Bayesian Gabor Network With Uncertainty Estimation for Pedestrian Lane Detection in Assistive Navigation

Hoang Thanh Le, Son Lam Phung, *Senior Member, IEEE*, and Abdesselam Bouzerdoum, *Senior Member, IEEE*

***Abstract*— Automatic pedestrian lane detection is a challenging problem that is of great interest in assistive navigation and autonomous driving. Such a detection system must cope well with variations in lane surfaces and illumination conditions so that a vision-impaired user can navigate safely in unknown environments. This paper proposes a new lightweight Bayesian Gabor Network (BGN) for camera-based detection of pedestrian lanes in unstructured scenes. In our approach, each Gabor parameter is represented as a learnable Gaussian distribution using variational Bayesian inference. For the safety of vision-impaired users, in addition to an output segmentation map, the network provides two full-resolution maps of aleatoric uncertainty and epistemic uncertainty as well-calibrated confidence measures. Our Gabor-based method has fewer weights than the standard CNNs, therefore it is less prone to overfitting and requires fewer operations to compute. Compared to the state-of-the-art semantic segmentation methods, the BGN maintains a competitive segmentation performance while achieving a significantly compact model size (from 1.8× to 237.6× reduction), a fast prediction time (from 1.2× to 67.5× faster), and a well-calibrated uncertainty measure. We also introduce a new lane dataset of 10,000 images for objective evaluation in pedestrian lane detection research.**

***Index Terms*— Bayesian Gabor Network, uncertainty estimation, variational inference, assistive and autonomous navigation, pedestrian lane detection.**

I. Introduction

TRAVELING safely and independently in unknown scenes is a major challenge for vision-impaired people. Numerous assistive and autonomous navigation methods have been developed to enable a vision-impaired person to navigate in a crowded area. A navigation system needs to perform several vital micro-navigation tasks such as lane detection, danger zone detection, lane condition recognition, and depth sensing. In this paper, we focus on the most significant task of pedestrian lane detection, which assists vision-impaired users to identify walkable paths and maintain their balance while walking. It is also a core component of a lane departure warning system that allows an assistive device (e.g., smart wheelchair and rollator) to operate autonomously with little guidance from disabled users. Automatic detection of pedestrian lanes can be applied to safe driving [1] and robots [2] in sensing off-limit areas and avoiding collision with pedestrians.

There are several research gaps in this area. *First*, despite the importance of assistive navigation, only a few automatic systems are developed for pedestrian lane detection. Some methods have been proposed to identify the lanes in structured environments based on the white markers [3], [4]. However, signalized intersections or pedestrian crossings with the white markers are only special cases, while most pedestrian paths have arbitrary lane surfaces. Other existing algorithms for unmarked lane detection rely on estimating vanishing points [5], [6] or extracting manually-designed features [7], [8], which are often sensitive to the scene variations. Recently, an uncertainty estimation method has been designed for pedestrian lane detection using a hybrid deep learning-hierarchical Gaussian process (DL-HGP) to produce both segmentation map and uncertainty map [9]. Although it yields promising segmentation performance, this method still has a high computational cost and a large model size, which are not well-suited to deployment on edge computers.

Second, although recent road-detection methods have achieved the state-of-the-art performance [10], [11], they are mostly designed for vehicle road lane detection. They often fail to identify pedestrian lanes, which is generally a more challenging problem. The surface textures, shapes, and illumination conditions of walking paths typically vary more significantly than vehicle roads.

Third, model uncertainty is important for preventing unintended behaviors of an autonomous system. This is a fundamental aspect of AI safety. Although recent deep learning (DL) models are able to learn powerful representations for mapping

Manuscript received 17 October 2021; revised 26 December 2021 and 12 January 2022; accepted 14 January 2022. Date of publication 18 January 2022; date of current version 4 August 2022. This work was supported in part by the Australian Research Council through the Discovery Project on "Assistive micro-navigation for vision impaired people" under Grant DP190100607. This article was recommended by Associate Editor D. Grois. *(Corresponding author: Son Lam Phung.)*

Hoang Thanh Le is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia, and also with the Faculty of Information Technology, Nha Trang University, Nha Trang 650000, Vietnam (e-mail: tlhoang@uow.edu.au).

Son Lam Phung is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: phung@uow.edu.au).

Abdesselam Bouzerdoum is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia, and also with the Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar (e-mail: bouzer@uow.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3144184.

Digital Object Identifier 10.1109/TCSVT.2022.3144184

high-dimensional inputs to an array of outputs, these mappings are treated as a black box and the networks may have limited awareness of their own competence. Note that, in a point-estimate neural network, predictive probabilities obtained by a *softmax* layer are often erroneously interpreted as the model confidence [12]. The network may extrapolate inputs far from the observed data and consequently exhibit unjustifiably high prediction confidence in scenarios that could be life-threatening to users. Hence, for the safety of vision-impaired users, a reliable algorithm must provide not only a detection decision but also a confidence measure with which users can trust the decision. An autonomous navigation system can also employ these confidence measures to sense off-limit regions and to keep users within the safe lane areas.

Gabor filtering is widely considered as an effective method for pattern analysis applications, e.g. face verification [14] and human gait recognition [15]. It is useful for analyzing the oriented-frequency information occurring in an image region. For road sensing, features constructed from the response of Gabor filters have been shown to be particularly suitable not only for detecting road boundaries [16], but also for analyzing road patterns [17]. The textural features of road surfaces can be efficiently extracted using a bank of Gabor filters. This is because road regions commonly correspond to smoother textures (i.e., low frequencies), whereas off-road regions have rough textures (i.e., high frequencies). Due to the nature of band-pass filters, a well-designed bank of Gabor filters effectively separates road textures from off-road textures. Furthermore, it can capture the properties of orientation selectivity, spatial locality, and spatial frequency selectivity to cope with the variations in illumination [16].

This paper addresses the aforementioned gaps in the existing approaches by proposing a cost-effective uncertainty estimation method for the camera-based detection of pedestrian lanes in unstructured scenes. To identify the lanes, we cast pedestrian lane detection as the task of semantic segmentation, where image pixels are classified into pedestrian-lane and background categories. This is because, unlike vehicle roads typically consisting of parallel boundaries with highly visible markers on asphalt surfaces, pedestrian lanes have various shapes with unclear boundaries. The pedestrian lanes refer to arbitrary areas where a visual-impaired person can walk safely, and they are not necessarily bounded by lines. Thus, identifying pedestrian lanes based on only the detection of lines is not practical. Fig. 1 shows examples in our dataset of pedestrian lanes without parallel lines. There are several studies devoted to the problem of lane detection using semantic pixel-wise classification [9], [18], [19].

The main contributions of this paper can be highlighted as follows.

1) We introduce a new lightweight Bayesian Gabor Network (BGN).[1] Each filter channel is represented by only ten learnable parameters (i.e., means and variances of the five Gabor parameters) regardless of the kernel size. This enables large receptive fields to be designed without a rapid growth in the number of learnable parameters. Reducing the number of learnable parameters is also an effective way to control overfitting and improve computational efficiency.
2) We apply *variational Bayesian inference* to Gabor neural network for semantic image segmentation. To the extent of our knowledge, this work is the first to parameterize a network with Gaussian distributions of Gabor parameters instead of point-estimates as the conventional Gabor neural networks [20], [21]. Due to the probabilistic Bayesian nature, BGN provides two full-resolution maps of aleatoric uncertainty and epistemic uncertainty as well-calibrated confidence measures for user safety.
3) We create a new benchmark dataset for quantitative evaluation in pedestrian lane detection research, named PLVP3. It is the largest pedestrian lane dataset in the literature.[2] Our dataset consists of 10,000 images with the manually-annotated ground-truth acquired from real indoor and outdoor scenes, at different times of day and in different weather conditions. It includes unmarked pedestrian lanes with various lane shapes, colors, textures, and surface materials (e.g., soil, concrete, brick, and carpet).

Fig. 1. Pedestrian lanes vs. road lanes. Top row: Pedestrian lanes in PLVP3 dataset with various shapes and unclear boundaries. The traditional vehicle road detection methods based on lines are not practical in these cases. Bottom row: Road lanes in Cityscapes dataset [13].

The remainder of the paper is organized as follows. Section II introduces the related work on the automatic detection of pedestrian lanes. Section III describes the proposed Bayesian Gabor Network for lane segmentation. Section IV presents the experimental results and analysis, and finally, Section V gives the concluding remarks.

II. Related Work

In this section, we provide a review on traditional pedestrian lane detection methods and CNN-based semantic segmentation methods in computer vision, which can be applied for lane detection. The existing Bayesian methods for uncertainty estimation and Gabor neural networks are then discussed.

A. Traditional Lane Detection Methods

1) Lane-Boundary Detection Approach: It is generally assumed that pedestrian lane boundaries have distinctive

[1] https://github.com/hthanhle/Bayesian-Gabor-Networks

[2] http://documents.uow.edu.au/ phung/plvp3.html

appearances with the lane regions. Several studies have identified the boundaries based on white stripes, which are commonly found at traffic intersections [3], [22]. In general, these methods do not cope well with unmarked pedestrian lanes in unstructured scenes. To address this problem, several studies have detected edge features using Hough Transform [23], [24] or the gradients of the intensity value [25], [26]. However, the performance of these methods highly depends on the variations of lane appearance and illumination conditions. They are not effective when other objects with strong contrast occur on the lane surfaces, such as shadows and brick patterns. To overcome this issue, several methods have detected lane borders among the edges pointing to the vanishing point [5], [6], [17]. In [5], Phung *et al.* employed local orientations of color edge pixels to estimate vanishing points, which are then utilized for sample region selection. The lane regions are determined using a matching score that combines color, edge, and shape features.

2) Lane Segmentation Approach: The lane regions are segmented using different color models [27], [28] or applying semantic pixel-wise classification [9], [18], [19]. The main limitation of the color-model-based methods is the use of offline trained models, which is sensitive to the variation of lane surfaces. To overcome this problem, a few CNN-based semantic segmentation methods have been proposed recently. In [9], Nguyen *et al.* introduced an uncertainty estimation method combining a convolutional encoder-decoder network and a hierarchical Gaussian Process (HGP) classifier. In addition to a segmentation map, this Bayesian method provides a map of calibrated uncertainty. In [18], Selim *et al.* employed a feature pyramid network (FPN) with the pre-trained network ResNet-50 for the bottom-up pathway. In [19], Zhang *et al.* proposed a multiple-task learning framework, where lane segmentation and lane boundary detection are performed simultaneously. Two geometric prior constraints are used to regularize the problem of lane detection.

