IEEE Open Journal of
Intelligent Transportation
Systems

Received 15 July 2021; revised 10 November 2021 and 21 January 2022; accepted 17 March 2022. Date of publication 21 March 2022;
date of current version 29 March 2022.

Digital Object Identifier 10.1109/OJITS.2022.3160888

LiCaNet: Further Enhancement of Joint Perception and Motion Prediction Based on Multi-Modal Fusion

YASSER H. KHALIL (Member, IEEE), AND HUSSEIN T. MOUFTAH (Life Fellow, IEEE)

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada

CORRESPONDING AUTHOR: H. T. MOUFTAH (e-mail: mouftah@uottawa.ca)

This work was supported by the Natural Sciences and Engineering Research Council of Canada through the
Discovery Grant Project under Grant RGPIN/1056-2017.

ABSTRACT The safety and reliability of autonomous driving pivots on the accuracy of perception and motion prediction pipelines, which reckons primarily on the sensors deployed onboard. Slight confusion in perception and motion prediction can result in catastrophic consequences due to misinterpretation in later pipelines. Therefore, researchers have recently devoted considerable effort towards enhancing perception and motion prediction models. However, targeting pixel-wise joint perception and motion prediction using different sensor modalities are often ignored. In this paper, we push performance even further by leveraging a multi-modal fusion network. We propose a novel LIDAR Camera Network (LiCaNet) that achieves accurate pixel-wise joint perception and motion prediction in real-time. LiCaNet expands on our earlier fusion network by incorporating a camera image into the fusion of LIDAR sourced sequential bird's-eye view (BEV) and range view (RV) images. We present a comprehensive evaluation using nuScenes dataset to validate the outstanding performance of LiCaNet compared to the state-of-the-art. Experiments reveal that utilizing a camera sensor results in a substantial gain in perception and motion prediction. Moreover, most of the improvements achieved fall within the camera range, with the highest registered for small and distant objects, confirming the significance of incorporating a camera sensor into a fusion network.

INDEX TERMS Autonomous driving, deep learning, motion prediction, multi-modal fusion, perception, sensor fusion.

I. INTRODUCTION

THE FIELD of autonomous driving had secured incremental progress over the past few years, especially around 2014, when deep learning blossomed. At that time, researchers started regaining hopes that the impediments in autonomous driving could be resolved with the help of innovative deep learning. Typically, an autonomous vehicle consists of several pipelines ranging from perception, motion prediction, planning to control [1]. The two pivotal pipelines in autonomous vehicles are perception and motion prediction, as they allow the vehicle to observe the environment and forecast the dynamics in its surroundings. All subsequent pipelines rely on the accuracy of both perception and motion prediction. Without these pipelines, an autonomous vehicle cannot operate safely and reliably. Moreover, fundamental to perception and motion prediction pipelines are their input data provided by sensors.

In general, an autonomous vehicle is equipped with a suite of different sensors (e.g., LIDAR, camera, ultrasonic, and RADAR) [2]. Although each sensor has advantages and disadvantages, combining data features from several sensors provides complementary information by reaping the benefits of all employed sensors and mitigating the inherent challenges of individual sensors. As our paper focuses on fusing LIDAR and camera features, we briefly compare LIDAR and camera sensors and then show how multi-modal fusion leads to performance advancement. A LIDAR sensor is designed to capture precise depth and physical information of the surrounding environment. On the other hand, a camera sensor acquires color information offering rich semantic images.

The review of this article was arranged by Associate Editor Winnie Daamen.

Unlike LIDAR, a camera is ineffective at capturing object ranges and physical sizes.

Fusing precise range and geometric measurements from a LIDAR and rich semantic information from a camera, we yield an integral set of features resistant to the limitations manifested by individual sensors. For example, when a camera captures semantic features concerning an object, the existence of LIDAR data complements those features by adding the object's depth and physical dimensions. Additionally, small and distant objects are naturally represented by few LIDAR points, and even a camera captures inadequate semantics for such objects. However, when these features are aggregated, the representation of such objects is strengthened.

LIDAR is the most common sensor employed in autonomous vehicles, and several representations exist in the literature for its data. The prominent LIDAR data representations include point-based form [3], [4], 3D voxelization [5], [6], bird's-eye view (BEV) [7], [8], [9], and range view (RV) [10], [11], [12]. In this paper, on top of multi-sensor fusion, we take advantage of fusing multi-view LIDAR data representation. Undoubtedly, each LIDAR representation has its benefits and drawbacks. However, the two most efficient and effective are BEV and RV forms. These two representations overcome most of the limitations found in the other LIDAR-based representations, and further, they encompass additional properties that are valuable to the learning model.

BEV and RV are compact 2D image projections of the LIDAR point cloud, inexpensive to generate, and efficient to process using 2D convolutions. In addition, BEV simplifies the process of adding historical information and preserves object dimensions making learning easier. Lastly, RV preserves occlusion and high-resolution point information. Accordingly, exploiting both BEV and RV form in a fusion network is computationally inexpensive and offers valuable, constructive features procured from one sensor, enabling a deeper understanding of the scene.

Based on recent works [7], [9], [13], [14], [15], [16], it has been shown that the benefits of exploiting multi-modal fusion for perception and motion prediction in autonomous driving are substantial. To the best of our knowledge, no other work explores the multi-modal fusion of historical BEV, RV, and camera features to address the issue of pixel-wise joint perception and motion prediction in real-time. In light of the above observations and inspired by [9], we propose a novel LIDAR Camera Network (LiCaNet), which expands on the fusion network of our earlier work [7] to involve a camera in addition to a LIDAR sensor. Hence, a new camera module is added to extract relevant semantic features from camera images, enabling fusion with BEV and RV features. Fig. 1 illustrates the architecture of our proposed LiCaNet multi-modal fusion network. LiCaNet aims to generate rich and complementary features constituting: temporal, depth, and object sizes encoded in BEV form; occlusion and high-resolution point information embodied in RV; and semantic information characterized in a camera image.

Model-based and data-driven are two main strategies for approaching multi-modal fusion [17], [18]. The model-based approach relies heavily on real-time data associations of different sensors for precise multi-target tracking. Kalman filter-based methods are the most popular among model-based approaches [19]. Such approaches are challenging, time-consuming, and are difficult to apply in complex environments. Conversely, data-driven approaches use deep learning to perform their assigned task. Even though deep learning-based algorithms require huge amounts of annotated data for training, they are fast at learning, generate accurate solutions, and do not require learning at runtime. Therefore, we design LiCaNet to produce outcomes based on a data-driven fusion approach.

After engendering our multi-modal features, they are fed into a backbone network to attain an enhanced joint perception and motion prediction model, especially for small and distant objects. These overarching set of multi-modal features engendered by LiCaNet feeds the backbone network a vivid and knowledgeable image of the surrounding scene, resulting in improved accuracy. The backbone network used in this paper is MotionNet [20] which is a pixel-wise model that perceives and predicts motion in real-time. Our primary contributions are summarized as follows:

- To propose a multi-modal fusion network that reaps the benefits of LIDAR data in historical BEV and RV representations, along with a camera image. The rich and comprehensive features attained are used to perform accurate pixel-wise joint perception and motion prediction in real-time.
- To enhance the accuracy performance for small and distant objects.
- To provide an extensive study that verifies the effectiveness of our proposed approach using nuScenes dataset [21]. We also show that our proposed approach accomplishes competitive performance compared to our earlier work, MotionNet, and other state-of-the-art models.

The remainder of this paper is organized as follows. Section II provides an overview of the related work. Section III explains our proposed fusion scheme. The analysis of our performed experiments is discussed in Section IV. Finally, Section V concludes our paper.

II. RELATED WORK

With the advent of deep learning, significant progress has been made towards perception and motion prediction. In this section, we review the existing prominent literature on perception and motion prediction in the field of autonomous vehicles. Research works have established various methodologies for formulating their input to the learning models, whether the generated input features are sourced from single or multiple sensors. As our proposed work involves two

