LDNet: End-to-End Lane Marking Detection Approach Using a Dynamic Vision Sensor

Farzeen Munir, *Graduate Student Member, IEEE*, Shoaib Azam, *Graduate Student Member, IEEE*, Moongu Jeon, *Senior Member, IEEE*, Byung-Geun Lee, *Member, IEEE*, and Witold Pedrycz, *Life Fellow, IEEE*

***Abstract*— Modern vehicles are equipped with various driver-assistance systems, including automatic lane keeping, which prevents unintended lane departures. Traditional lane detection methods incorporate handcrafted or deep learning-based features followed by postprocessing techniques for lane extraction using frame-based RGB cameras. The utilization of frame-based RGB cameras for lane detection tasks is prone to illumination variations, sun glare, and motion blur, which limits the performance of lane detection methods. Incorporating an event camera for lane detection tasks in the perception stack of autonomous driving is one of the most promising solutions for mitigating challenges encountered by frame-based RGB cameras. The main contribution of this work is the design of the lane marking detection model, which employs the dynamic vision sensor. This paper explores the novel application of lane marking detection using an event camera by designing a convolutional encoder followed by the attention-guided decoder. The spatial resolution of the encoded features is retained by a dense atrous spatial pyramid pooling (ASPP) block. The additive attention mechanism in the decoder improves performance for high dimensional input encoded features that promote lane localization and relieve postprocessing computation. The efficacy of the proposed work is evaluated using the DVS dataset for lane extraction (DET). The experimental results show a significant improvement of 5.54% and 5.03% in $F1$ scores in multiclass and binary-class lane marking detection tasks. Additionally, the intersection over union (IoU) scores of the proposed method surpass those of the best-performing state-of-the-art method by 6.50% and 9.37% in multiclass and binary-class tasks, respectively.**

***Index Terms*— Lane marking detection, event camera, attention network.**

Manuscript received 30 January 2021; revised 15 June 2021 and 3 July 2021; accepted 1 August 2021. Date of publication 19 August 2021; date of current version 8 July 2022. This work was supported in part by the Information Communication and Technology (ICT) Research and Development Program of MSIP/Institute of Information and Communications Technology Planning and Evaluation (IITP) (Development of Global Multi-Target Tracking and Event Prediction Techniques Based on Real-Time Large-Scale Video Analysis) under Grant 2014-3-00077, in part by the National Research Foundation of Korea (NRF) Grant through the Korean Government (MSIT) under Grant 2019R1A2C2087489, in part by the Ministry of Culture, Sports and Tourism (MCST), and in part by Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research and Development Program 2021 under Grant R2020070004. The Associate Editor for this article was S. Wan. *(Corresponding author: Moongu Jeon.)*

Farzeen Munir, Shoaib Azam, Moongu Jeon, and Byung-Geun Lee are with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea (e-mail: farzeen.munir@gist.ac.kr; shoaibazam@gist.ac.kr; mgjeon@gist.ac.kr; bglee@gist.ac.kr).

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada, also with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland (e-mail: wpedrycz@ualberta.ca).

Digital Object Identifier 10.1109/TITS.2021.3102479

I. Introduction

ADVANCEMENTS in the development of sensor technology have made a tremendous impact on autonomous driving in terms of environmental perception [1]. In the context of autonomous vehicles, the architecture mainly comprises a sensor layer, perception layer, planning layer, and control layer [2]. The sensor layer includes the integration of exteroceptive and proprioceptive sensors. The perception layer utilizes the information obtained through the sensor layer for environment understanding [3]. The decision from the perception is fed to the planning layer that devises the optimal trajectories for the autonomous vehicle [2]. Finally, the control layer is responsible for the safe execution of control commands applied to the vehicle through lateral and longitudinal control [4]–[6].

The primary goal is to understand the environment surrounding the autonomous vehicle through the fusion of exteroceptive and proprioceptive sensor modalities [7]. The perception of the surrounding environment includes many challenging tasks, for instance, lane extraction, object detection, and traffic mark recognition, which provides the foundation for the safety of autonomous vehicles as standardized by the Safety of the Intended Functionality SOTIF-ISO/PAS-21448.[1] The fundamental task in the hierarchy of perception is the extraction of lane information, as it assists an autonomous vehicle in precisely determining its position between the lanes. Accurate lane extraction forms the basis for the robust plans of autonomous vehicles, which includes lane departure and trajectory planning.

In the literature, much promising research has been proposed based on either using handcrafted features or using an end-to-end deep neural network for lane detection using conventional frame-based RGB cameras [8]–[11]. Conventional frame-based RGB camera performance is limited in various extreme and complex scenes [12]. For instance, by using conventional frame-based RGB cameras, the variation in illumination conditions can affect the performance of the lane detection algorithm because of unclear scenes in the input. Moreover, motion blur is typical for frame-based images when acquired from moving vehicles. The development of event cameras offers a promising solution to overcome uncertainty in conventional frame-based cameras caused by capturing the image at regular intervals. Event cameras capture per-pixel brightness changes, and each pixel streams the data asynchro-

[1] https://www.daimler.com/innovation/case/autonomous/safety-first-for-automated-driving-2.htm

Fig. 1. A sequence of images captured while coming out of a tunnel (T1-T2-T3-T4-T5). The top row shows the grayscale images, and the bottom row shows the corresponding event camera images. RGB cameras are highly affected by illumination variations due to their low dynamic range. The figure is borrowed from [13] to illustrate the difference between event cameras and frame-based RGB cameras.

nously. Compared to the frame-based camera, the event camera provides a significant advantage in higher temporal resolution, high dynamic range and less motion blur. Event cameras have two primary characteristics: i) a low latency rate and ii) a high dynamic range. An event camera captures the environment by the change in events, and its low latency rate helps generate the image faster than conventional frame-based cameras [12]. Additionally, this characteristic ensures that the image quality is not affected by motion blur. The high dynamic range of event cameras addresses the effect of illumination. Compared to conventional frame-based cameras having a dynamic range of 60 dB, event cameras (for the DET dataset CeleX-V) provide a high dynamic range of 120 dB that mitigates the illumination variation problem that appears in conventional frame-based cameras for lane detection [12], [14]. Fig. 1 illustrates the difference between event cameras and standard conventional cameras.

The perception of the environment plays an essential role in the architecture of the autonomous vehicle by determining the surrounding traffic entities, for instance, object detection. The inclusion of event cameras in the sensor suite of the autonomous vehicle provides an extra edge in the perception pipeline of the autonomous vehicle [15]. Changes in illumination, sun glare, and motion blur are detrimental to frame-based cameras and lead to performance degradation of the perception module and may lead to autonomous vehicle fatalities. Moreover, in contrast to frame-based cameras, event cameras with low latency can benefit the perception module [12]. Notably, different exteroceptive sensors have pros and cons, but their integration them for in the autonomous vehicles provides a complement to different sensor modalities. This redundant integration of sensors contributes to the safety of autonomous vehicles in the environment.

In this work, inspired by the utilization of event cameras in autonomous driving for lane detection tasks, as illustrated in [13], an encoder-decoder neural architecture is designed for lane detection using an event camera. The architecture of the network is composed of three core blocks: i) an encoder, ii) an atrous spatial pyramid pooling (ASPP) block [16], [17], and iii) an attention-guided decoder. The encoder of the proposed network is a combination of convolutional layers followed by a DropBlock layer and a max-pooling layer for the fast encoding of the input data. The ASPP block processes the encoded feature maps for the extraction of long-range features to ameliorate the spatial loss. The proposed decoder is based on an attention-guided decoder and is followed by fully connected layers to produce the lane detection predictions. The generality of the proposed network is experimentally validated on an event camera dataset. The event camera dataset contains various lane types, for instance, a single solid line, a single dashed line, a parallel solid line and a dashed line, and parallel dashed lines, etc. In addition, different numbers of lanes in the event dataset are collected by driving on roads with various carriageways. The event dataset also includes scene sparsity in addition to lane diversity. Therefore, the dataset is collected in various traffic scenes, for instance, driving on overpasses and bridges and in tunnels and urban areas, etc. Furthermore, in real-world scenarios, the viewpoint of the image plays an essential role in scene understanding. In this context, the event camera dataset is collected by changing the camera's location to increase the dataset's intraclass variance. The proposed lane marking detection network (LDNet) trained on this dataset handles complex scenes by incorporating the generalization of the scene sparsity and lane sparsity. The experimental evaluation of the proposed method is extensively tested on the event camera dataset, the dynamic vision sensor (DVS) dataset for lane extraction (DET), for multiclass and binary-class lane detection tasks and evaluated using the F-measure ($F1$ score) and intersection over union (IoU) metrics. The proposed method achieves a significant improvement of 5.54% and 5.03% on the mean $F1$ scores in the multiclass and binary-class tasks, respectively, surpassing the best-performing state-of-the-art method. In the case of the IoU scores, the proposed method surpasses the best-performing state-of-the-art method

by 6.50% and 9.37% in multiclass and binary-class tasks, respectively.

Moreover, an ablation study is conducted on the Carla-DVS dataset and Event-Segmentation dataset. Carla-DVS is a synthetic dataset collected using the open-source Carla simulator. The data consist of event data and binary labels for lane detection. The dataset is evaluated on the proposed algorithm and compared with other state-of-the-art algorithms. The LDNet is evaluated for generalization over the Event-Segmentation dataset.

In summary, the main contributions of this work are as follows:

1) The novelty of this work is in the design of a convolutional encoder-decoder network for lane segmentation using the event camera dataset. We studied the encoder-decoder architecture for the lane detection task using the frame-based RGB camera as a sensor modality. We designed the encoder-decoder architecture for the novel application of lane detection using an event camera as the sensor modality based on the relevant literature. We present a detailed comparative analysis of our encoder-decoder framework and other state-of-the-art frameworks in Section II Related Work.
2) In this work, we have proposed a convolutional encoder-attention-guided decoder architecture in LDNet for lane marking detection using an event camera. The encoder architecture is composed of four convolutional layers followed by DropBlock layers to handle the event data lanes and scene sparsity. In the proposed method, the reason for using few convolutional layers is to ensure that the feature size computable (to avoid gradient explosion) because of the sparsity of event data. In addition, to retain the spatial resolution of the encoded features, a dense ASPP block is employed. The additive attention mechanism is utilized in the decoder part because of its better performance for high dimensional input encoded features that help improve lane localization and relieve postprocessing computation.
3) In the proposed work, we employed the ASPP module to retain the spatial resolution by increasing the receptive field. The ASPP allows capturing valuable features as well as objects at multiple scales. The novelty of ASPP in the proposed work is the application to the event camera dataset for lane marking detection. Since using deeper convolutional neural networks (CNN) causes loss of spatial information at multiple scales and due to the sparsity of input data, it reduces the network performance. Furthermore, in contrast to DeepLabv3, we used the deep ASPP block for the feature extraction for lane marking detection using the event camera data. The deep ASPP block enables learning the feature representation at multiple scales and is followed by the attention-guided decoder module for lane marking detection. We have evaluated our method against DeepLabv3 and achieved obtained improvement of 15.82% in the mean $F1$ score and 15.49% in the mean IoU score.

