Although the FCN has shown benefits, it has two main limitations. First, the FCN only uses a single-layer interpolation to reconstruct the original input size from a coarse heat map, which can lead to a loss of detailed boundary information.

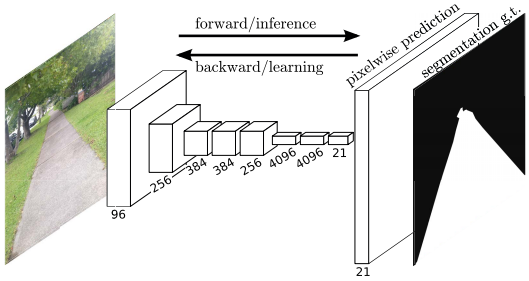

FIGURE 8. Architecture of the FCN. Figure is adapted from [33].

In lane detection, the network might misclassify roads in the distance, preventing the users from planning their routes ahead. The network might also misjudge small obstacles or tripping hazards as roads, which could endanger the user. Second, the FCN focuses on more local information than the global context, which makes prediction outputs lack spatial consistency. This might cause the network to predict lanes in the incorrect places, such as on water or in the sky.

Similarly, there are two challenges in applying classification networks for semantic segmentation. The first challenge is the lost of image resolution. Classification networks utilise down-sampling and max-pooling operations to extract low-resolution feature maps. However, semantic segmentation tasks require precise location information, which can only be achieved with high-resolution feature maps. The second challenge is the lack of multi-scale context. Objects in an image are at multiple scales, and convolutions with fixed-size kernels are not enough to capture both local and global contexts. Over the years, many network structures have been developed to address these two challenges.

2) ENCODER-DECODER NETWORKS

E-D networks address the lost resolution problem in FCNs by introducing a decoder. The decoder upsamples the coarse feature map and incorporates the spatial information from the shallower layers of the encoder. Some E-D networks also fuse the feature maps of different layers to retains more local information and reduces the risk of vanishing gradients while training the network. As a result, the encoder-decoder network can generate high-resolution lane segmentation maps.

Deconvnet [49] proposes a deconvolution network to produce dense predictions. The encoder of Deconvnet adopts the first 13 convolution layers and two fully connected layers in VGG-16. The decoder is identical to the encoder but hierarchically opposite. It also uses composed layers of unpooling, deconvolution, and rectification operations. Deconvnet records the locations of maximum activations (during the pooling operation) in switch variables and uses them for unpooling operations to improve interpolation accuracy. Since the output of an unpooling layer is enlarged but sparse, deconvolution layers produce dense predictions by using multiple learned filters to associate a single input activation with multiple outputs.

SegNet [35] (Fig. 9) adopts the 13 convolutional layers in VGG-16 as the encoder. The decoder has 13 layers that correspond to the encoder. SegNet records the pooling indices in the max-pooling steps, and uses them for the corresponding upsampling steps in the decoder. As a result, the spatial resolution of the extracted feature maps is improved. The outputs of the decoder are fed to a softmax layer to generate pixel-wise segmentation maps. Compared to Deconvnet, SegNet requires fewer computational resources because it does not include the final fully connected layer in VGG-16.

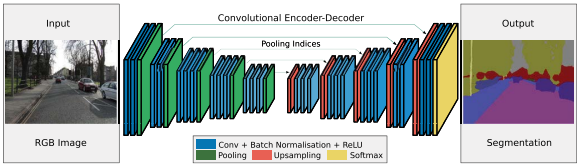

FIGURE 9. Architecture of SegNet. Figure is from [35].

U-Net [34] is a U-shaped symmetrical network that consists of a contracting path and an expansive path, see Fig. 10. The contracting path consists of repeated modules of two 3×3 convolutions (each followed by a rectified linear unit (ReLU)) and a 2×2 max-pooling layer for downsampling. The expansive path consists of repeated modules of a 2×2 convolution (also called up-convolution which halves the number of feature map channels), a feature map concatenation operation, and two 3×3 convolutions (each followed by a ReLu). The higher-resolution feature maps from the contracting path are concatenated with the corresponding upsampled feature maps from the expansive path. This process can yield higher accuracy, but needs more memory than in SegNet because it requires transferring the entire feature map instead of only pooling indices. Finally, a 1×1 convolution is used to generate pixel-wise segmentation.

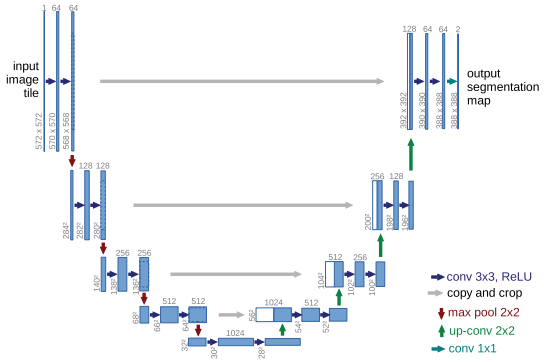

FIGURE 10. Architecture of U-Net. Figure is from [34].

UNet++ [50] is an image segmentation network developed based on the U-Net structure. UNet++ is designed to include multiple nested sub-networks between the original contracting and expansive path; therefore, the semantic gap can be bridged by the skip-pathways, shown in green and blue in Fig. 11. Moreover, because UNet++ produces full resolution predictions at multiple semantic levels ($X^{0,1}$, $X^{0,2}$, $X^{0,3}$, $X^{0,4}$), the network can be trained using deep supervision. At inference time, UNet++ can operate in a fast mode, where the final segmentation map is generated from one of the intermediate semantic levels (e.g., $X^{0,1}$, $X^{0,2}$, or $X^{0,3}$).

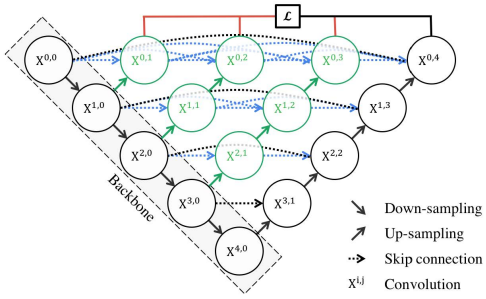

FIGURE 11. Architecture of UNet++. Figure is from [50].

In summary, the above E-D networks attempt to retain the visual information during upsampling by using different techniques. Notably, Deconvnet [49] stores the locations of maximum activations during pooling. SegNet [35] records the pooling indices during max-pooling. U-Net [34] concatenates the corresponding feature maps from the encoder and the decoder. These strategies can help a lane detection network better detect road regions in the distance and small tripping hazards on the road.

3) MULTISCALE AND PYRAMID NETWORKS

The multiscale and pyramid networks are proposed to address the multiscale challenge in semantic segmentation by utilising features at different scales. Two methods are often used to achieve pyramid structures: (i) combining feature maps from different levels of a deep CNN [36], [51]; and (ii) scaling a feature map to different scales and then concatenating the results to produce the final feature representation [37]. These methods can combine global and local contexts better, which reduces situations where lanes are predicted in the wrong places, such as in the sky or on the wall.

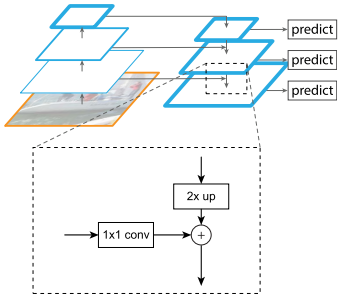

FIGURE 12. Architecture of the FPN. Figure is from [36].

