IEEE Open Journal of
Intelligent Transportation
Systems

Received 30 June 2023; revised 2 September 2023, 16 October 2023, and 7 November 2023; accepted 7 November 2023. Date of publication 9 November 2023; date of current version 1 December 2023.

Digital Object Identifier 10.1109/OJITS.2023.3331437

Radar Translation Network Between Sunny and Rainy Domains by Combination of KP-Convolution and CycleGAN

JINHO LEE [1] (Graduate Student Member, IEEE), GEONKYU BANG [1], TOSHIAKI NISHIMORI [2], KENTA NAKAO [3], AND SHUNSUKE KAMIJO [1] (Senior Member, IEEE)

[1] Graduate School of Interdisciplinary Information Studies, The University of Tokyo, Tokyo 113-8654, Japan
[2] Infrastructure Facilities Business Division, Electricity & Control Systems Engineering Department, Mitsubishi Heavy Industries Machinery Systems Ltd., Kobe 652-8585, Japan
[3] Digital Innovation Headquarters, CIS Department, Mitsubishi Heavy Industries Ltd., Kobe 652-0854, Japan

CORRESPONDING AUTHOR: J. LEE (e-mail: leejinho@kmj.iis.u-tokyo.ac.jp).

ABSTRACT Recently, research on autonomous driving has focused on the advent of various deep learning algorithms. The main sensors for autonomous driving include cameras, LiDAR, and radar, but these algorithms primarily focus on image and LiDAR data. This is because radar data is limited compared to image and LiDAR data. To address the lack of data problem, GAN-based translation methods have been proposed. However, these methods also focus only on image and LiDAR data, such as day-to-night translation or sunny-to-adverse weather translation. Since radar data differs depending on radar sensors and radar points are too sparse to learn patterns compared to LiDAR, translation with radar data is a challenging task. Radar is usually utilized as a sensor that is nearly unaffected by the weather. However, it has been confirmed through JARI data collected by us that rain has a negative effect. CycleGAN is useful for data translation in traffic scenes where pair data is difficult to acquire, since CycleGAN is a network specialized in style translation. KP-Convolution is a module specialized in feature extraction of points while maintaining location information. Therefore, we propose a radar translation network between sunny and rainy domains by combining KP-Convolution and CycleGAN. In this process, we address the adverse effects of radar data by rain, establishing the training format of radar data, KP-Convolution which can learn patterns despite a small number of points, and CycleGAN which is the basis of the translation method.

INDEX TERMS Radar translation, autonomous driving, driving simulator, weather condition, deep learning, data-driven approach.

I. INTRODUCTION

AUTONOMOUS driving research has been focused on the advent of various technologies. Among these technologies, deep learning is considered a representative one as it can be used to implement various tasks such as object detection [1], [2], [3], segmentation [4], [5], [6], and depth estimation [7], [8], [9]. While these methods have been developed to take images and LiDAR as input, deep learning-based algorithms with radar data are relatively scarce. This scarcity is primarily due to the fact that radar data released is less abundant compared to images and LiDAR.

To address the lack of data problem, several data augmentation methods have been proposed, including cropping, flipping, adding noise, MixUp, and GAN-based methods. However, these methods are predominantly utilized with image data. Point-based data such as LiDAR and radar have limitations when applying such methods due to the absence of a standardized format like RGB values in images. In particular, radar data is challenging to augment because it exhibits significant variation depending on the sensor type,

The review of this article was arranged by Associate Editor Xin Xia.

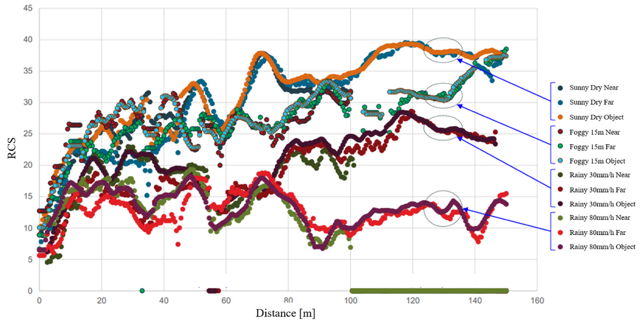


FIGURE 1. Adverse effects of rain and foggy in the JARI data. The RCS value is attenuated in proportion to the amount of precipitation. In particular, there is a big difference between sunny and rainy 80mm/h. RCS difference values are between 15 and 20. With these results, we confirmed that the radar has a great influence of rain.

and the number of points representing a scene is typically smaller than that of LiDAR.

In autonomous driving, LiDAR and radar sensors are commonly used in applications such as Adaptive Cruise Control (ACC) and Emergency Brake Assist (EBA) for both automotive passenger vehicles and commercial vehicles. LiDAR relies on light to measure distances. It leads that it can provide high reliable and richer environmental information, compared to other onboard sensors. This advantage is leveraged for various tasks in autonomous driving, such as integrated inertial LiDAR-based map matching localization for varying environments [10], [11]. However, one of the major disadvantages of LiDAR is that it can be significantly affected by environmental factors. Radar sensor which relies on radio wave to measure distances has the advantage of being unaffected by climate changes. This is widely recognized by the public. However, we have observed the adverse effects of rain in the JARI data collected by our team. These adverse effects indicate that the Radar Cross Section (RCS) value is attenuated in proportion to the amount of precipitation, as shown in Figure 1. The RCS (i.e., reflection intensity) is the ratio of back-scatter power to the power density received by the target.

In this paper, we propose a radar translation method with handling the adverse effects of radar data by rain. To implement this radar translation, we first formulate the radar data format for training and apply KP-Convolution [12] to CycleGAN [13], which serves as the foundation of our translation method. The KP-Convolution is capable of learning patterns in radar data, even with a small number of points. In summary, the contribution of our proposed method is the first radar translation based on deep learning that specifically tackles the adverse effects of rain on radar data. We also anticipate that the proposed method can not only contribute to data augmentation but also aid in the development of driving simulators for the verification of various autonomous driving algorithms.

The remainder of this paper is organized into the following sections: Section II introduces the related work, Section III presents our proposed method, Section IV provides the experimental results, Section V discusses more in detail, and finally, in Section VI, we conclude and address directions for future work.

II. RELATED WORK

In this section, we address related researches with the proposed method. Firstly, we introduce common deep learning-based translation methods that are widely applied in various fields. After that, we discuss several research studies related to radar data augmentation. Also, we explore the phenomenon of radar data being affected by weather by introducing relevant research. Finally, We describe general radar researches for cooperative and autonomous driving. The advantages of the proposed method are specified through a comparison with related researches.

A. DEEP LEARNING BASED TRANSLATION METHOD

When the translation method was initially proposed, the applied main data was images, primarily for data augmentation purposes. Translation research gained momentum with the advent of GAN [14]. GAN-based methods have made significant contributions to various tasks, including image generation [15], [16] and image editing [17], [18]. Inspired by a similar concept, there are other image applications such as text-to-image [19] and image inpainting [20]. The core factor of GAN is the adversarial loss, which enforces the generated images to be indistinguishable from real images.

Following the success of GAN [14], many research studies on paired image translation have been proposed, with pix2pix [21] being a representative one. Pix2pix can generate different output images based on an input image by training pairs of input images and corresponding output images. This idea has been adopted in various tasks, including image generation from sketches [22]. While paired image translations have made significant contributions, it is challenging to apply them in traffic scenes. This difficulty arises from the inability to obtain fully paired images in a traffic scene, such as paired images between day and night. To overcome this issue, unpaired image translation methods have been introduced with the emergence of CycleGAN [13]. In these methods, paired images are not required as CycleGAN learns the style of the source domain and transfers it to the target domain using the concept of cycle-consistency. This idea has been utilized in diverse fields, such as voice conversion [23], MR-to-CT for medical purposes [24], and video translation [25]. Given that the proposed method for the traffic scene also utilizes unpaired data, we have designed a network architecture based on CycleGAN [13].

The aforementioned research has primarily focused on image data, but there are also several deep learning-based translation methods for data augmentation with LiDAR. One of the most representative methods is [26], which considers various variations of LiDAR. However, it has a significant limitation in that the height information is lost, as the method utilizes 3D point cloud data represented as a 2D Bird-Eye-View (BEV). Furthermore, it solely focuses on data translation between simulated and real LiDAR data, without considering weather conditions. In addition to this, several research works have explored denoising and translating LiDAR point clouds using CNNs, GANs, and statistical filters [27], [28]. Recent studies have started considering weather changes and height information in LiDAR translation, as demonstrated in [29], [30]. While image and LiDAR data have been the subject of numerous translation research, radar data has received less attention.