This work addresses the novel problem of lane detection using an event camera by designing a convolutional encoder-attention-guided decoder architecture. The design of the encoder network consists of four convolutional blocks followed by DropBlock layers to address the lane and scene sparsity. In addition, to retain the spatial resolution of the encoded features, we have employed the deep ASPP module. Finally, we have added the attention-guided decoder that helps the proposed method better generalize for the lane detection task and relieve post-processing computations. The efficacy of the proposed method is extensively evaluated on the DET dataset and showing better performance in contrast to other state-of-the-art methods. The remainder of the paper is organized as follows: Section II introduces the event camera and its principle of operation related to the proposed method. Section III covers the related work. Section IV discusses the proposed methodology. Section V focuses on the experiments and results. The experimental analysis is discussed in Section VI. The ablation study is performed in Section VII and finally, Section VIII concludes the paper.

II. Event Camera and Principle of Operation

Event cameras are operated asynchronously in contrast to traditional frame-based cameras. The event camera captures the change in brightness (events) for each pixel independently in addition to capturing dense brightness, as in frame-based cameras at a fixed rate. In event cameras, the light is sampled by considering the scene dynamics with no dependencies related to the external clock (for instance, 30 fps (frame per second)) for the viewed scene. When measuring brightness changes, event cameras generate the sparse signals that are asynchronous in space and time, usually encoding moving image edges. This enables the event camera to obtain an advantage over traditional frame-based cameras in terms of high temporal resolution, low latency, low power computation and high dynamic range ((140 dB and 120 db) vs 60 dB of standard cameras) [12], [18].

The event camera generates the output in the form of events or spikes. The usability of these data to the convolutional neural network is to transform it into an opposite representation (for instance, images). In this context, the stream of event data is converted to an image where independent pixels correspond to a change in brightness, specifically, the logarithmic brightness signal $H(\mathbf{v_i}, t_i) \doteq logI(\mathbf{v_i}, t_i)$. For the pixel location $\mathbf{v_i} = (x_i, y_i)^T$ and time t_i, an event is recorded when the change in the brightness reaches the threshold (σ) [19]:

$$H(\mathbf{v_i}, t_i) - H(\mathbf{v_i}, t_i - \triangle t_i) \geq p_i \sigma \tag{1}$$

where $p_i \in \{-1, 1\}$ corresponds to the polarity of brightness change and $\triangle t_i$ represents the time since the last event triggered at location v_i. A sequence of events $E(t_N) = e_i{}_{i=1}^{N} = (x_i, y_i, t_i, p_i)_{i=1}^{N}$ is generated in the time interval $\triangle t_i$. Eq.1 represents the event generation model for the ideal sensor. In this work, we used the event dataset generated using the CeleX-V event camera [20]. In this camera, instead of polarity, a new event packet is introduced $E(t_N) = e_i{}_{i=1}^{N} = (x_i, y_i, t_i, a_i)_{i=1}^{N}$ where "a" corresponds to an in-pixel time-stamp or pixel logarithmic gray level value. The CeleX-V

encodes the events to the image representation by accumulating the event along the time interval $\triangle t_i$ that is set to 30 ms for this dataset.

The event camera is a new sensor modality in contrast to frame-based traditional cameras, requiring the same maturity level of research as conducted on frame-based traditional cameras. The challenge in utilizing the event camera relies on processing event data, as agreement on the best method for representing the events has not yet been reached [12], [21]. The processing of event data is performed based on the application. As the event camera operates on the illumination variations in the scene, the utilization of the event camera inside a static scene will limit its usability. Notably, when the event camera is placed on a stationary vehicle with respect to the road and the scene dynamics are constantly changing, the camera generates data according to changes in brightness regardless of the stationary position of the vehicle.

III. Related Work

In autonomous driving, lane detection serves as a fundamental component, and much research has been focused on the development of robust lane detection algorithms [22]. In the literature, two types of mainstream techniques have been used for lane detection: traditional image processing methods and deep learning-based segmentation methods [23], [24] [25].

A. Traditional Image Processing Methods for Lane Detection

Traditional vision-based lane detection methods follow pipelines that include image preprocessing, feature extraction, lane model fitting and lane tracking. In traditional approaches, image preprocessing is a necessary step in determining the quality of features for lane detection tasks. For this purpose, image preprocessing includes region of interest (ROI) generation, image enhancement for extracting lane information and removal of non-lane information. The extraction of ROIs is an efficient method for reducing redundant information by selecting the lower part of the image [26]–[28], and in some works, ROIs are generated using vanishing point detection techniques [29]–[31]. Reference [32] has proposed a global way to estimate dense vanishing points using dynamic programming for multiple lane detection with horizontal and vertical curves. Inverse perspective mapping (IPM) [33], [34] or warp perspective mapping [35] is used after ROI generation based on the parallel line assumption to reduce the effect of noise and to conveniently extract lanes. Lane enhancement is performed by using either color-based techniques or edge detection methods, such as hue-saturation-intensity (HSI) [36], luma, blue-difference and red-difference chroma components (YCbCr) [29], and lightness, red/green and yellow/blue coordinates (LAB) [37] as color-based models for transformation, and the Sobel operator [38], [39] and Canny detector [40], [41] as edge-based techniques. Hybrid methods comprising color and edges are also used in research [28]. ROI generation reduces the noise in images, but it is not robust to shadows and vehicles. Filters are used in some works to eliminate non-lane information [26], [42], [43]. In traditional approaches, lanes can also be modeled in the form of lines [42], [44], parabolas [39], [45], splines [24], [42], [46], hyperbolas [31], and so on. Reference [47] solved the lane detection problem by formulating it as a two-dimensional graph search problem. They designed a graph model that incorporates the continuous structure of lanes and roads. Furthermore, dynamic programming is used to solve the shortest path problem for the lane detection defined as the graph model. Additionally, tracking is used as the postprocessing step to overcome illumination variations. Kalman filtering and particle filtering are the most widely used approaches for tracking lane detection [35], [44]. In addition to tracking, the authors also utilized Markov and conditional random fields as a postprocessing approach for lane detection [48]. Reference [49] used a normal map for lane detection. The authors utilized the depth information for the generation of normal maps and used adaptive threshold segmentation for lane extraction.

The traditional image processing methods for lane detection are unreliable for event camera data due to the nature of the data. The event camera data are sparse and consist of spikes, which indicate a change in brightness at each pixel. It lacks color information and complex information of scenes present in frame-based RGB images. The aforementioned techniques cannot be directly applied to event camera data, such as edge detectors, line fitting, Hough transforms, etc. They require human supervision and fail to extract valuable features that would affect the robustness of the lane detection task.

B. Deep Learning-Based Segmentation Methods

Recent advances in neural network architectures have exhibited a tremendous impact on refining the extracted features for lane detection tasks. The fine-tuning step of traditional methods in ROI generation, filtering and tracking has been solved by the use of neural networks. The deep neural networks formalize the lane detection problem as a semantic segmentation task. The vanishing point guided network (VPGNet) is guided by vanishing points for road and lane marking detection [50]. Reference [51] proposed LaneNet, which performs detection in two stages: i) lane edge proposal generation and ii) lane localization. PolyLaneNet uses a front-facing camera for lane detection by generating the polynomials for each lane in the image via deep polynomial regression [52], [53]. In [54], the authors formulated lane detection as a row-based selection problem using global features. The use of row-based selection has reduced the computational cost of lane detection tasks. Moreover, the self-attention distillation (SAD) approach is also used in lane detection tasks that allow model self-learning with any additional labels [55]. Reference [56] used two cascaded neural networks in an end-to-end lane detection system. Reference [57] proposed a lane line detection technique. The network consists of two parts. First, a simple module follows an encoder-decoder architecture that learns features and predicts reasonable lanes. To handle more complex scenes, a second multitarget segmentation module is developed based on Wasserstein generative adversarial network (GAN).

In the literature, the encoder-decoder architecture is broadly used for semantic segmentation using the image data. Lane marking detection, as the peculiar task of semantic

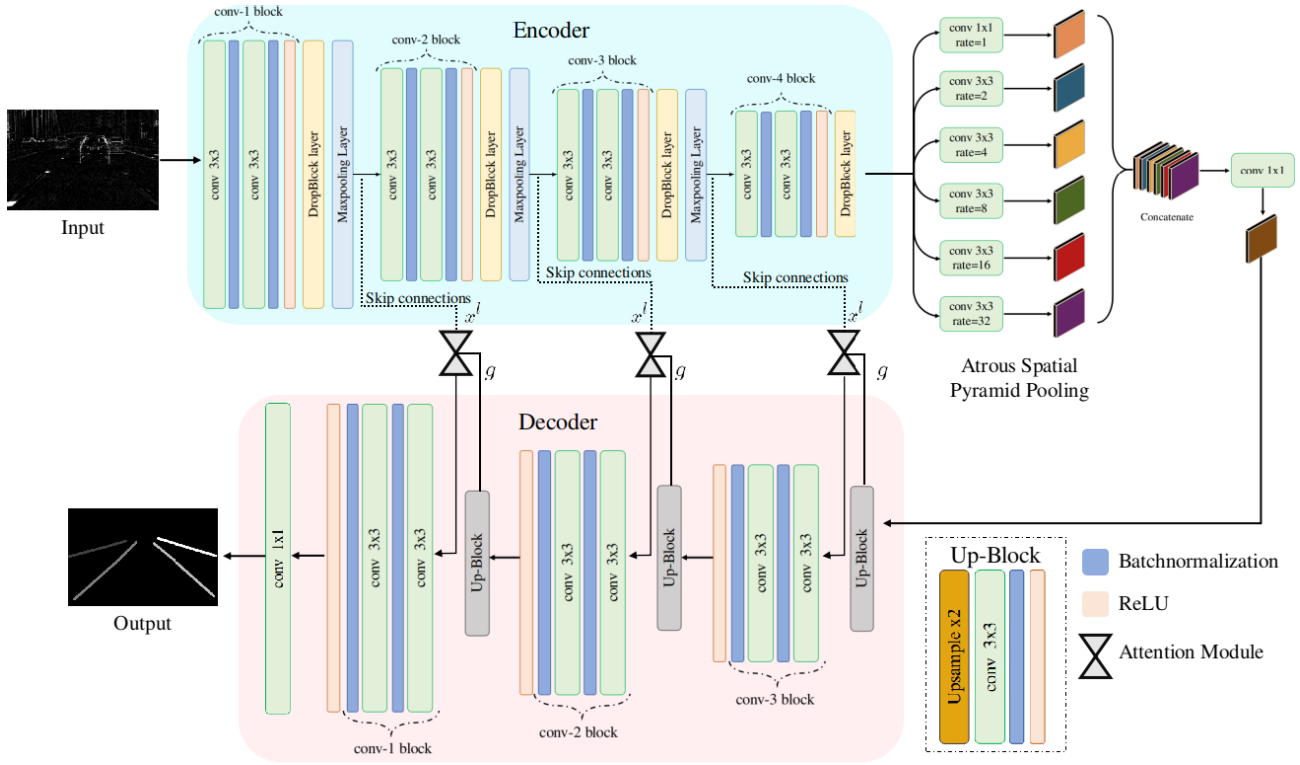

Fig. 2. The proposed LDNet architecture. The network is composed of three core components: i) a convolutional encoder to learn the feature representation, ii) a deep atrous spatial pyramid pooling (ASPP) block to retain the spatial resolution of encoded features, and iii) an attention-guided decoder. The attention-guided decoder is comprised of Up-block layers followed by the same convolutional block of the encoder part. The network takes an event camera image and predicts the lane marking detection.