The Feature Pyramid Network (FPN) [36] was originally proposed for object detection tasks, but it can be extended

for semantic segmentation tasks. The FPN consists of a bottom-up pathway, a top-down pathway, and several lateral connections (Fig. 12). The extracted feature maps from the bottom-up pathway first undergo a 1×1 convolution. Then, they are fused with the upsampled feature maps from the top-down pathway by element-wise addition via the lateral connections. The final output of the FPN is a feature pyramid that has semantically rich representations at all scales.

Pyramid Scene Parsing Network (PSPNet) [37] is a multiscale network proposed to better learn the global context (Fig. 13). Instead of using global average pooling to capture global contextual information which may cause the loss of spatial information, PSPNet proposes a pyramid pooling module. This module divides the final-layer feature map into four sub-regions and then performs the average pooling for each region. A 1×1 convolution is then used after each pyramid level to reduce the dimension of context representation. Finally, the four feature maps are concatenated to form a global feature. This global feature covers different sub-region representations and different levels of contextual information.

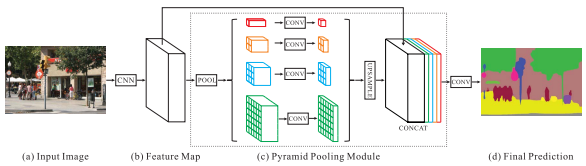

FIGURE 13. Architecture of PSPNet. Figure is from [37].

4) DILATED CONVOLUTION NETWORKS

Dilated convolution, also known as atrous convolution, is an effective way to extract multiscale features. The dilated convolution can enlarge receptive fields without increasing the computational or memory costs while preserving the spatial resolution. Computation efficiency is very important to lane detection in assistive navigation, which requires real-time inference and small models for deployment on mobile devices.

DeepLabv1 [52] is the first version in the DeepLab family. This network adopts dilated convolution to address the decreased resolution problem caused by subsampling. It also uses a fully connected pairwise CRF to refine segmentation by encouraging the same label for similar pixels. VGG-16 is used as the backbone of this network.

DeepLabv2 [53] improves DeepLabv1 by introducing the atrous spatial pyramid pooling (ASPP) module. ASPP exploits multiscale features by applying multiple parallel filters at different dilation rates to the input feature maps. The final results are generated by fusing the output feature maps from the ASPP module. Moreover, DeepLabv2 adopts ResNet-101 [54] as the encoder, which gives a better result than VGG-16.

DeepLabv3 [55] improves DeepLabv2 by exploring deeper network structures (Fig. 14). Four parallel atrous convolutions with different rates and batch normalization are applied in ASPP. Furthermore, the ASPP module incorporates the global context of the image by applying global average pooling on the last feature map. In DeepLabv3, ResNet-101 with depth-wise separable convolutions is used as the feature extractor. To improve the computational efficiency, the dense CRF is removed from the network.

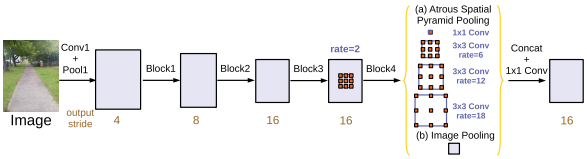

FIGURE 14. Architecture of DeepLabv3. Figure is adapted from [55].

DeepLabv3+ [38] is the latest extension with the highest performance in the DeepLab family. Compared to DeepLabv3, DeepLabv3+ adopts the encoder-decoder architecture to restore the spatial resolution (Fig. 15). The output feature maps from the encoder are bilinearly upsampled by a factor of 4 and then concatenated with the low-level feature maps. DeepLabv3+ also uses the atrous separable convolution, which consists of an atrous depthwise convolution and a 1×1 convolution. Atrous separable convolution reduces the computation complexity significantly while maintaining similar or better performance. In summary, dilated convolution is effective for enlarging the receptive field and handling multiscale features.

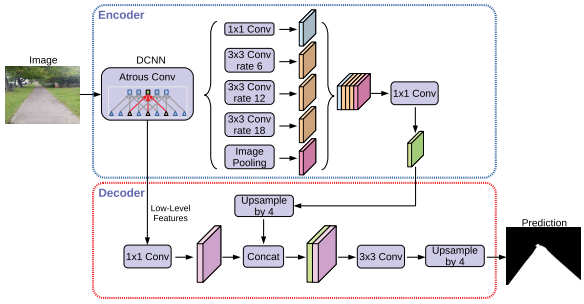

FIGURE 15. Architecture of DeepLabv3+. Figure is adapted from [38].

5) ATTENTION-BASED NETWORKS

Attention mechanism is designed to improve segmentation performance by placing a stronger emphasis on important features. It is inspired by the way way humans perceive images by focusing on the important parts rather than the entire image. This mechanism is helpful to the lane detection task as it enforces segmentation networks to focus on roads instead of other non-related classes.

In [39], a dual attention network is proposed to capture rich contextual information based on the self-attention mechanism. The network has two modules: the position module and the channel module, as in Fig. 16. The encoder of this network is ResNet-101 with dilated convolution layers in the last two blocks. The feature maps extracted by the encoder are passed

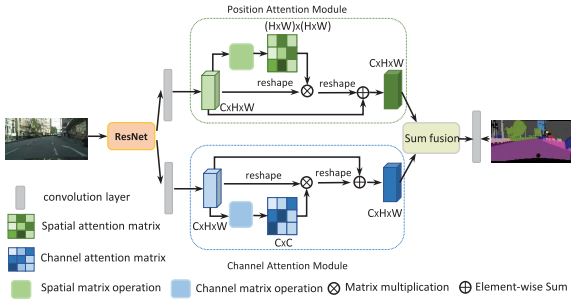

FIGURE 16. Architecture of the dual attention network. Figure is from [39].

to the two parallel attention modules. The two modules are designed to learn the spatial and channel interdependencies. This network aggregates outputs from the two modules to obtain a better feature representation for dense prediction.

In [40], an attention-based approach is proposed to hierarchically fuse multiscale information during training and inference, see Fig. 17. At training time, a relative attention mask between adjacent scales is learned. At inference time, the learned attention mask is used to fuse predictions of adjacent scales. This method enables a flexible number of scales during inference. Moreover, with multiple scales at inference time, only two scales are needed during training, which improves the training efficiency. This method achieves SOTA performance using HRNet [56] and OCR method [57] as the network structure.

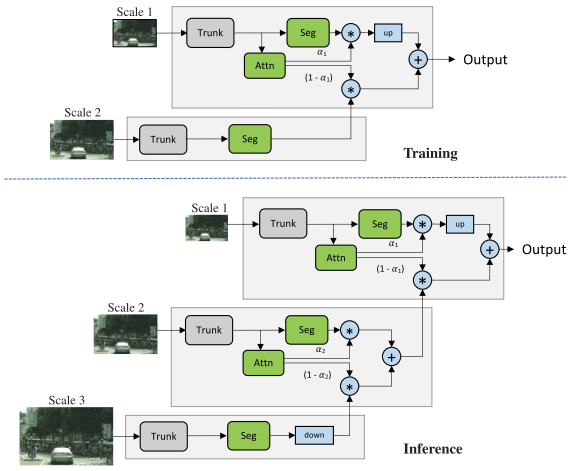

FIGURE 17. Architecture of the multiscale attention method. Figure is from [40].

C. DESIGN PATTERN IN DEEP LEARNING MODELS

This section briefly discusses common design patterns in deep learning models. Note that the design patterns are the well-recognized best practices that researchers can use when developing models to solve a problem [58].

1) EMBEDDINGS

Embeddings are utilized to represent image data as a dense feature map so that the deep network can find patterns in the input images. Encoder networks with down-sampling operations are commonly applied to embed the image data.

2) TRANSFER LEARNING

Training deep learning models often requires a huge amount of computation time and labeled data. Using encoders with pre-trained weights found on the ImageNet dataset is a common practice when designing deep networks for downstream tasks such as lane segmentation. Many road detection methods also pre-train the entire network on other large road-scene datasets, such as the Cityscapes [59] and Mapillary dataset [60].