B. RADAR DATA AUGMENTATION

There are very few research studies focusing on radar data augmentation. A representative research in radar data augmentation is [31], which utilizes CNN with domain-specific data augmentation operations. The augmentation operations include target translation, randomization of speckle noise in different observations, and the inclusion of pose images in the training data. However, this method has limitations as it is time-consuming and suitable for only limited targets. Another method for radar data augmentation [32], introduced by Sheeny, M. in 2020, is the parameterized radar data augmentation (RADIO) technique, which generates realistic radar samples from small datasets. It leverages the physical properties of radar signals, such as attenuation, azimuthal beam divergence, and speckle noise. While it is meaningful for radar augmentation using deep learning methods, it only tests simple target objects in small datasets and lacks stability while not fully considering road environments. Despite other methods related to radar data augmentation [33], [34], there is still no radar data augmentation method that takes weather conditions into consideration.

C. PHENOMENON OF RADAR DATA AFFECTED BY WEATHER

In general, radar sensors have the advantage of being unaffected by weather conditions in autonomous driving. However, compared to other sensors, radar is still susceptible to weather effects, especially in rainy conditions. The research study [35] provides an in-depth analysis of the changes in radar data due to weather effects. This work offers insights into the different physical principles that contribute to radar signal disturbance and presents theoretical investigations for estimating weather influence. According to this study, water vapor in the air interferes with the radar signal during rain and fog, resulting in a weakening of the reflection intensity (RCS value) or scattering of radar points. Another research study [36] also analyzes the impact of weather on radar data, specifically focusing on rain, which has a direct influence on traffic accidents compared to previous studies. Additionally, there is a research study [37] that conducts a detailed analysis of the released radar dataset (i.e., nuScenes) independent of weather conditions. This research [37] quantitatively investigates radar properties, such as detection thresholds and detection probabilities depending on objects, environment, and radar parameters, as well as object properties, such as reflection behavior depending on object type. However, it is important to note that the aforementioned studies are focused on analysis rather than radar translation.

In the previously mentioned research studies, there is a lack of translation methods based on deep learning (i.e., data-driven approaches) that can accurately replicate weather changes in radar data. Therefore, we propose a radar translation method that addresses the adverse effects of rain on radar data. To achieve this, we employ KP-Convolution [12] in conjunction with CycleGAN [13], which forms the foundation of our translation approach. We anticipate that the proposed method will make significant contributions to radar data augmentation and driving simulators for autonomous driving.

D. GENERAL RADAR RESEARCHES FOR COOPERATIVE AND AUTONOMOUS DRIVING

Radar, due to its ability to provide relative positional information between vehicles, has been a foundation for various researches in cooperative and autonomous driving. One notable research is [38]. This paper introduces a secure cooperative localization system for connected automated vehicles (CAVs). It describes the consensus estimation, which integrates sensor data shared by multiple neighboring vehicles. The results demonstrate that collaborative information utilization from multiple vehicles leads to robust vehicle position estimation against attacks with higher accuracy and resilience. Another remarkable study is [39]. This study provides a comprehensive overview of the current state of vehicle control technologies in connected and automated vehicles, emphasizing key areas of interest and future research directions. Indeed, there are various researches [40], [41], [42], [43], [44] associated with cooperative and autonomous driving that are continually being introduced. Through our proposed method, we anticipate that providing diverse radar data will promote and invigorate cooperative and autonomous driving researches, such as the researches mentioned above.

III. PROPOSED METHOD

In this section, we provide a detailed explanation of the proposed method. The objective of this research is to generate and translate realistic radar data between sunny and rainy conditions. To achieve this, the proposed method incorporates KP-Convolution [12] and CycleGAN [13]. We provide a brief overview of these components before delving into the specifics of the proposed method.

Furthermore, we address the challenge of training radar data, which has a distinct format compared to the original image-based CycleGAN. While most image data consists of RGB values, radar data encompasses various components. In the proposed method, we utilize RCS (reflection intensity) and relative velocity values for training. It's worth noting that unlike RGB values, RCS values can be negative, and the scale of the introduced radar data format differs from that of images.

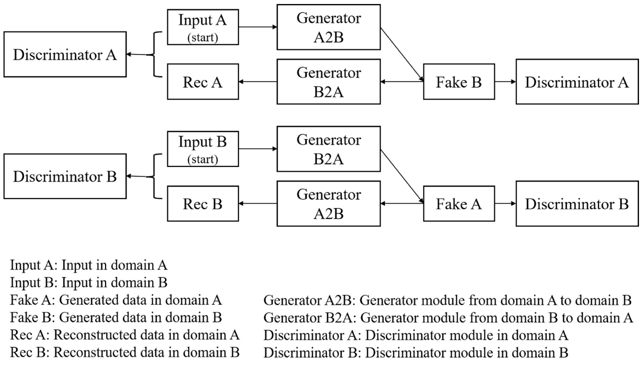

Discriminator A
Input A (start)
Generator A2B
Rec A
Generator B2A
Fake B
Discriminator A
Discriminator B
Input B (start)
Generator B2A
Rec B
Generator A2B
Fake A
Discriminator B
Input A: Input in domain A
Input B: Input in domain B
Fake A: Generated data in domain A
Fake B: Generated data in domain B
Rec A: Reconstructed data in domain A
Rec B: Reconstructed data in domain B
Generator A2B: Generator module from domain A to domain B
Generator B2A: Generator module from domain B to domain A
Discriminator A: Discriminator module in domain A
Discriminator B: Discriminator module in domain B

FIGURE 2. **The overall flow of radar translation based on CycleGAN [13] involves the utilization of KP-Convolution [12] for feature extraction from point data, unlike the original CycleGAN [13] designed for image translation that uses ResNet-based feature extraction modules.**

To address these challenges, we have designed a network by incorporating KP-Convolution [12] into the CycleGAN [13]. Figure 2 illustrates the overall architecture of the radar translation based on CycleGAN [13], while Figure 3 demonstrates the feature extraction module, which is based on KP-Convolution [12] and KP-FCNN, a segmentation network that utilizes KP-Convolution.

A. TRANSLATION NETWORK BASED ON CYCLEGAN

Image-to-image translation is a field of generative models that aims to map input images to output images within an image dataset. It enables various transformations such as colorizing black and white images [45], translating daytime images to nighttime images [46], and converting images with only borders into realistic objects [22]. Representative models in image-to-image translation include pix2pix [21], CycleGAN [13], and StyleGAN [47].

The initial model, pix2pix [21], was followed by CycleGAN [13], which aimed to address the limitations of pix2pix. The key objective of CycleGAN [13] is to perform image-to-image translation using unpaired image data. One of the common challenges in GAN models is the mode collapse problem, where the generator fails to produce diverse outputs and generates similar results repeatedly, resulting in a collapsed state. To mitigate this issue, CycleGAN introduces a cycle-consistency loss by incorporating a cycle structure into the existing GAN architecture.

The distinguishing characteristic of the CycleGAN structure is that when an input image is transformed in the forward direction, it should be able to revert back to the original input through the reverse transformation. In Figure 2, we present the overall architecture of the radar translation based on CycleGAN [13].

The proposed radar translation utilized JARI data acquired in the same environment with only different weather factors. However, in order to perform data translation in various environments for real road through the proposed method, unpaired data must be handled. To deal with this issue, the network architecture was designed based on CycleGAN [13].

Thanks to CycleGAN [13] based network, more natural translation is possible than pix2pix [21].

B. FEATURE EXTRACTION MODULE BASED ON KP-CONVOLUTION

Kernel Point Convolution (KP-Convolution) is a method that applies convolution directly to a point cloud without converting it to a graph or 3D voxel. It is among various methods used for processing 3D point cloud data. Prior to the introduction of KP-Convolution, PointNet, an MLP (i.e., multi-layer perceptron) [48] based network was representative. While KP-Convolution [12] applies convolution to the point cloud as it is without conversion (e.g., voxel or grid conversion), MLP does not. Thus, MLP based method leads to that positional information is somewhat lost.

