RESEARCH ARTICLE

A Spatiotemporal Deep Learning Architecture for Road Surface Classification Using LiDAR in Autonomous Emergency Braking Systems

JU WON SEO[1], (Graduate Student Member, IEEE),
JIN SUNG KIM[1], (Graduate Student Member, IEEE),
JIN HO YANG[1], (Graduate Student Member, IEEE),
AND CHUNG CHOO CHUNG[2], (Member, IEEE)
[1]Department of Electrical Engineering, Hanyang University, Seoul 04763, South Korea
[2]Division of Electrical and Biomedical Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Chung Choo Chung (cchung@hanyang.ac.kr)

ABSTRACT This paper proposes a spatiotemporal architecture with a deep neural network (DNN) for road surface conditions and types classification using LiDAR for autonomous emergency braking (AEB) systems. It is known that LiDAR provides information on the reflectivity and number of point clouds depending on a road surface. Thus, this paper utilizes the characteristics of LiDAR and the vehicle's speed obtained from in-vehicle sensors for classifying front road surface conditions for the AEB system application. First, we divided the front road area into four subregions to use spatial information from the vehicle's speed. Second, we constructed feature vectors using each subregion's reflectivity, number of point clouds, and in-vehicle information. Third, the DNN classifies road surface conditions and types for each subregion. Finally, the output of the DNN feeds into the spatiotemporal process to make the final classification reflecting vehicle speed and probability given by the outcomes of softmax functions of the DNN output layer. To validate the effectiveness of the proposed method, we performed a comparative study with five other algorithms. With the proposed DNN, we obtained the highest accuracy of 98.0% and 98.6% for two subregions near the vehicle. In addition, we implemented the proposed method on the NVIDIA Jetson TX2 board to confirm that it is applicable in real-time.

INDEX TERMS Autonomous vehicles, road surface classification, LiDAR, spatiotemporal architecture.

I. INTRODUCTION

In automated driving, accurate detection of road surface conditions and types are essential because the friction between the road surface and the tire has a relationship for driver safety [1]. Hydroplaning by heavy rain or low friction between tires and slippery roads influenced by heavy snow can cause fatal car accidents. In addition, knowing the types of roads provides essential information regarding

The associate editor coordinating the review of this manuscript and approving it for publication was Junho Hong.

safety, fuel efficiency, and driver convenience [2]. Since all these factors significantly influence ego vehicle stability, e.g., braking distance under/oversteer, the recognition of road surface conditions plays a vital role in improving the collision avoidance performance of the ego vehicle. Hence, the ability to classify road surfaces is critical for properly functioning autonomous emergency braking (AEB) systems. The AEB system consists of driving environment assessment, braking decision & deceleration profile generation, and actuation of speed reduction. For exact decision-making and reference generation for the AEB system, evaluation of surrounding

road property should be preceded. Using the AEB system, we can avoid a forward collision with an unexpected cut-in of a neighboring vehicle [3]. Thus, many techniques for classifying the conditions and types of road surfaces have been studied by using various sensors.

Generally, in-vehicle sensors have been widely used to detect road surface conditions and types. One technique is to classify the road surface conditions using in-vehicle sensors with a fuzzy logic block and an artificial neural network [4]. In addition, a recurrent neural network architecture using audio of tire-surface interaction was used for automated road surface wetness detection [5]. However, it cannot predict the forward road surface conditions required for the AEB system during driving by using in-vehicle sensors to determine the road surface's condition directly contacting the vehicle's tires. Therefore, due to the necessity of predicting and assessing road surface conditions along the vehicle's intended path, research is being conducted on utilizing cameras to evaluate the road surface ahead for forward driving [6], [7], [8]. Although the utilization of cameras to classify the front road surfaces has become possible, challenges like illumination continue to remain a concern [9]. Therefore, research has been conducted using another sensor to cope with these problems.

Various studies have been conducted to classify road surfaces using the characteristics of LiDAR that can be obtained according to road surfaces. LiDAR can be effective in determining road surface conditions and types. Using LiDAR is robust against various factors, such as the shadow of objects and illumination changes [9]. It is known that LiDAR provides information on the reflectivity and range depending on the roughness of objects. Using these characteristics of LiDAR, many studies have applied them to classifying road surfaces. One study classified road conditions using regular reflection on the wet road due to water film and irregular reflection on dry roads [10]. Likewise, there is a study to classify the types of roads by utilizing the differences in the range and remission of LiDAR data since the roughness varies depending on the type of road [11].