B. CNN-Based Semantic Segmentation Methods

Pedestrian lane segmentation can be considered as a subset of semantic image segmentation. This subsection presents a brief review on CNN-based semantic image segmentation methods in computer vision.

One of the first CNN-based works is the fully convolutional network (FCN) [29], which does not include any dense layers as in the traditional CNNs. Instead, a final 1×1 convolution is used to perform the task of pixel-wise prediction. Inspired by the FCNs, several methods have been proposed using the fully convolutional encoder-decoder architecture, notably SegNet [30] and U-Net [31]. To obtain a better representation ability, several methods either aggregate contextual features at multiple scales [32]–[34] or maintain high-resolution representations through the whole process [35]. In [32], Zhao *et al.* proposed a pyramid scene parsing network (PSPNet) by exploiting the global context through the pyramid pooling modules. In [34], Tao *et al.* introduced a Hierarchical Multi-scale Attention method (HMSA) by which the network learns to predict a relative weighting between adjacent scales. In [35], Wang *et al.* proposed a High-Resolution Network (HRNet), which connects the high-to-low resolution streams in parallel and repeatedly exchange the information across the resolutions.

Dilated convolution (a.k.a atrous convolution) is an effective way to handle multi-scale features and enlarge the receptive field without increasing the computational cost. Chen *et al.* applied the atrous convolution to develop the DeepLab models [36]–[39] for semantic image segmentation. To refine segmentation maps, DeepLabv1 [36] employs fully-connected conditional random fields (CRFs), which are probabilistic models for predicting values given the conditional input of surrounding pixels. In addition to the atrous convolutions and the CRFs, DeepLabv2 [37] uses the atrous spatial pyramid pooling (ASPP) for exploiting multi-scale features with multiple parallel filters at different rates. DeepLabv3 [38] and DeepLabv3+ [39] are the improved versions, where a backbone (e.g., Xception [40]) with the depth-wise separable convolutions is utilized as the main feature extractor.

C. Bayesian Methods for Uncertainty Quantification

1) Bayesian Neural Networks: In recent years, there is an increasing interest in applying Bayesian learning to neural networks for quantifying uncertainty of predictions. Various Bayesian methods have been proposed to approximate the intractable true posterior probability distribution. In [41], Blundell *et al.* first introduced the variational inference for learning probability distribution on the weights of a multilayer perceptron, called *Bayes by Backprop* algorithm. The distributions are obtained by minimising the evidence lower bound (ELBO) on the marginal likelihood. With the advances in CNNs, several studies have placed probability distributions over convolutional kernels [12], [42], [43]. In [42], Gal and Ghahramani first showed that dropouts can be cast as approximate Bernoulli variational inference in Bayesian CNNs. This method produces a considerable improvement in classification accuracy on the CIFAR-10 dataset as compared to the non-Bayesian methods. Instead of randomly drawing Bernoulli random variables, in [43], Shridhar *et al.* applied the Bayes by Backprop algorithm to CNNs with Gaussian variational posterior probability distributions. To estimate the predictive uncertainties in a coherent manner, softplus normalization is used in the last fully-connected layer.

2) Bayesian Semantic Segmentation Methods: A few studies have been conducted recently to predict pixel-wise class labels and provide a measure of model uncertainty [44]–[46]. In [44], Kendall *et al.* introduced a probabilistic pixel-wise semantic segmentation, called Bayesian SegNet. The Monte Carlo (MC) sampling with dropout at both training and test time is used to evaluate the posterior distribution of the softmax outputs over the SegNet's weights. Inspired by Bayesian SegNet, Mukhoti and Gal modified DeepLabv3+ (with Xception backbone) to create a Bayesian counterpart using the MC dropout and the concrete dropout as inference techniques [45]. Unlike Bayesian SegNet using the variance of the softmax outputs, Bayesian DeepLabv3+ utilizes the mutual information and the predictive entropy to estimate the model uncertainty. The main limitations of these dropout-based methods are the huge model size of the backbone and the manually-tuned dropout rates. Furthermore, using MC dropout does not allow us to capture aleatoric uncertainty and epistemic uncertainty explic-

itly. Although several Bayesian neural networks have been proposed for image classification task, there is few research on Bayesian methods for semantic segmentation task using variational inference.

D. Gabor Neural Networks

In recent years, several studies have applied Gabor filtering to neural networks and visual recognition. In [20], Luan *et al.* proposed Gabor convolutional networks (GCNs), where standard convolutional filters are modulated with Gabor filters, called Gabor orientation filters (GoFs), to produce enhanced feature maps. The number of Gabor orientations and scales are fixed before the modulation and training processes. In [47], Yao *et al.* extracted Gabor features at three particular directions (0°, 45°, and 90°), which are then employed as a pseudo 3-channel image for classification using a standard CNN. A limitation of the existing Gabor-based methods is that they mainly utilize manually-designed Gabor parameters for feature extraction. Finding appropriate Gabor parameters for a certain problem is time-consuming and requires significant domain expertise. Most existing networks embed Gabor modules into CNNs or modulate standard convolutional kernels with Gabor filters. They are not full deep networks constructed from only Gabor filters with an end-to-end training algorithm. To the extent of our knowledge, there are no Gabor neural networks with Bayesian learning reported in the literature.

E. Salient Object Detection

Pedestrian lane detection is also related to salient object detection (SOD), where lane regions are separated from the backgrounds. Although SOD is class-agnostic, high-level semantics are often employed in saliency modeling. Several methods have applied binary semantic segmentation with the saliency aggregation modules to produce an output SOD map [48], [49]. To reduce the computational cost, a few lightweight models have recently been developed using adversarial networks [50] and hierarchical visual perception learning [51]. Although these methods have shown state-of-the-art performance on the benchmark SOD datasets, they are not specifically designed for the detection of pedestrian lanes. Furthermore, due to the nature of class-agnostic segmentation, the SOD methods may not be suitable for navigation systems which usually require multi-task capability. Besides the category of pedestrian lanes, the systems typically consider other objects of interest (e.g., danger zones and lane conditions).

III. Proposed Bayesian Gabor Network

This section presents the proposed Bayesian Gabor Network, including network architecture of BGN (Section III-A), the probabilistic Gabor modeling with variational inference (Section III-B), training BGN (Section III-C), and uncertainty estimation in BGN (Section III-D).

Gabor filtering has attracted considerable research interest due to its biological motivations. It is widely accepted that the Gabor-like spatial functions are closely related to the mammalian vision systems, particularly in the perception of texture [52]. Note that such orientation-sensitive functions may also be learned by many machine learning algorithms, such as spike-and-slab sparse coding [53] and CNNs [54] when applied to natural images. Inspired by the biological and computational evidence of the Gabor filtering, we propose a new Gabor neural network with variational inference.

For clarity, a brief background on Gabor filters is warranted. Let λ be the wavelength of the sinusoidal component, α be the orientation of the normal to the parallel stripes, ϕ be the phase offset, δ be the standard deviation of the Gaussian envelope, and γ be the spatial aspect ratio. A complex Gabor filter channel with real and imaginary components representing orthogonal directions is defined as

$$g_{\lambda,\alpha,\phi,\delta,\gamma} = \exp\{-\frac{\tilde{x}^2 + \gamma^2\,\tilde{y}^2}{\delta^2}\}\ \exp\{i(2\pi\frac{\tilde{x}}{\lambda} + \phi)\}, \quad (1)$$

where $\tilde{x} = x\cos\alpha + y\sin\alpha$ and $\tilde{y} = -x\sin\alpha + y\cos\alpha$ are the transformed coordinates.

A. Network Architecture

Inspired by the panoptic feature pyramid network [55], BGN is designed as a compact model where the rich multi-scale features are merged into a single output (see Fig. 2). The network comprises 13 Bayesian Gabor layers, which are the probabilistic models of Gabor filters. To compute the outputs at each layer, we sample the five Gabor parameters of a Gabor filter (i.e., λ, α, ϕ, δ, and γ) from Gaussian distributions represented by learnable means and learnable variances. The detailed description of Bayesian Gabor layer is presented in Section III-B.

We use the larger kernel sizes (15×15 and 7×7) in the early layers to capture the global fine-grained information, and the smaller kernel sizes (5×5 and 3×3) in the successive layers to extract the local semantic information. At each scale, we perform several progressively downsampling stages, followed by a bilinear upsampling (except the first scale) to reach the same scale of 1/4. The output feature maps at two adjacent scales are then element-wise summed. A bilinear upsampling by a factor of 4 and a final Bayesian Gabor layer with kernel size of 1×1 are utilized to generate the segmentation map at the original image resolution.

Note that high-resolution feature maps in the early Bayesian Gabor modules are well-suited to capture the fine structures, but they contain semantically weak features. By contrast, low-resolution feature maps in the successive Bayesian Gabor modules contain semantically strong features to accurately predict the per-pixel class labels, but the spatial locations are not precise due to the pooling effects. Hence, BGN builds a feature hierarchy where the semantically weak features are aggregated with the strong features to enhance the contextual semantic information. The compact backbone with a small number of Bayesian Gabor modules is designed for memory and computation efficiency.

B. Inference for Bayesian Gabor Network

We introduce the proposed Bayesian modeling to Gabor filters. A Gabor neural network can be considered as a probabilistic model, which is parameterized by the Gabor parameters

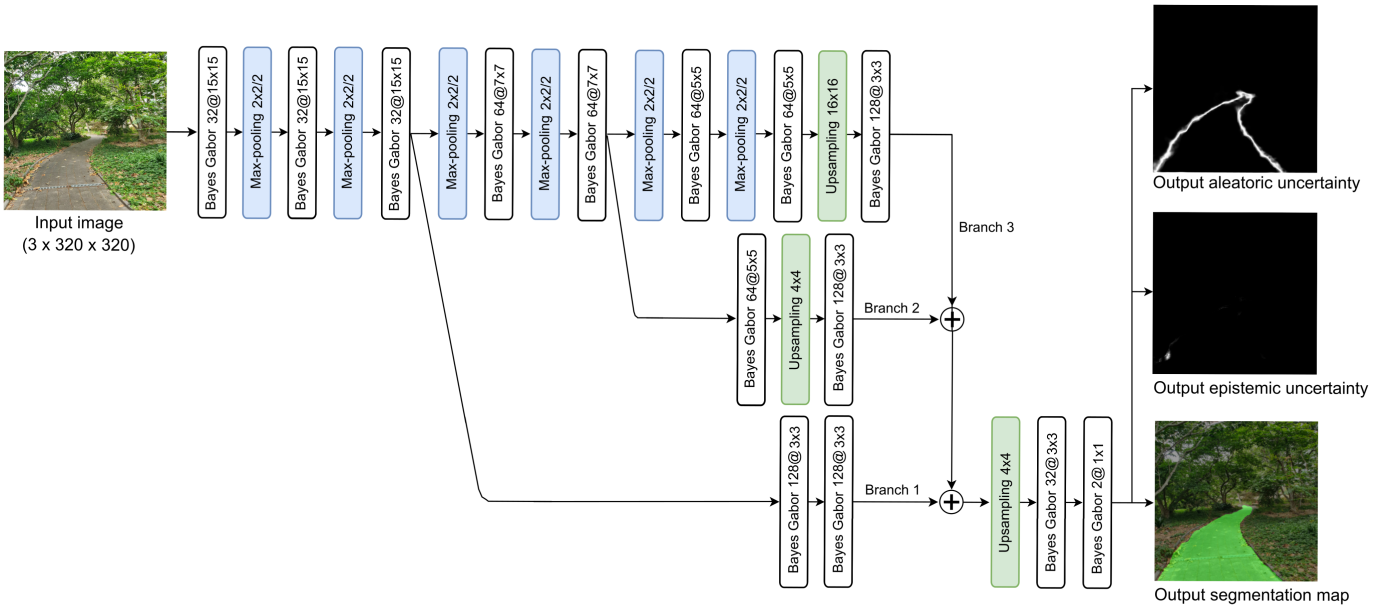

Fig. 2. The network architecture of BGN for pedestrian lane detection.