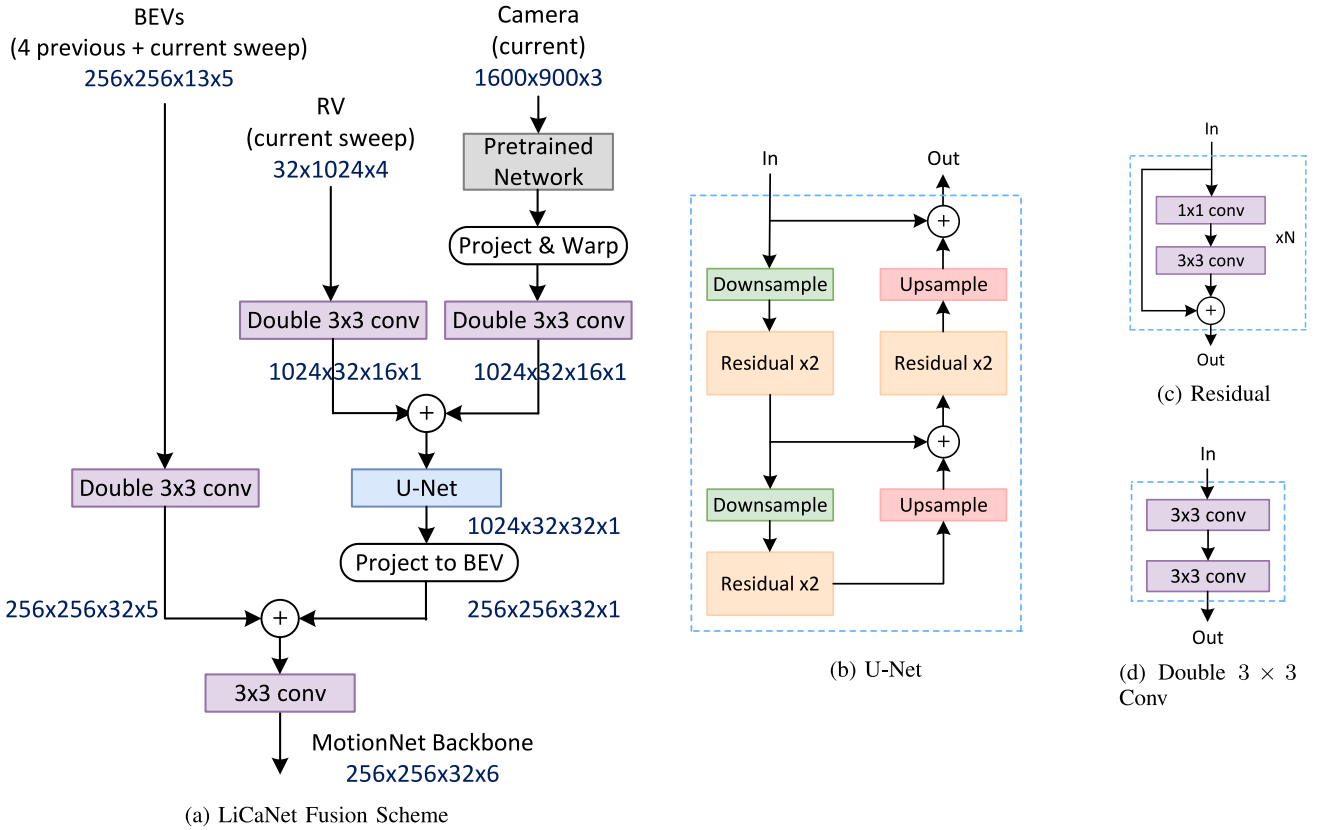

FIGURE 1. LiCaNet architecture. The input is composed of 5 sequential BEV images, an RV, and a camera image. LiCaNet is composed of three modules. The BEV and RV modules consist of double 3 × 3 convolution layers (d). The camera module consists of a pretrained network, projecting and warping the camera features into RV form and double 3 × 3 convolution layers. The RV features from both the RV and camera modules are concatenated and fed into U-Net (b). The U-Net consists of residual blocks (c) and upsample and downsample blocks of scale 2. The output of the U-Net, in RV form, is projected into BEV and concatenated with the features from the BEV module to be finally fed into a single 3 × 3 convolution layer. Finally, the LiCaNet output is fed into MotionNet backbone for joint perception and motion prediction.

sensors, we target works that deployed a LIDAR, camera, and LIDAR and camera combined.

A. PERCEPTION AND MOTION PREDICTION USING LIDAR SENSOR

There exist many models that depend on a single LIDAR sensor, with each representing its input features differently. Some of the prior works directly processed the raw 3D point cloud without applying any transformation. To begin with, PointRCNN [3] is a point-based method that generates 3D proposals for 3D object detections using a two-stage method: the bottom-up 3D proposal generation and refining the proposals. Shi *et al.* [4] extends PointRCNN to achieve 3D object detections using part-aware and aggregation neural network. Another common approach is to transform the point cloud into 3D voxels. VoxelNet [5] predicts 3D detections using voxel encodings. Additionally, fast point R-CNN [6] and PV-RCNN [22] performed 3D object detection by incorporating voxel-based and point-based for better point cloud feature learning.

Although perceiving the environment through point-based or 3D voxels has benefits. However, the runtime requirement suffers due to the processing of sparse representations and the existence of 3D convolutions. Typically, point-based models are data-intensive, and so in addition to being time-consuming, they face computation and memory bottlenecks; thus cannot be scaled easily. Recently, researchers have considered transforming point clouds into 2D images, such as BEV and RV, for their compactness, efficiency in processing, and effectiveness in improving the performance of point cloud classification and segmentation. There exist few works that used 2D LIDAR-based image representations but ended up generating 3D object-level proposals [12], [23], [24].

All works reviewed so far engender 3D object-level predictions, even though different input representations were used. The generation of 3D proposals requires 3D convolutions, which are computationally inefficient compared to 2D convolutions. The following works adopt 2D input images for their input features and use 2D convolutions to perform pixel-wise predictions. Khalil and Mouftah [7] proposed combining BEV and RV representations to perceive and predict motion in real-time. Khalil and Mouftah [8] used reinforcement learning to train an autonomous vehicle in an urban environment with perception and motion prediction guidance. SqueezeSegV3 [11] used RV images to perceive the driving environment through segmentation. SalsaNext [10] a real-time, uncertainty-aware semantic segmentation model that is based on RV representations. AMVNet [25] use

RV images to perform semantic segmentation using an assertion-guided sampling strategy. Lastly, MotionNet [20] used sequential BEV images to perform pixel-wise joint perception and motion prediction in real-time.

We can conclude that fusing LIDAR-based 2D image representations is more effective than other LIDAR-based representations because they are compact, efficient to generate and process, and 3D convolutions can be avoided. Furthermore, it is possible to generate real-time pixel-wise predictions with such representations.

B. PERCEPTION AND MOTION PREDICTION USING CAMERA SENSOR

Perception algorithms that utilize deep learning and depend on a camera sensor fall under the category of image-based detections based on convolutional neural network (CNN) architectures. Typically, such frameworks are divided into two streams: one-stage and two-stage object detections. One-stage detectors (e.g., YOLO [26] and SSD [27]) directly map input features to class probabilities and bounding box coordinates via a single-stage CNN model. Whereas two-stage detectors (e.g., Faster R-CNN [28] and R-FCN [29]) firstly extract region proposals, then they are refined down the pipeline to generate object classification and regression. Predominantly, one-stage detectors are faster than two-stage detectors but are inferior to two-stage detectors in terms of detection accuracy. An excellent performance assessment of one-stage and two-stage detectors in autonomous driving can be found in [30]. Over the past decade, image-based object detections have picked a staggering pace in the field of autonomous driving [31], [32], [33], [34].

A typical problem with image-based CNN detectors is that accuracies for large and small objects are unbalanced. Large objects are represented by sufficient features permitting them to be classified correctly with high confidence. In contrast, small objects are usually represented by inadequate features and thus left undetected or classified with low confidence. In the field of autonomous driving, it is essential to detect small objects (e.g., pedestrians and bicyclists) to maintain their safety. Recently, researchers proposed image-based CNN algorithms with a focus on detecting small objects. FPN [35] is a two-stage multiscale network that achieved high detection accuracy for small objects. The technique adopted in FPN is the fusion of multiscale features. PNA [36] is an enhanced version of FPN that also specializes in detecting small objects. YOLOv4-5D [31] proposed an improvement to the PNA backbone network to increase the detection accuracy for small objects. Even though these detectors were able to achieve better detection results for small objects, they are still bounding box-based methods.

Several successful attempts exist that build a pixel-wise detector using only camera sensors, registering challenging outcomes. Porzi *et al.* [37] proposed semantic segmentation using a single backbone network, outperforming UPSNet [38] which uses parameter-free panoptic head for segmentation. Yang *et al.* [39] proposed an end-to-end unsupervised learning framework to perform depth estimation and camera motion prediction. Results showed that using stereo image sequences surpasses scale ambiguity for depth estimation and increases motion prediction accuracy for temporal image sequences.

To conclude this subsection, we note that pixel-wise predictions are essential for perceiving small and distant objects in autonomous driving; however, the inference time requirement remains an issue. Furthermore, motion prediction using only a camera sensor is a recent research topic, and results are still not ideal mainly due to the lack of depth information in camera semantics.

C. FUSION OF LIDAR AND CAMERA SENSORS

Presently, the utilization of multi-modal fusion for perception and motion prediction has gained much attention among researchers in autonomous driving. Multi-modal fusion is used to exploit the complementary properties of different sensors and representations. Feng *et al.* [13] presented a survey on different multi-modal methodologies for object detection and segmentation in autonomous driving. An in-depth review on the fusion of point clouds and images can be found in [40]. Liang *et al.* [41] fused BEV and camera images to perform 3D object detection using a continuous fusion layer. Moreover, Liang *et al.* [15] enhanced the 3D object detection model in [41] by reasoning about 2D and 3D object detections, ground estimation, and depth completion. LaserNet++ [14] is another model that performs 3D object detections; however, by fusing RV and camera features. LaserNet++ reported good performance results, especially for small and distant objects. PointPainting [42] applied sequential fusion for semantic segmentation using the painting technique where the point cloud is augmented with image semantics. Later, the 3D detections are extracted by applying the painted point cloud to a LIDAR detector. Lastly, MVX-Net [43] proposed a 3D object detector using two methods that fuse a point cloud and a camera image in a point-wise or voxel-wise fashion.