segmentation, is used to classify the lanes in binary and multiclass categories. In the research domain, the most effective method to employ for lane marking detection is encoder-decoder architecture. In [58], the authors designed SegNet, an encoder-decoder architecture for image semantic segmentation. The encoder of the architecture consists of 13 convolutional layers inspired by the VGG-16 network followed by batch normalization, rectified linear unit (ReLU) activation function and a max-pooling layer. Each encoder has a corresponding decoding layer for upsampling the encoded feature map to full input resolution feature maps for semantic segmentation. The use of successive convolution results in spatial information loss, and due to sparsity of event data, in our proposed network, we use fewer convolution blocks with atrous spatial pooling to cater to information loss and improve lane detection. Reference [59] proposed an encoder-decoder network for lane detection that is similar to SegNet [58] architecture. They removed the pooling layers and fully connected layers from the Segnet architecture and used the low-resolution image to achieve real-time lane detection. Furthermore, they trained the network by generating their dataset as a set of points for the lanes from the TuSimple dataset. The reduction in the resolution of the input image in the case of event data causes significant information loss, which is unfavorable for a robust lane detector.

In addition, some works have employed two decoders and semantic segmentation in an encoder-decoder architecture. For instance, [60] used the VGG16 network as a base model for the encoder followed by dilated convolution layers and two separate decoders for the semantic and instance segmentation for the lane detection task. In [61], inspired by spatial pyramid pooling, the authors designed an encoder-decoder network. The main focus of their work is on designing the encoder network that includes an efficient dense module of depthwise separable convolution (EDD) and a dense spatial pyramid (DSP) module. For the decoder, they utilized bilinear interpolation and deconvolution for upsampling the encoder feature maps. Reference [62] designed an encoder-decoder network on the top of LaneNet architecture by replacing the LaneNet encoder with a sequential combination of atrous ResNet-101 and the spatial pyramid pooling (SPP) network. The decoder module consists of two decoder networks for embedding the feature map and binary segmentation map.

This work employed the encoder-attention-guided decoder architecture for lane marking detection using event camera data. In contrast to the abovementioned encoder-decoder architecture, we used the DropBlock layer with convolutional blocks to retain the network generality for the lane detection task. In the SegNet [58] architecture and [59], the spatial resolution is lost in the succession of the encoder architecture. We employed the ASPP module to retain the spatial resolution followed by the attention-guided decoder, which improves the localization of lanes. Moreover, in our work, the attention-guided decoder is beneficial to lane marking detection by not performing additional postprocessing steps and in contrast

to [60], [61] [62] optimizes the localization of features in the feature map.

C. Datasets for Lane Marking Detection

The most common datasets used by the traditional and deep learning-based methods include the Caltech dataset [23], TuSimple dataset [63], and CULane dataset [64]. These datasets are based on RGB images generated by conventional cameras. The change in illumination and motion blur in the images will affect the performance of the lane detection algorithm. Event cameras are a type of novel sensor that address the problem of standard cameras by having a dynamic range and low latency. In the literature, several event camera datasets have been published, including the Synthesized Dataset [14], Classification Dataset [65], Recognition Dataset [66], and Driving Dataset [67]. The aforementioned event camera datasets are for general purposes, and none of them are explicitly dedicated to the lane detection task. Additionally, these datasets have low spatial resolutions. The two main applications that have been published in the research on the event camera include steering angle prediction [68] and car detection [69]. Reference [13] proposed an event camera dataset for lane detection tasks. The authors evaluated their dataset with different lane detection algorithms, including DeepLabv3 [17], a fully convolutional network (FCN) [70], RefineNet [71], LaneNet [51] and a spatial convolution neural network (SCNN) [64], and published the benchmark for lane detection tasks using event cameras. In their lane detection benchmark, the SCNN [64] outperformed all the other algorithms and achieved a better mean IoU and mean $F1$ score. Inspired by [13], in this work, we use their dataset in the lane detection task, and the experimental evaluation of the proposed method surpasses the abovementioned benchmark in terms of both the mean $F1$ and IoU scores.

IV. Methods

In this section, we describe in detail the proposed framework for lane marking detection, as illustrated in Fig. 2. The framework consists of three modules: a convolutional encoder module, which extracts the features from the input image; a deep ASPP block to extract global features, and an attention-guided decoder module. In addition, skip connections are added from the encoder to the decoder to retain high-frequency spatial features.

A. Encoder

The CNNs outperform traditional computer vision and image processing techniques that incorporate handcrafted features for lane marking detection using standard RGB cameras [26]–[28]. However, lane marking detection with event cameras is a new research domain, and many state-of-the-art deep learning-based lane detection algorithms, such as SCNN [64], LaneNet [51], and FCN [70], are implemented on event camera images but require further improvements in robustness [13]. Fig.2 shows the design of the encoder for the event camera for lane marking detection. The related work gives an insight into the motivation of the designed encoder. We adopted and modified the encoder from the SegNet architecture [58]. SegNet uses convolution blocks similar to the VGG architecture [72]. In contrast to SegNet, the LDNet encoder has four operation blocks so that the feature map size is sufficient for ASPP to extract features, and on the other hand, it reduces the number of parameters in the LDNet encoder from 14.7M to 583.07 k. Each block consists of a convolution stack, a max-pooling layer and an additional DropBlock layer that improves the regularization of the LDNet. However, the last operational block does not include a max-pooling layer to match the filter size of the decoder. The encoder details of each layer are given in Table I.

TABLE I
THE DETAILED ARCHITECTURE OF THE PROPOSED LDNET

	Layer	Output size
	Input data	
Encoder	Conv-1 block	32*256*256
	DropBlock +Max pooling	32*128*128
	Conv-2 block	64*128*128
	DropBlock + Max pooling	64*64*64
	Conv-3 block	128*64*64
	DropBlock + Max pooling	128*32*32
	Conv-4 block	256*32*32
	DropBlock	256*32*32
Astrous spatial Pooling block	Conv 1X1 (rate=1)	256*32*32
	Conv 1x1 (rate=2)	256*32*32
	Conv 1x1 (rate=4)	256*32*32
	Conv 1x1 (rate=8)	256*32*32
	Conv 1x1 (rate=16)	256*32*32
	Conv 1x1 (rate=32)	256*32*32
	Concatenate + Conv 1X1	256*32*32
Decoder	Up-Block	128*64*64
	Attention Module	256*64*64
	Conv-3 block	128*64*64
	Up-Block	64*128*128
	Attention Module	128*128*128
	Conv-2 block	64*128*128
	Up-Block	32*256*256
	Attention Module	64*256*256
	Conv-1 Block	32*256*256
Output	Conv 1X1	5*256*256

Since the convolution stack of the encoder architecture is adopted from the VGG architecture [72], the convolutional layer parameters in terms of the receptive filter size and stride are kept the same as those of the VGG architecture: 3×3 and 1, respectively. To increase the detailed representation of low-level feature encoding, the convolution stack consists of two convolutional layers followed by batch normalization. A nonlinear activation function is employed after the second convolutional layer, which makes the decision function discriminative. Let x^l be the higher-dimensional image representation extracted from convolutional layers by progressively processing local features layer by layer. This process categorizes pixels in higher-dimensional space corresponding to their semantics. However, the model predictions are conditioned on the features extracted from the receptive field. For each convolutional layer l, a feature map x^l is obtained by sequentially applying a linear transformation realized by a nonlinear activation function. The ReLU function is chosen

as a nonlinear activation function, as shown in Eq.2:

$$\sigma(x_{i,c}^{l}) = max(0, x_{i,c}^{l}), \tag{2}$$

where c represents the channel dimension, i denotes the spatial channel dimensions and σ corresponds to the activation function. Eq.3 represents the feature map activation formulation.

$$x_{c}^{l} = \sigma_1(\sum_{c' \in F_l} x_{c'}^{l-1} * k_{c',c}), \tag{3}$$

$*$ represents the convolution operation, F_l is the number of feature maps in layer l, and k is the convolution kernel. The subscript i is ignored for notational clarity in the equation. The function $f(x^l; \Phi^l) = x^{l+1}$ is applied to convolutional layer l, where ϕ^l is a trainable kernel parameter. These parameters are learned by minimizing the objective function during training.

The DropBlock layer is introduced after each convolution stack in the operational block, as inspired by [73]. It is a structured form of dropout that is particularly efficient in regularizing the CNN. The notable difference between DropBlock and dropout is that DropBlock drops the contiguous regions from a feature map rather than random independent values. The pseudocode of DropBlock is illustrated as Algorithm 1. BS and γ are the two main tuning parameters. The BS represents the size of the block to be dropped, while γ is a control parameter for the number of activation connections to be dropped. DropBlock is not applied during evaluations, similar to dropout. A max-pooling layer is incorporated in each operational block to reduce the size of the feature map.

Algorithm 1: DropBlock Layer

Input: feature map obtained from convolutional layer x^l, BS, γ, $mode$
if $mode == evaluation$ **then**
  return x^l
end
Randomly generate mask $M : M_{i,j}\ Bernoulli(\gamma)$
For each zero position $M_{i,j}$, a spatial square mask is created with size equal to BS and centered at $M_{i,j}$
Set all the values inside the spatial square mask equal to zero
Apply the mask $x^l = x^l \times M$ Normalize the feature map $x^l = x^l \times count(M)/count - ones(M)$

B. Atrous Spatial Pyramid Pooling Block

In CNNs, reducing the receptive field size results in the loss of spatial information, which is associated with repeated usage of max pooling and strided convolution completed in the successive layers. One possible way to decrease the spatial loss is the addition of deconvolutional layers [70], [74], but it is computationally intensive. The notion of atrous convolution was introduced by [16], [17] to overcome the spatial loss problem. The dilated convolution operation increases the receptive field without increasing the training parameters or feature map resolution. Table I gives the details of the ASPP module. Here, we used a deeper ASPP module than those in [16], [17], which helps LDNet in learning higher-dimensional features across the entire feature map and refining full-resolution lane marking detection for event data. The event data are sparse in nature, and reducing the feature map in successive convolution causes information loss. However, we used a deep ASPP module that helps to extract deeper features without reducing the feature map size.