3) PYRAMID STRUCTURES

Capturing multiscale contexts is a challenge caused by the fixed kernel sizes of CNNs. Pyramid structures can address this challenge by merging feature maps at different scales. As a result, networks can capture contexts at multiple scales and learn both global and local features better.

D. DISCUSSION

This section discusses the conceptual differences between traditional and deep learning methods. The key difference is in the feature extraction approach. Traditional lane detection methods use hand-engineered feature extractors, designed based on some prior knowledge. For example, methods in [8], [9], and [10] are based on the observation that the difference in lane positions between two adjacent video frames is usually negligible. Methods in [17], [18], and [19] are based on the assumption that the lane shape is a simple arc and has no branches. However, these heuristics and assumptions work under limited circumstances, and the hand-engineered feature extractors have difficulty in coping with different lane types.

In comparison, deep learning methods use CNNs to learn the lane features from large amounts of lane images. CNNs learn low-level features such as lines and edges in the early layers, and high-level features such as lanes in the later layers. Consequently, they can extract more complex and generalized features of the lane compared to traditional methods. However, the quality of the training data plays an important part in creating an accurate lane detection network. For example, if a pedestrian lane dataset mostly contains lanes surrounded by grass, the network will learn to associate the presence of surrounding grass with lane. Therefore, the dataset used for training pedestrian lane detection must have variations in lighting conditions, lane surface textures, and background surroundings.

The ability of the deep learning methods to learn from large-scale image datasets has enabled them to achieve better accuracy in various environments. However, traditional methods can still offer valuable insights in designing a deep model for pedestrian lane detection. Some prior knowledge can be used to design CNNs and improve model accuracy. For example, Zhang *et al.* [30] used the geometric prior that the lane boundaries form the outer contours of lane regions to design a multi-task network and improve detection accuracy.

Furthermore, the difference in lane positions between two adjacent video frames is usually small, and this property can be used to stabilize lane segmentation predictions for video.

IV. PERFORMANCE EVALUATIONS AND ANALYSIS

This section presents experimental evaluations of representative methods that can be used for lane detection. Section IV-A describes the image datasets for pedestrian lane detection. Section IV-B describes the performance measures and experimental setup used in this survey. Section IV-C presents the method performances on the PLVP3 dataset. Section IV-D presents evaluations of method robustness under different scene types. Section IV-E provides an analysis of different encoders in U-Net.

A. IMAGE DATASETS FOR PEDESTRIAN LANE DETECTION

We evaluated different methods on the pedestrian lane dataset PLVP3 [28]. This dataset is extended from the PLVP dataset [24] and the PLVP2 dataset [27], which have 2,000 and 5,000 images, respectively. The PLVP3 dataset contains 10,000 pedestrian lane images with the corresponding ground-truth masks, see Fig. 18. Each ground-truth mask is a binary image, where pixels are manually labeled as either lane or background. The color images are taken from numerous real indoor and outdoor scenes at different times of the day. They include unmarked pedestrian lanes with multiple surface textures (e.g., brick, concrete, and soil) and shapes (e.g., straight and curve). Many images are taken under unfavorable lighting variations, e.g. weak illumination and strong shadows. The statistics of the PLVP3 dataset are shown in Table 2.

TABLE 2. Statistics of the PLVP3 dataset.

Description	Number of images	Percentage
Brick surfaces	2,917	29.17
Concrete surfaces	4,860	48.60
Pavement surfaces	1,164	11.64
Indoor surfaces	734	7.34
Other surfaces	325	3.25
Normal lighting	7,845	78.45
Extreme lighting	2,155	21.55

Note that other datasets exist for the visually impaired, but they do not include pixel-wise annotations for pedestrian lanes or sidewalks. For example, the dataset in [61] contains only bounding box annotations for common sidewalk obstacles, and boundary line annotations for blind sidewalks. The dataset in [62] provides only pixel-wise and bounding box annotations for common sidewalk objects. The dataset in [63] only has the floor semantic annotations for indoor scenes. Hence, these datasets are not used for the experimental evaluation in this survey.

Other benchmark datasets such as Mapillary [60] and Cityscapes [59] contain pixel-wise annotations of the sidewalk class. However, these datasets are created for self-driving vehicles, and hence not suitable for pedestrian lane detection. In these datasets, the images are taken near the center of the vehicle roads; the pedestrian regions are often on the side with relatively small areas. In other words, there is a domain gap between these datasets and our desired application of assistive navigation for blind people.

B. PERFORMANCE MEASURES AND EXPERIMENTAL SETUP

1) PERFORMANCE MEASURES

To measure model performances, we use three quantitative metrics which have been widely accepted for semantic segmentation research: 1) pixel accuracy, 2) mean intersection over union, and 3) F1 score. To obtain the overall evaluation score on the test set, the metrics are computed for individual images and then averaged over the entire test set.

1) *Pixel accuracy* is the ratio between the correctly-classified pixels versus the total number of pixels.

2) *Mean intersection over union* (mIoU) computes the average IoU over all semantic classes. Let S be a machine-predicted segmentation map, and G be the corresponding ground-truth mask. Intersection over union (a.k.a. Jaccard Index) is defined as the area of overlap between S and G, divided by the area of union between S and G:

$$\text{IoU} = \text{Jaccard}(S, G) = \frac{|S \cap G|}{|S \cup G|} = \frac{TP}{TP + FP + FN}, \quad (1)$$

where TP, FP, and FN refer to the numbers of true positives, false positives, and false negatives.

3) *F1 score* (a.k.a. *Dice Coefficient*) is defined as the harmonic mean of precision and recall:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}. \quad (2)$$

Here, recall is the ratio of correctly-detected lane pixels versus all lane pixels. Precision is the ratio of correctly-detected lane pixels versus all machine-detected lane pixels.

2) EXPERIMENTAL SETUP

We employed the 5-fold cross-validation to evaluate the representative methods. The dataset was divided randomly into five equal-sized partitions. For each fold, one partition was used as the test set, and the remaining four partitions are used as the training set. This step was repeated five times for different choices of the test partition, and the segmentation measures were then averaged. Note that each training set was further divided into 90% images for training, and 10% images for validation. Collectively, each cross-validation fold consisted of 7200 training images, 800 validation images, and 2000 test images. The images were resized to 320×320 pixels.

To train the deep neural networks, we used the Adam optimizer [64] with a learning rate of 0.001. The exponential decay rates for the first and the second moment estimates were set to 0.9 and 0.999, respectively. All models used pretrained weights on ImageNet for the encoders, and