These works utilized multi-modal fusion to perform just perception using 3D object proposals. Fadadu *et al.* [9] proposed a multi-modal fusion model (BEV, RV, camera, and HD maps) for perception and motion prediction using LIDAR and camera. However, again, 3D object-level predictions were computed. It is evident that no work has yet been conducted that investigates pixel-wise joint perception and motion prediction using multi-modal fusion, which is essential for small and distant objects as they provide fine-grained, pixel-level precision.

D. RELATED WORK CONCLUDING REMARKS

In contrast to the reviewed works above, we propose a multi-modal fusion network named LiCaNet to generate accurate pixel-wise perception and motion prediction. LiCaNet fuses camera features with LIDAR-based historical BEV and RV features. LiCaNet engenders multi-modal features that

embrace: 1) temporal, depth, and physical object dimensions in BEV form; 2) occlusion and high-resolution point information in RV form; and 3) semantics in camera images. These rich and integral features enhance the accuracy of perception and motion prediction, especially for small and distant objects.

III. PROPOSED METHODOLOGY

The overview of our LiCaNet model is shown in Fig. 1. We designed LiCaNet to incorporate features from LIDAR and camera sensors to produce complementary multi-modal features. We represent the LIDAR data in sequential BEV and RV images. LiCaNet consists of three modules: BEV, RV, and camera. The BEV and RV modules represent the network of our earlier work [7], and the camera module is the proposed expansion resulting in LiCaNet. The camera module accepts camera images and extracts relevant features to be fused with the outcomes of the BEV and RV modules for further performance enhancement. The multi-modal features generated by LiCaNet are then used as input to MotionNet backbone network for pixel-wise joint perception and motion prediction. LiCaNet is evaluated on nuScenes dataset [21] which consists of large amounts of high-quality LIDAR and camera data designed for autonomous driving. The methodologies used to analyze LiCaNet predictions are classification accuracy for perception and displacement error for motion prediction. LiCaNet performance outperforms our earlier work, MotionNet, and other state-of-the-art models. Furthermore, LiCaNet operates in real-time, making it suitable for autonomous driving.

In Sections III-A and III-B, we explain the formulation of the sensors' data representation specifically BEV, RV and camera images; basically, our LiCaNet input. We define the architecture of LiCaNet that generates the integral features in Section III-C. Last, we discuss in Section III-D the MotionNet backbone network used for learning from the generated integral features to realize accurate pixel-wise joint perception and motion prediction in real-time.

A. LIDAR INPUT REPRESENTATION

A LIDAR operates by scanning its entire field-of-view (FOV) using laser beams. The LIDAR measures the time difference between firing a focused laser beam and detecting its reflection. This collected data is used to compute the distance to objects, which can be further used to compute the xyz-coordinates of objects. We represent the LIDAR data provided by the nuScenes dataset in BEV and RV forms.

1) BIRD'S-EYE VIEW

Bird's-eye view (BEV) is formed by projecting the 3D LIDAR points into 2D images of dimensions $R_x \times R_y \times H$ meters, and grid cell resolution of $\Delta r_x \times \Delta r_y \times \Delta h$ meters in the xyz-axis. The 2D grid images represent the top-down view of the point cloud, where $R_x R_y H$ denotes the region-of-interest in the xyz-direction. We set R_x and R_y to each cover a range of 64m in length and width, respectively. The covered range in length is 32m long from the front- and back-side of the vehicle ($R_x \in [-32, 32]$). Similarly, the width range is divided equally between the left- and right-side of the vehicle ($R_y \in [-32, 32]$). Height dimension H covers a total range of 5 meters ($H \in [-3, 2]$). We set the resolution of each 2D grid cell to $\Delta r_x = 0.25$, $\Delta r_y = 0.25$ and $\Delta h = 0.4$. Discretizing all 3D points into evenly spaced cells results in a BEV image of dimensions $256 \times 256 \times 13$. In other words, the entire height dimension of the point cloud is discretized into 13 image channels ($H \Delta h^{-1}$), each with a size of 256×256 ($R_x \Delta r_x^{-1} \times R_y \Delta r_y^{-1}$). Upon discretization, a grid cell is considered occupied if at least one LIDAR point is mapped to it and is labeled with a value of 1; otherwise, -1 is assigned.

The procedure outlined above computes one BEV image from a single LIDAR sweep. On the other hand, our BEV input requires a sequence of 5 BEV images. As a result, we incorporate a sequence of 4 historical LIDAR sweeps in addition to the current sweep to provide the capacity to generate motion predictions. The historical sweeps are exploited in BEV form because of their simplicity in stacking sequential images. So, on top of computing the BEV image of the current sweep, we need to convert all 4 historical sweeps into BEV form by the same discretization process explained above. Before discretizing the 4 historical sweeps, we must first synchronize them with the current sweep's coordinate system. The synchronization process is done through coordinate transformation. This step is necessary to compensate for the autonomous vehicle's motion across time. After the synchronization and discretization steps, all 4 historical BEV images are stacked on top of the current BEV image. Thus, the overall BEV input dimensions become $256 \times 256 \times 13 \times 5$. Fig. 2a illustrates a sample of our sequential BEV input.

2) RANGE VIEW

Range view (RV) representation is created by projecting each point $p_i = (x, y, z)$ in the LIDAR point cloud to a pixel in the projected RV image. A spherical projection is used to speed up computation. The transformation used for projecting to RV image is defined in Eq. (1).

$$\begin{pmatrix} u_l \\ v_l \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left[1 - \arctan(y, x)\, \pi^{-1}\right] w_l \\ \left[1 - \left(\arcsin\left(z r^{-1}\right) + f_{up}\right) f^{-1}\right] l_l \end{pmatrix}, \tag{1}$$

where (u_l, v_l) represents the angular coordinates denoting the pixels in the RV image. w_l and l_l indicates the width and length of the RV image, respectively. The vertical FOV of the LIDAR sensor is defined as $f = f_{up} + f_{down}$, and $r = \sqrt{x^2 + y^2 + z^2}$ is the range of point p_i.

We design the projected RV image dimensions to be $1024 \times 32 \times 4$. The length value usually reflects the number of LIDAR beams. The LIDAR sensor used in nuScenes dataset is Velodyne HDL32E and has 32 beams. In contrast, the width and channel values are determined by the designer. In our earlier work [7], it was proven that the wider the RV image, the better the performance. However, we selected a width dimension of 1024 because that is the maximum

our limited hardware can handle to load and train LiCaNet. Similar to [7], we construct the 4 channels for each (u_l, v_l) pixel by projecting the range r, height, intensity of p_i, and the last channel is a binary value indicating if the pixel is occupied by at least one p_i. If no p_i is projected to a (u_l, v_l) then all 4 channels are filled with -1. Unlike the input BEV representation, our RV image only constitutes the current LIDAR sweep with no historical information. Compared to BEV form, where past sweeps are simply stacked together, concatenating past sweeps in RV form involves more complicated processing. The LiDAR sweep used to generate the RV image is the same as the one used to compute the current BEV image. A sample range, height and intensity of the current sweep in RV form are illustrated in Fig. 2c, 2d, and 2e, respectively.

B. CAMERA INPUT REPRESENTATION

A camera sensor captures information with color encodings, offering semantically rich images. From the nuScenes dataset, the front camera images are used as input to the LiCaNet camera module. The RGB camera images are of dimensions $1600 \times 900 \times 3$. Fig. 2b shows a sample camera image captured at the exact timestamp as the LIDAR sweep that is used to compute the current BEV and RV images.

C. LICANET ARCHITECTURE

As aforementioned, the significance of fusing multi-modal features is to extract complementary information that contributes to producing improved perception and motion prediction. A LIDAR is adopted in LiCaNet primarily due to its capability in capturing precise depth information. Moreover, a front camera is employed for its dense semantic features. To avoid the sparsity of the LIDAR data and the inefficient processing of the vast number of points, LIDAR data is processed in its BEV and RV representations. We select BEV representation to represent LIDAR data because they are handled readily and efficiently by 2D convolutions. In addition, BEV representations preserve physical object sizes offering vital prior information to the learning model. Further, a sequence of historical data encoded in BEV form can be easily concatenated. Along with BEV representations, we use RV images to represent LIDAR data. They are generated from a single viewpoint, making them the most informative to portray a point cloud. Another advantage of RV images is that they preserve occlusion information. Consequently, concatenating features from BEV, RV, and camera representations produces complementary features that leverage all representations' benefits and mitigate the drawbacks of individual representations. Therefore, the performance advancements reported in this paper are attributed to the fusion of camera semantics into BEV and RV features.