$\boldsymbol{\theta}$. Following the Bayesian inference method for pixel-wise classification, we can predict the output $\hat{\mathbf{y}}$ for a new pixel $\hat{\mathbf{x}}$ using the marginal likelihood, which is the distribution of the observed data $\mathscr{D}$ marginalized over the Gabor parameters $\boldsymbol{\theta}$:

$$p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \mathscr{D}) = \int p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \boldsymbol{\theta})\ p(\boldsymbol{\theta}|\mathscr{D})\ d\boldsymbol{\theta}. \tag{2}$$

Note that the predictive distribution (2) is intractable for the Gabor networks of any practical size, because the computation is equivalent to the high-dimensional integrals from an ensemble of possible Gabor parameters. Moreover, the true posterior distribution $p(\boldsymbol{\theta}|\mathscr{D})$ cannot be evaluated analytically.

To overcome this problem, we approximate the posterior distribution by a computationally tractable function, called variational distribution $q(\boldsymbol{\theta}|\boldsymbol{\omega})$, where $\boldsymbol{\omega}$ denotes mean and standard deviation of a Gaussian distribution. In other words, instead of using the point-estimate Gabor parameters θ, we parameterize the neural network with the probability distributions of the Gabor parameters. This can be done by minimizing the loss function $\mathscr{L}(\boldsymbol{\omega}, \mathscr{D})$, which is the Kullback-Leibler (KL) divergence between $q(\boldsymbol{\theta}|\boldsymbol{\omega})$ and $p(\boldsymbol{\theta}|\mathscr{D})$:

$$\begin{aligned}\mathscr{L}(\boldsymbol{\omega}, \mathscr{D}) &= \mathrm{KL}[q(\boldsymbol{\theta}|\boldsymbol{\omega}) \parallel p(\boldsymbol{\theta}|\mathscr{D})] \\ &= \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\omega})} \log q(\boldsymbol{\theta}|\boldsymbol{\omega}) - \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\omega})} \log p(\boldsymbol{\theta}) \\ &\quad - \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\omega})} \log p(\mathscr{D}|\boldsymbol{\theta}).\end{aligned} \tag{3}$$

The loss function $\mathscr{L}(\boldsymbol{\omega}, \mathscr{D})$ describing in Eq. (3) is also known as the variational free energy. It comprises three terms: the first two prior-dependent terms denoting the *KL loss* between $q(\boldsymbol{\theta}|\boldsymbol{\omega})$ and $p(\boldsymbol{\theta})$, and the last data-dependent term denoting the *negative log-likelihood loss*. Because the loss function involves the expectations with respect to the variational distribution $q(\boldsymbol{\theta}|\boldsymbol{\omega})$, it can be evaluated by using Monte Carlo sampling. Consequently, the loss function is obtained in the following tractable form:

$$\mathscr{L}(\boldsymbol{\omega}, \mathscr{D}) \approx \sum_{i=1}^{n} [\log q(\boldsymbol{\theta}_i|\boldsymbol{\omega}) - \log p(\boldsymbol{\theta}_i) - \log p(\mathscr{D}|\boldsymbol{\theta}_i)], \tag{4}$$

where n is the number of samples, and θ_i denotes the i-th sample drawn from $q(\boldsymbol{\theta}|\boldsymbol{\omega})$.

Since we work with mini-batch optimization, the loss can be summed over the random partitioning of mini-batches. For each mini-batch, we sample once from the variational distribution and then re-weight the KL loss. The exact loss form is given by

$$\mathscr{L}(\boldsymbol{\omega}, \mathscr{D}) \approx \sum_{i=1}^{N} [\lambda_i\ (\log q(\boldsymbol{\theta}|\boldsymbol{\omega}) - \log p(\boldsymbol{\theta})) - \log p(\mathscr{D}_i|\boldsymbol{\theta})], \tag{5}$$

where N is the number of mini-batches, and $\lambda_i = \frac{2^{N-i}}{2^N - 1}$ is the weight denoting the trade-off between the KL loss and the negative log-likelihood loss on the i-th mini-batch $\mathscr{D}_i$. That way, the KL loss significantly affects the first few mini-batches (when data are slight), and the negative log-likelihood loss greatly affects the later mini-batches (when more data are seen).

In Eq. (5), the prior distribution $p(\boldsymbol{\theta})$ can be evaluated using a scale mixture of two zero-mean Gaussian densities as in [41]. The prior over the Gabor parameters is computed as

$$p(\boldsymbol{\theta}) = \prod_j [0.5\mathscr{N}(\boldsymbol{\theta}_j|0, \sigma_1^2) + 0.5\mathscr{N}(\boldsymbol{\theta}_j|0, \sigma_2^2)], \tag{6}$$

where θ_j is the j-th Gabor parameter of the network, and $\mathscr{N}(\theta_j|0, \sigma_k^2)$ is the evaluation of the k-th Gaussian component at θ_j. The first mixture component has a large standard deviation $\sigma_1 > 1$, which provides a heavier tail than the normal distribution. While the second mixture component has a small standard deviation $\sigma_2 \ll 1$, so that the priori tightly concentrates around zero. This paper utilizes a weight of 0.5 to balance the two Gaussian components.

C. Training Bayesian Gabor Network

BGN can be considered as a non-deterministic mapping function. Training it involves learning the parameters of the

variational distributions $q(\boldsymbol{\theta}|\boldsymbol{\omega})$ instead of the Gabor parameters directly. During a forward pass, Gabor parameters are sampled from the variational distributions to evaluate the loss function. Note that the data-independent KL loss in Eq. (5) is evaluated layer-wise, while the data-dependent log-likelihood loss is computed at the end of the forward pass. During a backward pass, the gradients with respect to the variational parameters $\boldsymbol{\omega}$ (i.e., μ and σ) are calculated by the back-propagation algorithm and then updated by an optimizer.

To reduce the variance of the gradients caused by the stochastic sampling step in the forward pass, we utilize the *local reparameterization trick* [56]. Instead of sampling directly from the variational distribution, we draw a sample s from the parameter-free distribution $\mathcal{N}(0, I)$, then shift it by the mean μ and scale by the standard deviation σ. In practice, we represent the network with a learnable parameter ρ instead of σ directly, then transform ρ using a softplus function to obtain $\sigma = \log(1 + \exp(\rho))$. This ensures that the standard deviations of the variational distributions are always non-negative during the training process. Collectively, a posterior sample of the Gabor parameter can be computed as

$$\theta = \mu + s \otimes \log[1 + \exp(\rho)], \tag{7}$$

where $\otimes$ is the point-wise multiplication.

D. Uncertainty Quantification in Bayesian Gabor Network

Let $\hat{\mathbf{x}}$ be an input pixel, $\boldsymbol{\omega}^*$ be the optimized variational parameters after training, and $q(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \boldsymbol{\omega}^*)$ be the variational predictive distribution which approximates the predictive distribution $p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \mathscr{D})$. At segmentation inference time, the predictive variance for $\hat{\mathbf{x}}$ is computed as

$$\begin{aligned}\mathrm{Var}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}(\hat{\mathbf{y}}) &= \mathbb{E}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}([\hat{\mathbf{y}} - \mathbb{E}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}(\hat{\mathbf{y}})]^2) \\ &= \mathbb{E}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}(\hat{\mathbf{y}}^{\otimes 2}) - \mathbb{E}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}(\hat{\mathbf{y}})^{\otimes 2}, \end{aligned} \tag{8}$$

where $v^{\otimes 2} = v\ v^{\mathrm{T}}$ is the outer product of v with itself. Following the definition of expectation and the Fubini's Theorem [46], Eq. (8) can be decomposed into aleatoric and epistemic quantities as follows:

$$\begin{aligned}&\mathrm{Var}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}(\hat{\mathbf{y}}) \\ &= \int \mathrm{Var}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}})\ q(\boldsymbol{\theta}|\boldsymbol{\omega}^*)\ d\boldsymbol{\theta} \\ &\quad + \int [\mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}}) - \mathbb{E}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}(\hat{\mathbf{y}})]^{\otimes 2} q(\boldsymbol{\theta}|\boldsymbol{\omega}^*)\ d\boldsymbol{\theta}. \end{aligned} \tag{9}$$

The first term in Eq. (9) is the expectation of the variance of the predicted outputs. In other words, it denotes the aleatoric uncertainty regarding the input pixel $\hat{\mathbf{x}}$ (i.e., heteroscedastic aleatoric uncertainty), which refers to the inherent randomness of $\hat{\mathbf{y}}$. Note that, because $\hat{\mathbf{y}}$ is one-hot encoded, the variance $\mathrm{Var}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}})$ can be rewritten as

$$\begin{aligned}\mathrm{Var}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}}) &= \mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}}^{\otimes 2}) - \mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}})^{\otimes 2} \\ &= \mathrm{diag}\{\mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}})\} - \mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}})^{\otimes 2}, \end{aligned} \tag{10}$$

where $\mathrm{diag}\{\mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}})\}$ is the diagonal matrix with elements of the expected outputs. The aleatoric uncertainty captures noise inherent in the data.

The second term in Eq. (9) is the expectation of the difference between the predicted outputs and the averaged prediction that is caused by the variability of $\boldsymbol{\theta}$. In other words, it refers to the uncertainty in the Gabor parameters sampled from the variational distribution $q(\boldsymbol{\theta}|\boldsymbol{\omega}^*)$. This quantity is also known as the epistemic uncertainty. Since BGN can be considered as an ensemble of Gabor networks, the epistemic uncertainty captures the unawareness about which network generated the training data. This type of uncertainty can be reduced as the size of training data increases.