Besides MLP [48], there is another layer that can be applied to points for convolution-based feature extraction, which is the 1x1 convolution [49]. Unlike MLP [48], which requires a fixed input size, 1x1 convolution [49] layers are more flexible as they can accommodate varying input sizes. 1x1 convolution [49] layers have three key advantages. Firstly, they allow reducing the number of channels while keeping the size of the feature map unchanged, using relatively few parameters. Secondly, they enable increasing depth while significantly reducing the number of parameters, preventing the model from becoming overly complex and reducing the risk of overfitting. Finally, by using 1x1 convolution [49] layers, depth can be increased, allowing for the use of activation functions at each layer, thereby increasing non-linearity and enhancing the model's ability to handle more complex problems. Despite these advantages, there is a concern that 1x1 convolution [49] layers might lose spatial information, as well.

A convolution kernel of KP-Convolution [12] is composed of several kernel points, and the convolution operation for the surrounding points is performed by arranging kernel weights with continuous values for each kernel point. The network capacity can be adjusted by flexibly setting the number and location of kernel points, and the shape of the kernel can be optimized based on the geometry of the point cloud. KP-Convolution makes significant contributions to tasks such as 3D point cloud classification and segmentation, with maintaining spatial information.

We show briefly the comparison results of coordinate error with MLP [48], 1x1 convolution [49] layers and KP-Convolution [12] for radar translation in Table 1. As shown in Table 1, we confirmed that KP-Convolution can learn more location information than MLP [48] and 1x1 convolution [49] layers (i.e., the same convolution mechanism with the propose method), through the fact of small coordinate error value for KP-Convolution [12]. It results in the generation of natural radar data. Therefore, we adopted KP-Convolution [12] as feature extraction module for radar points.

TABLE 1. Comparison results of coordinate error with MLP, 1x1 convolution layers and KP-Convolution for radar translation. The MLP consisted of two layers with 32 channels each. The 1x1 convolution layers are also composed of two layers, the first with 16 channels and the second with 32 channels. For KP-Convolution, we used two layers with the same size of 1x1 convolution layers. These configurations adopted the Leaky ReLU activation function in our experiments.

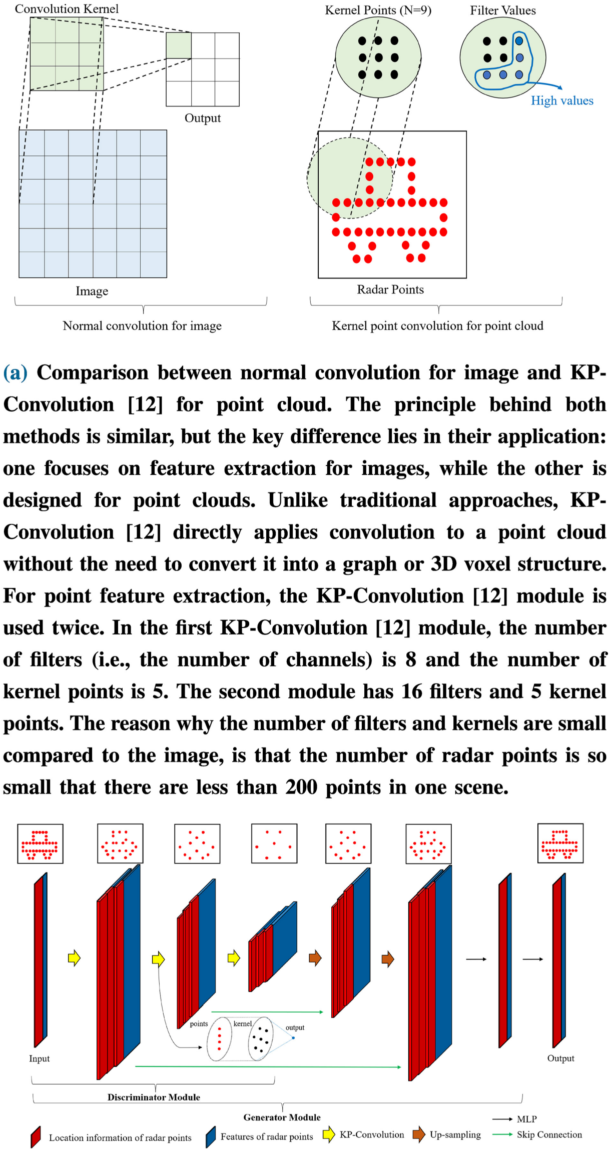
Feature Extraction Module	MLP [48]	1x1 convolutions [49]	KP-Convolution [12]
Coordinate Error [cm]	13.28	11.62	6.181

Similar to U-Net [50], the KP-Convolution network extracts multi-scale feature vectors using pooling and upsampling layers. A sub-sampling process is required during pooling, and a grid subsampling method, independent of the density of the input point cloud, is utilized. This method samples only the representative point from each grid after placing the points on a voxel-shaped grid. During the pooling process, the feature vectors for the points in each cell are integrated through KP-Convolution after reducing the number of output points by doubling the cell size of the previously defined grid. This process is commonly referred to as strided KP-Convolution.

The network with KP-Convolution is divided into two parts: KP-CNN for classification and KP-FCNN for segmentation. KP-FCNN shares the encoder part of KP-CNN. Due to the segmentation task's characteristics, point-wise features need to be extracted, which involves adjusting the number of points through the nearest upsampling process. In contrast, KP-CNN directly extracts class information from the encoder feature vector using a fully-connected layer. Figure 3(a) illustrates the comparison between KP-Convolution [12] for feature extraction from point clouds and normal convolution for feature extraction from image. Additionally, in Figure 3(b), we present KP-FCNN with KP-Convolution, which includes a generator module and a discriminator module for the proposed method.

C. RADAR TRANSLATION BASED ON COMBINATION OF CYCLEGAN AND KP-CONVOLUTION

We propose the first deep learning-based radar translation method by applying KP-Convolution [12] to CycleGAN [13]. However, the proposed radar translation is not simply performed by combining these components. Several modifications and additions have been made through verification. To address this in detail, we first explain how radar data is collected, including an explanation of the radar sensor, radar format, and the differences compared to already released radar datasets. After that, we introduce the architecture of the proposed method. Finally, we provide an explanation of the learning process involved in the proposed method.



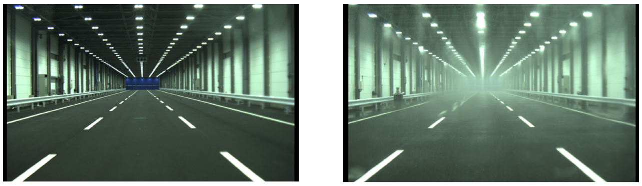
(a) **Comparison between normal convolution for image and KP-Convolution [12] for point cloud. The principle behind both methods is similar, but the key difference lies in their application: one focuses on feature extraction for images, while the other is designed for point clouds. Unlike traditional approaches, KP-Convolution [12] directly applies convolution to a point cloud without the need to convert it into a graph or 3D voxel structure. For point feature extraction, the KP-Convolution [12] module is used twice. In the first KP-Convolution [12] module, the number of filters (i.e., the number of channels) is 8 and the number of kernel points is 5. The second module has 16 filters and 5 kernel points. The reason why the number of filters and kernels are small compared to the image, is that the number of radar points is so small that there are less than 200 points in one scene.**

(b) **Radar point generator and discriminator module based on KP-FCNN. Originally used for point cloud segmentation, KP-FCNN is a network known for its effective feature extraction and restoration capabilities. In the proposed method, the KP-FCNN module was adopted as the generator and discriminator for CycleGAN [13]. The red parts in the network indicate the location information of radar points, while the blue parts represent the features of radar points, including multi-channel data such as RCS and relative velocity values. The yellow arrows represent KP-Convolution [12], and the green arrows represent Skip-Connections [51]. Skip-Connections [51] are employed to maintain and effectively propagate local information.**

FIGURE 3. Outline of the proposed method.

1) JARI DATASET FOR RADAR TRANSLATION

The purpose of the proposed method is to generate realistic radar data between sunny and rainy conditions. One representative and released radar dataset [52] is the nuScenes dataset, which is the first dataset to include a full suite of autonomous vehicle sensors: 6 cameras, 5 radars, and 1 LiDAR. All of these sensors have a full 360-degree field of view. The nuScenes dataset [52] consists of 1000 scenes and



JARI-Sunny

JARI-Rainy

FIGURE 4. Scene examples of the JARI dataset for radar translation. From left to right, they depict sunny and rainy data for the JARI dataset. The JARI dataset is collected with adjusted precipitation levels.

is fully annotated with 3D bounding boxes for 23 classes and 8 attributes. However, this dataset presents a significant challenge when utilized in the proposed method. In order to achieve accurate weather translation, a well-balanced amount of data is required in both the sunny and rainy domains. Unfortunately, the nuScenes dataset contains very limited rainy radar data. Apart from this dataset, the Oxford Radar RobotCar [53] and Astyx HiRes2019 datasets [54] also include radar data. However, these datasets have their limitations. The Astyx HiRes2019 dataset contains sparse radar points instead of raw radar spectra, and the Oxford Radar RobotCar dataset supplies radar spectra without any annotation. In summary, until now, there has not been a high-quality public radar dataset available.