The architecture of LiCaNet is depicted in Fig. 1. The proposed fusion scheme consists of three key modules: BEV, RV, and camera. A sample of input data to the LiCaNet modules is illustrated in Fig. 2. Starting with the BEV module,

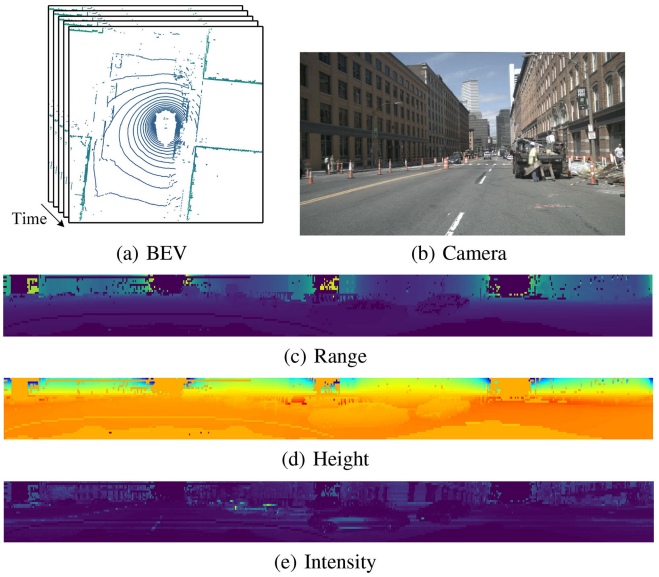

FIGURE 2. A sample of LiCaNet input features. (a) illustrates the historical BEV images; (b) the camera image; (c), (d), and (e) represent three channels of the RV image.

once all LIDAR sweeps are transformed into BEV representation through synchronization and discretization, the aggregated BEVs are sent down a two-layer 3×3 convolution layers, named *Double* 3×3 *conv*. Concurrently, the RV image and camera features representing only the current timestamp are also passed down a *Double* 3×3 *conv* independently. However, before applying the RGB camera image directly to *Double* 3×3 *conv*, the RGB image is first passed to a small pretrained network to generate high-level camera features. Then the high-level features are projected and warped into RV representation. Next, the resulting features from the RV and camera modules are concatenated and applied to a U-net [44]. U-Net is an encoder-decoder network with a strong representation ability mainly because of the skip connections that combine shallow features from the encoder path with deep features from the decoding path at their respective stages. The features resulting from the U-Net are in RV form and thus need to be projected to BEV representation to complete the fusion process. The subsequent step is to concatenate the projected features, in BEV representation from the RV and camera modules, with the features from the BEV module. The last step in the fusion process is to feed the resulting multi-modal features into a single 3×3 convolution layer. This proposed LiCaNet fusion process generates rich complementary features that enable us to achieve enhanced performance. Finally, the generated multi-modal features are then applied to MotionNet backbone network to perform accurate pixel-wise joint perception and motion prediction in real-time.

Directly feeding the fusion network with raw RGB features causes the learning network to discard most features as they do not comprise valuable high-level information. Thus, the camera image is first passed to a pretrained network

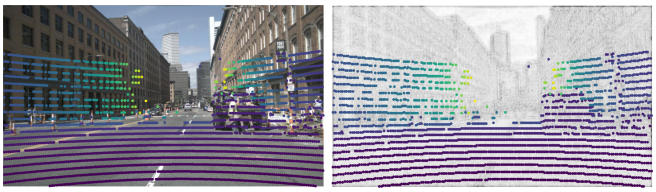

FIGURE 3. Examples of LIDAR points' range values projected onto the camera image. The left image represents the LIDAR points projected on the raw camera image. The right image shows the LIDAR points projected on the outcome of the lightweight pretrained network (in grey scale).

to extract high-level features, which are later fed into the fusion network. Now, to use the extracted high-level camera features in our fusion process, we need a mapping from the LIDAR points to the camera image. This mapping permits the retrieval of the high-level camera features corresponding to LIDAR points residing in the camera's FOV. In summary, once we have a mapping from LIDAR points to camera features, we can warp and project features from the camera image into the RV image. The mapping of each LIDAR point p_i onto a camera image is achieved by the transformation $\mathrm{T}_{c\leftarrow l}$ defined in Eq. (2).

$$\mathrm{T}_{c\leftarrow l} = \mathrm{K}\left(\mathrm{T}_{c\leftarrow v_{t_c}} * \mathrm{T}_{v_{t_c}\leftarrow v_{t_l}} * \mathrm{T}_{v_{t_l}\leftarrow l}\right), \tag{2}$$

where subscripts c, l, and v stands for camera, LIDAR, and vehicle, respectively. K is the intrinsic calibration matrix of the camera. In nuScenes dataset, LIDAR and camera sensors have different operational frequencies and so before transforming LIDAR points to the camera's coordinate system they need to be mapped to the vehicle's coordinate system to compensate for the time-shift between the two sensors. $\mathrm{T}_{v_{t_l}\leftarrow l}$ transforms LIDAR points to the vehicle's frame at LIDAR capture time t_l, $\mathrm{T}_{v_{t_c}\leftarrow v_{t_l}}$ transforms the points from vehicle's frame at LIDAR capture time t_l to camera capture time t_c. Last, $\mathrm{T}_{c\leftarrow v_{t_c}}$ transforms the points from vehicle's frame at t_c to the camera's coordinate system.

The complete mapping equation that maps LIDAR points onto the camera coordinate system is defined in Eq. (3).

$$[u_c \; v_c \; 1]^T = \mathrm{T}_{c\leftarrow l}\,(p), \tag{3}$$

where (u_c, v_c) are the mapped points from the LIDAR's coordinate system onto the camera. An example of LIDAR points' range values being mapped using Eq. (3) and projected onto camera coordinate system is shown on the left image of Fig. 3. The projection algorithm is explained later.

Ultimately, we need to fuse the extracted high-level camera features with RV features. Thus, using the mapping computed in Eq. (3) between the LIDAR points and the camera features, we can now project the camera features into RV representation. Up to this point, we assumed that the features extracted from the pretrained network have the same dimensions as the original camera image. Unfortunately, this is not always the case, so to resolve this issue, we need to update the mapping between LIDAR points and camera image pixels

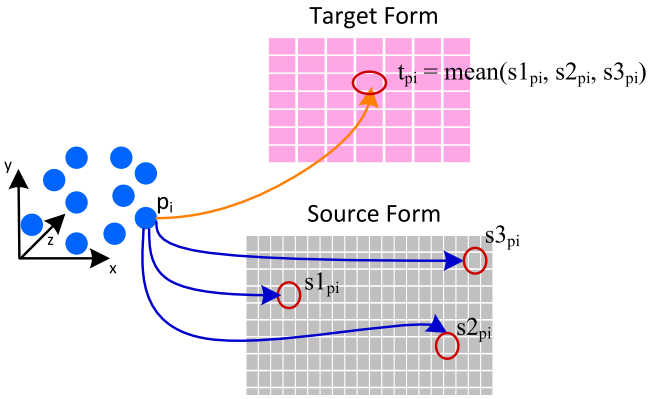

FIGURE 4. Illustration of the projection algorithm from source to target form. t_{p_i} is the average of all features that LIDAR point p_i is mapped to in the source form ($s1_{p_i}$, $s2_{p_i}$, and $s3_{p_i}$).

with a scale factor as expressed in Eq. (4).

$$\begin{aligned} u_{c_{scaled}} &= u_c * s_l^{-1} \quad \text{and} \quad s_l = \left(l_c / l_{hlcf}\right), \\ v_{c_{scaled}} &= v_c * s_w^{-1} \quad \text{and} \quad s_w = \left(w_c / w_{hlcf}\right), \end{aligned} \tag{4}$$

where $(u_{c_{scaled}}, v_{c_{scaled}})$ are the scaled mapped points on the camera image. (l_c, w_c) represents the length and width of the original camera image, while (l_{hlcf}, w_{hlcf}) denotes the resolution of the high-level camera features resulting from the pretrained network. The right image of Fig. 3 illustrates the projected LIDAR points' range values on the high-level features resulting from the pretrained network, using mapping in Eq. (3) and scaling in Eq. (4). The lightweight pretrained network used to extract high-level features from camera images is described in Section IV-B.

The projection of features from one form to another is accomplished using the painting technique [42]. Briefly, the algorithm takes the mean of all features from the source representation that corresponds to a LIDAR point p_i and projects them to the cell position in the target representation where p_i is linked. If no features are projected into a cell in the target representation, the cell value remains -1. The same algorithm is repeated for all points that have a mapping between the source and target forms. Fig. 4 demonstrates an illustrative example of the adopted projection algorithm. Firstly, the mapping for each LIDAR point p_i is computed in both the source and target forms (denoted by blue and orange arrows). Next, all features in the source form where p_i is linked to (denoted by $s1_{p_i}$, $s2_{p_i}$, and $s3_{p_i}$) is averaged. Eventually, the averaged features are positioned in the target's cell where p_i is linked (orange arrow). Fig. 5 displays features from the camera image projected into RV form. The number of LIDAR points mapped between the camera and RV forms determine the resolution of the target RV image.

D. BACKBONE NETWORK

The backbone network used in this work is MotionNet [20]. MotionNet is a novel model that performs pixel-wise joint perception and motion prediction in real-time. MotionNet

FIGURE 5. Example of the front camera image projected into RV representation. The RV image has been cropped to present only the area that contains the projected camera features. The rest of the RV image is empty because the camera FOV is 70°, while the horizontal LIDAR FOV is 360°.

architecture consists of an encoder-decoder named spatio-temporal pyramid network (STPN) and three output heads. MotionNet is considered one the fastest models in performing joint perception and motion prediction due to its lightweight STPN. This is because STPN lacks 3D convolutions and depends merely on 2D and pseudo-1D convolutions. Fig. 6 presents the architecture of MotionNet. The main element of STPN is the spatio-temporal convolution (STC) that constitutes two 2D convolutions followed by one pseudo-1D convolution. STPN builds an encoder with STC blocks to extract features at different stages, leveraging multi-scale spatial and temporal feature learning. Global temporal pooling is used to assist in fusing multi-stage temporal features while going up the decoder part of the STPN. This design promotes the extraction of local and global spatio-temporal information.