For the dense-pixel classification, we substitute $\mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\theta})}(\hat{\mathbf{y}}) = \mathrm{softmax}[f_{\theta}(\hat{\mathbf{x}})]$ into Eq. (9), and then sample from $q(\boldsymbol{\theta}|\boldsymbol{\omega}^*)$. Collectively, the expectations in Eq. (9) is approximated as

$$\begin{aligned}\mathrm{Var}_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}},\boldsymbol{\omega}^*)}(\hat{\mathbf{y}}) \approx \frac{1}{M}\sum_{i=1}^{M}[\mathrm{diag}(\mathbf{z}_i) - \mathbf{z}_i\ \mathbf{z}_i^{\mathrm{T}}] \\ + \frac{1}{M}\sum_{i=1}^{M}(\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^{\mathrm{T}}, \end{aligned} \tag{11}$$

where $\mathbf{z}_i = \mathrm{softmax}[f_{\theta_i}(\hat{\mathbf{x}})]$ is the softmax-generated vector in the i-th prediction, and $\bar{\mathbf{z}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{z}_i$ is the averaged vector. Here, M is the number of repetitive predictions for $\hat{\mathbf{x}}$.

Aleatoric and epistemic uncertainties allow us to evaluate the room for improvements (data vs. model). This has a major significance for the model design. For deployment, because blind users may not need the individual uncertainty quantities, we can combine the quantities to produce a final uncertainty map as Eq. (9). That way, blind users can trust the output segmentation map and decide which regions of the scene (with high uncertainties) should be avoided.

IV. Experiments and Analysis

This section presents the experiments and analysis, including lane data acquisition and annotation (Section IV-A), performance evaluation measures (Section IV-B), comparisons with Bayesian segmentation methods (Section IV-C), comparisons with feature-based lane detection methods (Section IV-D), comparisons with the state-of-the-art segmentation methods (Section IV-E), ablation study (Section IV-F), and discussion (Section IV-G).

A. Image Dataset

1) Data Acquisition: We acquired a new pedestrian lane dataset, named PLVP3, from real indoor and outdoor scenes. The images were taken at different times of day and in different weather conditions (e.g., rain, fog, and cloud cover). Many images have extreme lighting variations, such as strong shadows and low illumination. PLVP3 includes unmarked pedestrian lanes with various shapes, colors, textures, and surface materials. For quantitative performance evaluation, we created the ground-truth masks by manually annotating the lane regions. Since PLVP3 is specifically designed for the detection of pedestrian lanes, we exclude other objects of interest from the ground-truth segmentation masks (e.g., pitfalls, pedestrians, and stairs). The dataset has been manually

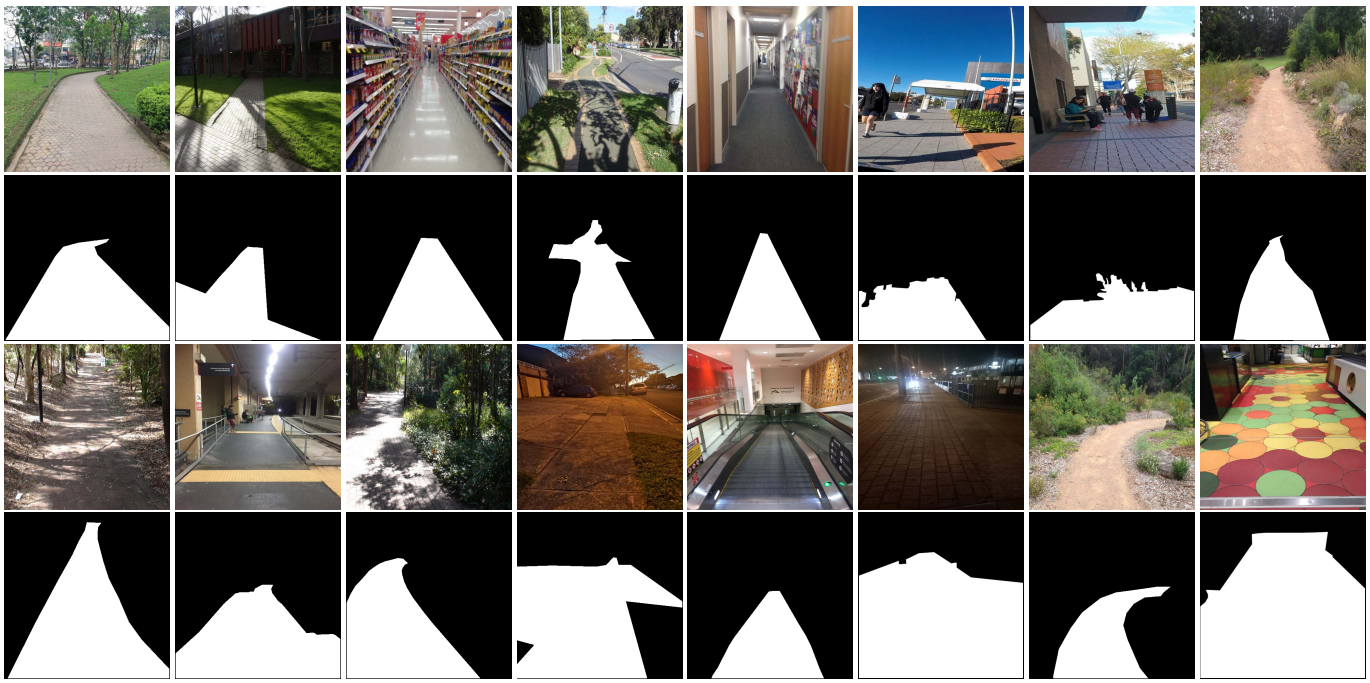

Fig. 3. Examples from PLVP3 dataset. Rows 1 and 3: Input colour images. Rows 2 and 4: The corresponding lane segmentation ground-truth.

annotated to include only the walkable pedestrian lane regions, which are free from obstacles and false positives. Examples of images and the corresponding ground-truth from the lane dataset PLVP3 are shown in Fig. 3.

Unlike the existing dataset of forest trails with various mountain scenes [57], PLVP3 aims to study the problem of perceiving pedestrian lanes in urban scenes (e.g., sidewalks, parks, shopping centers, and offices). They are the common scenes of daily living which support blind people in navigation. Furthermore, due to the objective of forest trail tracking for hikers and mountain bikers, the existing dataset includes several narrated video sequences with the GPS information, whereas PLVP3 consists of static color images.

2) Dataset Statistics: The lane dataset PLVP3 is extended from PLVP2 dataset, which has been previously introduced in [9]. In total, it has 10,000 labelled images, including 5,000 images from PLVP2 and 5,000 new images. Compared to PLVP2, we increased the percentage of images with concrete and pavement surfaces (48.6% and 11.6%), which are typical walking paths commonly found in urban scenes. We also collected more images with the normal lighting conditions (78.5%), so that models can learn the lane patterns reliably. Statistics of PLVP2 and PLVP3 datasets are presented in Table I. To the extent of our knowledge, PLVP3 is the largest public dataset for pedestrian lane detection research in the literature.

3) Experimental Setup: The lane images are resized to the designed input shape of 320×320 pixels. To evaluate the performance measures, we employ the five-fold cross-validation scheme where the lane dataset is divided randomly into five equal-sized partitions. For each fold, one partition is used as the test set, and the remaining four partitions are used as the training set. This step is repeated five times for different choices of the test partition. Each training set is further split into 90% of the samples for training and 10% of the samples for validation. Collectively, each cross-validation fold consists of 7200 training images, 800 validation images, and 2000 test images.

TABLE I
STATISTICS OF PLVP2 AND PLVP3 LANE DATASETS

Description		PLVP2		PLVP3	
		#Images	%	#Images	%
Surface	Brick	1,558	31.16	2,917	29.17
	Concrete	2,335	46.70	4,860	48.60
	Pavement	431	8.62	1,164	11.64
	Indoor	432	8.64	734	7.34
	Others	244	4.88	325	3.25
Lighting condition	Normal	3,485	69.70	7,845	78.45
	Extreme	1,515	30.30	2,155	21.55

B. Performance Evaluation Measures

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

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

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

by the area of union between S and G:

$$\text{IoU} = \text{Jaccard}(S, G) = \frac{|S \cap G|}{|S \cup G|} = \frac{TP}{TP + FP + FN}, \quad (12)$$

where TP, FP and FN refers to the number of true positives, the number of false positives, and the number of false negatives, respectively.

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

$$\text{F1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2\,TP}{2\,TP + FP + FN}. \quad (13)$$

Here, recall is the percentage of actual lane pixels that are detected correctly, and precision is the percentage of the machine-detected lane pixels that are actually correct.

2) Model Calibration Metrics: We use the *Expected Calibration Error* (ECE), which is a standard confidence calibration metric for classification models [58], [59]. A well-calibrated model produces predictive scores matching the accuracy; in other words the model must not be overconfident for incorrectly-classified pixels. To this end, we partition the predictive scores into K interval bins, and then calculate the gap between the accuracy and the average predictive score within a bin B_k. Let N be the total number of pixels and y_i be the true class label for the i-th pixel, the ECE can be computed as

$$\text{ECE} = \frac{1}{N} \sum_{k=1}^{K} \left| \sum_{i \in B_k} \mathbb{1}(\hat{y}_i = y_i) - \sum_{i \in B_k} \hat{p}_i \right|, \quad (14)$$

where $\hat{y}_i$ is the predicted class label, and $\hat{p}_i$ is the predictive score for the i-th pixel.

To evaluate the quality of model uncertainty estimation, we follow the approaches in [60], [61] and compute the *Area Under the Sparsification Error* (AUSE) metric. This measure reveals how well the estimated uncertainty matches the prediction error (the difference between the predictive score and the ground-truth). To obtain a so-called *sparsification plot*, we first sort the pixels according to the estimated uncertainties. An increasing percentage of the pixels is then removed, and the total errors of the remaining pixels are calculated. To obtain a so-called *oracle*, we sort and remove the pixels according to the prediction errors. The AUSE measure is defined as the area between the sparsification plot and its oracle. This paper utilizes the mean squared error (MSE) as the measure of error.