The atrous convolution operation is employed for one-dimensional or two-dimensional input data. Considering one-dimensional input data first, an atrous convolution is formalized as the output $y[i]$ of the input signal $x[i]$ with a kernel filter $w[k]$ of length K, as shown in Eq.4:

$$y[i] = \sum_{k=1}^{K} x[i + r.k]w[k], \tag{4}$$

where r is the rate parameter that corresponds to the stride through which the input signal is sampled. The standard convolution is an atrous convolution with a rate of $r = 1$. The variable i represents the location on the output signal $y[i]$ when the atrous convolution having kernal filter $w[k]$ is applied on the input image $x[i]$. k represents the indices of the atrous convolution kernel. The increase in the rate parameter increases the receptive field of the feature map at any convolutional layer without the increase in computation power and number of parameters. It introduces $r - 1$ zeros in the consecutive filter values in the feature map, efficiently increasing the kernel size of the $K \times K$ filter to $K_d = k + (k-1)(r-1)$ without increasing the number of parameters or increasing the computational complexity. Therefore, it offers an effective mechanism to control the receptive field of view and find the best compromise between the localization of an object of interest and context assimilation. In this work, the feature enhancer module consists of a deep ASPP block. The feature map obtained from the encoder module is convolved with the deep ASPP block. It consists of six layers with a rate ranging from $r = 2^0$ to 2^5. The output from each layer is concatenated and given to the attention-guided decoder block.

C. Attention-Guided Decoder

The semantic contextual information is captured efficiently by the acquisition of a large receptive field, and for this step, the feature map is gradually downsampled in a typical CNN. The features on the coarse spatial grid model the location and their relationship with different features at the global level. However, reducing false-positive predictions for small objects with large variability is a challenging task. In this work, we propose a novel attention-guided decoder. Generally, in the literature, attention (additive [75] and multiplicative [76]) and self-attention are used. In the proposed method, we have used additive attention [75] to transfer the information from the encoder to the decoder. Choosing this attention in the proposed method enhances the feature representation and localization that progressively reduces the feature response in unrelated background regions without extracting the ROI. In this attention mechanism, the decoder neurons receive the additional

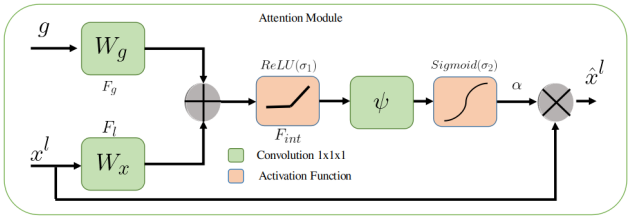

Fig. 3. The working operation of the attention-guided decoder is illustrated. The term g corresponds to the vector taken from the lowest layer of the decoder. In our case, it is taken from the Up-Block layer. x^l represent the encoded features from the encoder network.

input through attention from encoder states/activation providing more flexibility in terms of what to focus on from a regional basis when combined with gating signal from the coarsest scale activation map. In addition, the use of additive attention in comparison to multiplicative attention is employed because the additive attention tends to perform better for high dimensional input features.

We have not employed the self-attention mechanism in the proposed method because, in the self-attention mechanism, only the attention is applied within one component. The objective of the proposed method is to detect the lane marking, and for this task, an encoder-decoder architecture with the addition of the ASPP module as a spatial feature enhancer is employed. In the proposed method, if self-attention is employed, then the decoders usability is limited, or no decoder will be used, as in bidirectional encoder representations from transformers (BERT) [77] architecture.

The attention coefficient $\alpha_i \in [0, 1]$ in the attention-guided decoder distinguishes prominent image regions and prunes features from task-specific activations. The output of the attention module is the elementwise multiplication of attention coefficients and input feature maps described in Eq. 5:

$$\hat{x}^l_{i,c} = x^l_{i,c} \cdot \alpha^l_i, \tag{5}$$

For each pixel vector $x^l_i \in \mathbb{R}^{F_l}$, a single scalar attention value is calculated. Reference [78] proposed a multidimensional attention coefficient to learn sentence embeddings. Since lane marking detection is a multiclass problem, we utilize multidimensional attention coefficients to learn the semantic context in the image. Fig. 3 shows the attention module. The input vector $g_i \in \mathbb{R}^{F_g}$ determines the focus region for each pixel i. Eq.6 shows the additive attention formulation.

$$\begin{aligned} q^l_{att} &= \psi^T(\sigma(W_x^T x^l_i + W_g^T g_i + b_g)) + b_\psi, \\ \alpha^l_i &= \sigma_2(q^l_{att}(x^l_i, g_i; \Theta_{att})), \end{aligned} \tag{6}$$

Here, $\sigma_2(x_{i,c}) = \frac{1}{1+exp(-x_{i,c})}$ represents the sigmoid activation function. Θ_{att} characterizes a set of parameters including the linear transformation $W_x \in \mathbb{R}^{F_l \times F_{int}}$, $W_g \in \mathbb{R}^{F_g \times F_{int}}$, $\psi \in \mathbb{R}^{F_{int} \times 1}$ and bias term $b_\psi \in \mathbb{R}$, $b_g \in \mathbb{R}^{F_{int}}$. The term g represent the vector taken from the lowest layer of the network. The channel-wise $1 \times 1 \times 1$ convolutions compute the linear transformation for the input tensors, which is called "vector concatenation-based attention" and involves concatenating the features x^l and g and linearly mapping to a $\mathbb{R}^{Fint}$ multi-dimensional space [79]. There are three operational blocks in the attention-guided decoder. Each block consists of a convolutional stack that is similar to the encoder, an Up-Block layer to increase the feature map size, and the attention module. The Up-Block layer includes an upsampling layer followed by convolutional, ReLU and batch normalization layers. The attention module highlights the salient features that are carried through the skip connections, as shown in Fig 2. The features obtained at the coarse scale are used in gating to remove irrelevant and noisy skip connections. Gating is performed before concatenation to add only relevant activations, as in Fig. 3. A fully connected layer is added following the decoder module, which classifies each pixel in the feature map and is further compared with the corresponding ground truth to calculate the loss during training.

The proposed encoder-decoder network is jointly trained in end-to-end manner for lane marking detection. During the training, the loss is backpropagated to optimize the weight of the network. We incorporated the cross-entropy function given by Eq.7:

$$Loss = -\frac{1}{N}\sum_{N}^{j=1}\sum_{M}^{c=1} y_{c,j} \ln(\hat{y}_{c,j}) \tag{7}$$

where N is number of pixels in the ground truth and M is the number of classes. $y_{c,j}$ defines the ground truth of a pixel belonging to the correct class, and $\hat{y}_{c,j}$ defines whether the predicted pixel belongs to correct class c.

V. Experiments and Results

The effectiveness of the proposed method for lane marking detection in event camera-based images (DET dataset [13]) is evaluated using multiclass and binary-class labels. The results are compared with the state-of-the-art algorithm benchmark on the DET dataset. The proposed method is evaluated in terms of the mean $F1$ score and the mean IoU. The details are described below.

A. DET Dataset

In our experiments, we use the benchmark developed by [13]. A high-resolution dynamic vision sensor dataset for lane detection is published. The dynamic vision sensor is a type of event-based sensor that responds to variations in brightness. It does not follow the principle of frame-based RGB cameras, but individual pixels are incorporated in the sensor function individually and asynchronously, recording variations in brightness. The DET dataset is collected using a CeleX-V event camera with a resolution of 1280×800 mounted on a car. The dataset is recorded at different times of day and comprises various traffic scenes, such as urban roads, tunnels, bridges, and overpasses. The dataset also includes various lane types, such as parallel dashed lines, single lines, and single dashed lines. The DET dataset consists of a total of 5424 images with binary and multiclass labels. In the case of multiple classes, the labels are categorized into five classes, where four labels correspond to different lane types and the

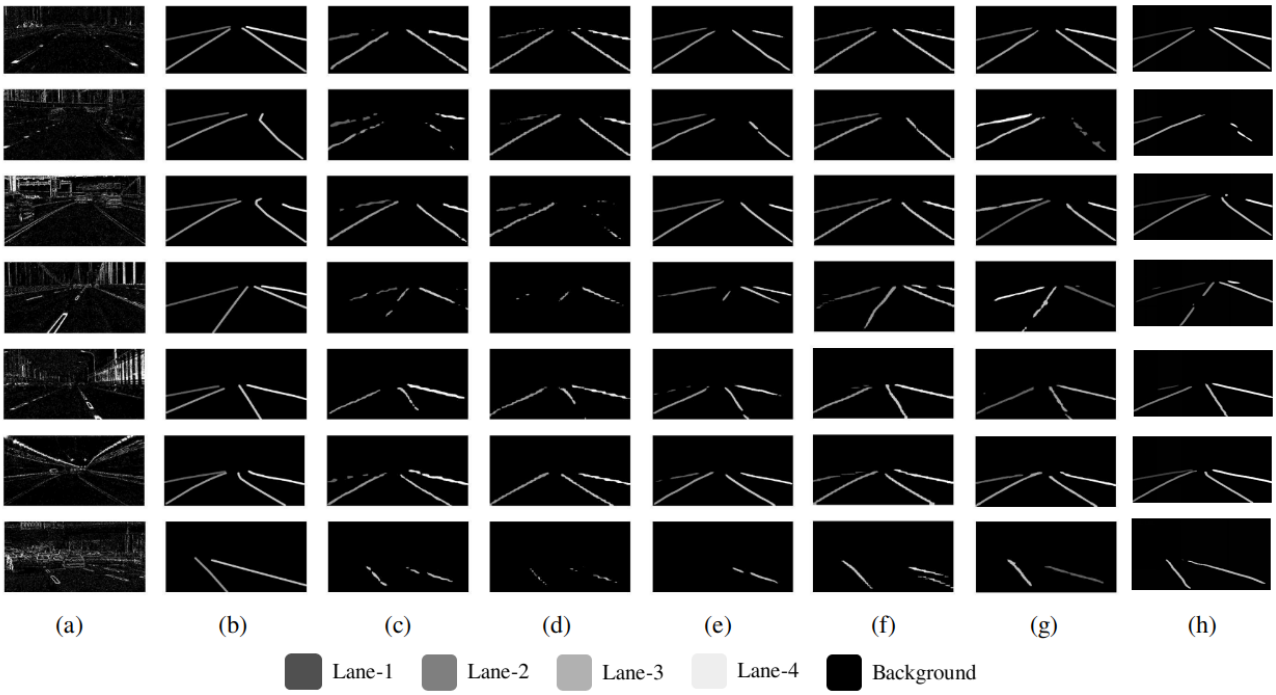

Fig. 4. The qualitative comparison between different lane detection methods using multiclass labels. There are 5 classes: background, lane-1, lane-2, lane-3 and lane-4. (a) shows the input image. (b) shows the ground-truth labels. (c-h) show the results for FCN, DeepLabv3, RefineNet, SCNN, LaneNet and LDNet (ours), respectively.

last label is for the background. In this work, we use both (lanes and background) types of labels to evaluate the proposed method. For the experimental evaluation, the dataset is split into training, validation and test data at percentages of 50%, 16%, and 33%, respectively.