The MotionNet output heads are: 1) cell classification – for perceiving the category of pixels; 2) motion prediction – for predicting pixels motion; 3) state estimation – for predicting whether the pixels are static or dynamic. The three output heads constitute two 2D convolutions each to acquire BEV pixel-wise predictions. The cell classification head classifies pixels from 5 category groups: background, vehicles, pedestrians, bicyclists, and others. The *others* category is assigned to detect objects that are not categorized in any of the remaining four groups. Therefore, the output dimension of the cell classification head is $256 \times 256 \times 5$. The motion prediction head predicts pixel positions for a sequence of 20 frames into the future (translating into 1 second); thus, the dimension of its output head is $256 \times 256 \times 2 \times 20$. Lastly, the output dimension of the state estimation head is $256 \times 256 \times 2$ because it predicts whether each pixel in the BEV image is static or dynamic.

MotionNet loss function defined in Eq. (5) consists of six components. Three of which are dedicated for global regularization of network training. These components are linked to the three output heads (cell classification loss $\mathcal{L}_{class}$, motion prediction loss $\mathcal{L}_{motion}$, and state estimation loss $\mathcal{L}_{state}$). Spatial consistency loss $\mathcal{L}_s$, foreground $\mathcal{L}_{ft}$ and background $\mathcal{L}_{bt}$ temporal consistency losses are the other three components that are dedicated for local regularization.

$$\mathcal{L} = \mathcal{L}_{class} + \mathcal{L}_{motion} + \mathcal{L}_{state} + \alpha\ \mathcal{L}_s + \beta\ \mathcal{L}_{ft} + \gamma\ \mathcal{L}_{bt}, \quad (5)$$

where α, β, and γ are balancing factors. $\mathcal{L}_{class}$ and $\mathcal{L}_{state}$ use weighted cross-entropy loss, while $\mathcal{L}_{motion}$ uses weighted smooth L_1 loss. Different weights are used for each class category to counteract the class imbalance issue. $\mathcal{L}_s$ uses smooth L_1 loss to constrain predicted motion between adjacent pixels of the same object. Similarly, $\mathcal{L}_{ft}$ limits the predicted motion for each object, but temporally rather than spatially. In other words, the motion of objects between consecutive frames should not have abrupt changes. Unlike $\mathcal{L}_{ft}$ that focuses on foreground objects, $\mathcal{L}_{bt}$ concentrates on background cells and tries to minimize the temporal loss of static cells by overlapping them across adjacent frames.

IV. RESULTS

We begin this section with a brief description of the dataset used. Second, we define the set of experiments performed to validate the significance of multi-modal fusion for joint perception and motion prediction. Details about the training setup are illustrated next. Extensive quantitative and qualitative results are provided to confirm the effectiveness of our proposed approach. Last, we include an ablation study to compare LiCaNet to other state-of-the-art models.

A. DATASET

The dataset used to conduct our experiments is nuScenes [21]. It consists of 850 annotated scenes, with each having a continuous sequence of sweeps. Our training set consists of 500 scenes (17,065 sequences), while our validation and test set has 100 (1,719 sequences) and 250 scenes (4,309 sequences), respectively. The LIDAR sensor used in the nuScenes dataset is Velodyne HDL32E, consisting of 32 beams, and operates at 20Hz. The horizontal FOV of the LIDAR is 360°, while its vertical FOV ranges from −30.67° to 10.67°. The front camera sensor used in the nuScenes dataset captures images at 12Hz with an opening angle of 70°.

B. EXPERIMENTAL SETUP

MotionNet confirms that using bigger BEV dimensions than $256 \times 256 \times 13$ accumulates additional computational cost without promising any performance improvements. Similarly, using 5 frames to reflect the temporal informal manifests a good trade-off between efficiency and accuracy. Thus, our BEV input dimension is $256 \times 256 \times 13 \times 5$. In addition, it was proven in [7] that using RV images of dimensions 2048×32 for fusion purposes leads to better joint perception and motion prediction results compared to smaller sizes. However, in this paper, we select the RV image dimensions to be 1024×32 due to our limited computational power.

We begin our experiments by training MotionNet on our machine to record its potential. Second, we include the experiment conducted in our earlier work [7] that fuses historical BEV data with RV representation (1024×32). In [7], the *Double* 3×3 *conv* block in the RV module uses 32 channels to encode features. In order to load and train LiCaNet on our machine, we narrow down the depth of the RV module's

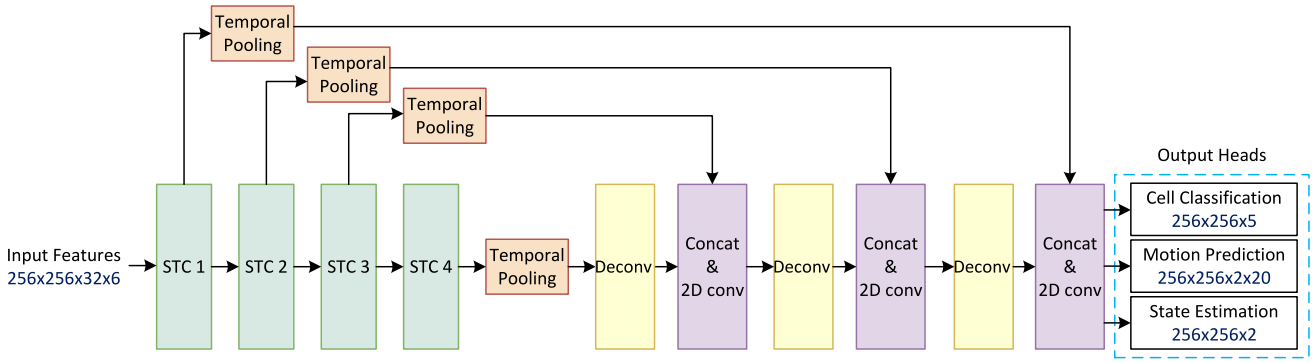

FIGURE 6. MotionNet Architecture. The encoder part consists of 4 STC blocks, each constituting two 2D convolution layers and one pseudo-1D convolution layer. The temporal pooling is used to diminish the temporal dimension of the features to 1. The decoder part consists of deconvolution, concatenation, and two 2D convolutions layers. The deconvolution layers are used to scale up the input features to allow concatenation with the features coming from the encoder. MotionNet has three output heads where each has two 2D convolution layers.

TABLE 1. Comparison of perception and motion prediction results between MotionNet, multi-view LIDAR-based fusion, and our proposed LiCaNet model.

Method	Static		Slow ($< 5\mathrm{m/s}$)		Fast ($\geq 5\mathrm{m/s}$)		Classification Accuracy (%)							Time
	Mean	Median	Mean	Median	Mean	Median	Bg	Vehicle	Ped.	Bike	Others	MCA	OA	(ms)
MotionNet [20]	0.0236	0	0.2534	**0.0959**	1.0778	0.7346	97.5	91.2	74.9	22.1	65.9	70.3	96.3	**20.5**
LIDAR fusion [7]	0.0227	0	0.2497	0.0967	1.0360	0.7056	97.9	92.5	82.2	23.4	70.9	73.4	96.8	28.3
LiCaNet (LIDAR only)	0.0230	0	0.2531	0.0964	1.0547	0.7305	97.6	92.0	80.6	22.6	69.2	72.4	96.6	28.1
LiCaNet (MobileNetv2_6)	0.0224	0	**0.2504**	0.0964	**1.0432**	0.7304	97.9	92.8	82.7	23.0	70.7	73.4	96.8	30.7
LiCaNet (VGG16_6)	0.0224	0	0.2527	0.0969	1.0456	0.7300	97.8	92.4	84.0	23.2	71.9	**73.9**	**96.9**	31.0
LiCaNet (ResNet50_11)	0.0223	0	0.2530	0.0963	1.0479	0.7295	97.8	92.6	81.9	23.4	70.4	73.2	96.8	32.9
LiCaNet (ResNeXt50_11)	**0.0220**	0	0.2529	0.0963	1.0461	**0.7289**	98.0	92.7	81.5	23.6	70.6	73.3	96.9	33.2

Double 3×3 *conv* block to 16 channels. Thus, in order to compare LiCaNet with [7], we retrain the fusion of historical BEV and RV images with a depth configuration of 16 for the RV module's *Double* 3×3 *conv* block. It is worth mentioning that LiCaNet minus the camera module matches the fusion network of [7]. Thus, we name this experiment LiCaNet (LIDAR only). The comparison with these two experiments is essential to verify that LiCaNet (fusion of BEV, RV, and camera) outperforms both our earlier work (fusion of BEV and RV) and MotionNet (BEV only).

The next set of experiments analyzes the performance of LiCaNet under various lightweight pretrained networks. These experiments are used to monitor whether the performance gain of LiCaNet is consistent across all pretrained networks. All selected lightweight networks dedicated to extracting high-level features from camera images are pretrained on ImageNet. The four pretrained networks investigated are MobileNetv2, VGG16, ResNet50, and ResNeXt50. These pretrained networks were chosen as they have shown challenging performance in image classification. As our camera module requires only a lightweight pretrained network, we only employ 6 convolution layers from MobileNetv2 and VGG16; and 11 layers from ResNet50 and ResNeXt50.