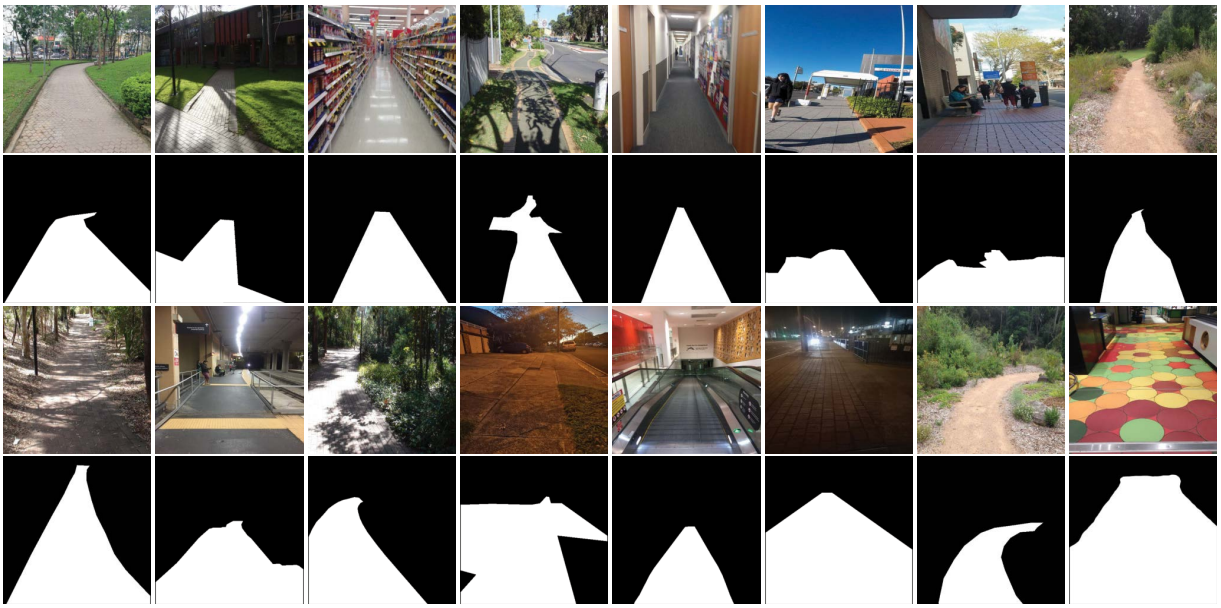

FIGURE 18. Examples from the PLVP3 dataset. Rows 1 and 3: Input color images. Rows 2 and 4: The corresponding lane segmentation ground-truth.

Kaiming uniform weight initialization [65] for the decoders. All experiments were conducted on a computer with Intel Xeon Gold 5115 2.4 GHz processor and NVIDIA TITAN Xp GP102 graphics card. All methods were implemented using the PyTorch framework [66]. We trained each network for 150 epochs. The early stop strategy was used to prevent overfitting: The training was stopped if the network's performance on the validation set had not improved for 50 epochs.

C. COMPARISONS OF REPRESENTATIVE METHODS ON PEDESTRIAN LANE DETECTION

In this section, we evaluate: (i) two feature-based methods and three deep-learning-based methods specifically designed for pedestrian lane detection, and (ii) eight deep networks for generic semantic segmentation. Although the eight segmentation networks are not explicitly proposed for lane detection, they can be applied to scene perception and their performances on the PLVP3 dataset can provide insights on designing effective lane detection networks. The implementation details of the examined methods are as follows.

1) *Border-based method* [18]: The number of orientation-consistent points for computing the orientation consistency ratio was set to 16. We used the MATLAB code provided by Kong *et al.* [18].

2) *Combined (color & border) method* [24]: We used the MATLAB code provided by Phung *et al.* [24].

3) *DL-HGP* [27]: The backbone SegNet with five encoder/decoder units (total 26 convolutional layers) was employed for feature extraction. The number of inducing points and the initial number of local Gaussian process experts were set to 50 and 9, respectively. We used the Python code provided by Nguyen *et al.* [27].

4) *BGN* [28]: We implemented the network structure presented in the reference. As recommended by the reference, the local re-parameterization trick [67] was used.

5) *NAS method* [32]: We implemented the network structure presented in the reference. The search phase of this method was conducted with 16 layers of nodes and five levels of scales. The base channel size for the output feature maps was set to eight.

6) *FCN-8s* [33]: We implemented this model based on VGG-16 as suggested in the reference. The final feature map was upsampled by a factor of 8, and then element-wise summed with the feature maps from the third and fourth pooling layers.

7) *U-Net* [34]: We implemented the network structure presented in the reference. The input images are downsampled four times before upsampling.

8) *UNet++* [50]: We implemented the network structure presented in the reference. The input images were downsampled at most four times before upsampling. This method was trained without deep supervision.

9) *SegNet* [35]: We implemented the network structure presented in the reference. The backbone VGG-16 with 13 convolutional layers was employed as the encoder.

10) *PSPNet* [37]: The backbone ResNet-101 was used as the encoder. We used the Python code provided by Zhao *et al.* [37].

11) *DeepLabv3+* [38]: We implemented the network structure presented in the reference. We used Xception as the encoder, where the depthwise separable convolutions were applied to both ASPP and decoder modules.

12) *HRNet with multiscale attention* [40]: We used the original HRNet architecture as the reference [56]. We applied

TABLE 3. Performance of the representative methods on the PLVP3 dataset.

Category	Method	mIoU	Accuracy	F1-score	Inference speed (Images/s)	#Params (M)	Model Size (MB)
Traditional methods	Border-based method [18]	63.12	79.23	37.04	0.009	-	-
	Combined method [24]	75.69	89.64	70.79	0.762	-	-
Deep learning methods	NAS-based method [32]	92.29	97.00	95.91	142.857	4.716	113.36
	DL-HGP [27]	92.61	96.40	95.17	1.008	29.4	19.0
	SegNet (VGG-16) [35]	92.57	96.45	95.30	14.450	29.4	112.43
	FCN-8s (VGG-16) [33]	94.88	97.20	96.81	27.778	14.7	56.24
	U-Net [34]	93.64	96.89	96.42	34.722	7.8	30.02
	BGN [28]	94.92	97.34	96.90	68.027	0.3	1.18
	UNet++ [50]	94.95	97.57	97.36	16.420	9.2	36.80
	DeepLabv3+ (Xception) [38]	94.99	97.42	97.28	40.322	54.7	219.83
	PSPNet (ResNet-101) [37]	95.40	97.76	97.24	30.303	68.0	272.90
	Multiscale HRNet [40]	95.59	97.87	97.70	3.269	69.8	280.00
	Multiscale HRNet-OCR [40]	**95.68**	**97.92**	**97.75**	2.610	72.1	289.30

a simple segmentation head to produce dense prediction from output feature maps of HRNet. The segmentation head consisted of $(3 \times 3$ conv$) \rightarrow$ (BN) $\rightarrow$ (ReLU) $\rightarrow (3 \times 3$ conv$) \rightarrow$ (BN) $\rightarrow$ (ReLU) $\rightarrow (1 \times 1$ conv$)$. In this experiment, we used two scales ($0.5\times$ and $1\times$) for training and three scales ($0.5\times$, $1\times$, and $2\times$) for inference as suggested in [40].

13) *HRNet-OCR with multiscale attention* [40]: We employed the original OCR structure as in [57], and the original HRNet structure as in [56]. The scale configurations for training and inference were the same as HRNet with multiscale attention.

Table 3 shows the experimental results of the representative methods. For feature-based lane detection methods, the performance of the combined method are better than that of the border-based method, with an mIoU improvement of 12.57% and an F1-score improvement of 33.75%. The inference time of the combined method is 84.67 times faster than that of the border-based method. The results indicate that combining border features with color features are more robust for coping with variations of lane appearance and illumination conditions. Because these feature-based methods are built on the fly, we do not compare their model sizes.

The deep learning methods significantly outperform the feature-based methods in both segmentation accuracy and inference speed. Among the 11 examined methods, multiscale HRNet-OCR achieves the best performance, with an mIoU of 95.68% and an accuracy of 97.92%. Note that this method also has the largest model (289.30 MB) and the lowest inference speed (2.61 images/s). Multiscale HRNet achieves the second-best performance with an mIoU of 95.59%. Due to the huge backbone with multiple inference scales, multiscale HRNet-OCR and multiscale HRNet are dramatically slower than other deep networks. The NAS-based method achieves the fastest inference of 142.857 images/s and an mIoU of 92.29%. This method has the most favorable speed-accuracy trade-off among all examined methods. The BGN has the second-best speed-accuracy trade-off, with an mIoU of 94.92%, and an inference speed of 68.027 images/s.