B. Training Details

The proposed LDNet is implemented using the PyTorch deep learning library. The network is trained from scratch in an end-to-end manner. A 256×256 size image is input to the network. The input images does not have any preprocessing or filtering step in our network. The network learns to filter out white noise present in the images. The Adam optimizer is adopted for training the network with the learning rate schedule policy given by Eq.8. The initial learning rate of 5^{-4}; epsilon is set to 1^{-8}, and the weight decay is set to 1^{-4}. In Eq.8, the value of $power$ in training the network is set to 0.9. The tuning of neural network parameters involves heuristics, or some parameters are architecture-specific, but in the literature, some parameters are considered to have been perfected after years of studies. In this work, we conducted extensive experimentation and evaluation to determine the best parameters for the proposed method. In all of our experiments, we kept the same parameter settings.

$$LR = initial - LR \times (\frac{1 - epoch}{max - epochs})^{power}, \tag{8}$$

In addition, the DropBlock parameters mentioned in Algorithm. 1, i.e., BS and γ, are also fixed when training the proposed network. The value of BS is fixed at 5, whereas the γ value is determined by Eq.9 for controlling the features to be dropped during training.

$$\gamma = \frac{1 - kP}{BS^2} \frac{feat^2}{(feat - BS + 1)^2}, \tag{9}$$

where kP defines the probability of keeping a unit. In our experiments, the value of kP is linearly increased from 0.0 to 0.5. $feat$ denotes the feature map size. These hyperparameter values are inspired by [73] and were selected empirically by using grid search.

The proposed method is evaluated with both label categories, i.e., multi-class labels and binary class labels. However, in experimenting with both labels, the training parameters of the proposed network are kept the same. The training process runs for a total of 100 epochs, with batch size of 4 using PyTorch deep learning library on an Nvidia RTX 2060 GPU.

C. Evaluation Metrics

The research society has matured and standardized the evaluation metric for lane marking detection. The public frame-based lane detection benchmarks have utilized $F1$ and IoU scores to evaluate lane marking detection [80], [81]. Moreover, in the literature, to evaluate event camera segmentation [82] and lane marking detection [13], $F1$ and IoU scores are adopted. Notably, in evaluating the proposed LDNet, the image size is kept at 256×256. In this work, we have also used the mean $F1$ and IoU scores to evaluate the proposed

TABLE II

COMPARISON OF EVALUATION RESULTS OF LDNET WITH OTHER STATE-OF-THE-ART METHODS ON THE DET DATASET. THE MEAN $F1$ SCORES (%) AND MEAN IoUS (%) ARE USED AS EVALUATION METRICS FOR THE MULTICLASS LABELS. THE VALUES IN BOLD ARE THE BEST SCORES

Model	Mean $F1$ (%)	Mean IoU (%)
FCN [13]	60.39	47.36
DeepLabv3 [13]	59.76	47.30
RefineNet [13]	63.52	50.29
LaneNet [13]	69.79	53.59
SCNN [13]	70.04	56.29
LDNet-multiclass (ours)	**75.58**	**62.79**

TABLE III

COMPARISON OF THE EVALUATION RESULTS OF LDNET WITH OTHER STATE-OF-THE-ART METHODS ON THE DET DATASET. THE MEAN $F1$ SCORES (%) AND MEAN IoUS (%) ARE USED AS EVALUATION METRICS FOR THE BINARY CLASS LABELS. THE VALUES IN BOLD ARE THE BEST SCORES

Model	Mean $F1$ (%)	Mean IoU (%)
FCN	72.65	58.51
DeepLabv3	71.93	58.45
RefineNet	75.78	61.44
LaneNet	79.21	64.74
SCNN	80.15	67.34
LDNet-binary class (ours)	**85.18**	**76.71**

method. The $F1$ score is expressed in Eq.11:

$$F1 = 2 \times \frac{P \times R}{P + R}, \tag{10}$$

$$P = \frac{TP}{TP + FP}, \tag{11}$$

$$R = \frac{TP}{TP + FN}, \tag{12}$$

where TP, FP and FN represent the number of true positives, false positives, and false negatives, respectively. The IoU is given by Eq. 13:

$$IoU(S_m, S_{gt}) = \frac{\mathbb{N}(S_m \bigcap S_{gt})}{\mathbb{N}(S_m \bigcup S_{gt})}, \tag{13}$$

where S_m represents the predicted lane detection output and S_{gt} denotes the ground-truth labels. $\bigcap$, $\bigcup$, and $\mathbb{N}$ represent intersection, union and number of pixels, respectively. We evaluated the mean $F1$ and IoU scores for both multiclass and binary-class lane detection.

D. Results

The DET dataset is benchmarked on typical lane detection algorithms, which include the FCN [70], RefineNet [71], SCNN [64], DeepLabv3 [17] and LaneNet [51] algorithms. The FCN algorithm is one of the earliest works to perform semantic segmentation by classifying every pixel in an image. An end-to-end FCN is trained to predict the segmentation map. DeepLabv3 investigates ASPP by upsampling a feature map to extract dense and global features. RefineNet explores a multipath refinement network that extracts features along the downsampling process to allow high-resolution predictions using long residual connections.

However, LaneNet and SCNN were specifically designed for lane detection tasks. SCNNs achieve state-of-the-art accuracy on the TuSimple dataset [63]. They use slice-by-slice convolutions within feature maps to enable message crossing between pixels across rows and columns. LaneNet applies a learned perspective transformation trained on the images. For each predicted lane, a third-degree polynomial is fitted, and lanes are reprojected onto the images.

The aforementioned methods are considered baseline methods and compared with the proposed network. Table II and Table III show the evaluation of the proposed method to the baseline methods. LaneNet and SCNN outperform typical semantic segmentation algorithms such as FCN, DeepLabv3 and RefineNet. However, LDNet (the proposed method) outperforms the best-performing state-of-the-art SCNN with an improvement of 5.54% on the mean $F1$ score and 6.5% on the mean IoU for multiclass lane detection, and an improvement of 5.03% on the mean $F1$ score and 9.37% on the mean IoU for binary-class lane detection. This comparison provides insight into how the use of the ASPP module with an attention-guided decoder improves the detection of lane markings. It should be noted that no postprocessing step is utilized in our framework. Fig. 4 shows the qualitative results of the proposed algorithm with the baseline methods in multiclass lane detection.

We calculated the FLOPs (floating point operations per second) and number of parameters for the proposed method and the state-of-the-art methods. Table IV illustrates the computational cost in FLOPs and the number of parameters. The proposed network has $5.71M$ parameters and 12.49 GMac[2] FLOPs, second-best compared to other state-of-the-art algorithms. As the proposed LDNet has utilized the dense ASPP (the initial variant introduced by DeepLabV3) in an encoder-attention-guided decoder architecture, the proposed model has a lower computational cost and higher accuracy than DeepLabv3.

VI. EXPERIMENTAL ANALYSIS

In this section, we investigate the effect of the different factors (using a backbone network before the encoder network, the addition of the DropBlock layer and the attention-guided decoder) on the performance of the proposed method.

We experiment with the proposed network with a deeper encoder by utilizing six different backbone

[2]MACS is the abbreviation for the number of fixed-point multiply-accumulate operations performed per second. It is a measure of the fixed-point processing capacity of a computer. This amount is often used in scientific operations that require a large number of fixed-point multiply-accumulate operations. A GMACS: equal to 1 billion (= 10^9) fixed-point multiply-accumulate operations per second.

TABLE IV

COMPARISON OF THE COMPUTATIONAL COSTS OF THE PROPOSED LDNET AND OTHER STATE-OF-THE-ART METHODS IN TERMS OF FLOPS AND NUMBER OF PARAMETERS

Model	Number of parameters	FLOPS
FCN	132.27 M	62.79 GMac
DeepLabv3	39.05 M	30.91 GMac
RefineNet	99.02 M	46.51 GMac
LaneNet	0.526 M	0.64 GMac
SCNN	25.16 M	90.98 GMac
LDNet-binary class (ours)	5.71 M	12.49 GMac

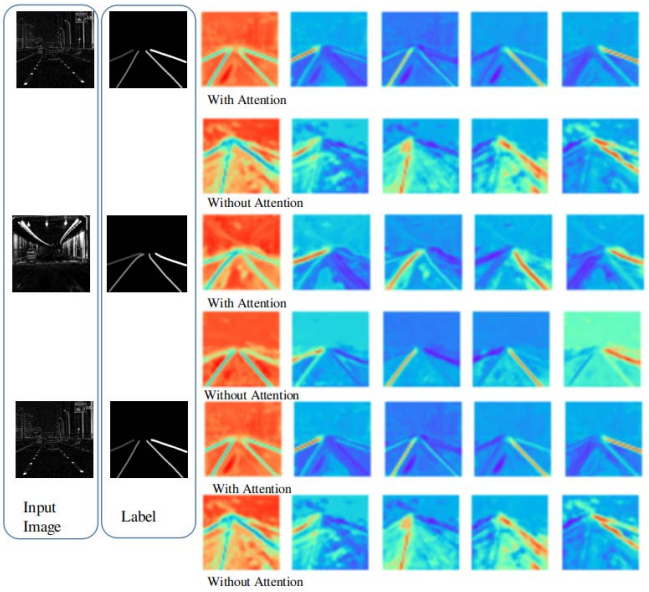

Fig. 5. The visualization of feature activation with and without the attention module in the decoder. The input image and the corresponding labels are also shown. The orange color shows the predicted class, and blue is the nonpredicted class. The first row from left shows the predicted background, lane 1, lane 2, lane 3 and lane 4.

networks: VGG16 [72], ResNet-18 [83], ResNet-50 [83], MobileNetV2 [84], ShuffleNet [85] and DenseNet [86]. The image is fed to the backbone network, and the feature map is given to the proposed encoder. The pretraining weights are used for the backbone networks. Table V shows the results when using the deeper encoder in the proposed network. The evaluation results show no significant gain from incorporating the backbone network compared to the proposed network. This finding justifies the use of shallow encoders in LDNet.