C. TRAINING SETUP

For fair comparisons, we follow the same setup as our earlier work and MotionNet. Each scene in the dataset is divided into several clips, where each clip consists of 5 consecutive (1 current and 4 previous) sweeps. The current sweeps are sampled at 2Hz for training and 1Hz for testing. The current sweeps for testing are sampled at a lower frequency to reduce the similarity between the clips. Additionally, the period between all sweeps in a clip is 0.2s. All experiments are trained with a batch size of 4. The initial learning rate is set at 1.6×10^{-3} and it decays every 10 epochs to end at 0.8×10^{-3}. All experiments are implemented using PyTorch and trained on a single NVIDIA Quadro RTX5000 GPU with Intel Xeon 3.9-GHz CPU.

D. QUANTITATIVE RESULTS

Table 1 unveils the perception and motion prediction results of our conducted experiments. To evaluate motion prediction, we measure the mean and median displacement errors of pixels based on three-speed groups. The speed groups are static, slow, and fast. Pixels with speed $< 5m/s$ are assigned to the slow speed group, while pixels with predicted motion $\geq 5m/s$ are assigned to the fast group. If pixels are predicted to have 0 motion, then they are assigned to the static group. Due to the large proportion of staticity in a scene, distinguishing static from dynamic pixels becomes essential to avoid biased displacement errors. The displacement error is measured using L_2 distances between the predicted and the ground-truth displacements. In terms of perception, the metrics used are classification accuracy per category, mean classification accuracy (MCA), and overall pixel accuracy (OA).

It is evident from the collected results in Table 1 that all LiCaNet experiments, with the different pretrained networks, compare favorably to MotionNet and LiCaNet (LIDAR only) - a narrowed version of the multi-view LIDAR-based fusion network [7]. For fair comparisons, the second experiment in Table 1 is excluded from our analysis as the depth of its RV module is wider than LiCaNet. Nevertheless, it was included to show that its narrower version has inferior performance in both perception and motion prediction. According to Table 1, the use of 6 convolution layers from pretrained VGG16 resulted in the best perception. Even though VGG16 did not attain the lowest displacement errors; however, VGG16 still achieved competitive motion predictions compared to the other pretrained networks. Comparing perception accuracy of LiCaNet (VGG16_6) with LiCaNet (LIDAR only) experiments, we see that the addition of the camera module achieved a substantial increase of 1.5% and 0.3% in MCA and OA, respectively. Moreover, a total gain of 3.6% in MCA and 0.6% in OA is registered relative to MotionNet.

In addition, examining the classification accuracy per category, we notice that of all the selected pretrained networks, VGG16_6 is considered the best at detecting small objects. With only 6 convolution layers, VGG16_6 secured the highest detection accuracy for pedestrians and a competitive accuracy for bikes. In comparison, ResNeXt50_11 used 11 convolution layers to procure the maximum accuracy for bikes (23.6%), which is only 0.4% higher than what VGG16_6 accomplished. Also, the use of ResNeXt50_11 did not perform as well as VGG16_6 in detecting the remaining categories. Overall, VGG16_6 outperformed the other pretrained networks in perception. Further investigation into the classification accuracy results reveals that smaller objects have the highest perceptual gain compared to the other categories. LiCaNet (VGG16_6) resulted in a jump of 0.4% for vehicles, but a rise of 3.4% and 0.6% is procured for pedestrians and bikes, respectively, compared to LiCaNet (LIDAR only) experiment.

The presented results in Table 1 confirm that LiCaNet experiments, for all different pretrained networks, achieved an enhancement in perception accuracy and prominent decrease in displacement error (i.e., increase in motion prediction) compared to our earlier work and MotionNet. Furthermore, as LiCaNet (VGG16_6) attained the best perception accuracy and competitive displacement errors, we summarize in Table 2 its standard deviation (STD) and root mean square error (RMSE) compared to MotionNet and LiCaNet (LIDAR only) experiments. LiCaNet (VGG16_6) secured the lowest motion prediction errors for the three-speed groups compared to the base algorithms in terms of mean, STD, and RMSE. Thus, we can confidently say that the exploitation of camera images in the fusion process assists in achieving an enhanced perception and motion prediction model, with the highest advancement dedicated to small objects.

Table 3 further investigates the success of LiCaNet experiments by restricting the perception accuracy to within the camera FOV. Furthermore, the results are measured based on the distance from the camera sensor. For each object category, we measure perception within the camera FOV at three distance ranges: short-range (S) defined from 0m-10m, medium-range (M) from 11m-20m, and far-range (F) from 21m-30m. Considering the perception accuracy between LiCaNet, LiCaNet (LIDAR only), and MotionNet experiments, we recognize that within the camera FOV, the accuracy is higher for LiCaNet experiments, with the highest rise recorded for small and distant objects.

TABLE 2. Further motion prediction analysis of MotionNet, LiCaNet (LIDAR only), and LiCaNet (VGG16_6) experiments.

Error		Method		
		MotionNet [20]	LiCaNet (LIDAR only)	LiCaNet (VGG16_6)
Static	Mean	0.0236	0.0230	**0.0224**
	STD	0.4222	0.4186	**0.4114**
	RMSE	0.4229	0.4192	**0.4120**
Slow	Mean	0.2534	0.2531	**0.2527**
	STD	0.4418	0.4416	**0.4410**
	RMSE	0.5094	0.5035	**0.5027**
Fast	Mean	1.0778	1.0547	**1.0456**
	STD	1.3302	1.2340	**1.2188**
	RMSE	1.7120	1.6133	**1.6059**

To begin with, comparing the gain of vehicles between LiCaNet (VGG16_6) and LiCaNet (LIDAR only), we note that a gain of 0.7% and 1.4% is attained for short- and far-range, respectively. Similarly, for pedestrians, an increase of 1.7% and 1.9% is procured in the same range groups, respectively. This shows that our proposed model can gather higher accuracy for distant objects. Moreover, smaller objects collected even greater gain in perception within the camera FOV. For example, in the far-range 1.4% improvement is registered for vehicles, while 1.9% for pedestrians and 4.0% for bikes. This proves the potential of LiCaNet in detecting small and distant objects. Furthermore, a natural characteristic in any model is that the detection accuracy decreases with a farther distance from the sensor; nevertheless, the drop with LiCaNet is lower. The accuracy drop between the short- and far-range of vehicles, pedestrians, and bikes is 8.4%, 6.5%, and 4.8% for LiCaNet (LIDAR only); whereas, for LiCaNet (VGG16_6) the drop is merely 7.7%, 6.3%, and 0.2%, respectively.

To summarize, results in Table 3 confirm that the perception accuracy of LiCaNet within camera FOV is substantially better than our earlier work, especially for small and distant objects. Although most of our analysis is limited between LiCaNet (VGG16_6) and LiCaNet (LIDAR only), comparing our proposed LiCaNet with MotionNet leads to even more significant gains.

The inference time is naturally compromised when a fusion network is expanded by adding a camera module. Thus, a trade-off should be made between accuracy and inference time. Although our proposed LiCaNet model

TABLE 3. Evaluation of classification accuracy within the camera FOV based on distance ranges from the camera sensor. The three distance groups are short-range (S) defined from 0m-10m, medium-range (M) from 11m-20m, and far-range (F) from 21m-30m.

Method	Classification Accuracy in Camera FOV (%)														
	Background			Vehicle			Pedestrian			Bike			Others		
	S	M	F	S	M	F	S	M	F	S	M	F	S	M	F
MotionNet [20]	98.5	97.0	94.8	95.7	93.1	83.3	83.9	81.4	76.0	20.0	16.2	33.2	81.2	71.6	57.7
LIDAR fusion [7]	98.7	97.5	95.7	96.0	93.9	87.0	89.4	81.1	83.5	30.5	21.3	31.3	84.9	76.2	63.3
LiCaNet (LIDAR only)	98.6	97.1	95.2	94.5	93.3	86.1	88.6	77.0	82.1	26.2	20.1	31.0	83.4	75.2	61.7
LiCaNet (MobileNetv2_6)	98.8	97.5	95.6	95.2	94.6	87.7	90.0	86.6	**84.1**	28.6	21.9	**38.2**	83.3	75.6	66.4
LiCaNet (VGG16_6)	98.7	97.2	95.3	**95.2**	**94.6**	87.5	**90.3**	**86.7**	84.0	**35.2**	**23.6**	35.0	85.5	**79.5**	**67.7**
LiCaNet (ResNet50_11)	98.8	97.5	95.7	94.3	87.8	87.6	89.3	78.5	83.4	26.9	22.3	34.0	86.5	75.7	66.0
LiCaNet (ResNeXt50_11)	**98.8**	**97.5**	**96.0**	94.8	94.4	**87.9**	89.0	77.9	82.6	21.0	22.7	34.6	**87.4**	75.8	65.4