Assistive navigation for blind people requires real-time algorithms. In Table 3, five methods achieve inference speed higher than 30 FPS: U-Net, BGN, DeepLabv3+, PSPNet, and the NAS-based method. Among them, the BGN and the NAS-based method achieve the highest inference speed. The BGN uses Bayesian Gabor layers instead of the common convolutional layers, which significantly reduces the number of trainable weights. The NAS-based method searches directly on the pedestrian lane dataset instead of relying on networks found with other image datasets. Consequently, it obtains a deep network with an optimized structure for the lane detection task. Note that the two methods with the highest segmentation accuracy (multiscale HRNet and multiscale HRNet-OCR) have the lowest inference speed (below 4 FPS).

Fig. 19 presents examples of pedestrian lane detection results produced by different segmentation methods. The visual results indicate that the border-based method does not cope well with the variations of lane shapes. This is because it assumes that all lanes are formed by two straight edges pointing to the vanishing point. The combined method performs better than the border-based method. However, it only has medium performances, especially when the lane region has varying textures, or the lane region has a similar color to the background. This is because the combined method uses the color model constructed from the lower half of the lane to detect the entire lane regions. This method also relies substantially on the accuracy of the detected vanishing point. For example, in Row 3, Column 4, this method misses the true lane, and miss-classifies the handrail as lanes

The deep learning methods achieve significant improvements compared to the traditional methods in both inference speed and detection accuracy. However, they still produce segmentation errors, especially when the background has similar textures as lane regions (Rows 6 and 8). This finding indicates that, despite their high performances as shown in Table 3, the deep-learning-based methods are still not robust enough to maintain high accuracy in complex scenes. For example, as demonstrated in Fig. 19, even the

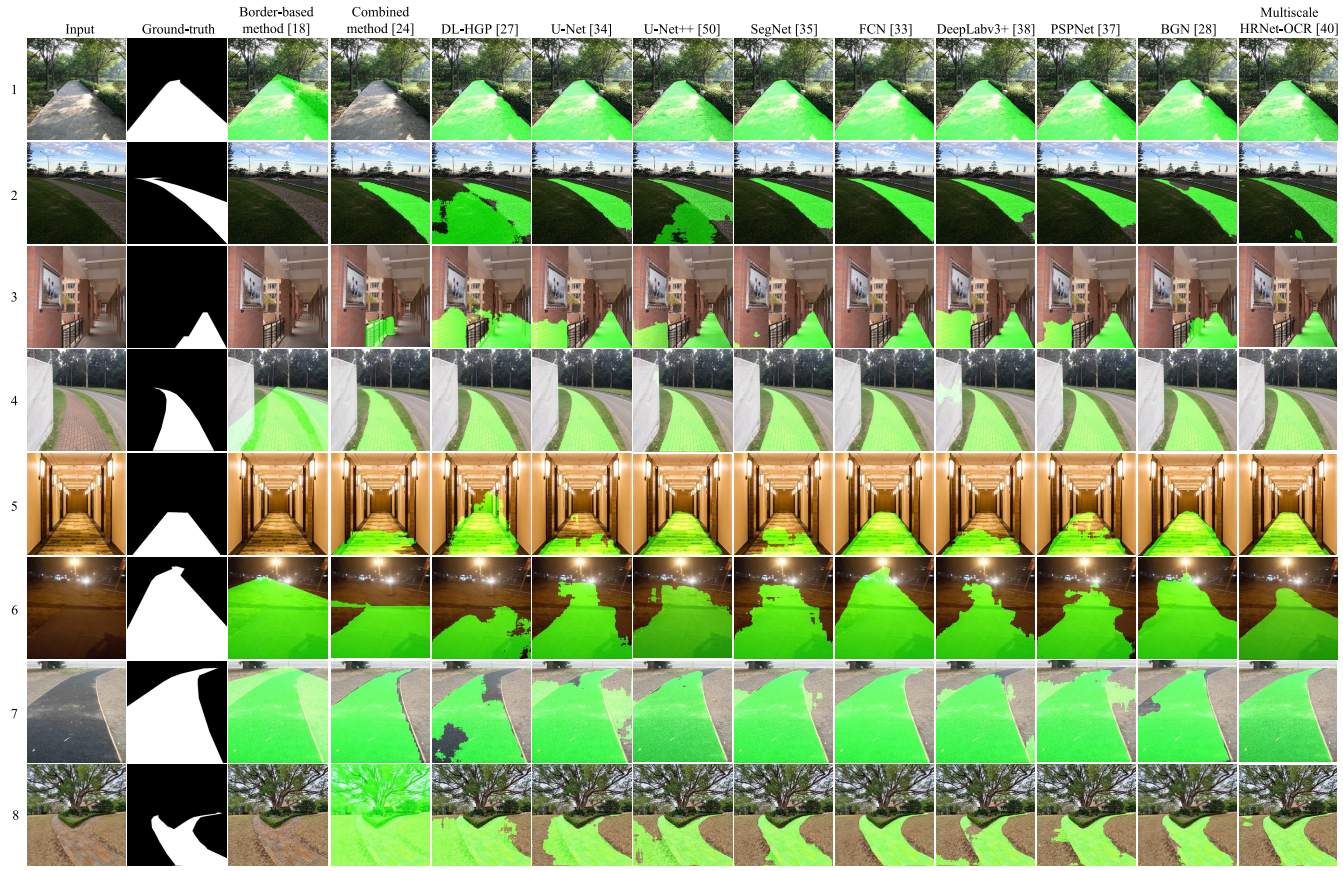

FIGURE 19. Visual results of representative methods on the PLVP3 dataset.

TABLE 4. Segmentation accuracy (mIoU) of the representative methods under different conditions in the PLVP3 dataset.

Category	Method	Indoor	Night-time	Campus	Park	Suburban	City
Traditional methods	Border-based method [18]	58.50	67.72	66.57	68.43	62.12	64.53
	Combined method [24]	69.26	55.38	75.02	78.07	72.44	69.91
Deep learning methods	NAS-based method [32]	90.36	86.70	91.89	95.61	91.42	86.09
	DL-HGP [27]	92.91	86.39	93.42	95.04	92.7	89.43
	SegNet (VGG-16) [35]	91.35	86.27	93.35	94.87	92.23	88.46
	FCN-8s (VGG-16) [33]	93.80	87.94	94.61	95.13	93.28	89.16
	U-Net [34]	92.15	87.06	93.07	95.06	92.03	89.54
	BGN [28]	93.17	88.14	94.91	95.15	93.46	90.77
	UNet++ [50]	92.69	88.36	95.31	96.47	93.93	90.17
	DeepLabv3+ (Xception) [38]	93.26	88.69	94.97	96.60	94.08	90.24
	PSPNet (ResNet-101) [37]	94.21	89.57	95.40	96.68	94.26	90.60
	Multiscale HRNet [40]	**94.56**	89.96	95.70	96.85	94.65	91.21
	Multiscale HRNet-OCR [40]	94.47	**90.58**	**95.77**	**96.91**	**94.76**	**92.12**

best-performing method has some false positive predictions (Rows 2 and 8). Because these incorrectly-predicted lane regions can endanger blind people, we will discuss this challenge further in Section V.