C. Comparisons With Bayesian Segmentation Methods

BGN is compared to *three* state-of-the-art Bayesian segmentation methods. The configurations for these methods are as follows:

1) *DL-HGP* [9]: This method is specifically designed for pedestrian lane detection. In this experiment, the SegNet backbone with five encoder/decoder units (total 26 convolutional layers) are employed for feature extraction. The number of inducing points and the initial number of local Gaussian Process experts are set to 50 and 9, respectively. We use the Python code provided by Nguyen *et al.* [9].
2) *Bayesian SegNet* [44]: We implement the network architecture as suggested in the reference, where the dropout layers are inserted to the central *six* encoders and decoders. The mean and the variance of the predictions are considered as the segmentation map and the uncertainty map, respectively. A dropout rate of 0.5 is used for every dropout layer.
3) *Bayesian DeepLabv3+* [45]: We implement this Bayesian model based on the Xception backbone as suggested in the reference. A Monte Carlo dropout is inserted into the middle flow of the backbone after every *four* Xception modules. The mutual information and predictive entropy are used to capture epistemic uncertainty and aleatoric uncertainty, respectively. We also set the dropout rate to 0.5.

All experiments are conducted on a computer with Intel Xeon Gold 5115 2.40 GHz processor and NVIDIA TITAN Xp GP102 graphics card.

Table II shows the performance of the evaluated methods. In terms of segmentation metrics, BGN achieves an mIoU score of 94.92%, which is higher than DL-HGP (by 2.31%) and the Bayesian SegNet (by 1.48%). The t-tests confirms that there is a statistically significant difference in the segmentation measures between our methods and these methods. Furthermore, BGN also maintains the baseline performance of Bayesian DeepLabv3+. The measures produced by our method are statistically similar to Bayesian DeepLabv3+.

In terms of computational complexity and inference time, BGN is significantly more efficient than other Bayesian methods. It achieves a giga floating-point operations per second (GFLOPS) of 0.15, which is 11,397× lower than DL-HGP. Its model size is only 1.18 MB, which is at least 95.3× smaller than other methods. A possible explanation for this finding is the computational efficiency of the Gabor modules due to their significantly small number of learnable parameters. BGN can operate at a speed of 68.027 images/s, which is 67.4× faster than DL-HGP, 4.2× faster than Bayesian SegNet, and 1.7× faster than Bayesian DeepLabv3+.

Next, the four methods are analyzed in terms of the quality of the estimated uncertainty. To produce uncertainty maps, we make 15 repetitive predictions for each input image. For BGN, the aleatoric uncertainty is combined with the epistemic uncertainty as Eq. (9) to obtain a single uncertainty map for comparison. For Bayesian DeepLabv3+, the predictive entropy is combined with the mutual information. Representative visualizations of uncertainty maps produced by different Bayesian methods are shown in Fig. 6. Clearly, the uncertainty maps produced by BGN are more meaningful than those produced by the other methods. Its significantly less-noisy uncertainty maps indicate that our model is more certain about its predictions. The regions of high aleatoric uncertainty are mostly along the lane boundaries and the lane surfaces far from the camera, where the presence of noise in the dataset is highly probable (Fig. 4). When the output segmentation map disagrees with the ground-truth, the aleatoric uncertainty map reflects a lack of confidence around the incorrectly-classified

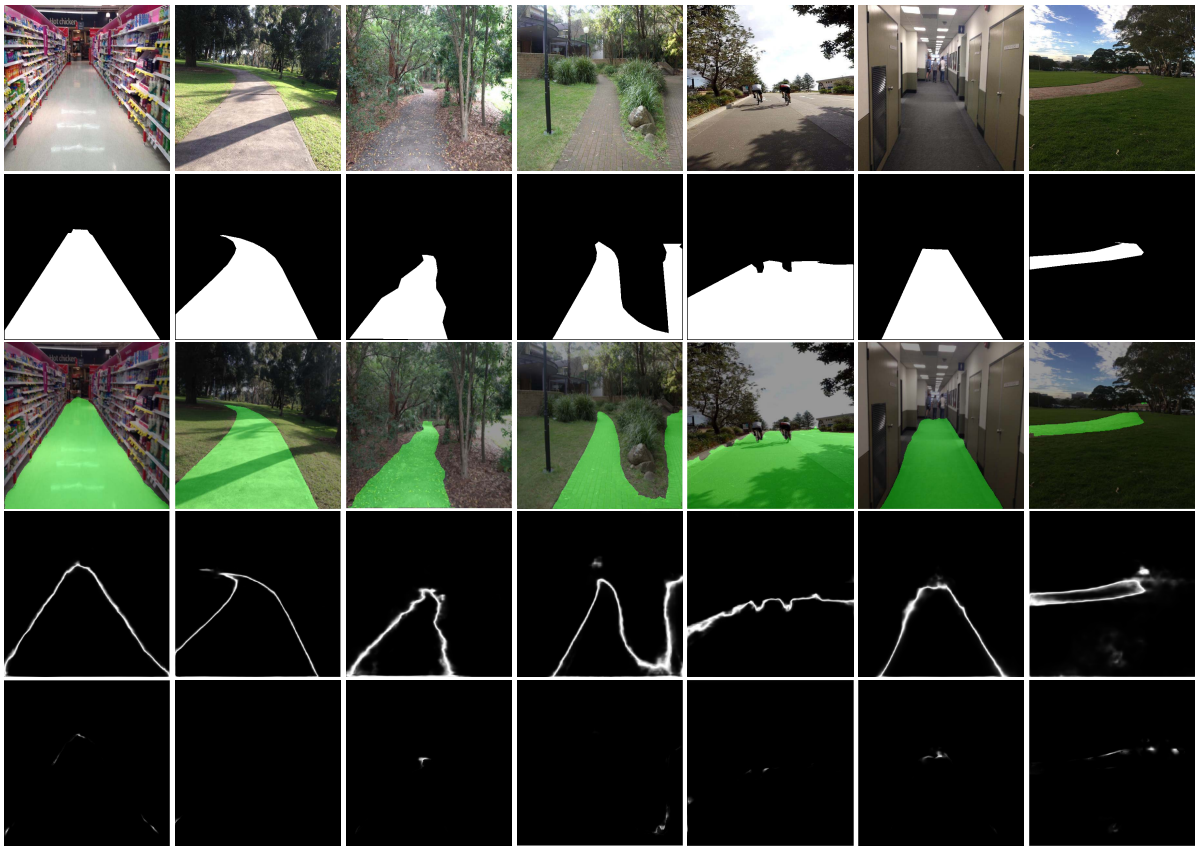

Fig. 4. Examples of pedestrian lane-detection results produced by BGN. A brighter intensity in the uncertainty maps indicates a higher uncertainty level. Row 1: Input images. Row 2: Ground-truth. Row 3: Output segmentation maps. Row 4: Output aleatoric uncertain maps. Row 5: Output epistemic uncertain maps.

TABLE II

PERFORMANCE OF BGN AND OTHER DL-BASED METHODS ON PLVP3 DATASET USING FIVE-FOLD CROSS-VALIDATION. WE COMPUTED TWO-SIDED STATISTICAL TESTS COMPARING BGN AND THE OTHERS

Method	mIoU↑	Accuracy↑	F1-score↑	ECE↓	AUSE↓	Images/s	#Params. (M)	Model size (MB)	GFLOPS
BGN (ours)	94.92	97.34	96.90	**0.0164**	**0.056**	**68.027**	**0.31**	**1.18**	**0.15**
DL-HGP [9]	92.61*	96.40*	95.17*	0.0259	0.063	1.008	29.43	113.36	1709.67
Bayesian SegNet [44]	93.44*	96.80*	96.34*	0.0308	0.285	16.181	29.41	112.46	125.10
Bayesian DeepLabv3+ [45]	94.86^{ns}	97.37^{ns}	96.84^{ns}	0.0261	0.087	37.878	54.75	219.83	40.21
SegNet (VGG-16) [30]	92.57*	96.45*	95.30*	0.0373	-	14.450	29.40	112.43	125.09
FCN-8s (VGG-16) [29]	94.88^{ns}	97.20^{ns}	96.81^{ns}	0.0496	-	27.778	14.72	56.24	86.60
U-Net [31]	93.64*	96.89*	96.42*	0.0285	-	34.722	7.87	30.02	7.97
DeepLabv3+ (Xception) [39]	94.99^{ns}	97.42^{ns}	97.28^{ns}	0.0207	-	40.322	54.70	219.83	40.21
PSPNet (ResNet-101) [32]	95.40*	97.76*	97.24^{ns}	0.0189	-	30.303	68.00	272.90	1065.43
HRNetv2-W32 [35]	94.67^{ns}	97.44^{ns}	97.20^{ns}	0.0190	-	40.742	9.98	40.28	14.53
HMSA [34]	**95.59***	**97.87***	**97.70***	0.0277	-	3.269	69.83	280.34	1860.14
CGNet-M3N21 [62]	95.13^{ns}	97.68^{ns}	97.29^{ns}	0.0352	-	54.78	0.49	2.18	1.35

(*): We reject the null hypothesis $\mathscr{H}_0 : m_{\text{BGN}} = m_{\text{other}}$ at a confidence level of 98%. That is, there is a significant difference compared to BGN.
(ns): We accept the null hypothesis. That is, there is no statistically significant difference compared to BGN.
The mark "-" means that the evaluation metric is not applicable. An up (or down) arrow indicates a higher (or lower) measure is better. The entire model is saved into a .PT file.

pixels. Furthermore, our method exhibits decreased epistemic uncertainty, which captures the ignorance about which model generated the training data. The low epistemic uncertainties reflect the reliability in the Gabor parameters due to a sufficiently large dataset used for training. Our finding agrees with [46], [63] in that a reliable segmentation model must provide low epistemic uncertainty and high aleatoric uncertainty for semantically challenging pixels. A quantitative evaluation of uncertainty estimation using the metric AUSE is discussed next.

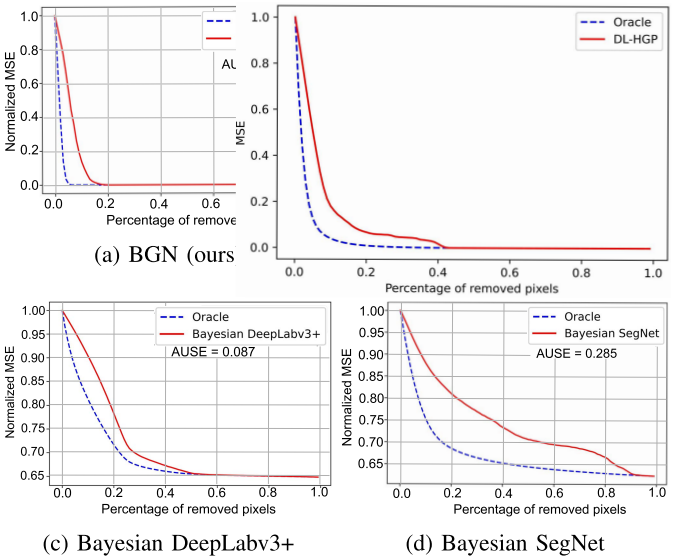

Fig. 5. Sparsification plots of different Bayesian segmentation methods over the five cross-validation folds. The plots show the normalized MSE for each percentage of pixels having the highest uncertainties removed. The oracles show the lower bounds by removing each percentage of pixels sorted by the prediction errors.