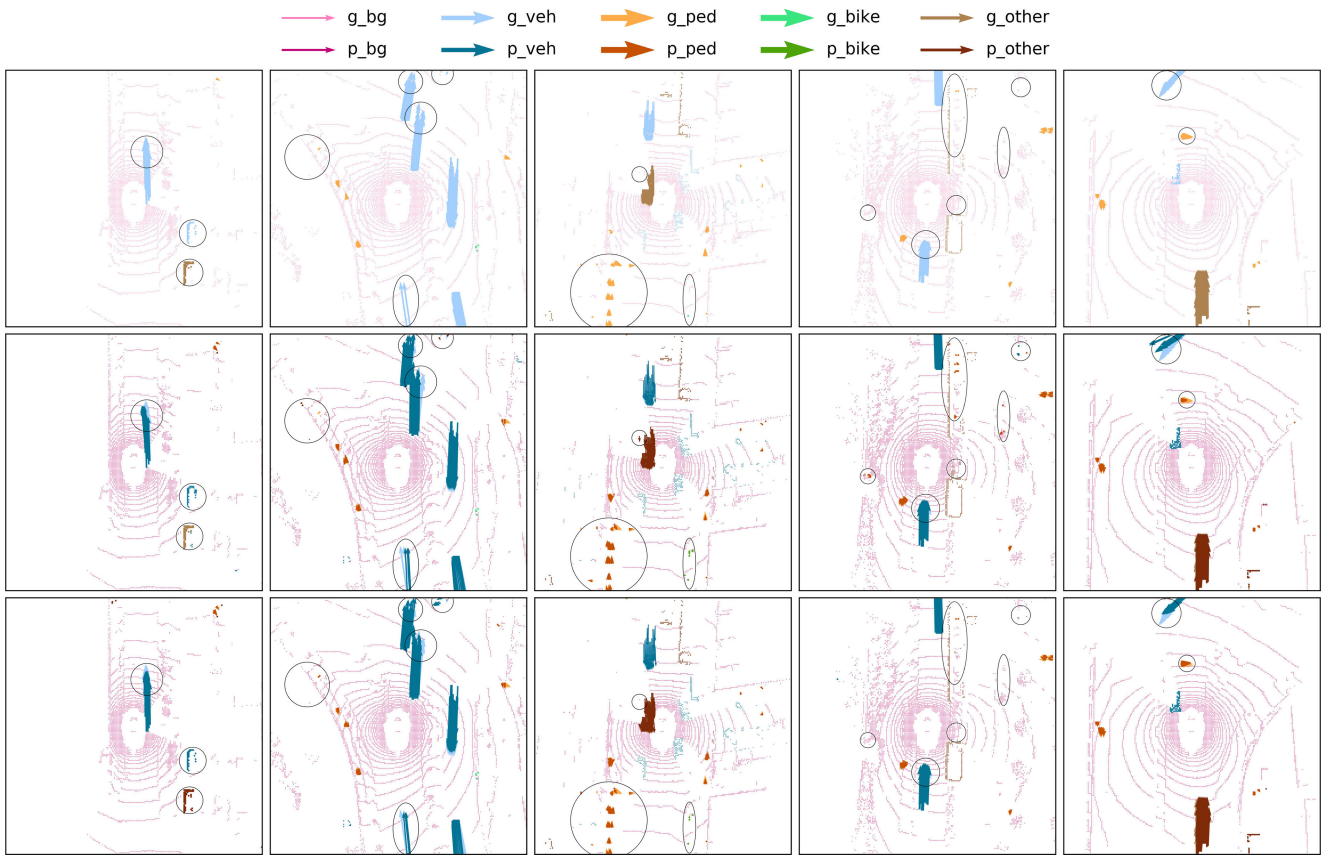

FIGURE 7. Qualitative comparison of perception and motion prediction. Top row: ground truth. Middle row: LiCaNet (LIDAR only). Bottom row: LiCaNet (VGG16_6). Ground truth is also present in the second and third row for easier visual comparison. Color codes are presented at the top of the figure. *g_* and *p_* denotes ground truth and prediction colors, respectively.

involves a camera module, all experiments with the different pretrained networks did not exceed the real-time requirement. Therefore, we can safely conclude that LiCaNet is suitable for autonomous driving applications. According to the results in Table 1, incorporating a camera module onto our fusion network results in a minimum increase of 2.6ms. This is recorded for MobileNetv2, where 6 convolution layers are utilized. VGG16_6 contains more parameters than MobileNetv2 and thus is heavier than MobileNetv2_6, explaining the reason behind the additional 0.3ms in inference time. Furthermore, the use of 11 convolution layers will undoubtedly consume additional time compared to 6 layers. As ResNeXt50 has a denser architecture than ResNet50, its inference time is the maximum (33.2ms) compared to the rest of the used pretrained networks.

Lastly, an advantage of our proposed LiCaNet solution is that it is transferable. The operation of LiCaNet does not depend on the sensor's specifications. For instance, if the LIDAR sensor generates a different number of points, it will not affect the operation of LiCaNet, as LIDAR points are converted to 2D image representations (BEV and RV). The

TABLE 4. Performance comparison with other state-of-the-art methods.

Method	Static		Slow		Fast		Class. Acc. (%)	
	Mean	Median	Mean	Median	Mean	Median	MCA	OA
FlowNet3D (pretrained) [46]	2.0514	0	2.2058	0.3172	9.1923	8.4923	-	-
FlowNet3D [46]	0.0410	0	0.8183	0.1782	8.5261	8.0230	-	-
HPLFlowNet (pretrained) [47]	2.2165	1.4925	1.5477	1.1269	5.9841	4.8553	-	-
HPLFlowNet [47]	**0.0041**	0.0002	0.4458	**0.0960**	4.3206	2.4881	-	-
PointRCNN [3]	0.0204	0	0.5514	0.1627	3.9888	1.6252	55.4	96.0
LSTM-Encoder-Decoder [48]	0.0358	0	0.3551	0.1044	1.5885	1.0003	69.6	92.8
LiCaNet (VGG16_6)	0.0224	**0**	**0.2527**	0.0969	**1.0456**	**0.7300**	**73.9**	**96.9**

transferability has been proven in [45], where the network has been applied to an autonomous driving simulator.

E. QUALITATIVE RESULTS

Qualitative results for LiCaNet are displayed in Fig. 7. We present five scenes to visually compare LiCaNet (LIDAR only) predictions with LiCaNet (VGG16_6). Each scene is displayed in a column; the first row displays the ground truth, the second row shows LiCaNet (LIDAR only) predictions, and the last row depicts LiCaNet (VGG16_6) predictions. The ground truth is also included in the second and third rows for easier visual comparison. Arrows represent motion predictions. The presented examples demonstrate many prediction differences between the experiments; yet, we only label the most apparent ones with circles to simplify the comparison process for the reader. It is obvious that the overlap between LiCaNet (VGG16_6) predictions and the ground truth is higher compared to LiCaNet (LIDAR only) and the ground truth, indicating better accuracy attained by LiCaNet (VGG16_6). Indeed, these examples vividly illustrate that our proposed LiCaNet model has enhanced perception and motion prediction than LiCaNet (LIDAR only). To that end, in addition to quantitative analysis, we qualitatively proved that incorporating camera images in the fusion network enhances performance.

Furthermore, Fig. 8 provides visual comparison on perception and motion prediction confined to the camera FOV (70°). Examining the overlap between the predictions and the ground truth, we observe that the accuracy level is higher within the camera FOV for LiCaNet (VGG16_6), especially for small and distant objects. Thus, this comparison further validates the positive effect of adding semantic camera features to the fusion network.

F. ABLATION STUDIES

We conduct extensive ablation experiments to prove the effectiveness of LiCaNet (VGG16_6) against other state-of-the-art methods. We compare LiCaNet performance to FlowNet3D [46], HPLFlowNet [47], PointRCNN [3], and LSTM-Encoder-Decoder [48]. In addition to the finetuned FlowNet3D and HPLFlowNet, we include the results of their pretrained models. The two scene flow datasets used for the pretrained models are FlyingThings3D and KITTI Scene, while finetuned on nuScenes. FlowNet3D and HPLFlowNet

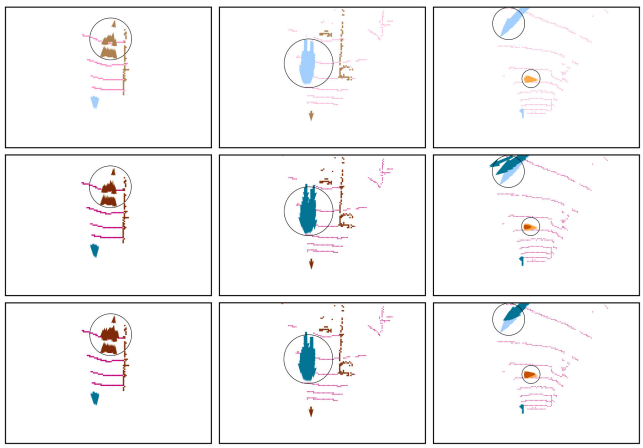

FIGURE 8. Examples of perception and motion prediction within camera range. Ground truths are presented in all rows. The second and third rows include the outcomes of LiCaNet (LIDAR only) and LiCaNet (VGG16_6), respectively.

models estimate the scene flow between two point clouds, while PointRCNN predicts directly from a point cloud. LSTM-Encoder-Decoder estimates the multi-step 2D grid map representation of the point cloud. Table 4 reveals that LiCaNet (VGG16_6) exceeds the state-of-the-art in joint perception and motion prediction. Even though the mean error in the static and the median in the slow groups are not the lowest; nonetheless, the overall LiCaNet performance largely outperforms the other methods, especially in the fast group. All these comparisons collectively show the potential of LiCaNet in joint perception and motion prediction.