Table VI shows the evaluation of the proposed network with DropBlock, spatial dropout, and no dropout. The dropout layer is added to the network to regularize the network and to prevent overfitting. The addition of the DropBlock shows improved results on the test dataset compared to no dropout or spatial dropout. The contiguous regions in the feature map are highly correlated; dropping random units still allows information flow but is not efficient in regularizing the

TABLE V

QUANTITATIVE ANALYSIS OF LDNET WITH DIFFERENT BACKBONE NETWORKS. THE EXPERIMENTAL ANALYSIS IS PERFORMED ON BOTH THE MULTICLASS AND BINARY-CLASS TASKS AND THE RESULTS ARE EVALUATED IN TERMS OF THE MEAN $F1$ AND IoU SCORES

Model	multiclass		Binary Class	
	Mean $F1$	Mean IoU	Mean $F1$	Mean IoU
LDNet	**75.80**	**62.79**	**85.18**	**76.71**
LDNet-VGG16	74.42	61.16	83.75	74.98
LDNet-ResNet-18	73.92	60.56	84.127	75.62
LDNet-ResNet-50	74.48	60.20	84.71	76.124
LDNet-MobileNetv2	74.15	60.79	84.03	75.30
LDNet-DenseNet	74.90	61.69	84.11	75.62
LDNet-ShuffleNet	72.72	59.17	83.52	74.70

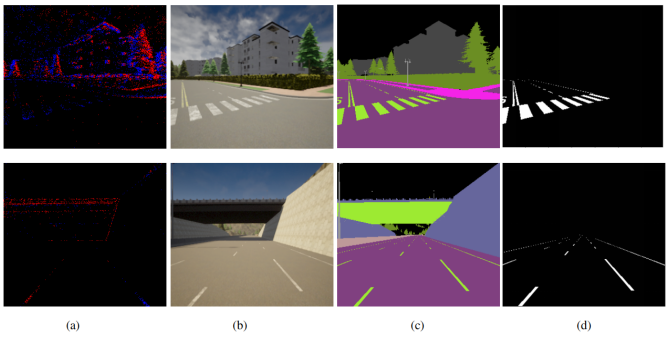

Fig. 6. The Carla-DVS dataset: (a) shows the event information. The blue and red colors in the event information show the increase and decrease of brightness of that specific pixel, respectively. (b) shows the frame-based RGB image, (c) shows the semantic ground-truth labels, and (d) shows the lane binary labels obtained from semantic labels.

network. The DropBlock helps the network retain semantic information required for lane marking detection.

Fig. 5 shows the visualization of the feature activations. The output of the LDNet is a feature map with 5 layers. Each layer predicts 5 classes, four labels corresponding to different lanes and the background. The orange color shows the class predicted in each image. The first row from left shows the predicted background, lane-1, lane-2, lane-3 and lane-4. The blue color shows the remaining pixels. The comparison between using an attention-guided decoder with a convolution decoder is illustrated. The attention-guided decoder shows improved localization of features, which eliminates the need for external localization of the features and postprocessing steps.

VII. ABLATION STUDY

A. Perfomance of LDNet on Carla-DVS Data

For the proposed LDNet method's efficacy, a synthetic dataset using the Carla open-source driving simulator is utilized [87]. The dataset consists of event data, semantic labels and frame-based RGB images. Fig.6 illustrates a sample of data that is used for the evaluation of LDNet. In the dataset, all the map data having different weather conditions are utilized. Furthermore, we sampled the data that contain the

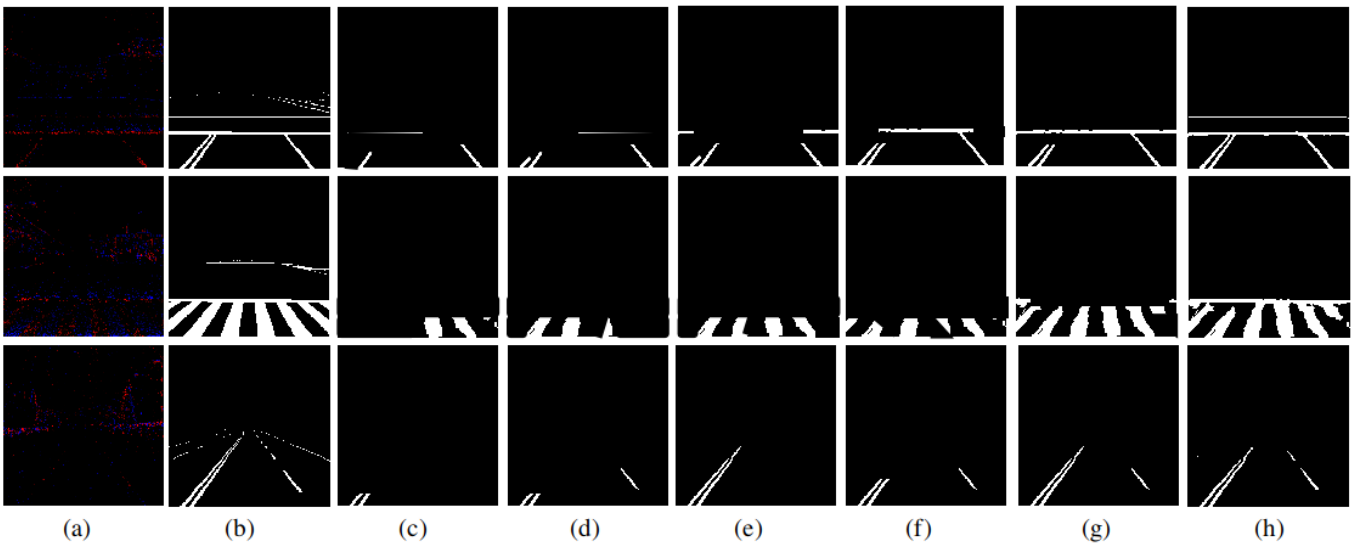

Fig. 7. Qualitative comparison between different semantic segmentation methods on Carla dataset (a) shows the input image. (b) shows the ground-truth image. (c-h) show the results for FCN, DeepLabv3, RefineNet, SCNN, LaneNet and LDNet (ours), respectively.

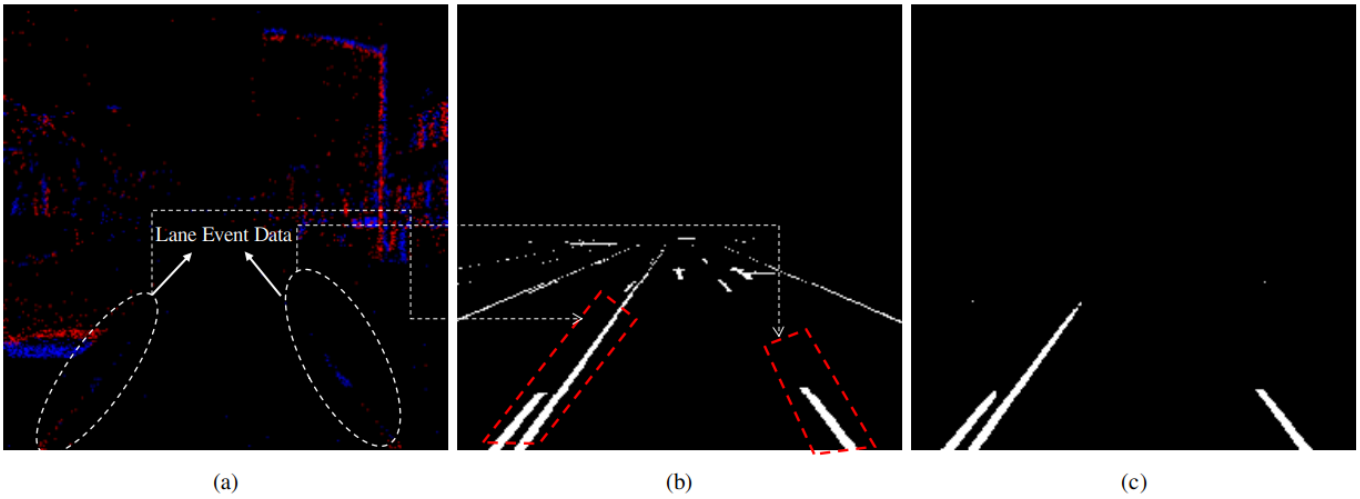

Fig. 8. Illustrations of the case-study of having no predictions by the proposed network in some regions. (a) explains the lane event dataset is available near the simulated vehicle. (b) shows the projection visualization that is learned by the network. (c) illustrates the prediction results by the proposed LDNet.

TABLE VI

QUANTITATIVE ANALYSIS ILLUSTRATING THE EFFECT OF DROPOUT AND THE DROPBLOCK ON LDNET. THE EVALUATION IS PERFORMED FOR BOTH MULTICLASS AND BINARY CLASS LABELS, AND THE MEAN $F1$ SCORE AND IoU ARE EVALUATED FOR EACH CASE AND LABEL

Model	multiclass		Binary Class	
	Mean $F1$	Mean IoU	Mean $F1$	Mean IoU
LDNet-no dropout	74.36	61.09	84.17	75.70
LDNet-dropout2d	72.47	58.94	83.25	72.80
LDNet-DropBlock	**75.80**	**62.79**	**85.18**	**76.71**

lane information for the evaluation of LDNet. The training details for LDNet are similar to the descriptions in section V-C with training samples of 2522 and test samples of 1220, respectively.

The quantitative evaluation of the proposed LDNet is performed with the same aforementioned lane detection algorithms on the Carla-DVS dataset. Table VII illustrates the mean F1 and IoU scores of the LDNet along with other state-of-the-art algorithms. Fig.7 shows the qualitative results of LDNet in comparison to the other algorithms.

The qualitative and quantitative results depict the efficacy of the proposed LDNet method, indicating that it surpasses the state-of-the-art lane detection algorithms. However, as the proposed method is applied on simulated data, it is assumed that the results will be better than a real-world dataset. In contrast, the results could be improved compared to the real-world dataset. To analyze this case-study, we review the dataset and find that the number of event data points is not sufficient at