Table II shows that BGN achieves the best AUSE of 0.056 among the Bayesian methods. Fig. 5 presents the sparsification plots of the evaluated methods. The small gap between the oracle and the sparsification plot by BGN shows that its estimated uncertainties are usually consistent with the *normalized* MSE (i.e., the prediction errors), and the measure of uncertainty is well-calibrated. The results also indicate a low prediction error level in the oracle of BGN, where the total prediction errors of the remaining pixels reach 0 after removing 10% of the highest-error pixels.

Finally, we compare the Bayesian models in terms of model confidence calibration. In this experiment, the number of interval bins M is set to 10. Table II shows that BGN significantly outperforms the others; it has an ECE of 0.0164, which is 1.9× smaller than DL-HGP. It means that the average predictive score (within a bin) by BGN is usually consistent with the accuracy. Our model is not overconfident for the pixels having high predictive scores.

D. Comparisons With Feature-Based Lane-Detection Methods

In this experiment, BGN is compared to *two* representative traditional feature-based methods for pedestrian lane detection. These methods detect the boundaries of unmarked lanes by finding edges pointing to the vanishing point of the scene.

1) *Color & geometric based method* [5]: This method determines lane regions using the vanishing points combined with the geometric and color features of lane boundaries and surfaces. We use the MATLAB code provided by Phung *et* . [5].
2) *Edge based method* [17]: This method detects a vanishing-point constrained group of dominant edges based upon an orientation consistency ratio (OCR) feature. In this experiment, the number of orientationally consistent points for computing the OCR is set to 16. We use the MATLAB code provided by Kong *et al.* [17].

TABLE III

PERFORMANCE OF BGN AND TRADITIONAL LANE DETECTION METHODS ON PLVP3 DATASET USING FIVE-FOLD CROSS-VALIDATION

Method	mIoU	Accuracy	F1-score	Images/s
BGN (ours)	**94.92**	**97.34**	**96.90**	**68.027**
Geometric-based [5]	75.69	89.64	70.79	0.762
Edge-based [17]	63.12	79.23	37.04	0.009

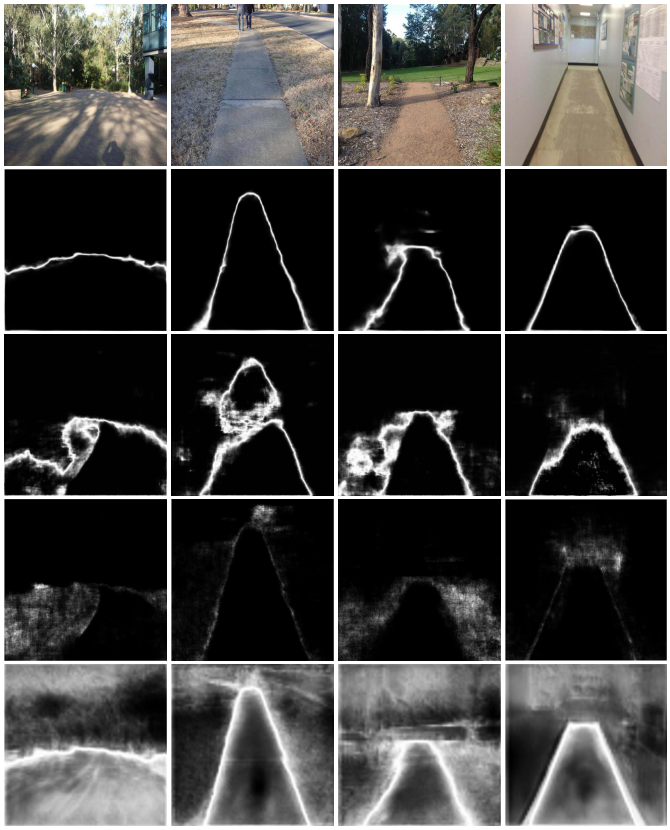

Fig. 6. Examples of uncertainty maps produced by different Bayesian segmentation methods. Row 1: Input images. Row 2: BGN (combined from both aleatoric and epistemic uncertainties). Row 3: DL-HGP [9]. Row 4: Bayesian SegNet [44]. Row 5: Bayesian DeepLabv3+ [45].

Table III shows that our method significantly outperforms the traditional lane-detection methods in terms of both segmentation performance and inference time. Compared to Method (1) and Method (2), BGN achieves an mIoU improvement of 19.23% and 31.8%, respectively. Our method also demonstrates at least an accuracy improvement of 7.7% and an F1-score improvement of 26.11%. The inference time of BGN is 89.3× faster than Method (1), and 6963.9× faster than Method (2). Since these feature-based methods are non-Bayesian and built on the fly, we do not compare the model size and the metrics of model calibration.

E. Comparisons With Non-Bayesian Segmentation Methods

In this experiment, BGN is compared to *eight* state-of-the-art non-Bayesian methods for generic semantic segmentation:

1) *DeepLabv3+* [39]: For a fair comparison with Bayesian DeepLabv3+, we also use the Xception backbone [40],

where the depthwise separable convolutions are applied to both the atrous spatial pyramid pooling (ASPP) and the decoder modules.

2) *SegNet* [30]: As with Bayesian SegNet, the VGG-16 network with 13 convolutional layers is also used as the encoder.
3) *FCN-8s* [29]: We implement this model based on the VGG-16 backbone as suggested in the reference. The final feature map is upsampled by a factor of 8 and then element-wise summed with the feature maps from the third and fourth pooling layers.
4) *U-Net* [31]: The encoder downsamples input image by a factor of 16 as suggested in the reference.
5) *PSPNet* [32]: The pre-trained ResNet-101 network is used as the encoder. A pyramid pooling module is applied then to fuse the features from four different pyramid scales. We use the GitHub repository provided by Zhao *et al.*[3]
6) *HRNet* [35]: We implement HRNetv2-W32, where the low-resolution representations are upsampled and then concatenated with the high-resolution representation. We use the GitHub repository provided by Wang *et al.*.[4]
7) *HMSA* [34]: We use the HRNet backbone with two scales (0.5× and 1×) for training and three scales (0.5×, 1×, and 2×) for inference as suggested in the reference.
8) *CGNet* [62]: The number of context-guided blocks in stage 2 and stage 3 are set to 3 and 21, respectively. We use the GitHub repository provided by Wu *et al.*.[5]

Table II shows that our segmentation performance is competitive with the others. Compared to HMSA, PSPNet, CGNet-M3N21 and DeepLabv3+, BGN shows a slight mIoU decrement of 0.67%, 0.48%, 0.21% and 0.07%, respectively. However, the *t*-tests indicate that there is no statistically significant difference between our mean measures and those of CGNet-M3N21 and DeepLabv3+. Compared to HRNetv2-W32, and FCN-8s, BGN produces a slight mIoU improvement of 0.25%, 0.04%, respectively. The *t*-tests also confirm that the null hypothesis of identical mean measures is rejected for SegNet and U-Net; in other words, our method is better than these methods statistically.

In terms of computational complexity, BGN significantly outperforms other methods. Compared to HMSA, BGN shows an improvement in both GFLOPS and model size with 12,400× lower and 237× smaller, respectively. Compare to the lightweight network CGNet-M3N21, BGN is 9× lower in GFLOPS and 1.8× smaller in model size. In terms of the inference time, our method shows a processing speed of 20.8× faster than HMSA and 1.6× faster than DeepLabv3+. The experimental results also indicate that BGN alleviates the overconfidence issue. The ECE produced by BGN is better than those of PSPNet and DeepLabv3+; the improvement in percentage is 13.2% and 20.7%, respectively.

[3]https://github.com/hszhao/PSPNet

[4]https://github.com/HRNet/HRNet-Semantic-Segmentation

[5]https://github.com/wutianyiRosun/CGNet

F. Ablation Study

In this section, we ablate important design elements in the proposed method. For a fair comparison, all examined models use the same network architecture as BGN.

1) Bayesian vs. Non-Bayesian: For ablations of the Bayesian inference, we design two non-Bayesian models: a Gabor network counterpart (GN) and a plain convolutional neural network counterpart (CNN). For the GN, we replace all Bayesian Gabor layers with standard Gabor layers. The GN is modeled by point-estimate Gabor parameters without Bayesian learning as in [21]. For the plain CNN, we replace all Bayesian Gabor layers with standard convolutional layers.

Ablations of the Bayesian inference approach are reported in Table V. The results demonstrate that BGN with Bayesian inference outperforms the non-Bayesian counterparts in terms of segmentation performance. Compared to the GN and the CNN, BGN produces an mIoU improvement of 2.37% and 1.95%, respectively. This indicates the effectiveness of using Bayesian inference to represent Gabor parameters with probability distributions. BGN can be considered as an ensemble of the GNs, and its final prediction is made by the committee. In terms of confidence calibration, BGN also significantly outperforms the CNN (1.5× smaller ECE).

2) Gabor vs. Convolution: To investigate the impact of the Gabor filtering approach, we compare BGN with two non-Gabor network counterparts: a Bayesian CNN (BCNN) and the plain CNN as described above. For the BCNN, we replace all Bayesian Gabor layers with Bayesian convolutional layers, which are introduced in [43]. The BCNN parameterizes the weights in each convolutional filter with probability distributions using variational Bayesian inference.

Table V shows that the segmentation performance produced by BGN is better than those of the plain CNN and BCNN (by 1.95% mIoU and 16.04% mIoU, respectively). In terms of inference speed, the CNN counterpart shows a processing time of 2.7× faster BGN. The results indicate that the computational cost of a single Bayesian Gabor module is more expensive than a single convolutional module. This is because the Bayesian Gabor module performs an additional Gabor computation before the convolutional computation. However, a lightweight BGN with a *small* number of Bayesian Gabor modules has a favorable speed-accuracy trade-off, compared to the *huge* CNNs.

3) Number of Bayesian Gabor Modules: To investigate the impact of the Bayesian Gabor modules, we analyze two variants of BGN, named BGN2 and BGN3, where the number of filters in every Bayesian Gabor layer (except the output branch) is doubled and halved, respectively. Table IV presents the detailed architect specifications of the BGN variants.