To develop the proposed method, radar data was collected in both sunny and rainy domains using a 77 GHz radar sensor equipped with a digital beamforming scanning antenna. The radar offers two independent scans for far and near ranges. The JARI dataset was collected by this radar sensor. The JARI dataset was acquired in a laboratory provided by the Japan Automobile Research Institute (JARI), where the data collection process involved adjusting the amount of rainfall. More detailed information about the JARI dataset will be provided in Section IV. Figure 4 illustrates scene examples from the JARI dataset.

There are various radar values used to represent radar points. Among these values, the proposed method adopts four elements: x coordinates, y coordinates, RCS values, and relative velocity values. The input format of the radar translation network is formulated as (N, 4), where N indicates the number of radar points in each scene. The range for each value is as follows: $-200 \sim 200$ for x coordinates, $-200 \sim 200$ for y coordinates, $-65 \sim 65$ for RCS values, and $-7 \sim 2.5$ for relative velocity values. The proposed method is trained by separating the location information (i.e., x and y coordinates) and the feature information (i.e., RCS and relative velocity values). When the ranges of the feature information values are significantly different, such as $-65 \sim 65$ for RCS values and $-7 \sim 2.5$ for relative velocity values, the learning loss does not converge and diverges greatly. Therefore, we conducted a normalization process during the creation of the dataset, making the relative velocity values similar to the range of RCS values. It is important to note that we restore the original range of the relative velocity values for quantitative evaluation. This process is not required for images since they maintain the same range of RGB values. For LiDAR translation [29], [30], the above process improves the learning results.

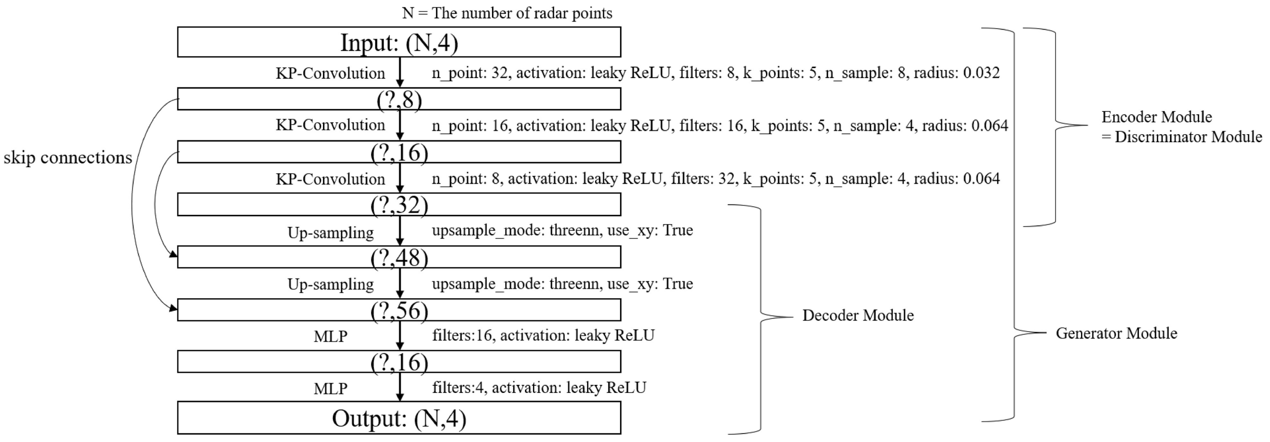
2) ARCHITECTURE OF THE PROPOSED NETWORK

The proposed method focuses on radar translation between sunny and rainy domains. To achieve point-based translation, we apply KP-Convolution [12] to the CycleGAN [13]. The detailed architecture of the proposed network is described in Figure 5. There are several differences compared to the original CycleGAN [13] for image translation.

The first difference is the choice of activation function. While the image-based CycleGAN adopted the Rectified Linear Unit (ReLU) [55], the proposed network utilizes the Leaky Rectified Linear Unit (Leaky ReLU) [56], which is a variant of ReLU. This decision is made because radar data, unlike RGB values in images, often includes many negative values in RCS and relative velocity. Both Leaky ReLU [56] and Parametric ReLU (PReLU) [57] can handle negative values. The main difference between them is that the negative slope of PReLU [57] is determined as a parameter value during the learning process. However, PReLU [57] has a drawback where the negative slope converges to the average value of 0 as the learning epoch progresses, especially when there are as many negative values as positive values, such as in radar data. Since the fixed negative slope of Leaky ReLU [56] is not affected by the learning process, the proposed method is designed with Leaky ReLU [56].

The second difference is in terms of normalization. In image-based networks with fixed input sizes and linear values, various normalization methods such as min-max [58] and batch normalization [59] are often utilized to enhance performance. However, these normalization methods are not effective for the proposed method. There are several reasons for this. Firstly, radar data has an unstable input size and a significantly smaller number of valid data compared to images. The presence of a reflective object determines the validity of a radar point, resulting in a dynamic change in the input size. Secondly, when normalization is applied to radar data, which consists of discrete values unlike the continuous RGB values of images, a significant amount of noise data is generated.

The final difference lies in the decoder part, which is responsible for restoring the encoded features. In image-based networks, the deconvolution method [60] is commonly used to restore encoded features. Deconvolution fills in the empty spaces as the size of the feature map increases, typically in the case of an image. However, when restoring encoded data for radar translation, it is necessary to sample representative points instead of using filling methods. To address this issue, we have adopted the three nearest neighbors search method [61], also known as threeNN, as the upsampling method. This method allows us to sample representative points for the decoded features. Finally, the output, which is of the same size as the input, is generated through two multi-layer perceptron (MLP) [48] layers.


N = The number of radar points
Input: (N,4)
KP-Convolution
n_point: 32, activation: leaky ReLU, filters: 8, k_points: 5, n_sample: 8, radius: 0.032
(?,8)
KP-Convolution
n_point: 16, activation: leaky ReLU, filters: 16, k_points: 5, n_sample: 4, radius: 0.064
(?,16)
KP-Convolution
n_point: 8, activation: leaky ReLU, filters: 32, k_points: 5, n_sample: 4, radius: 0.064
(?,32)
Up-sampling
upsample_mode: threenn, use_xy: True
(?,48)
Up-sampling
upsample_mode: threenn, use_xy: True
(?,56)
MLP
filters:16, activation: leaky ReLU
(?,16)
MLP
filters:4, activation: leaky ReLU
Output: (N,4)
skip connections
Encoder Module
= Discriminator Module
Decoder Module
Generator Module

n_point: Number of points sampled in farthest point sampling
(i.e., number of output points)
activation: Activation function of the kernel point convolution
filters: The dimensionality of the output space
(i.e., the number of output filters in the convolution)

k_points: Number of kernel points (similar to the kernel size)
n_sample: Maximum number of points in each local region
radius: Search radius in local region
upsample_mode: Method used to associate the former features with the latter points
use_xy: Whether to concat x y with the output point features

FIGURE 5. To handle the negative values of radar RCS and relative velocity, the proposed network utilizes Leaky ReLU activation [56]. Normalization methods such as min-max [58] and batch normalization [59] are effective when the input size is fixed and consists of linear values, such as images. However, the proposed radar translation method does not employ normalization methods because the input size dynamically changes and the point values are discrete. Additionally, the three nearest neighbors search method is utilized as the upsampling method.

3) TRAINING STRATEGY OF THE PROPOSED METHOD

The proposed radar translation method is based on CycleGAN [13], and therefore, the loss calculation process is similar to the existing CycleGAN. Two generators, G_{AB} and G_{BA}, are utilized for translation from domain A to B and from B to A, respectively. Two discriminators, D_A and D_B, are defined to determine whether the output is real or fake in each domain, A and B. During training, multiple losses are calculated, including adversarial loss and cycle consistency loss.