In general, support vector machine (SVM) and K-nearest neighbor (KNN) techniques have been utilized to classify the condition and type of road surface using LiDAR [11], [12], [13]. After extracting the reflectivity and range of LiDAR point cloud as a feature, methods of classifying the types of roads through SVM have been studied [11], [12]. Furthermore, KNN is used to classify the road condition using reflectivity information from LiDAR [13]. Another approach is to transform LiDAR data into an image. In [14], the authors proposed a method of determining the road condition and weather with a convolutional encoder by transforming the LiDAR point cloud into a range image. These studies determined the condition and type of roads using only the characteristics of LiDAR, such as reflectivity and range. However, it is necessary to utilize the spatiotemporal information of sensor data to classify the road surface in an
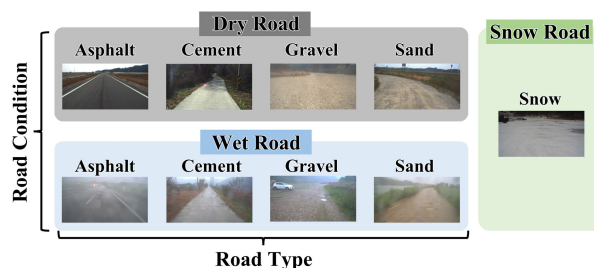
FIGURE 1. Road conditions (dry, wet, and snow) and types (asphalt, cement, gravel, and sand) with obtained sample images.

autonomous vehicle since the feature of LiDAR varies as the vehicle moves [15]. Moreover, for normal operation of the AEB system, it is essential to know the ego vehicle's speed. Hence, the use of in-vehicle sensors in conjunction becomes imperative.

This paper proposes a deep learning spatiotemporal architecture with the LiDAR and in-vehicle sensor data for classifying road conditions and types to use on the AEB system. To get spatial information, we divide the front road area into four subregions: left near region, right near region, left far region, and right far region. We classify the road conditions and types for each region using a DNN. The features are reflectivity, number of point clouds, and ego vehicle speed. The selected features are stacked using the time-windowing method to utilize the temporal information, and this feature vector constructs the input vector of the deep neural network (DNN). We are focusing on the regions near the vehicle, as they are notably more important than the road further away, considering the braking distance for passenger safety against forward collisions. Therefore, we construct a spatiotemporal formulation with past DNN results in regions far from the vehicle, calculating the final results of regions close to the vehicle. We compared the results with five other methods (SVM, KNN, etc.) to show the effectiveness of the proposed algorithm. The experimental results show that the proposed method outperforms the other methods by obtaining the accuracies, 98.0% and 98.6%, for each near left and right subregion, respectively. Furthermore, we implemented the proposed method on the Jetson TX2 board to confirm that it is applicable in real-time for the AEB system. It was similar to the results implemented on a desktop; the average computation time was about 10 ms.

II. APPROACH FOR ROAD SURFACE CONDITION AND TYPE CLASSIFICATION

A. SCENARIO DESCRIPTION

This study considers three road conditions (dry, wet, and snow) and four road types (asphalt, cement, gravel, and sand). Figure 1 shows the obtained sample images of each road condition and type. In an urban area, it is common to discover that different types of roads exist, and rain or snow may affect the condition of the road surface. Therefore,

(a) Experiment vehicle

(b) Sensor logging system
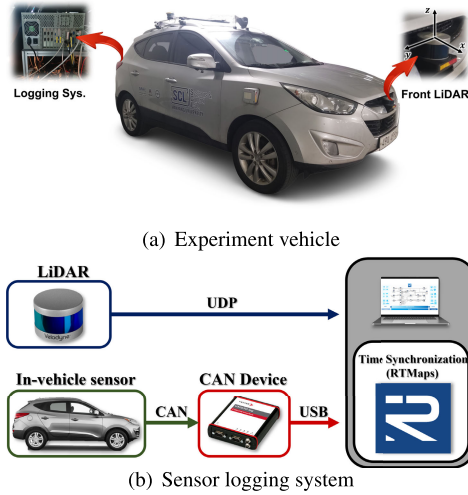
FIGURE 2. Configuration of the logging system in our test vehicle. We use the test vehicle with LiDAR and in-vehicle sensors for obtaining the data to make the training data set.

TABLE 1. Data set for training and validation obtained by the logging system.