Table V shows that BGN produces the best mIoU and F1-score among the examined BGNs (94.92% and 96.90%, respectively), whereas BGN2 achieves the best accuracy (97.51%). However, the *t*-tests indicate that there is no statistically significant difference between the mean measures of BGN and BGN2. The simplified model BGN3 with very small number of Bayesian Gabor modules has the most compact model size of 321.9 KB.

TABLE IV
DETAILED ARCHITECTURE SPECIFICATIONS OF THE BGNS

	BGN	BGN2	BGN3
Branch 1	$\begin{Bmatrix}\text{BayesGabor: } 32@15\times15\\ \text{Maxpooling: } 2\times2,\text{s2}\end{Bmatrix}\times2$ BayesGabor: 32@15×15 $\{\text{BayesGabor: } 128@3\times3\}\times2$	$\begin{Bmatrix}\text{BayesGabor: } 64@15\times15\\ \text{Maxpooling: } 2\times2,\text{s2}\end{Bmatrix}\times2$ BayesGabor: 64@15×15 $\{\text{BayesGabor: } 256@3\times3\}\times2$	$\begin{Bmatrix}\text{BayesGabor: } 16@15\times15\\ \text{Maxpooling: } 2\times2,\text{s2}\end{Bmatrix}\times2$ BayesGabor: 16@15×15 $\{\text{BayesGabor: } 64@3\times3\}\times2$
Branch 2	$\begin{Bmatrix}\text{Maxpooling: } 2\times2,\text{s2}\\ \text{BayesGabor: } 64@7\times7\end{Bmatrix}\times2$ BayesGabor: 64@5×5 Upsampling: 4×4 $\{\text{BayesGabor: } 128@3\times3\}\times2$	$\begin{Bmatrix}\text{Maxpooling: } 2\times2,\text{s2}\\ \text{BayesGabor: } 128@7\times7\end{Bmatrix}\times2$ BayesGabor: 128@5×5 Upsampling: 4×4 $\{\text{BayesGabor: } 256@3\times3\}\times2$	$\begin{Bmatrix}\text{Maxpooling: } 2\times2,\text{s2}\\ \text{BayesGabor: } 32@7\times7\end{Bmatrix}\times2$ BayesGabor: 32@5×5 Upsampling: 4×4 $\{\text{BayesGabor: } 64@3\times3\}\times2$
Branch 3	$\begin{Bmatrix}\text{Maxpooling: } 2\times2,\text{s2}\\ \text{BayesGabor: } 64@5\times5\end{Bmatrix}\times2$ Upsampling: 16×16 BayesGabor: 128@3×3	$\begin{Bmatrix}\text{Maxpooling: } 2\times2,\text{s2}\\ \text{BayesGabor: } 128@5\times5\end{Bmatrix}\times2$ Upsampling: 16×16 BayesGabor: 256@3×3	$\begin{Bmatrix}\text{Maxpooling: } 2\times2,\text{s2}\\ \text{BayesGabor: } 32@5\times5\end{Bmatrix}\times2$ Upsampling: 16×16 BayesGabor: 64@3×3
Output	Upsampling: 4× 4 BayesGabor: 32@3×3 BayesGabor: 2@1×1		

TABLE V
ABLATION STUDY ON THE BAYESIAN INFERENCE, GABOR FILTERING APPROACH, AND THE NUMBER OF BAYESIAN GABOR MODULES

Method	Bayesian inference	Gabor	mIoU↑	Accuracy↑	F1-score↑	ECE↓	AUSE↓	Images/s	#Params.	Model size	GFLOPS
CNN			92.97*	96.62*	96.07*	0.0255	-	**189.1**	1.4 M	5.6 MB	15.57
BCNN	✓		78.88*	88.71*	87.45*	0.0024	0.208	29.4	2.8 M	11.2 MB	15.57
GN		✓	92.55*	96.45*	95.85*	0.0047	-	80.6	256.8 K	1.0 MB	0.15
BGN	✓	✓	**94.92**	97.34	**96.90**	0.0164	0.056	68.1	308.1 K	1.2 MB	0.15
BGN2	✓	✓	94.85ns	**97.51**ns	96.74ns	**0.0132**	**0.047**	45.2	1.2 M	4.9 MB	0.31
BGN3	✓	✓	76.62*	87.67*	86.00*	0.0142	0.297	79.8	**77.2 K**	**321.9 KB**	**0.08**

(*): We reject the null hypothesis $\mathscr{H}_0 : m_{\text{BGN}} = m_{\text{other}}$ at a confidence level of 98%. That is, there is a significant difference compared to BGN.
(ns): We accept the null hypothesis. That is, there is no statistically significant difference compared to BGN.

4) Training Behavior: In this experiment, we investigate the training behavior of the examined networks. Figure 7 illustrates the validation mIoU of the networks during the training process. The non-Gabor networks (the plain CNN and BCNN) has a faster speed of convergence than BGN and BGN2, which often converge from the epoch of 40.

Compared to the GN, BGN demonstrates a significant improvement in both the mIoU metric and the speed of convergence. This indicates that the inclusion of Bayesian inference benefits the training process of the BGNs. Among the proposed BGNs, BGN3 has an unstable training behavior with a poor segmentation performance for the first 120 epochs. It may be prone to underfitting due to its too simple network configuration with regard to the data.

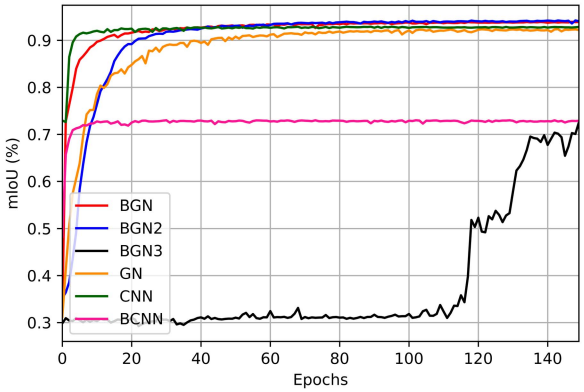

Fig. 7. Ablation study on training behaviour. The performance is averaged on the validation datasets of PLVP3 dataset.

G. Discussion

Compared to our previous work, DL-HGP [9], BGN has several conceptual merits. First, DL-HGP is a two-stage uncertainty estimation method, whereas BGN is a one-stage method. As a hybrid architecture, DL-HGP replaces the last 1×1 convolution of a full segmentation CNN, typically SegNet [30], with an HGP module. The CNN is used as a backbone to generate multi-dimensional features for each pixel in the input image, and the HGP (placed at the top of the CNN) is separately used for pixel-wise classification and uncertainty quantification. In contrast, BGN estimates uncertainty from its model weights without additional GP modules, and the KL loss is accumulated layer-wise in a single forward pass.

Second, the computational cost of BGN is lower than DL-HGP. Considering a single HGP module, its complexity in training time is $\mathscr{O}(NM^2)$, where N is the total number of image pixels in training set,[6] and M is the number of inducing points. Thus, increasing image size or dataset size leads to a rapid growth of computational cost. The choice of the

[6]For mini-batch optimization, the complexity is $\mathscr{O}(BM^2)$, where B is the total number of image pixels in a mini-batch.

TABLE VI
PER-CLASS IOU RESULTS ON CITYSCAPES *val* SET

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU	#Params.
BGN	95	**85.5**	90.7	44.6	59.1	66.6	68.8	77.4	89.2	55.4	92.6	**85.3**	68.1	92.6	73.8	81.7	81.6	**76.6**	72.3	76.6	**308 K**
BGN2	**97.4**	83.5	88.2	53.4	57.5	65.7	69.6	77.9	**91.7**	64	64.3	79.8	59.9	94.3	**86.9**	**90**	83.3	64.8	75.2	77.7	1.2 M
FCN-8s [29]	96.4	77.4	88.2	33.9	43.2	46.4	59.1	64	90.4	68.3	92.9	76.1	50.4	91.6	34.3	47.6	45.5	50.6	65.8	64.3	7.8 M
DeepLabv3+ [39]	95.7	84	90.9	**56.5**	60.7	**68.4**	**75.2**	**79.2**	91	70	92.8	85	70	93.4	75	87.9	80.9	70.8	**75.9**	79.1	54.7 M
PSPNet [32]	96.7	84.9	**91.5**	56.4	**61.7**	65.7	74.1	78.5	91.6	**70.2**	**93.3**	84.8	**69.9**	**94.4**	75.7	89.5	**87.3**	68.8	75.5	**79.5**	68.1 M

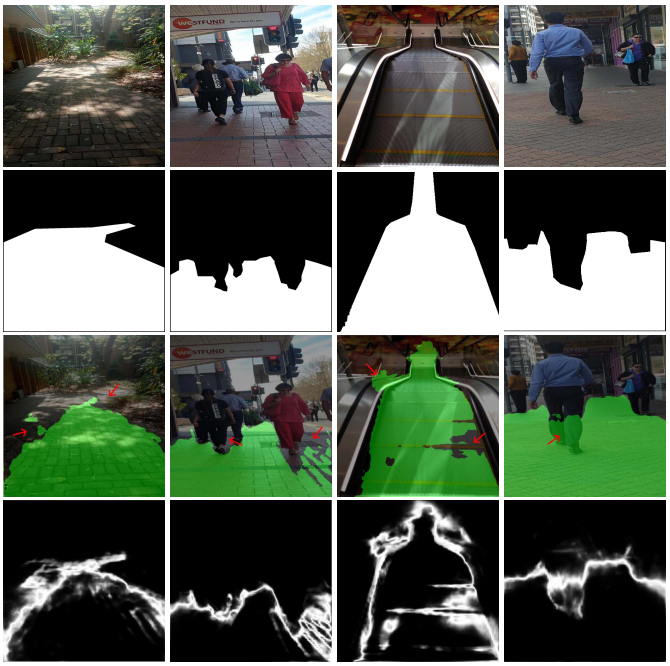

Fig. 8. Examples of failed segmentation by BGN. Row 1: Input images. Row 2: Ground-truth. Row 3: Output segmentation maps. Row 4: Output uncertain maps combined from both aleatoric and epistemic uncertainties.

hyper-parameter M is also a trade-off between the complexity and the approximation capability. In contrast, BGN has a simple CNN-like network architecture, so its complexity is comparable to the backbone of DL-HGP (excluding the HGP module). Note that the complexity of a typical CNN is just linear in the number of input pixels. The results in Section IV-C show that BGN outperforms DL-HGP in both segmentation accuracy and (especially) computational complexity.

Third, DL-HGP does not provide separate well-calibrated measures for aleatoric uncertainty and epistemic uncertainty. Its final uncertainty is the predictive variance using the Gauss-Hermite quadrature method. In contrast, BGN decomposes the predictive variance into aleatoric and epistemic quantities, which can be used to identify improvements.