Adversarial loss is utilized in a binary classifier that differentiates between ground truth data and generated data predicted by the generative network. Note that it calculates least square loss instead of cross-entropy loss, like a normal GAN.

$$\begin{aligned}\mathcal{L}_{GAN_1} &= \mathbb{E}_{x_b \sim X_b}\left[\log(D_B(x_b))\right] \\ &\quad + \mathbb{E}_{x_a \sim X_a}\left[\log(1 - D_B(G(x_a)))\right], \qquad (1)\\ \mathcal{L}_{GAN_2} &= \mathbb{E}_{x_a \sim X_a}\left[\log(D_A(x_a))\right] \\ &\quad + \mathbb{E}_{x_b \sim X_b}\left[\log(1 - D_A(F(x_b)))\right]. \qquad (2)\end{aligned}$$

Cycle consistency loss is used in generative adversarial networks for unpaired data translation. It addresses the problem of mode collapse by enforcing diversity. $G_{BA}(G_{AB}(x_a)) \approx x_a$ and $G_{AB}(G_{BA}(x_b)) \approx x_b$ are achieved by calculating the L_1 loss between the original and reconstructed data:

$$\begin{aligned}\mathcal{L}_{cycle} &= \mathbb{E}_{x_a \sim X_a}[||G_{BA}(G_{AB}(x_a)) - x_a||_1] \\ &\quad + \mathbb{E}_{x_b \sim X_b}[||G_{AB}(G_{BA}(x_b)) - x_b||_1]. \qquad (3)\end{aligned}$$

Identity loss is a mechanism used to maintain common attributes within data groups. In other words, it ensures that the output remains consistent when data from the target domain is input:

$$\begin{aligned}\mathcal{L}_{id} &= \mathbb{E}_{x_a \sim X_a}[||G_{BA}(x_a) - x_a||_1] \\ &\quad + \mathbb{E}_{x_b \sim X_b}[||G_{AB}(x_b) - x_b||_1]. \qquad (4)\end{aligned}$$

Therefore, we can obtain the final loss as:

$$\mathcal{L}_{total} = \lambda_{GAN}\mathcal{L}_{GAN_1} + \lambda_{GAN}\mathcal{L}_{GAN_2} + \lambda_{cycle}\mathcal{L}_{cycle} + \lambda_{id}\mathcal{L}_{id}, \qquad (5)$$

λ_{GAN}, λ_{cycle} and λ_{id} indicate hyper-parameters of each loss calculation.

In normal image translation, the image sizes of the source and target domains are the same. Even if they are not identical, resizing them to make them identical is one-sided. This process is very important because it is possible to calculate losses when the sizes are the identical. However, radar translation cannot conduct resizing like an image when the number of radar points in the source and target domains is different. Therefore, for calculating losses, we propose the point padding to match the number of radar points in both the source and target domains. This enables the loss calculations described above. General image padding [62] places the minimum RGB value of 0 at the edges. Inspired by this idea, in the point padding, we also place the minimum RCS and minimum relative velocity values (i.e., –65 for RCS and –7 for relative velocity) at the maximum and minimum x, y coordinates, which represent the edge coordinates of the radar points. This ensures that the padded points have the least possible impact on learning. The aforementioned process makes it possible to calculate the losses, which enables learning and even contributes to stable learning.

TABLE 2. Quantitative evaluation to compare error of coordinates, RCS, relative velocity, and the number of points, between the ground truth and generated result. Except the number of points error, the other error values were obtained by point-wise averaging the sum of errors. The number of points error is just a comparison of the number of points.

Sunny to Rainy (80 mm/h)	X Error [cm]	Y Error [cm]	Rec Error	Relative Velocity Error [km/h]	The Number of Points Error
Error Value	2.183	5.173	1.713	0.001513	+9.7

Rainy (80 mm/h) to Sunny	X Error [cm]	Y Error [cm]	Rec Error	Relative Velocity Error [km/h]	The Number of Points Error
Error Value	1.982	6.449	3.121	0.0009815	-7.2

IV. EXPERIMENTAL RESULTS

As mentioned earlier, we conducted experiments using the JARI dataset. The proposed method is evaluated in two ways: qualitative and quantitative evaluations. Qualitative evaluation involves comparing the generated radar data by projecting it into a two-dimensional space and visualizing it. Since objective evaluation alone is not possible through qualitative evaluation, we also conducted quantitative evaluation. Quantitative evaluation was performed by comparing the difference between the x,y coordinates, RCS, relative velocity, and the number of points between the ground truth and generated data. We also compared the maximum and minimum values of RCS for real data and generated data. Finally, the RCS value distribution of radar data using histograms was conducted for qualitative evaluation.

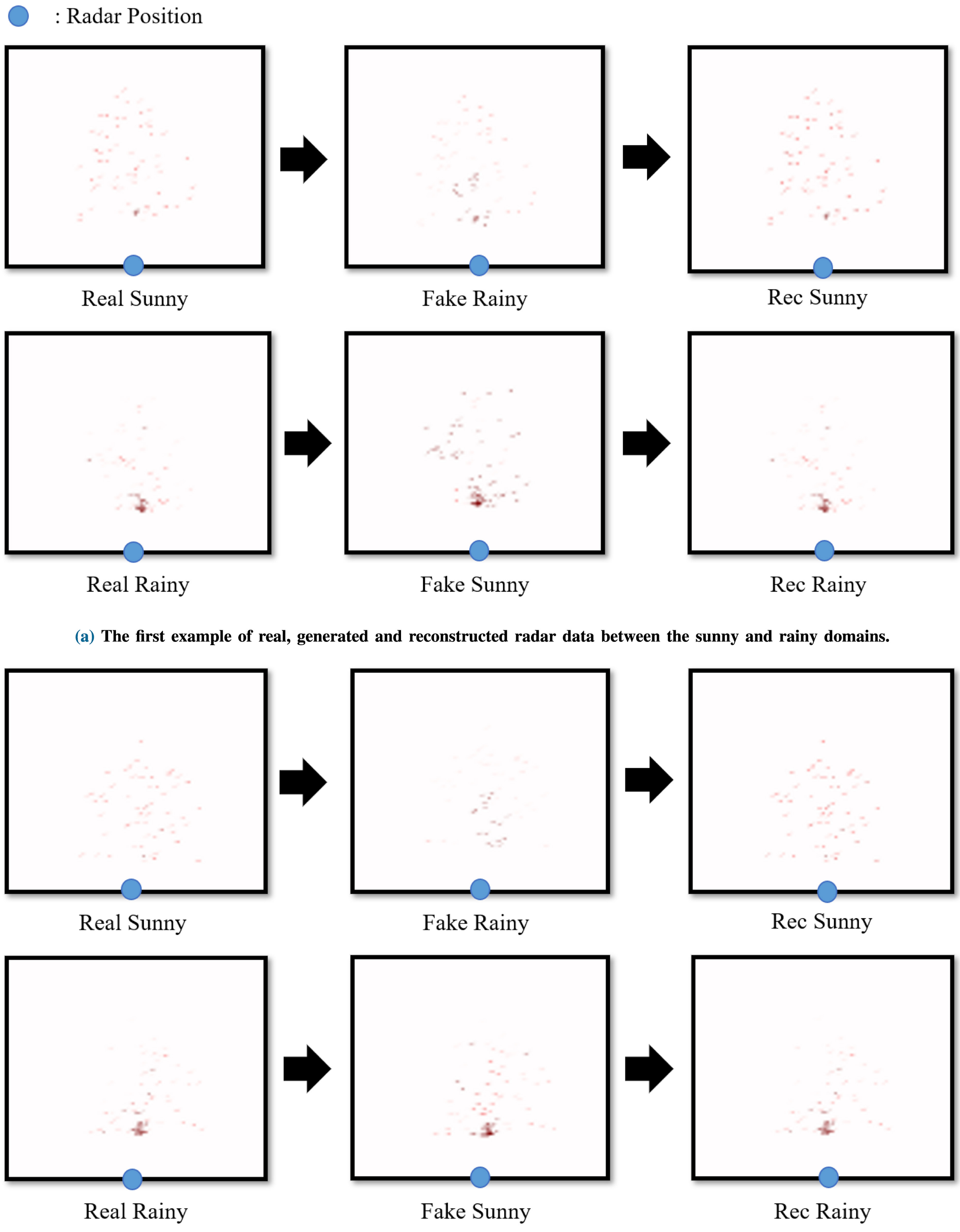
It is important to note that there is no comparable related research among the state of the art researches, because the proposed method is a newly pioneering research field. However, there are LiDAR translation studies [29], [30] that performed quantitative evaluation in a similar way. The LiDAR translations [29], [30] focus on converting LiDAR data from clear weather conditions into data simulating rainy or foggy conditions. In these works, quantitative evaluations were conducted by comparing differences in x, y, z coordinates, distance, and intensity between the ground truth LiDAR data and the generated LiDAR data. For LiDAR data, distance and intensity values play crucial roles. Similarly, our proposed method aims to convert radar data from clear weather conditions into data representing rainy weather. This translation process bears some similarities to the LiDAR translations [29], [30], specifically in terms of weather-induced data translation. In this study, we focus on x, y coordinates, RCS (Radar Cross Section), and relative velocity values for quantitative evaluation as these are the key attributes in radar data.