D. COMPARISONS OF MODEL PERFORMANCE IN DIFFERENT SCENE CONDITIONS

A good algorithm for pedestrian lane detection should maintain high accuracy in various scene conditions. In this section, we evaluate the representative methods under six different scene types in the PLVP3 dataset: indoor scenes, scenes during night-time, university campus scenes, park scenes, suburban scenes, and city scenes. As shown in Table 4, the model performances for different scenes vary by a large margin. All models show performance drops when tested in night-time and city scenes, indicating that these scene types are specifically hard for pedestrian lane detection and require further considerations when designing lane detection

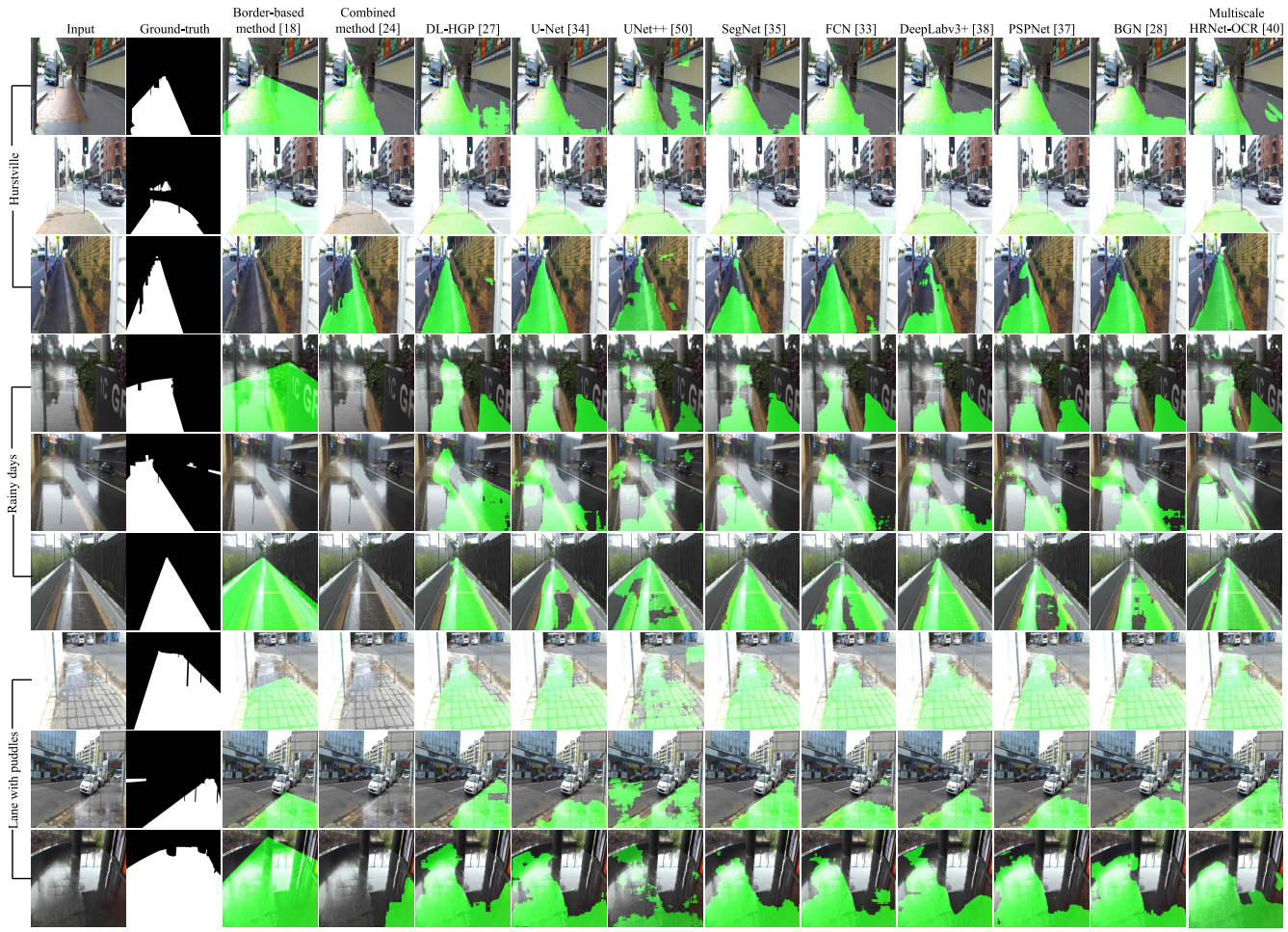

FIGURE 20. Visual results of representative methods on unseen images.

algorithms. All deep learning methods have high accuracy for park scenes, in which man-made pedestrian paths differ significantly from the natural scene elements. Moderate performances are observed in indoor, university campus, and suburban scenes. Multiscale HRNet-OCR demonstrates the best performance among all methods.

To further evaluate the robustness, we test the above methods on three scenarios outside the PLVP3 dataset: a new city (Hurstville), rainy days, and roads with puddles. As shown in Fig. 20, all methods exhibit performance drop compares to Fig. 19. Among the 11 methods, multiscale HRNet-OCR, BGN, SegNet, and DL-HGP show better stability. The traditional methods do not perform as well as the deep learning methods. Notably, most deep learning models are prone to incorrect segmentation for reflective surfaces (rainy days or lanes with puddles). The results show that the current methods achieve reasonable detection performances, but they still need to be improved for unseen situations.

E. ANALYSIS OF ENCODERS USING U-NET

We selected U-Net as the base network architecture to investigate the impact of the encoder component. Fourteen SOTA backbones were implemented and used as the encoder of U-Net. The experimental results are presented in Table 5.

In terms of segmentation accuracy, MobileNetV3 achieves the best performance with an mIoU of 96.10% and an F1-score of 97.98%. EfficientNet-b6 achieves the second-best accuracy with an mIoU of 95.92%. Compared to the original encoder in U-Net [34], SOTA backbones improve the mIoU score by 1.34% to 2.46%. In terms of inference speed and model size, MobilenetV3 has the smallest model size (19.80 MB) and the second-highest inference speed (64.516 images/s). MobilenetV2 achieves the highest inference speed (64.935 images/s), and the second smallest model size (26.80 MB).

V. FUTURE DIRECTIONS

Despite the high performances that the deep networks achieve, many aspects still need to be tackled before a practical assistive navigation system is possible. This section discusses future research directions to address the current technical limitations for pedestrian lane detection.

1) *Developing integrated methods that efficiently remove incorrectly-detected lane regions:* The evaluation

TABLE 5. Performance comparison of different encoders using U-Net on the PLVP3 dataset.

Backbone	mIoU	Accuracy	F1-score	Inference speed (Images/s)	#Params (M)	Model Size (MB)
Encoder in original U-Net [34]	93.64	96.89	96.42	34.722	7.8	30.02
NFNet-f0 [68]	94.98	97.57	97.37	17.825	80.9	324.00
ResNet-34 [54]	95.31	97.75	97.56	58.824	24.4	97.90
ResNet-152 [54]	95.28	97.73	97.54	20.367	67.2	269.60
MobileNetV2 [69]	95.64	97.91	97.73	64.935	6.6	26.80
VGG-11 [46]	95.45	97.81	97.64	42.373	18.3	73.10
VGG-16 [46]	95.54	97.86	97.68	27.322	23.8	95.10
DenseNet-161 [70]	95.53	97.85	97.68	15.504	38.7	156.20
DenseNet-169 [70]	95.55	97.87	97.68	22.831	21.2	85.90
Inceptionv4 [71]	95.60	97.90	97.71	21.459	48.8	195.80
Xception [72]	95.77	97.98	97.80	28.490	28.8	115.40
EfficentNet-b2 [73]	95.71	97.95	97.78	34.722	10.0	40.70
SENet + ResNet-50 [74]	95.78	97.98	97.81	32.258	35.1	140.60
EfficentNet-b6 [73]	95.92	98.04	97.88	15.674	43.8	176.60
MobileNetV3 [75]	**96.10**	**98.14**	**97.98**	64.516	4.8	19.80

results in Section IV-C show that even the best-performing methods produce false positives. In assistive navigation, these errors are more severe than false negatives because they could endanger the blind user. The false positives could be addressed by using post-processing steps, e.g., combining segmentation results from two separately-trained models [31]. However, these heuristic steps are less robust than machine learning methods and may fail in complex scenes. To better address this limitation, an integrated deep learning method should detect pedestrian lanes and remove false positives in an end-to-end manner.