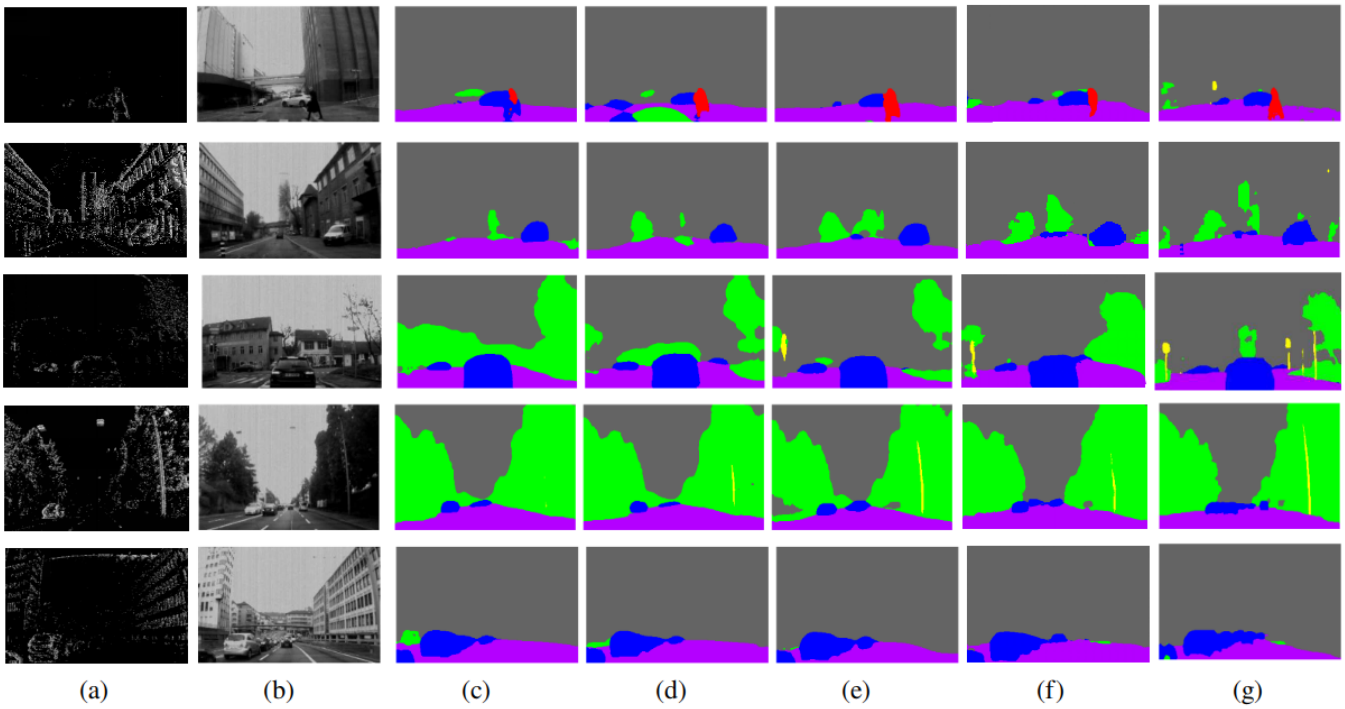

Fig. 9. The qualitative comparison between different semantic segmentation methods on the EV-Seg dataset, where (a) shows the input image and (b) shows the grayscale image. (c-g) show the results for Basic Dense encoding, Temporal dense encoding, EV-SegNet LDNet (ours) and the ground truth, respectively. In the semantic map, the blue color corresponds to the vehicle, the red color to a person, the green color to vegetation, the purple color to roads, the yellow color to poles and lamps and gray to the sky and buildings.

TABLE VII

COMPARISON OF THE EVALUATION RESULTS OF LDNET WITH OTHER STATE-OF-THE-ART METHODS ON THE CARLA-DVS DATASET. THE MEAN $F1$ SCORES (%) AND MEAN IoUS (%) ARE USED AS EVALUATION METRICS FOR THE BINARY CLASS LABELS. THE VALUES IN BOLD ARE THE BEST SCORES

Model	Mean $F1$ (%)	Mean IoU (%)
FCN	46.15	41.32
DeepLabv3	50.42	42.10
RefineNet	54.93	45.34
LaneNet	58.25	49.81
SCNN	59.15	53.14
LDNet-binary class (ours)	**63.50**	**58.45**

far distances from the simulated vehicle compared to locations near the vehicle. The network can learn the schematic at the front using the available event data points but is limited to no predictions at far distances using the same event data. Fig.8 illustrates this behavior for the prediction lanes using the Carla-DVS dataset.

B. Performance of LDNet on Event-Segmentation Data

We perform extensive experimentation of the proposed method on the dynamic vision sensor. Considering that dynamic vision sensor is a newly evolving sensor, not many public datasets are available. Due to the scarcity of lane detection datasets, we used the Event Segmentation dataset [82] to test the proposed algorithm's generalization. The Event Segmentation dataset is an extension of the DDD17 dataset [68], which added semantic segmentation ground truth to the original DDD17 dataset consisting only of grayscale images and event information. The semantic labels have six classes: buildings, objects, trees, person, vehicles and background. The LDNet is trained in an end-to-end manner on the Event Segmentation dataset. The training parameters are similar, as described in section V-C. For a fair comparison, the dataset has the standard split consisting of 15,950 training event images and 3890 test event images.

Table VIII shows the quantitative evaluation on the Event Segmentation dataset. We compared our model with already existing algorithms. The IoU and $Accuracy$ show the efficacy of LDNet. All the training and testing data were recorded at $50 - ms$ intervals. The proposed algorithm does not perform any encoding or preprocessing of the input data. The attention-guided decoder mechanism helps the network to learn the localization of the features. Fig. 9 shows the qualitative comparison of semantic segmentation.

C. Effect of Frame-Based Images on LDNet

We experimented with the effect of frame-based images on the proposed LDNet. To validate this claim, we performed experiments with LDNet on the TuSimple dataset. The TuSimple dataset provides the RGB images with the corresponding lane labels. The dataset has 3626 training images and 2782 testing images. First, we evaluated LDNet trained on

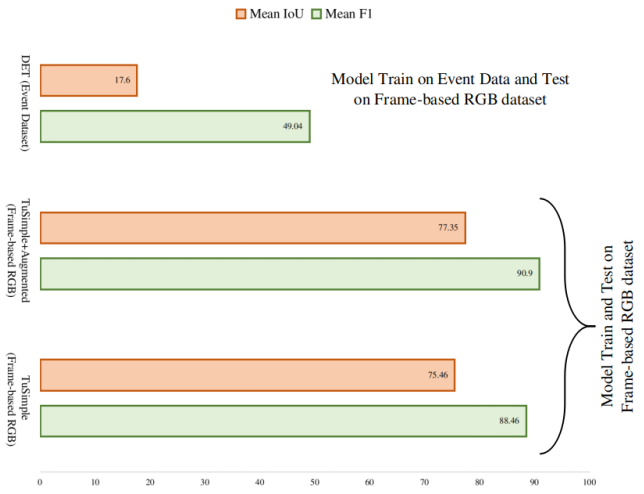

Fig. 10. The bar graph shows the quantitative analysis between the frame-based RGB TuSimple dataset and the Event Camera dataset on the LDNet. This graph illustrates the efficacy of the proposed method when trained on the TuSimple dataset. For a fair comparison with the event camera dataset, the proposed network parameters are kept the same in this experimental analysis for the frame-based RGB TuSimple dataset.

TABLE VIII

COMPARISON OF THE EVALUATION RESULTS OF LDNET WITH OTHER STATE-OF-THE-ART METHODS ON THE EVENT SEGMENTATION DATASET. THE MEAN F1 SCORES (%), MEAN IOUS (%) AND ACCURACY ARE USED AS EVALUATION METRICS FOR THE SEMANTIC LABELS. "−" INDICATES THE METRIC IS NOT INCLUDED IN THE EVALUATION. THE VALUES IN BOLD ARE THE BEST SCORES

Model	Accuracy	Mean $F1$ (%)	Mean IoU (%)
Basic Dense encoding [68]	88.85	-	53.07
Temporal dense encoding [88]	88.99	-	52.32
EV-SegNet [82]	89.76	-	54.81
LDNet (our)	**90.12**	69.09	**58.12**

event camera images with TuSimple testing images. Afterward, we trained the LDNet on the TuSimple dataset to make a fair analysis for the lane marking detection task. We did not optimize the network parameters for the TuSimple dataset, and the network is used in the same configuration as optimized on the event camera dataset. Fig. 10 illustrates the quantitative results of LDNet with TuSimple. The environmental conditions covered in the TuSimple dataset are limited; therefore, we also trained the LDNet with the augmented TuSimple dataset. The augmentations on the TuSimple dataset are sun glare, illumination variations and motion blur. Moreover, 30% of the images in the training dataset were augmented. The evaluation of the TuSimple test data result is shown in Fig. 10.

VIII. CONCLUSION

Both academia and industry have spent considerable resources and efforts to bring autonomous driving closer to real-world applications. The main challenge is to design reliable algorithms that work in diverse environmental scenarios. There has been extensive development at the algorithm level inspired by deep neural networks. Furthermore, the new sensor development work is progressing, and deployment in the autonomous driving sensor suite continues to mature. The sensor setup might be redundant, but different sensor modalities complement each other to achieve the safety of the autonomous vehicle. Event cameras are fast-growing sensors that provide information with precise timing. In contrast to the event cameras, frame-based cameras and Lidar are sampling-based sensors that oversample distant structures and undersample close structures. Moreover, event cameras capture the scene with precise timing when there is a change in brightness. Thus, they provide a very high dynamic range and low latency compared to standard conventional sensors.

In this paper, we proposed LDNet, a novel encoder-decoder architecture for lane marking detection in event camera images. LDNet extracts higher-dimensional features from an image, refining full-resolution detections. We introduced the ASPP block as the core of the network, which increases the respective field of the feature map without increasing the number of training parameters. The use of an attention-guided decoder improves the localization of features in the feature map, hence removing the need for the postprocessing step. The proposed network was evaluated on an event camera benchmark, and it was found to outperform the best-performing state-of-the-art methods in terms of the mean $F1$ and IoU scores. LDNet achieves mean $F1$ scores of 75.58% and 85.13% and mean IoUs of 62.79% and 76.71% for multiclass and binary-class tasks, respectively. Moreover, an ablation study is performed on two datasets, i.e. the Carla-DVS dataset and Event Segmentation dataset, which shows the efficacy of LDNet.

The utilization of an event camera in contrast to a frame-based camera is beneficial for the autonomous vehicle's perception of the environment because the event camera dataset is invariant to illumination conditions. In future work, one possible direction is to investigate the application of the current work with the planning and control module of autonomous driving [2], [89] for lane keeping and lane changing tasks.

REFERENCES

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, Oct. 2014.

[2] S. Azam, F. Munir, A. M. Sheri, J. Kim, and M. Jeon, "System, design and experimental validation of autonomous vehicle in an unconstrained environment," *Sensors*, vol. 20, no. 21, p. 5999, Oct. 2020.

[3] M. P. Muresan and S. Nedevschi, "Multi-object tracking of 3D cuboids using aggregated features," in *Proc. IEEE 15th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2019, pp. 11–18.

[4] H. Kim, J. Cho, D. Kim, and K. Huh, "Intervention minimized semi-autonomous control using decoupled model predictive control," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 618–623.

[5] A. Arikan *et al.*, "Control method simulation and application for autonomous vehicles," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–4.

[6] S. Azam, F. Munir, and M. Jeon, "Dynamic control system design for autonomous car," in *Proc. 6th Int. Conf. Vehicle Technol. Intell. Transp. Syst. (VEHITS)*, 2020, pp. 456–463.

[7] F. Munir, S. Azam, M. I. Hussain, A. M. Sheri, and M. Jeon, "Autonomous vehicle: The architecture aspect of self driving car," in *Proc. Int. Conf. Sensors, Signal Image Process. (SSIP)*, 2018, pp. 1–5.

[8] H. Deusch, J. Wiest, S. Reuter, M. Szczot, M. Konrad, and K. Dietmayer, "A random finite set approach to multiple lane detection," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 270–275.