Road		Training Dataset	Validation Dataset
Dry	**Asphalt**	27,000 (2,700 sec)	7,000 (700 sec)
	Cement	27,000 (2,700 sec)	7,000 (700 sec)
	Gravel	27,000 (2,700 sec)	7,000 (700 sec)
	Sand	27,000 (2,700 sec)	7,000 (700 sec)
Wet	**Asphalt**	27,000 (2,700 sec)	7,000 (700 sec)
	Cement	27,000 (2,700 sec)	7,000 (700 sec)
	Gravel	27,000 (2,700 sec)	7,000 (700 sec)
	Sand	27,000 (2,700 sec)	7,000 (700 sec)
Snow		27,000 (2,700 sec)	7,000 (700 sec)
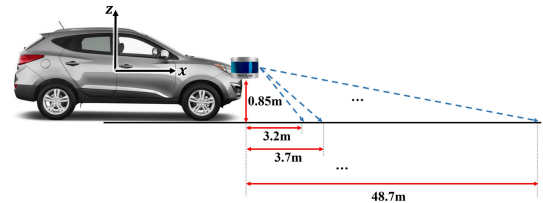
FIGURE 3. Detecting points at each LiDAR channel, which measures the road surface. The nearest distance is about 3.2m and the farthest distance is about 48.7m.

FIGURE 4. Divided regions of the front road according to the AEB test protocol and road regulations. Since the regions close to the vehicle is essential regarding the safety of the vehicle, *LN Region* and *RN Region* are mainly considered in this paper. We determined the size of the subregion based on the Euro NCAP's CCRb scenarios.

classifying the condition and type of road surface is a big issue in autonomous vehicles in urban areas. Thus, this paper considers the driving environment in an urban area with various conditions and types of road surfaces. We assumed the Car-to-Car Rear braking (CCRb) scenario of autonomous emergency braking (AEB) protocol from the Euro NCAP [16].

B. DATA ACQUISITION

In this study, we use a real car to obtain the dataset. Figure 2 shows the experimental vehicle and sensor system used for the data acquisition. The vehicle is a small SUV, Tucson ix from Hyundai Motors, and sensor data was obtained from the LiDAR and in-vehicle sensors. The LiDAR was mounted 0.85 m above the ground in the vehicle's front bumper so that the LiDAR could obtain data on the front road surface. The VLP16 from Velodyne, which has a rotation rate of 10 Hz and 16 vertical channels, is used. As the number of channels increases, accuracy improves; however, we employed a cost-effective LiDAR model for commercialization purposes. As shown in Fig. 2 (b), the VN1630 from Vector is adopted to obtain the data from the in-vehicle sensors through the controller area network (CAN) bus. The vehicle state data was obtained from the in-vehicle sensors, e.g., the vehicle's speed. A logging system is needed to obtain synchronized data from various communication methods. Here, the RTMaps from Intempora was used for acquiring LiDAR and in-vehicle data under time synchronization.
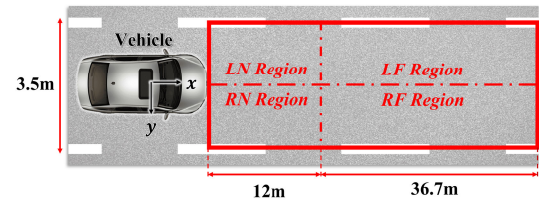
The dataset was obtained for three road conditions (dry, wet, and snow) and four road types (asphalt, cement, gravel, and sand) where the vehicle drives in various places. The point cloud obtained when LiDAR rotates once with a rotation rate of 10 Hz is regarded as one frame. Point cloud data of 34,000 frames from LiDAR and vehicle speed

data from the in-vehicle sensor were obtained for each road class, so a total of about 9 hours of point cloud data and vehicle speed data were obtained. After generating a feature vector from the obtained data (will be explained in Section II-D), 243,000 frames were used as a training dataset. The remaining 63,000 frames were used as a validation dataset, as shown in Table 1. The validation dataset consists of data acquired from different places on different days from the training data set.

C. REGION TESSELLATION

This paper proposes an algorithm that classifies road conditions and types with spatiotemporal information for

the AEB system. To obtain this information, we first need to set the region of interest (ROI) of the LiDAR sensor. We used the data in which the LiDAR detected the front road surface. Hence, the data was selected by the z-axis criteria below 0.1 m. In addition, the lateral distance was set from −1.75 m to 1.75 m according to the available road width of 3.5 m. The longitudinal distance was designed as 48.7 m, the longest detectable distance for LiDAR, as depicted in Fig 3. For spatial information, we divided the ROI of the front road into four subregions, as shown in Fig. 4. Here, *L* means *Left*, *R* means *Right*, *N* means *Near the vehicle*, and *F* means *Far from the vehicle*. For example, the *LF Region* means the *Region of the left and far from the vehicle*. In the lateral direction, the ROI was separated as the {*LN Region*, *LF Region*} group and the {*RN Region*, *RF Region*} group. This is because the road conditions can significantly differ between the left and right wheel paths, the so-called split friction. The split friction phenomenon can affect the safety of the vehicle's motion. Thus, we separated the ROI into the left and right regions to classify each wheel path's road conditions. Moreover, we divided the ROI into the near and far regions as the {*LN Region*, *RN Region*} group and the {*LF Region*, *RF Region*} group. According to the Car-to-Car Rear braking (CCRb) scenario of AEB test protocol from Euro NCAP [16], the near region in the longitudinal direction was set as 12 m to guarantee collision avoidance with the front car. In this paper, *N Regions* are mainly considered because the conditions of the road close to the vehicle are notably more important than the road further away. In addition, since we are proposing an algorithm assuming the AEB system, we only consider scenarios where the road curvature is not excessive.