Although BGN has shown several benefits, it could still be extended in two aspects. First, in our approach, Gabor filtering is utilized as the prior knowledge instead of randomly-initialized weights as convolutional kernels. Assuming the form of kernels endows BGN with a small number of weights and an enhanced computational efficiency, which is particularly useful for the problems with limited computational resources or scarce data. However, this constraint could make the weights less flexible than those of CNNs, and the network may focus only on specific texture features due to the nature of the Gabor filters. Examples of failed segmentation are shown in Fig. 8. To protect blind users, the false positives can be addressed using other modules in the assistive navigation system, e.g. obstacle detection module and danger-zone detection module. In this paper, we mainly focus on the pedestrian lane detection module. Note that, although the model may produce segmentation errors, its uncertainty map still provides us with a useful confidence measure for the predictive outputs. Incorrectly-classified pixels usually correspond with the regions of high uncertainty, where the model is less certain about its predictions. This has a major significance for user safety.

Second, the results in Section IV-F show that the computational cost may grow rapidly when BGN goes deeper, as it repetitively performs the Gabor computation for every filter channel after updating the weights. Hence, it is worth to replace standard Gabor filtering with a fast Gabor computation technique, notably the 2-D complex Gabor filtering with kernel decomposition [64]. We leave this plausible extension of our approach for a future study.

V. CONCLUSION

This paper presents a new Bayesian method for the camera-based detection of pedestrian lanes in unstructured scenes. The steerable Gabor filtering with variational Bayesian inference are embedded within the cascaded layers to enhance the scale and orientation decomposition, and improve the computational efficiency. Instead of learning the Gabor parameters directly as in the traditional Gabor networks, BGN can be trained in an end-to-end manner to learn the probability distributions of the Gabor parameters. Compared to the state-of-the-art DL-based methods in computer vision, our approach shows a competitive segmentation performance while achieving a significantly compact model size, a faster operation speed, and a well-calibrated confidence measure. Furthermore, due to the nature of Bayesian inference, BGN provides two full-resolution maps of aleatoric uncertainty and epistemic uncertainty for the safety of vision-impaired users.

APPENDIX

The main aim of this study is to develop a cost-effective uncertainty estimation method for the detection of pedestrian lanes. However, besides the lane dataset PLVP3, we run

extensive experiments on the benchmark segmentation dataset Cityscapes to evaluate the proposed method.

Cityscapes is a widely-used segmentation dataset, covering a broad range of semantic classes for urban scene understanding [13]. In this experiment, we use the *fine* annotations with 2,975 training images and 500 validation images. We retain 19 common classes for evaluation, and exclude the too-rare classes as suggested in [13]. The segmentation performance on the validation set is reported. For the BGNs, the pre-trained weights on PLVP3 dataset are utilized.

Table VI shows the performance of the examined methods on Cityscapes *val* set. Unlike the results on PLVP3 dataset, BGN2 with more complex network configuration slightly outperforms BGN (+0.8% mIoU). It shows a favorable trade-off between mIoU and model complexity. Among the examined methods, PSPNet achieves the best mIoU of 81.9%. Compared to PSPNet, BGN2 produces a slight mIoU decrement of 1.8%, but it shows a significant improvement in the number of model parameters (57× smaller).

ACKNOWLEDGMENT

The findings herein reflect the work and are solely the responsibility of the authors.

REFERENCES

[1] M. Jeong, B. C. Ko, and J.-Y. Nam, "Early detection of sudden pedestrian crossing for safe driving during summer nights," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1368–1380, Jun. 2017.

[2] C. Siagian, C.-K. Chang, and L. Itti, "Mobile robot navigation system in outdoor pedestrian environment using vision-based road recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 564–571.

[3] M. S. Uddin and T. Shioyama, "Bipolarity and projective invariant-based zebra-crossing detection for the visually impaired," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Sep. 2005, p. 22.

[4] M. C. Le, S. L. Phung, and A. Bouzerdoum, "Pedestrian lane detection for assistive navigation of blind people," in *Proc. Int. Conf. Pattern Recognit.*, 2012, pp. 2594–2597.

[5] 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.

[6] J. H. Yoo, S.-W. Lee, S.-K. Park, and D. H. Kim, "A robust lane detection method based on vanishing point estimation using the relevance of line segments," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3254–3266, Dec. 2017.

[7] S. Se and M. Brady, "Road feature detection and estimation," *Mach. Vis. Appl.*, vol. 14, no. 3, pp. 157–165, Jul. 2003.

[8] V. Ivanchenko, J. Coughlan, and H. Shen, "Detecting and locating crosswalks using a camera phone," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–8.

[9] 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.

[10] R. K. Satzoda and M. M. Trivedi, "On enhancing lane estimation using contextual cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1870–1881, Nov. 2015.

[11] C.-B. Wu, L.-H. Wang, and K.-C. Wang, "Ultra-low complexity block-based lane detection and departure warning system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 582–593, Feb. 2019.

[12] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[13] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[14] C. Low, A. B. J. Teoh, and C. Ng, "Multi-fold Gabor, PCA, and ICA filter convolution descriptor for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 115–129, Jan. 2019.

[15] H. Hu, "Enhanced Gabor feature based classification using a regularized locally tensor discriminant model for multiview gait recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1274–1286, Jul. 2013.

[16] Z.-Q. Li, H.-M. Ma, and Z.-Y. Liu, "Road lane detection with Gabor filters," in *Proc. Int. Conf. Inf. Syst. Artif. Intell. (ISAI)*, Jun. 2016, pp. 436–440.

[17] 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.

[18] S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets, "Feature pyramid network for multi-class land segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 272–275.

[19] 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.

[20] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357–4366, Sep. 2018.

[21] H. T. Le, S. L. Phung, P. B. Chapple, A. Bouzerdoum, C. H. Ritz, and L. C. Tran, "Deep Gabor neural network for automatic detection of mine-like objects in sonar imagery," *IEEE Access*, vol. 8, pp. 94126–94139, 2020.

[22] H. U. Khan, A. R. Ali, A. Hassan, A. Ali, W. Kazmi, and A. Zaheer, "Lane detection using lane boundary marker network with road geometry constraints," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1823–1832.

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

[24] 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. Int. Conf. Adv. Conc. Intell. Vis. Syst.*, 2005, pp. 76–83.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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. 6230–6239.

[33] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–13.

[34] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," 2020, *arXiv:2005.10821*.

[35] J. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[36] 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.

[37] 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. 2018.

[38] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.
[39] 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)*, Sep. 2018, pp. 801–818.
[40] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
[41] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1613–1622.
[42] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," in *Proc. Int. Conf. Learn. Represent. Workshop Track*, 2015, pp. 1–12.
[43] K. Shridhar, F. Laumann, and M. Liwicki, "Uncertainty estimations by softplus normalization in Bayesian convolutional neural networks with variational inference," 2018, *arXiv:1806.05978*.
[44] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *Proc. Brit. Mach. Vis. Conf.*, 2017, p. 57.
[45] J. Mukhoti and Y. Gal, "Evaluating Bayesian deep learning methods for semantic segmentation," 2018, *arXiv:1811.12709*.
[46] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation," *Comput. Statist. Data Anal.*, vol. 142, Feb. 2020, Art. no. 106816.
[47] H. Yao, L. Chuyi, H. Dan, and Y. Weiyu, "Gabor feature based convolutional neural network for object recognition in natural scene," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2016, pp. 386–390.
[48] Z. Yu, Y. Zhuge, H. Lu, and L. Zhang, "Joint learning of saliency detection and weakly supervised semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7223–7233.
[49] F. Huo, X. Zhu, L. Zhang, Q. Liu, and Y. Shu, "Efficient context-guided stacked refinement network for RGB-T salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Aug. 3, 2021, doi: 10.1109/TCSVT.2021.3102268.
[50] L. Huang, G. Li, Y. Li, and L. Lin, "Lightweight adversarial network for salient object detection," *Neurocomputing*, vol. 381, pp. 130–140, Mar. 2020.
[51] Y. Liu, Y.-C. Gu, X.-Y. Zhang, W. Wang, and M.-M. Cheng, "Lightweight salient object detection via hierarchical visual perception learning," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4439–4449, Sep. 2021.
[52] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 2, no. 7, pp. 1160–1169, 1985.
[53] I. J. Goodfellow, A. Courville, and Y. Bengio, "Scaling up spike-and-slab models for unsupervised feature learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1902–1914, Aug. 2013.
[54] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, pp. 369–371.
[55] 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.
[56] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2575–2583.
[57] A. Giusti *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 661–667, Jul. 2016.
[58] C. Guo and G. Pleiss, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 1321–1330.
[59] F. K. Gustafsson, M. Danelljan, and T. B. Schon, "Evaluating scalable Bayesian deep learning methods for robust computer vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1289–1298.
[60] O. M. Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow, "Learning a confidence measure for optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1107–1120, May 2013.
[61] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware CNNs for depth completion: Uncertainty from beginning to end," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12014–12023.
[62] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "CGNet: A light-weight context guided network for semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 1169–1179, 2021.
[63] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.
[64] J. Kim, S. Um, and D. Min, "Fast 2D complex Gabor filter with kernel decomposition," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1713–1722, Apr. 2018.

Hoang Thanh Le received the B.Eng. degree in computer science from Nha Trang University, Vietnam, in 2008, the M.Sc. degree in computer science from The University of Queensland, Australia, in 2012, and the Ph.D. degree in computer science from the University of Wollongong, Australia, in 2021. 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.

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 is currently a Professor with the University of Wollongong. He has also been a Visiting Senior Research Scientist at VinAI. He has published over 130 papers in journals and international conferences. His research interests include image and signal processing, neural networks, pattern recognition, and machine learning. He has served as the Chief Investigator for over 14 research projects funded by government agencies (research, defense, intelligence, foreign affairs, and trade) and industry. 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 of *Sensors*.

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

He has extensive experience in teaching, research, and academic leadership. He is currently an Associate Provost for Academic Affairs with Hamad Bin Khalifa University (HBKU), Doha, Qatar. Most recently, he served as the Head for the ICT Division, College of Science and Engineering, HBKU. 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 an Associate Dean (research) from 2007 to 2013. In 2015, he was promoted as a Senior Professor of computer engineering at UOW. His main research interests include signal and image processing, radar imaging, vision, machine learning, and pattern recognition. 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. He was a Distinguished Visiting Professor at several international institutions in France, USA, Germany, China, and New Zealand. He was a recipient of the Eureka Prize for Outstanding Science in Support of Defence or National Security in 2011, the Chester Sall Award of IEEE TRANSACTION ON CONSUMER ELECTRONICS in 2005, and a 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.