[9] H. Jung, J. Min, and J. Kim, "An efficient lane detection algorithm for lane departure detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 976–981.

[10] J. Kim and M. Lee, "Robust lane detection based on convolutional neural network and random sample consensus," in *Proc. Int. Conf. Neural Inf. Process*. Malaysia: Springer, Nov. 2014, pp. 454–461.

[11] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 690–703, Mar. 2016.

[12] G. Gallego *et al.*, "Event-based vision: A survey," 2019, *arXiv:1904.08405*. [Online]. Available: http://arxiv.org/abs/1904.08405

[13] W. Cheng, H. Luo, W. Yang, L. Yu, S. Chen, and W. Li, "DET: A high-resolution DVS dataset for lane extraction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1666–1675.

[14] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, Feb. 2017.

[15] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[16] 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.

[17] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

[18] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5633–5643.

[19] N. Messikommer, D. Gehrig, A. Loquercio, and D. Scaramuzza, "Event-based asynchronous sparse convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Springer, Aug. 2020, pp. 415–431.

[20] S. Chen and M. Guo, "Live demonstration: CeleX-V: A 1M pixel multi-mode event-based sensor," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1682–1683.

[21] R. Sun, D. Shi, Y. Zhang, R. Li, and R. Li, "Data-driven technology in event-based vision," *Complexity*, vol. 2021, pp. 1–19, Mar. 2021.

[22] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, 2014.

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

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

[25] B. Huval *et al.*, "An empirical evaluation of deep learning on highway driving," 2015, *arXiv:1504.01716*. [Online]. Available: http://arxiv.org/abs/1504.01716

[26] P.-C. Wu, C.-Y. Chang, and C. H. Lin, "Lane-mark extraction for automobiles under complex conditions," *Pattern Recognit.*, vol. 47, no. 8, pp. 2756–2767, Aug. 2014.

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

[28] D. C. Hernández, L. Kurnianggoro, A. Filonenko, and K. H. Jo, "Real-time lane region detection using a combination of geometrical and image features," *Sensors*, vol. 16, no. 11, p. 1935, 2016.

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

[30] C. Lee and J.-H. Moon, "Robust lane detection and tracking for real-time applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 4043–4048, Dec. 2018.

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

[32] U. Ozgunalp, R. Fan, X. Ai, and N. Dahnoun, "Multiple lane detection algorithm based on novel dense vanishing point estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 621–632, Mar. 2017.

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

[34] S. Jung, J. Youn, and S. Sull, "Efficient lane detection based on spatiotemporal images," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 289–295, Jan. 2016.

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

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

[37] C. Ma and M. Xie, "A method for lane detection based on color clustering," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, Jan. 2010, pp. 200–203.

[38] Y. Zhao, B. Zhou, and H. Chen, "An improved edge detection algorithm based on Canny operator," *J. Jilin Univ., Sci. Ed.*, vol. 50, no. 4, pp. 740–744, 2012.

[39] Y. Wang, N. Dahnoun, and A. Achim, "A novel system for robust lane detection and tracking," *Signal Process.*, vol. 92, no. 2, pp. 319–334, 2012.

[40] 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.

[41] J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, "Robust lane detection using two-stage feature extraction with curve fitting," *Pattern Recognit.*, vol. 59, pp. 225–233, Nov. 2016.

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

[43] X. An, E. Shang, J. Song, J. Li, and H. He, "Real-time lane departure warning system based on a single FPGA," *EURASIP J. Image Video Process.*, vol. 2013, no. 38, pp. 1–18, Jul. 2013.

[44] M. Liang, Z. Zhou, and Q. Song, "Improved lane departure response distortion warning method based on Hough transformation and Kalman filter," *Informatica*, vol. 41, no. 3, pp. 283–288, 2017.

[45] S. Kwon, D. Ding, J. Yoo, J. Jung, and S. Jin, "Multi-lane dection and tracking using dual parabolic model," *Bull. Netw., Comput., Syst., Softw.*, vol. 4, no. 1, pp. 65–68, 2015.

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

[47] J. Jiao, R. Fan, H. Ma, and M. Liu, "Using DP towards a shortest path problem-related application," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8669–8675.

[48] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, pp. 109–117, 2011.

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

[50] S. Lee *et al.*, "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1947–1955.

[51] 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.

[52] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "PolyLaneNet: Lane estimation via deep polynomial regression," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6150–6156.

[53] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key points estimation and point instance segmentation approach for lane detection," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 18, 2021, doi: 10.1109/TITS.2021.3088488.

[54] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," 2020, *arXiv:2004.11757*. [Online]. Available: http://arxiv.org/abs/2004.11757

[55] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection CNNs by self attention distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1013–1021.

[56] F. Pizzati, M. Allodi, A. Barrera, and F. García, "Lane detection and classification using cascaded CNNs," in *Proc. Int. Conf. Comput. Aided Syst. Theory*. Spain: Springer, Feb. 2019, pp. 95–103.
[57] Y. Zhang, Z. Lu, D. Ma, J.-H. Xue, and Q. Liao, "Ripple-GAN: Lane line detection with ripple lane line detection network and wasserstein GAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1532–1542, Mar. 2021.
[58] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[59] S. Chougule, A. Ismail, A. Soni, N. Kozonek, V. Narayan, and M. Schulze, "An efficient encoder-decoder CNN architecture for reliable multilane detection in real time," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1444–1451.
[60] L. Ding, H. Zhang, J. Xiao, C. Shu, and S. Lu, "A lane detection method based on semantic segmentation," *Comput. Model. Eng. Sci.*, vol. 122, no. 3, pp. 1039–1053, 2020.
[61] Y. Li and X. Lu, "Efficient dense spatial pyramid network for lane detection," *J. Phys., Conf. Ser.*, vol. 1575, Jun. 2020, Art. no. 012099.
[62] Y. Sun, L. Wang, Y. Chen, and M. Liu, "Accurate lane detection with atrous convolution and spatial pyramid pooling for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 642–647.
[63] *The Tusimple Lane Challange*. Accessed: Nov. 13, 2020. [Online]. Available: http://benchmark.tusimple.ai
[64] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 7276–7283.
[65] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "CIFAR10-DVS: An event-stream dataset for object classification," *Frontiers Neurosci.*, vol. 11, p. 309, May 2017.
[66] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Frontiers Neurosci.*, vol. 10, p. 405, Aug. 2016.
[67] Y. Hu, J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.
[68] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5419–5427.
[69] N. F. Y. Chen, "Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 644–653.
[70] 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.
[71] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.
[72] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[73] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," 2018, *arXiv:1810.12890*. [Online]. Available: http://arxiv.org/abs/1810.12890
[74] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 390–399.
[75] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473
[76] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*. [Online]. Available: http://arxiv.org/abs/1508.04025
[77] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[78] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "DiSAN: Directional self-attention network for RNN/CNN-free language understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 5446–5455.
[79] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[80] *Apolloscape*. Accessed: Jan. 11, 2021. [Online]. Available: http://apolloscape.auto/lane_segmentation.html
[81] S. Shirke and R. Udayakumar, "Lane datasets for lane detection," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2019, pp. 792–796.
[82] I. Alonso and A. C. Murillo, "EV-SegNet: Semantic segmentation for event-based cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1624–1633.
[83] 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.
[84] 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.
[85] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
[86] 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.
[87] J. Hidalgo-Carrió, D. Gehrig, and D. Scaramuzza, "Learning monocular dense depth from events," 2020, *arXiv:2010.08350*. [Online]. Available: http://arxiv.org/abs/2010.08350
[88] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," 2018, *arXiv:1802.06898*. [Online]. Available: http://arxiv.org/abs/1802.06898
[89] S. Azam, F. Munir, M. A. Rafique, A. M. Sheri, M. I. Hussain, and M. Jeon, "N^2C: Neural network controller design using behavioral cloning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4744–4756, Jul. 2021.

Farzeen Munir (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in system engineering from Pakistan Institute of Engineering and Applied Sciences, Pakistan, in 2013 and 2015, respectively. She is currently pursuing the Ph.D. degree in electrical engineering and computer science with Gwangju Institute of Science and Technology, South Korea. Her current research interests include machine learning, deep neural networks, autonomous driving, and computer vision.

Shoaib Azam (Graduate Student Member, IEEE) received the B.S. degree in engineering sciences from Ghulam Ishaq Khan Institute of Engineering Science and Technology, Pakistan, in 2010, and the M.S. degree in robotics and intelligent machine engineering from the National University of Science and Technology, Pakistan, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea. His current research interests include deep learning and autonomous driving.

Moongu Jeon (Senior Member, IEEE) received the B.S. degree in architectural engineering from Korea University, Seoul, South Korea, in 1988, and the M.S. and Ph.D. degrees in computer science and scientific computation from the University of Minnesota, Minneapolis, MN, USA, in 1999 and 2001, respectively. As a Post-Graduate Researcher, he worked on optimal control problems with the University of California at Santa Barbara, Santa Barbara, CA, USA, from 2001 to 2003, and then moved to the National Research Council of Canada, where he worked on the sparse representation of high-dimensional data and image processing until 2005. In 2005, he joined the Gwangju Institute of Science and Technology, Gwangju, South Korea, where he is currently a Full Professor with the School of Electrical Engineering and Computer Science. His current research interests are in machine learning, computer vision, and artificial intelligence.

Byung-Geun Lee (Member, IEEE) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2000, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 2004 and 2007, respectively. From 2008 to 2010, he was a Senior Design Engineer with Qualcomm, San Diego, CA, USA, where he was involved in the development of various mixed-signal ICs. Since 2010, he has been with Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. He is currently an Associate Professor with the School of Electrical Engineering and Computer Science, GIST. His research interests include high-speed data converters, CMOS image sensors, and neuromorphic system design.

Witold Pedrycz (Life Fellow, IEEE) received the M.Sc., Ph.D., and D.Sc. degrees from Silesian University of Technology, Gliwice, Poland. He is currently a Professor and the Canada Research Chair of computational intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also affiliated with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland. He has authored 17 research monographs and edited volumes covering various aspects of computational intelligence, data mining, and software engineering. His current research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data science, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He is a Foreign Member of the Polish Academy of Sciences and a fellow of the Royal Society of Canada. He also serves on the Advisory Board of IEEE TRANSACTIONS ON FUZZY SYSTEMS. He was a recipient of the Prestigious Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Society in 2007, the IEEE Canada Computer Engineering Medal, the Cajastur Prize for Soft Computing from the European Centre for Soft Computing, the Killam Prize, and the Fuzzy Pioneer Award from the IEEE Computational Intelligence Society. He is also a member of a number of editorial boards of other international journals. He is also vigorously involved in editorial activities. He is also the Editor-in-Chief of *Information Sciences*, *WIREs Data Mining and Knowledge Discovery* (Wiley), and the *International Journal of Granular Computing* (Springer).