V. CONCLUSION

We presented a new method, named LiCaNet, that fuses multi-modal features into a backbone network to perform accurate pixel-wise joint perception and motion prediction for autonomous driving. LIDAR and camera sensors are used to extract rich and complementary multi-modal features. The LIDAR data is represented in sequential BEV and RV forms. The predictions are attained in real-time, making LiCaNet suitable for real-world autonomous driving applications. Our experimental evaluation confirms that the involvement of camera information results in enhanced performance for joint perception and motion prediction. In addition, most accuracy improvement is registered within the

camera field-of-view region, with the highest recorded for small and distant objects. Overall, LiCaNet outperforms our earlier multi-view LIDAR-based fusion network, MotionNet, and other state-of-the-art models.

REFERENCES

[1] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," in *The DARPA Urban Challenge*. Heidelberg, Germany: Springer, 2009, pp. 1–59.

[2] J. Leonard *et al.*, "A perception-driven autonomous urban vehicle," *J. Field Robot.*, vol. 25, no. 10, pp. 727–774, 2008.

[3] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 770–779.

[4] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2021.

[5] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4490–4499.

[6] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 9775–9784.

[7] Y. H. Khalil and H. T. Mouftah, "End-to-end multi-view fusion for enhanced perception and motion prediction," in *Proc. 94th IEEE Veh. Techol. Conf.*, Norman, OK, USA, 2021, pp. 1–6.

[8] Y. H. Khalil and H. T. Mouftah, "Integration of motion prediction with end-to-end latent RL for self-driving vehicles," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2021, pp. 1111–1116.

[9] S. Fadadu *et al.*, "Multi-view fusion of sensor data for improved perception and prediction in autonomous driving," 2020, *arXiv:2008.11901*.

[10] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds," in *Proc. Int. Symp. Vis. Comput.*, 2020, pp. 207–222.

[11] C. Xu *et al.*, "SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 1–19.

[12] A. Laddha, S. Gautam, G. P. Meyer, C. Vallespi-Gonzalez, and C. K. Wellington, "RV-FuseNet: Range view based fusion of time-series LiDAR data for joint 3D object detection and motion forecasting," 2020, *arXiv:2005.10863*.

[13] D. Feng *et al.*, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[14] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Vallespi-Gonzalez, "Sensor fusion for joint 3D object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, 2019, pp. 1230–1237.

[15] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 7345–7353.

[16] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani, "FuseMODNet: Real-time camera and LiDAR based moving object detection for robust low-light autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 2393–2402.

[17] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and validation of 3D object position using multi-modal sensor fusion and semantic segmentation," *Sensors*, vol. 20, no. 4, p. 1110, 2020.

[18] J. Nie, J. Yan, H. Yin, L. Ren, and Q. Meng, "A multimodality fusion deep neural network and safety test strategy for intelligent vehicles," *IEEE Trans. Intell. Veh.*, vol. 6, no. 2, pp. 310–322, Jun. 2021.

[19] S. Jagannathan, M. Mody, J. Jones, P. Swami, and D. Poddar, "Multi-sensor fusion for automated driving: Selecting model and optimizing on embedded platform," in *Proc. IS T Int. Symp. Electron. Imag.*, vol. 2018, 2018, pp. 1–5.

[20] P. Wu, S. Chen, and D. N. Metaxas, "MotionNet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 11385–11395.

[21] H. Caesar *et al.*, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 11621–11631.

[22] S. Shi *et al.*, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10529–10538.

[23] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, "RangeDet: In defense of range view for LiDAR-based 3D object detection," 2021, *arXiv:2103.10039*.

[24] A. Bewley, P. Sun, T. Mensink, D. Anguelov, and C. Sminchisescu, "Range conditioned dilated convolutions for scale invariant 3D object detection," 2020, *arXiv:2005.09927*.

[25] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, "AMVNet: Assertion-based multi-view fusion network for LiDAR semantic segmentation," 2020, *arXiv:2012.04934*.

[26] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[27] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[29] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," 2016, *arXiv:1605.06409*.

[30] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sens.*, vol. 13, no. 1, p. 89, 2021.

[31] Y. Cai *et al.*, "YOLOv4-5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, Mar. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9374990

[32] E. Hassan, Y. Khalil, and I. Ahmad, "Learning feature fusion in deep learning-based object detector," *J. Eng.*, vol. 2020, May 2020, Art. no. 7286187.

[33] X. Hu *et al.*, "SINet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1010–1019, Mar. 2019.

[34] D. Guo, L. Zhu, Y. Lu, H. Yu, and S. Wang, "Small object sensitive segmentation of urban street scene with spatial adjacency between object classes," *IEEE Trans. Image Process.*, vol. 28, pp. 2643–2653, 2019.

[35] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8759–8768.

[36] C. Ma, Y. Guo, Y. Lei, and W. An, "Binary volumetric convolutional neural networks for 3-D object recognition," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 1, pp. 38–48, Jan. 2019.

[37] L. Porzi, S. R. Bulò, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 8277–8286.

[38] Y. Xiong *et al.*, "UPSNet: A unified panoptic segmentation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8818–8826.

[39] D. Yang, X. Zhong, D. Gu, X. Peng, and H. Hu, "Unsupervised framework for depth estimation and camera motion prediction from video," *Neurocomputing*, vol. 385, pp. 169–185, Apr. 2020.

[40] Y. Cui *et al.* "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 722–739, Feb. 2022.

[41] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Computer Vision (ECCV)*. Cham, Switzerland: Springer Int., 2018, pp. 663–678.

[42] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential fusion for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 4604–4612.

[43] V. A. Sindagi, Y. Zhou, and O. Tuzel, "MVX-Net: Multimodal VoxelNet for 3D object detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 7276–7282.

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

[45] Y. H. Khalil and H. T. Mouftah, "Exploiting multi-modal fusion for urban autonomous driving using latent deep reinforcement learning," submitted for publication.
[46] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 529–537.
[47] X. Gu, Y. Wang, C. Wu, Y. J. Lee, and P. Wang, "HPLFlowNet: Hierarchical permutohedral lattice FlowNet for scene flow estimation on large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3254–3263.
[48] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-term occupancy grid prediction using recurrent neural networks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 9299–9305.

YASSER H. KHALIL (Member, IEEE) received the B.Eng. degree (Hons.) in computer engineering from the American University of Kuwait in 2015, and the M.Sc. degree (Hons.) from Kuwait University in 2019. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Ottawa. He has two years of industrial experience and over four years in academia. He is currently appointed as a Research Assistant with the University of Ottawa. His current research interests include artificial intelligence, deep learning, reinforcement learning, autonomous driving, sensor fusion, perception, and motion prediction.

HUSSEIN T. MOUFTAH (Life Fellow, IEEE) received the B.Sc. degree in electrical engineering and the M.Sc. degree in computer science from the University of Alexandria, Egypt, in 1969 and 1972, respectively, and the Ph.D. degree in electrical engineering from Laval University, Canada, in 1975. He joined the School of Electrical Engineering and Computer Science (was School of Information Technology and Engineering), University of Ottawa in 2002 as a Tier 1 Canada Research Chair Professor, where he became a Distinguished University Professor in 2006. He was with the ECE Department, Queen's University from 1979 to 2002, where he was prior to his departure a Full Professor and the Department Associate Head. He has six years of industrial experience mainly with Bell Northern Research of Ottawa (Nortel Networks). He is the author or coauthor of 13 books, 73 book chapters and more than 1800 technical papers, 17 patents, 6 invention disclosures and 148 industrial reports. He is the joint holder of 25 best/outstanding paper awards. He has received numerous prestigious awards, such as the C. Gotlieb Medal in Computer Science and Engineering, the 2016 R.A. Fessenden Medal in Telecommunications Engineering of IEEE Canada, the 2015 IEEE Ottawa Section Outstanding Educator Award, the 2014 Engineering Institute of Canada K. Y. Lo Medal, the 2014 Technical Achievement Award of the IEEE Communications Society Technical Committee on Wireless Ad Hoc and Sensor Networks, the 2007 Royal Society of Canada Thomas W. Eadie Medal, the 2007–2008 University of Ottawa Award for Excellence in Research, the 2008 ORION Leadership Award of Merit, the 2006 IEEE Canada McNaughton Gold Medal, the 2006 EIC Julian Smith Medal, the 2004 IEEE ComSoc Edwin Howard Armstrong Achievement Award, the 2004 George S. Glinski Award for Excellence in Research of the University of Ottawa Faculty of Engineering, the 1989 Engineering Medal for Research and Development of the Association of Professional Engineers of Ontario, and the Ontario Distinguished Researcher Award of the Ontario Innovation Trust. He served as the Editor-in-Chief of the *IEEE Communications Magazine* from 1995 to 1997 and IEEE ComSoc Director of Magazines from 1998 to 1999, a Chair of the Awards Committee from 2002 to 2003, the Director of Education from 2006 to 2007, and a member of the Board of Governors from 1997 to 1999 and from 2006 to 2007. He has been a Distinguished Speaker of the IEEE Communications Society from 2000 to 2007. He is a Fellow of the Canadian Academy of Engineering in 2003, the Engineering Institute of Canada in 2005, and the Royal Society of Canada RSC Academy of Science in 2008.