2) *Exploring methods to cope with variations in pedestrian lanes:* As illustrated in Fig. 20, unseen situations such as new cities or extreme weather conditions will severely undermine the performance of deep learning methods. However, assistive navigation for blind people has not attracted similar attention as self-driving cars, which makes collecting a large-scale dataset from multiple countries and different weather conditions difficult. Consequently, domain shift is a technical limitation faced by most pedestrian lane detection models. Hence, investigating lane detection methods that can adapt to different environments is essential.
3) *Investigating methods that can learn from multiple different datasets:* Many existing lane detection methods only provide lane versus non-lane segmentation, which is useful but not enough for blind users to navigate safely. This limits the ability of the current pedestrian lane detection system. More detailed lane information is necessary for visually impaired users. A few datasets [61], [62] have been developed for obstacle detection in pedestrian scenes, but they do not contain pixel-wise annotations of pedestrian lanes. The PLVP3 dataset used in this study is the largest public pedestrian lane dataset in the literature, but it does not have annotations for other objects in traffic scenes. Note that creating a new large dataset containing all class labels required for assistive navigation is time-consuming and costly. Therefore, developing models that can learn from multiple existing road-scene datasets is useful to address the lack of labeled data.
4) *Developing methods to compress lane detection networks for real-time inference on smartphones or edge devices:* As shown in Section IV, most methods with

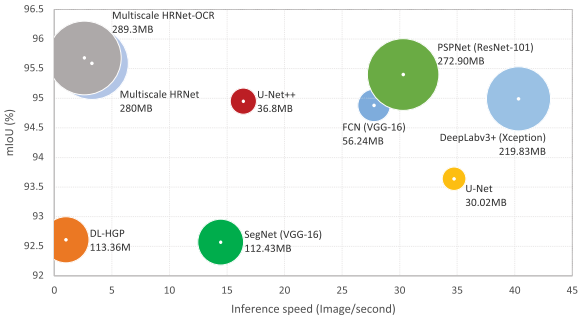

FIGURE 21. Segmentation accuracy versus inference speed produced by different lane detection methods. The area of the circles represents the model size.

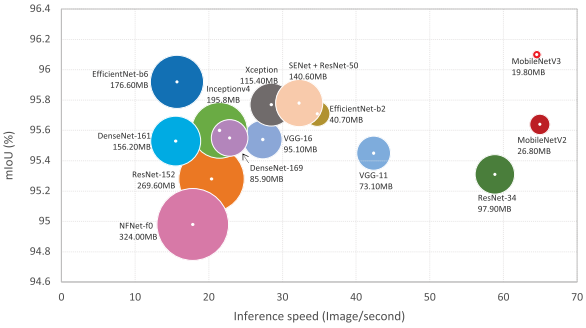

FIGURE 22. Segmentation accuracy versus inference speed produced by U-Net with different encoders. The area of the circles represents the model size.

high accuracy also require a longer inference time and larger storage. See also Fig. 21 and Fig. 22 for a summary. However, pedestrian lane detection needs to be accurate and fast to cope with real-time traffic situations. Furthermore, detection networks should be small to be deployed on mobile or edge devices, which often have limited storage, computation capability, and battery power. The current trade-off between detection accuracy, inference speed, and model size necessitates the development of methods to reduce the model size and increase the inference speed.

VI. CONCLUSION

Pedestrian lane detection is a vital task in an assistive navigation system for vision-impaired people. This paper provides a comprehensive survey of methods that can be used for pedestrian lane detection with two main categories: (i) traditional methods including color-based approaches, border-based approaches, and combined approaches; and (ii) deep learning methods including lane detection approaches and generic semantic segmentation approaches. The paper reports the quantitative performances of several notable methods on a large labeled dataset (PLVP3). Finally, we discuss the future research directions for pedestrian lane detection, which are guided by our theoretical review and experimental evaluations. We hope that this survey could serve as a baseline reference for assistive navigation, and facilitate the urgently-needed research for vision-impaired people.

ACKNOWLEDGMENT

The authors thank the five anonymous reviewers for the constructive comments which helped improving the paper. The findings herein reflect the work and are solely the responsibility of the authors.

REFERENCES

[1] R. R. A. Bourne *et al.*, "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: A systematic review and meta-analysis," *Lancet Global Health*, vol. 5, no. 9, pp. e888–e897, 2017.

[2] S. A. Bibby, E. R. Maslin, R. Mcilraith, and G. P. Soong, "Vision and self-reported mobility performance in patients with low vision," *Clin. Exp. Optometry*, vol. 90, no. 2, pp. 115–123, Mar. 2007.

[3] C. M. Patino, R. McKean-Cowdin, S. P. Azen, J. C. Allison, F. Choudhury, and R. Varma, "Central and peripheral visual impairment and the risk of falls and falls with injury," *Ophthalmology*, vol. 117, no. 2, pp. 199–206, 2010.

[4] T. Hong, P. Mitchell, G. Burlutsky, C. Samarawickrama, and J. J. Wang, "Visual impairment and the incidence of falls and fractures among older people: Longitudinal findings from the blue mountains eye study," *Investigative Ophthalmol. Vis. Sci.*, vol. 55, no. 11, pp. 7589–7593, 2014.

[5] H. Zhang and C. Ye, "An indoor wayfinding system based on geometric features aided graph SLAM for the visually impaired," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 9, pp. 1592–1604, Sep. 2017.

[6] B. Li, J. P. Muñoz, X. Rong, J. Xiao, Y. Tian, and A. Arditi, "ISANA: Wearable context-aware indoor assistive navigation with obstacle avoidance for the blind," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 448–462.

[7] M. C. Le, S. L. Phung, and A. Bouzerdoum, "Pedestrian lane detection for the visually impaired," in *Proc. Int. Conf. Digit. Image Comput. Techn. Appl. (DICTA)*, Dec. 2012, pp. 1–6.

[8] J. D. Crisman and C. E. Thorpe, "SCARF: A color vision system that tracks roads and intersections," *IEEE Trans. Robot. Autom.*, vol. 9, no. 1, pp. 49–58, Feb. 1993.

[9] M. A. Sotelo, F. J. Rodriguez, L. Magdalena, L. M. Bergasa, and L. Boquete, "A color vision-based lane tracking system for autonomous driving on unmarked roads," *Auto. Robots*, vol. 16, no. 1, pp. 95–116, Jan. 2004.

[10] O. Ramstrom and H. Christensen, "A method for following unmarked roads," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2005, pp. 650–655.

[11] C. Tan, T. Hong, T. Chang, and M. Shneier, "Color model-based real-time learning for road following," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2006, pp. 939–944.

[12] J. M. Á. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 184–193, Mar. 2011.

[13] B. Yu and A. K. Jain, "Lane boundary detection using a multiresolution Hough transform," in *Proc. Int. Conf. Image Process.*, 1997, pp. 748–751.

[14] V. Voisin, M. Avila, B. Emile, S. Begot, and J.-C. Bardet, "Road markings detection and tracking using Hough transform and Kalman filter," in *Proc. Adv. Concepts Intell. Vis. Syst.*, 2005, pp. 76–83.

[15] Y. Chen, M. He, and Y. Zhang, "Robust lane detection based on gradient direction," in *Proc. 6th IEEE Conf. Ind. Electron. Appl.*, Jun. 2011, pp. 1547–1552.