The JARI data was collected in the JARI laboratory, where the amount of rainfall can be controlled. This allowed us to analyze how the radar was affected by rainfall. It is important to note that the experiments were conducted using both sunny data, where there were no weather effects, and rainy data with a maximum controllable rainfall of 80 mm/h at the JARI laboratory. Since radar data is obtained through the reflection of objects, we compared the change in RCS values between sunny and rainy conditions from a reflective object. The radar sensor to collect data is a 77 GHz-based radar which is mainly used for autonomous driving. Since the radar of 60 GHz higher has strong straightness of radio waves, when the humidity is high, noise data is generated. It is because radio waves of a radar scatter and reflect by water vapor in the air. Also, attenuation of RCS value occurs. For real, the collected data demonstrated that the RCS values for rainy conditions were consistently 15 to 20 units lower than those for sunny conditions, representing the maximum difference in RCS values from the reflective object, as shown in Figure 1. Also, lots of noise data were observed for rainy conditions. In the JARI dataset, the relative speed value is not meaningful because the data was acquired while the vehicle was moving at a constant velocity.

Figure 6 presents examples of qualitative evaluation using the proposed method. The blue circle of each graph represents the location of the radar sensor, with the forward direction depicted towards the up of the graph. The redder, the higher the RCS value. In order from left to right, input, fake data (i.e., generated data), and reconstructed data are shown. As shown in Figure 6, the radar data generated by the proposed method accurately captures the characteristics of each domain. The proposed method is also capable of generating adequately reconstructed radar data, even when compared to the ground truth data. Particularly, in the generated rainy data from the proposed method, the RCS values of points located farther from the radar sensor are relatively lower compared to the sunny domain. It is because radio waves scatter and reflect by rainy vapor. Of course, the opposite phenomenon was reproduced in the sunny data generated by the proposed method. This observation aligns with the analysis results of the JARI data mentioned earlier. Therefore, we can confirm that the proposed method performs effectively.

We conducted three quantitative evaluation because the two-dimensional visualization results alone lack objective evaluation. Table 2 presents an error comparison of coordinates, RCS, relative velocity, and the number of points, between the ground truth and generated result. Except the number of points error, the other error values were obtained by point-wise averaging the sum of errors. The number of points error is just a comparison of the number of points. Those results are obtained between the ground truth and generated data for both sunny and rainy domains. It is possible due to that JARI data were obtained in the same environment, with only the weather elements varied.

The proposed network learns by dividing the x and y coordinate values and feature values, while also concatenating the x and y coordinate features of the encoder to the decoder. Due to these reasons, there are small coordinate errors as shown in Table 2. Considering that the measurable range of the radar is 100 m, the coordinate error values



(a) The first example of real, generated and reconstructed radar data between the sunny and rainy domains.

(b) The second example of real, generated and reconstructed radar data between the sunny and rainy domains.

FIGURE 6. Qualitative evaluation of the proposed method. Each example includes three panels arranged from left to right: 'Real,' 'Fake,' and 'Rec.' The 'Real' panel represents the ground truth data in the corresponding domain. The 'Fake' panel displays the generated radar data from the generator of the proposed method in the corresponding domain. The 'Rec' panel represents the reconstructed radar data derived from the generated data in the corresponding domain. The color of the points indicates the strength of the RCS, with higher values shown in red. The blue circles indicate the location of the radar sensor.

are quite small. For translating to sunny, the RCS error is larger than that for translating to rainy, it is because overall RCS value for sunny is simply larger than that for rainy. The relative velocity error is close to 0, because JARI data is acquired with a constant speed. Note that the number of points error for translating to sunny is lower than that

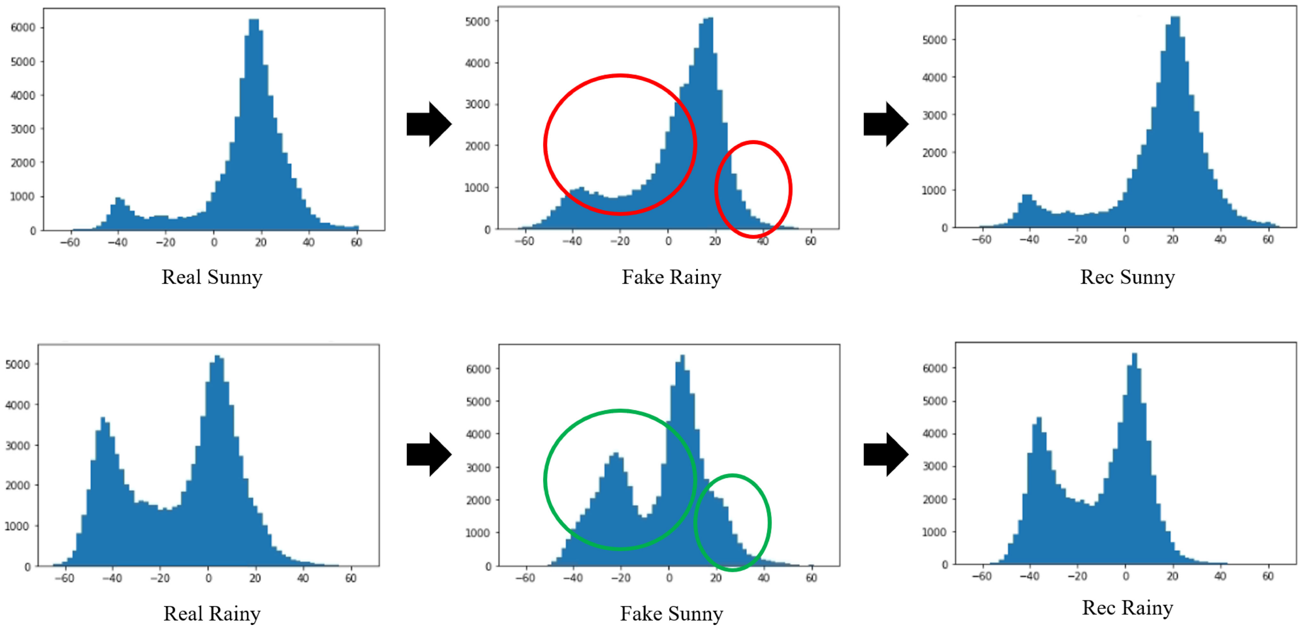


FIGURE 7. RCS distribution of radar data as a histogram between 'Real', 'Fake', and 'Rec' in a whole testset. In this context, 'Real' represents the ground truth data in the corresponding domain. 'Fake' indicates the generated radar data from the generator of the proposed method in the corresponding domain. 'Rec' refers to the reconstructed radar data obtained from the generated data in the corresponding domain. The X and Y axes represent RCS values and the number of RCS values, respectively. The red circle illustrates that RCS values decrease during translation from the sunny to rainy domain due to an increase in noisy data. In contrast, the green circle indicates that RCS values increase during translation from the rainy to sunny domain due to a decrease in noisy data.

TABLE 3. Quantitative evaluation between 'Real', 'Fake', and 'Rec' to compare the maximum and minimum RCS values in each domain. In this context, 'Real' refers to the ground truth data in the corresponding domain. 'Fake' represents the generated radar data from the generator of the proposed method in the corresponding domain. 'Rec' denotes the reconstructed radar data derived from the generated data in the corresponding domain.

Sunny to Rainy (80 mm/h)	Real Sunny	Fake Rainy	Rec Sunny
RCS Min	-49.86	-43.63	-46.17
RCS Max	51.96	35.93	54.98
Rainy (80 mm/h) to Sunny	Real Rainy	Fake Sunny	Rec Rainy
RCS Min	-44.73	-47.91	-38.93
RCS Max	33.46	47.20	34.31

for translating to rainy, due to more noise points in rainy domain.

As shown in Figure 1, the difference in RCS values between sunny and rainy domain is at most 15 to 20. In addition, since the RCS value includes many negative values, it is necessary to similarly reproduce the range of RCS values. To confirm the upper issues, we compared the maximum and minimum RCS values in each domain, as shown in Table 3. The results in Table 3 confirm that the RCS values for the rainy domain are consistently 15 to 20 units lower than those for the sunny domain. We confirmed that the ranges of maximum and minimum values of RCS between the ground truth and generated output are similar.

The RCS distribution of radar data, presented as a histogram, between the sunny and rainy domains is depicted in Figure 7. As depicted by the red and green circles in Figure 7, the RCS values decrease during translation from the sunny to rainy domain, indicating an increase in noisy data. Conversely, the RCS values increase during translation from the rainy to sunny domain, indicating a reduction in noisy data. These results provide evidence that the proposed method accurately represents the change in RCS values between the sunny and rainy domains.

It's worth emphasizing that there is no comparable existing research in the state-of-the-art literature, as the proposed method represents a pioneering foray into a new research field. Nevertheless, there have been studies on LiDAR translation [29], [30] that conducted quantitative assessments in a similar vein. These LiDAR translations [29], [30] primarily revolved around converting LiDAR data from clear weather conditions into data simulating rainy or foggy conditions. Similarly, our proposed method aims to translate radar data from clear weather conditions into data that simulates rainy weather. This translation process shares some commonalities with the LiDAR translations [29], [30], particularly in terms of weather-induced data translation. Therefore, similar to the quantitative evaluation conducted in the LiDAR translations [29], [30], we employed a comparable method to assess the effectiveness of our proposed approach. The above quantitative evaluation results validate that our proposed method can effectively achieve radar data translation.

V. DISCUSSION

The format of the generated radar data is different, but there are several state-of-the-art researches on generating radar data. Most of those researches [63], [64], [65] focus on generating radar data in the form of radar spectrum image representations. Converting radar raw data into radar spectrum images can lead to some data loss. However, the proposed method utilizes point information that is closer to raw radar data, which results in minimal data loss. This is one of its advantages. Additionally, there are other state-of-the-art researches [66], [67] in radar data generation that focus on creating radar signals. While radar signals also resemble raw radar data, there can be significant variations in the signals produced by different radar sensors. However, our proposed method offers the advantage of sensor independence, as it can generate radar data in a consistent format based on coordinate values and RCS, making it compatible with various sensors.

As mentioned in this study [68], a fundamental principle of deep learning is that training on diverse datasets leads to improved performance. In this sense, our proposed method, which can translate and generate a wide variety of radar data from relatively limited radar data compared to other sensors, can significantly enhance the performance of object detection or segmentation networks using radar data. In this study, we aimed to demonstrate performance improvements by referencing a specific radar recognition network through the proposed method. However, annotating the generated radar data requires a significant amount of human resources. Therefore, this is planned as a part of our future work. The generalizability and scalability of the proposed component to other networks are valid and well-established. Generalizability of its component has already been demonstrated in various fields, not only for object detection [69], [70] of 3D point cloud but also for point segmentation [12], through KP-Convolution [12]. As for scalability, it is flexible and can be adjusted according to the number of points in a scene, which can vary depending on the training dataset.

VI. CONCLUSION

We proposed the first radar translation network between the sunny and rainy domains by combining KP-Convolution [12] and CycleGAN [13]. Existing methods for radar translation rely on physical models, which require prior configuration information about radar sensors. In contrast, the proposed method utilizes a data-driven approach based on deep learning, eliminating the need for radar sensor configuration information. Furthermore, the proposed method is highly effective in terms of data augmentation. While traditional methods primarily focus on image and LiDAR data, the proposed method concentrates on radar data by directly applying convolution to a point cloud without the need for conversion into a graph or 3D voxel. Through diverse experiments, we formulated an optimal translation network architecture to effectively extract radar features.

In our experiments, we utilized the JARI dataset and successfully generated realistic radar data using the proposed method. Additionally, we anticipate that the proposed model will significantly contribute to driving simulators, enabling the examination of various autonomous driving algorithms before real-world implementation. By leveraging the proposed method, autonomous driving researchers can save considerable effort and time. For future work, we aim to enhance the proposed method to handle a broader range of environments. Since our testing was conducted on a limited dataset, we intend to gather diverse data from real roads and further optimize the proposed method for real-world conditions.

ACKNOWLEDGMENT

Words cannot express his gratitude to his professor for invaluable patience and feedback. The author also could not have undertaken this research without his professor, who generously provided knowledge and expertise. Additionally, this endeavor would not have been possible without the generous support of the Spring-GX scholarship. The author also grateful to his lab members for their help with editing, valuable feedback sessions, and unwavering moral support. Thanks are also due to my collaborators at MHI-MS (Mitsubishi Heavy Industries Machinery Systems, Ltd.) and MHI (Mitsubishi Heavy Industries, Ltd.) for their assistance in data collection and detailed discussions regarding this research. Lastly, The author would be remiss in not mentioning his family, especially his parents, brother, and spouse. Their belief in him has kept his spirits and motivation high during this process.

REFERENCES

[1] D. Wu et al., "YOLOP: You only look once for panoptic driving perception," *Mach. Intell. Res.*, vol. 19, pp. 1–13, Nov. 2022.

[2] J. Yu, and W. Zhang, "Face mask wearing detection algorithm based on improved YOLO-v4," *Sensors*, vol. 21, no. 9, p. 3263, 2021.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., 2020, pp. 213–229.

[4] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang, "A comparative study of real-time semantic segmentation for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 587–597.

[5] M. Treml et al., "Speeding up semantic segmentation for autonomous driving," in *Proc. NIPS*, 2016, pp. 1–7.

[6] H. Wang et al., "SFNet-N: An improved SFNet algorithm for semantic segmentation of low-light autonomous driving road scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21405–21417, Nov. 2022.

[7] Y. Ban et al., "Depth estimation method for monocular camera defocus images in microscopic scenes," *Electronics*, vol. 11, no. 13, p. 2012, 2022.

[8] X. Luo, J. B. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 71–81, 2020.

[9] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, Mar. 2022.

[10] X. Xia, N. P. Bhatt, A. Khajepour, and E. Hashemi, "Integrated inertial-LiDAR-based map matching localization for varying environments," *IEEE Trans. Intell. Veh.*, vol. 8, no. 10, pp. 4307–4318, Oct. 2023.

[11] K. Liu, X. Zhou, and B. M. Chen, "An enhanced LiDAR inertial localization and mapping system for unmanned ground vehicles," in *Proc. IEEE 17th Int. Conf. Control Autom. (ICCA)*, 2022, pp. 587–592.

[12] H. Thomas, C. R. Qi, J. E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.

[13] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[14] I. J. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[15] E. L. Denton, S. Chintala, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1486–1494.

[16] T. Qiao, J. Zhang, D. Xu, and D. Tao, "MirrorGAN: Learning text-to-image generation by redescription," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1505–1514.

[17] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020, pp. 592–608.

[18] A. Cherepkov, A. Voynov, and A. Babenko, "Navigating the GAN parameter space for semantic image editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3671–3680.

[19] W. Liao, K. Hu, M. Y. Yang, and B. Rosenhahn, "Text to image generation with semantic-spatial aware GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18187–18196.

[20] H. Liu, Z. Wan, W. Huang, Y. Song, X. Han, and J. Liao, "PD-GAN: Probabilistic diverse GAN for image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9371–9381.

[21] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[22] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2017, pp. 5400–5409.

[23] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC2: Improved cycleGAN-based non-parallel voice conversion," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 6820–6824.

[24] H. Yang et al., "Unpaired brain MR-to-CT synthesis using a structure-constrained CycleGAN," in *Proc. Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. 4th Int. Workshop (DLMIA)*, 2018, pp. 174–182.

[25] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, "Mocycle-GAN: Unpaired video-to-video translation," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 647–655.

[26] A. E. Sallab, I. Sobh, M. Zahran, and N. Essam, "LiDAR sensor modeling and data augmentation with GANs for autonomous driving," 2019, *arXiv:1905.07290*.

[27] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, "CNN-based LiDAR point cloud de-noising in adverse weather," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2514–2521, Apr. 2020.

[28] A. Kurup, and J. Bos, "DSOR: A scalable statistical filter for removing falling snow from LiDAR point clouds in severe winter weather," 2021, *arXiv:2109.07078*.

[29] J. Lee, D. Shiotsuka, T. Nishimori, K. Nakao, and S. Kamijo, "GAN-based LiDAR translation between sunny and adverse weather for autonomous driving and driving simulation," *Sensors*, vol. 22, no. 14, p. 5287, 2022.