[16] H. Yoo, U. Yang, and K. Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1083–1094, Sep. 2013.

[17] C. Rasmussen, "Texture-based vanishing point voting for road shape estimation," in *Proc. Brit. Mach. Vis. Conf.*, 2004, pp. 470–477.

[18] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2211–2220, Aug. 2010.

[19] M. C. Le, S. L. Phung, and A. Bouzerdoum, "Pedestrian lane detection in unstructured environments for assistive navigation," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2014, pp. 1–8.

[20] J. D. Crisman and C. E. Thorpe, "UNSCARF—A color vision system for the detection of unstructured roads," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1991, pp. 2496–2501.

[21] 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. 2003.

[22] O. Miksik, P. Petyovsky, L. Zalud, and P. Jura, "Robust detection of shady and highlighted roads for monocular camera based navigation of UGV," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 64–71.

[23] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot monocular vision navigation based on road region and boundary estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 1043–1050.

[24] S. L. Phung, M. C. Le, and A. Bouzerdoum, "Pedestrian lane detection in unstructured scenes for assistive navigation," *Comput. Vis. Image Understand.*, vol. 149, pp. 186–196, Aug. 2016.

[25] L. C. L. Bianco, J. Beltrán, G. F. López, F. García, and A. Al-Kaff, "Joint semantic segmentation of road objects and lanes using convolutional neural networks," *Robot. Auto. Syst.*, vol. 133, Nov. 2020, Art. no. 103623.

[26] Z. Cao, X. Xu, B. Hu, and M. Zhou, "Rapid detection of blind roads and crosswalks by using a lightweight semantic segmentation network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6188–6197, Oct. 2021.

[27] T. N. A. Nguyen, S. L. Phung, and A. Bouzerdoum, "Hybrid deep learning-Gaussian process network for pedestrian lane detection in unstructured scenes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5324–5338, Dec. 2020.

[28] H. T. Le, S. L. Phung, and A. Bouzerdoum, "Bayesian Gabor network with uncertainty estimation for pedestrian lane detection in assistive navigation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5331–5345, Aug. 2022.

[29] S. Yadav, S. Patra, C. Arora, and S. Banerjee, "Deep CNN with color lines model for unmarked road segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 585–589.

[30] J. Zhang, Y. Xu, B. Ni, and Z. Duan, "Geometric constrained joint lane segmentation and lane boundary detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 502–518.

[31] T. Almeida, B. Lourenço, and V. Santos, "Road detection based on simultaneous deep learning approaches," *Robot. Auton. Syst.*, vol. 133, pp. 1–10, Nov. 2020.

[32] S. P. Ang, S. L. Phung, A. Bouzerdoum, T. N. A. Nguyen, S. T. M. Duong, and M. M. Schira, "Real-time pedestrian lane detection for assistive navigation using neural architecture search," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 8392–8399.
[33] 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.
[34] 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.
[35] 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, Jan. 2017.
[36] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.
[38] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
[39] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3146–3154.
[40] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," 2020, *arXiv:2005.10821*.
[41] C. Oh, J. Son, and K. Sohn, "Illumination robust road detection using geometric information," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 1566–1571.
[42] M. C. Le, S. L. Phung, and A. Bouzerdoum, "Lane detection in unstructured environments for autonomous navigation systems," in *Proc. Asian Conf. Comput. Vis.*, 2015, pp. 414–429.
[43] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.
[44] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*.
[45] 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.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.
[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[49] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, May 2015, pp. 1520–1528.
[50] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. Int. Workshop Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support*, 2018, pp. 3–11.
[51] A. Kirillov, R. Girshick, K. He, and P. Dollar, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6392–6401.
[52] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[53] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[55] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.
[56] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.
[57] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 173–190.
[58] V. Lakshmanan, S. Robinson, and M. Munn, *Machine Learning Design Patterns*. Sebastopol, CA, USA: O'Reilly Media, 2020.
[59] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[60] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The Mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4990–4999.
[61] W. Tang, D.-E. Liu, X. Zhao, Z. Chen, and C. Zhao, "A dataset for the recognition of obstacles on blind sidewalk," *Universal Access Inf. Soc.*, vol. 2021, pp. 1–14, Aug. 2021.
[62] K. Park, Y. Oh, S. Ham, K. Joo, H. Kim, H. Kum, and I. S. Kweon, "SideGuide: A large-scale sidewalk dataset for guiding impaired people," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10022–10029.
[63] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2D–3D-semantic data for indoor scene understanding," 2017, *arXiv:1702.01105*.
[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–15.
[65] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2015, pp. 1026–1034.
[66] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
[67] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
[68] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," 2021, *arXiv:2102.06171*.
[69] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[70] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[71] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.
[72] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
[73] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[74] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[75] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

YUNJIA LEI received the B.Eng. degree (Hons.) from the University of Wollongong, Australia, in 2020, where she is currently pursuing the Ph.D. degree in computer engineering. Her research interests include machine learning, deep learning, image processing, and pattern recognition.

SON LAM PHUNG (Senior Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees in computer engineering from Edith Cowan University, Australia, in 1999 and 2003, respectively. He was invited as a Visiting Senior Research Scientist at VinAI and VinFAST, from 2020 to 2021. He is currently a Professor at the University of Wollongong. He has published over 130 papers in journals and international conferences. He has served as the Chief Investigator for over 16 research projects funded by government agencies (research, defense, intelligence, foreign affairs, and trade) and industry. His research interests include image and signal processing, neural networks, pattern recognition, and machine learning. He was awarded the University and Faculty Medals, in 2000. He is currently serving as an Associate Editor for IEEE ACCESS and a Section Editor for *Sensors*.

ABDESSELAM BOUZERDOUM (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, USA.

In 2004, he was appointed as a Professor and the Head of the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong (UOW), Wollongong, Australia, where he also served as the Associate Dean (Research), from 2007 to 2013. From 2009 to 2011, he was a member of the Australian Research Council College of Experts and served as the Deputy Chair for EMI Panel. In 2015, he was promoted to a Senior Professor of computer engineering with UOW. He was a Distinguished Visiting Professor at several international institutions in France, USA, Germany, China, and New Zealand. Most recently, he served as the Head for the ICT Division, College of Science and Engineering, Hamad Bin Khalifa University (HBKU), Doha, Qatar, where he is currently serving as an Associate Provost for Academic Affairs. His main research interests include signal and image processing, radar imaging, vision, machine learning, and pattern recognition. He was a recipient of the Eureka Prize for Outstanding Science in Support of Defence or the National Security, in 2011, the Chester Sall Award of IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, in 2005, and the Distinguished Researcher Award (Chercheur de Haut Niveau) from the French Ministry, in 2001. He served as an Associate Editor for five international journals, including IEEE TRANSACTIONS ON IMAGE PROCESSING.

HOANG THANH LE received the B.Eng. degree from Nha Trang University, Vietnam, in 2008, the M.Sc. degree from the University of Queensland, Australia, in 2012, and the Ph.D. degree from the University of Wollongong, Australia, in 2021, all in computer science. He is currently an Associate Research Fellow with the University of Wollongong. His research interests include image processing, pattern recognition, machine learning, and computer vision.

KHOA LUU is currently an Assistant Professor and the Director of the Computer Vision and Image Understanding Laboratory, Department of Computer Science and Computer Engineering, University of Arkansas, USA. His research has been published in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IJCV, CVPR, and ICCV. His research interests include image and video processing, deep learning, image segmentation, scene perception, and autonomous vehicles. He is a Co-organizer and the Chair of CVPR Precognition Workshop in 2019, 2020, and 2021; MICCAI Workshop in 2019 and 2020; and ICCV Workshop in 2021. He is currently serving as an Associate Editor for IEEE ACCESS.