[30] J. Lee, D. Shiotsuka, T. Nishimori, K. Nakao, and S. Kamijo, "LiDAR translation based on empirical approach between sunny and foggy for driving simulation," in *Proc. 25th Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, 2022, pp. 430–435.

[31] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, 26, pp. 364–368, Mar. 2016.

[32] M. Sheeny, A. Wallace, and S. Wang, "RADIO: Parameterized generative radar data augmentation for small datasets," *Appl. Sci.*, vol. 10, no. 11, 3861, 2020.

[33] N. Andriyanov and D. Andriyanov, "Pattern recognition on radar images using augmentation," in *Proc. Ural Symp. Biomed. Eng. Radioelectron. Inf. Technol. (USBEREIT)*, 2020, pp. 289–291.

[34] D. She, X. Lou, and W. Ye, "RadarSpecAugment: A simple data augmentation method for radar-based human activity recognition," *IEEE Sensors Lett.*, vol. 5, no. 4, pp. 1–4, Apr. 2021.

[35] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Adv. Radio Sci.*, vol. 9, no. 1, pp. 49–60, 2011.

[36] D. Jaroszweski and T. McNamara, "The influence of rainfall on road accidents in urban areas: A weather radar approach," *Travel Behav. Soc.*, vol. 1, no. 1, pp. 15–21, 2014.

[37] S. Muckenhuber, E. Museljic, and G. Stettinger, "Performance evaluation of a state-of-the-art automotive radar and corresponding modeling approaches based on a large labeled dataset," *J. Intell. Transp. Syst.*, vol. 26, no. 6, pp. 655–674, 2022.

[38] X. Xia, R. Xu, and J. Ma, "Secure cooperative localization for connected automated vehicles based on consensus," *IEEE Sensors J.*, vol. 23, no. 20, pp. 25061–25074, Oct. 2023.

[39] W. Liu et al., "A systematic survey of control techniques and applications in connected and automated vehicles," 2023, *arXiv:2303.05665*.

[40] C. Stadler, F. Montanari, W. Baron, C. Sippl, and A. Djanatliev, "A credibility assessment approach for scenario-based virtual testing of automated driving functions," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 45–60, 2022.

[41] S. Kitajima, H. Chouchane, J. Antona-Makoshi, N. Uchida, and J. Tajima, "A nationwide impact assessment of automated driving systems on traffic safety using multiagent traffic simulations," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 302–312, 2022.

[42] X. Guo, Q. Wang, and J. Zhao, "Data-driven vehicle rebalancing with predictive prescriptions in the ride-hailing system," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 251–266, 2022.

[43] M. C. Rademeyer, A. Barnard, and M. J. Booysen, "Optoelectronic and environmental factors affecting the accuracy of crowd-sourced vehicle-mounted license plate recognition," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 15–28, 2020.

[44] F. Bouali et al., "5G for vehicular use cases: Analysis of technical requirements, value propositions and outlook," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 73–96, 2021.

[45] S. Treneska, E. Zdravevski, I. M. Pires, P. Lameski, and S. Gievska, "GAN-Based image colorization for self-supervised visual feature learning," *Sensors*, vol. 22, no. 4, p. 1599, 2022.

[46] D. Shiotsuka et al., "GAN-based semantic-aware translation for day-to-night images," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, 2022, pp. 1–6.

[47] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[48] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, p. 386, 1958.

[49] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, Munich, Germany, Oct. 2015, pp. 234–241.

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

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

[53] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The Oxford radar robotcar dataset: A radar extension to the Oxford robotcar dataset," 2019, *arXiv:1909.01300*.

[54] M. Meyer and G. Kuschk, "Automotive radar dataset for deep learning based 3D object detection," in *Proc. 16th Eur. Radar Conf. (EuRAD)*, 2019, pp. 129–132.

[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[56] A. K. Dubey and V. Jain, "Comparative study of convolution neural network's Relu and leaky-Relu activation functions," in *Proc. Appl. Comput. Autom. Wireless Syst. Elect. Eng. (MARC)*, 2018, pp. 873–880.

[57] W. Q. Jie and W. W. Bin, "Research on image retrieval using deep convolutional neural network combining L1 regularization and PRelu activation function," *IOP Conf. Earth Environ. Sci.*, vol. 69, no. 1, 2017, Art. no. 12156.

[58] S. G. K. Patro, and K. K. Sahu, "Normalization: A preprocessing stage," 2015, *arXiv:1503.06462*.

[59] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, p. 31.

[60] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528.

[61] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[62] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Packing and padding: Coupled multi-index for accurate image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1939–1946.

[63] M. Rahnemoonfar, M. Yari, and J. Paden, "Radar sensor simulation with generative adversarial network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2020, pp. 7001–7004.

[64] V. Lekic and Z. Babic, "Automotive radar and camera fusion using generative adversarial networks," *Comput. Vis. Image Understand.*, vol. 184, pp. 1–8, Jul. 2019.

[65] J. Guo, B. Lei, C. Ding, and Y. Zhang, "Synthetic aperture radar image synthesis by using generative adversarial nets," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1111–1115, Jul. 2017.

[66] M. M. Rahman, S. Z. Gurbuz, and M. G. Amin, "Physics-aware generative adversarial networks for radar-based human activity recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 3, pp. 2994–3008, Jun. 2023.

[67] T. Truong and S. Yanushkevich, "Generative adversarial network for radar signal synthesis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–7.

[68] B. J. Son, "Analysis of the effect of training options on deep learning network for handwriting recognition," *J. Korea Soc. Adv. Comput. Struct.*, vol. 8, no. 2, pp. 73–79, 2017.

[69] I. de Gélis, S. Lefèvre, and T. Corpetti, "Siamese KPConv: 3-D multiple change detection from raw point clouds using deep learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 197, pp. 274–291, Mar. 2023.

[70] I. de Gélis et al., "Cliff Change detection using Siamese KPConv deep network on 3-D point clouds," *ISPRS Ann. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 3, no. 1, pp. 649–656, 2022.

JINHO LEE (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from The University of Kyushu, Fukuoka, Japan, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Institute of Industrial Science, The University of Tokyo. He was a Visiting Student Researcher with the University of California at Berkeley, Berkeley, in 2016. He began working with A.I. Matics. in 2018 as an Associate Research Engineer. His research interests include computer vision, localization, deep learning, and their applications to ITS and an autonomous vehicle. He was a recipient of the 69th IEEE Kyushu Section Excellent Presentation Award in 2016 and the 69th Meeting on Image Recognition and Understanding Interactive Presentation Award in 2017. He is a VTS Member.

GEONKYU BANG received the B.E. degree from the College of Engineering Systems, University of Tsukuba, Ibaraki, Japan, in 2018, and the M.S. degree in complexity science and engineering from The University of Tokyo, Chiba, Japan, in 2021. He is currently pursuing the Ph.D. degree in interdisciplinary information studies with The University of Tokyo, Tokyo, Japan. His research interests are computer vision, generative models, deep learning, and their applications to ITS.

TOSHIAKI NISHIMORI graduated from the National Institute of Technology, Kochi College, Kochi, Japan, in 1998. He has been working with Mitsubishi Heavy Industries Machinery Systems, Ltd. since 1998, where he was in charge of software for a facility that artificially reproduces weather conditions, such as rainfall, snowfall, and solar radiation and a wind tunnel facility for aerodynamic testing of automobiles, aircraft, and rockets.

KENTA NAKAO received the M.E. degree from The Himeji Institute of Technology, Hyogo, Japan, in 2005, and has been working with Mitsubishi Heavy Industries, Ltd. since 2005, where he engaged in research on computer vision for ITS and logistics equipment.

SHUNSUKE KAMIJO (Senior Member, IEEE) is an Associate Professor with The University of Tokyo. He has been engaged in video surveillance, sensor fusion, and their application to intelligent transport systems. His current research interests are autonomous vehicles, digital map, image processing and deep learning, and portable and wearable applications. He has been serving for the Board of Governors of the IEEE ITS Society since 2015 and a Vice President for Publication since 2022. He has many contributions to international conferences as a Vice-Chairman of the program committee for the ITS World Congress Tokyo 2013, a General Co-Chair of IEEE ICVES2015 and ITSC2017, and the International Program Chair of IEEE IV2017. He also has significant contributions to international journals as the Editor-in-Chief of *IATSS Research* (Elsevier) and an Editorial Board Member of *Multimedia Tools and Applications* (Springer). He is a Senior Member of the IEEE ITS Society and a member of IEICE and IATSS.