D. FEATURE VECTOR SELECTION

Data processing is required to obtain the feature data. The first step is to select the feature vectors from the LiDAR sensor. One of the feature vectors is the mean value of the reflectivity information obtained during one rotation of the LiDAR in each region. Another feature vector is the total number of point clouds obtained during one rotation of the LiDAR in each region. The reason for exploiting this information is that the reflectivity and the number of point clouds returned to the receiver of LiDAR are different according to the roughness of the road surface [14]. Regular reflection can occur when the road surface is polished smooth on a wet or snow road, resulting in few point clouds. However, mirror reflection occurs on the surface of the wet road, and point clouds return immediately so that some samples show high reflectivity. On the other hand, the number of point clouds and reflectivity may have been high on the dry road due to an irregular reflection [10]. To utilize this property of LiDAR data, the reflectivity and the number of point clouds are considered feature vectors. The feature data of the LiDAR are illustrated in Fig. 5. It shows that the distribution of features varies depending on the condition of the road. Similar to the method of analyzing each road condition, the LiDAR data feature by type of road was analyzed. Figure 6 (a)-(d) show the

feature of each type of road with the dry condition, and Figure 6 (d)-(e) show the feature of each type of road with the wet condition. Since the roughness of the road surface is different depending on the type of road, it can be seen that reflectivity and the number of points are different. In order to analyze the area close to the vehicle mainly covered in this paper, the characteristic of the *LN region* is represented.

The second step is to select the longitudinal speed obtained from the in-vehicle sensor as a feature vector. Most studies considered the characteristics of LiDAR, which varies depending on the road surface, but it is necessary to consider the feature that changes as the vehicle drives. Indeed, the data of LiDAR mounted on the vehicle is affected by vehicle speed since various motions occur as the vehicle moves [15]. Therefore, the vehicle speed is selected as another feature vector to utilize the information of LiDAR in which the vehicle speed was considered together. The ablation study for selecting and not selecting the longitudinal speed as a feature vector will be discussed in Section IV. Since the sample rate of the in-vehicle sensor is faster than that of LiDAR, unnecessary repetitive data for selecting the feature data must be prevented so that under-sampling of the in-vehicle sensor data is conducted with a sample rate of LiDAR (100 ms).

Finally, the feature vectors are stacked from the current time to the past one second using the time-windowing method, which causes the feature vector to contain the temporal information [17]. Then, an input vector of DNN is generated by this feature vector and entered into the neural network as shown in Fig. 7. To create the target data, we utilized the one-hot-encoding method to classify the nine classes, for example, $[1\ 0\ 0\ \dots\ 0]^T \in \mathbb{R}^{n_c \times 1}$ denotes dry asphalt road, and $[0\ 1\ 0\ \dots\ 0]^T \in \mathbb{R}^{n_c \times 1}$ denotes dry cement road, where n_c is the number of classes.

## III. DEEP NEURAL NETWORK WITH SPATIOTEMPORAL INFORMATION
A. DEEP NEURAL NETWORK AND OPTIMIZATION

We utilize the deep neural network to classify the road surface. The overall configuration of the network is shown in Fig. 7. The network consists of six hidden layers and one output layer. The transfer function for each layer is set to a non-linear function, e.g., the tangent-sigmoid function, since it has a very smooth curvature and does not cause gradient exploration affected by a sharp change in slope. The number of neurons in each layer is set to 100, 80, 40, 40, 20, and 10, respectively. The activation function of the output layer is utilized using the softmax method, in which the output is obtained as the probability corresponding to each class. Here, we design the neural network for each region, meaning there are four neural networks for classifying the road conditions at each region.

The scaled conjugate gradient (SCG) backpropagation algorithm was selected to optimize the network. SCG is a Conjugate Gradient Algorithm (CGA) variation [18]. Since most CGAs conduct a line search for every iteration, they
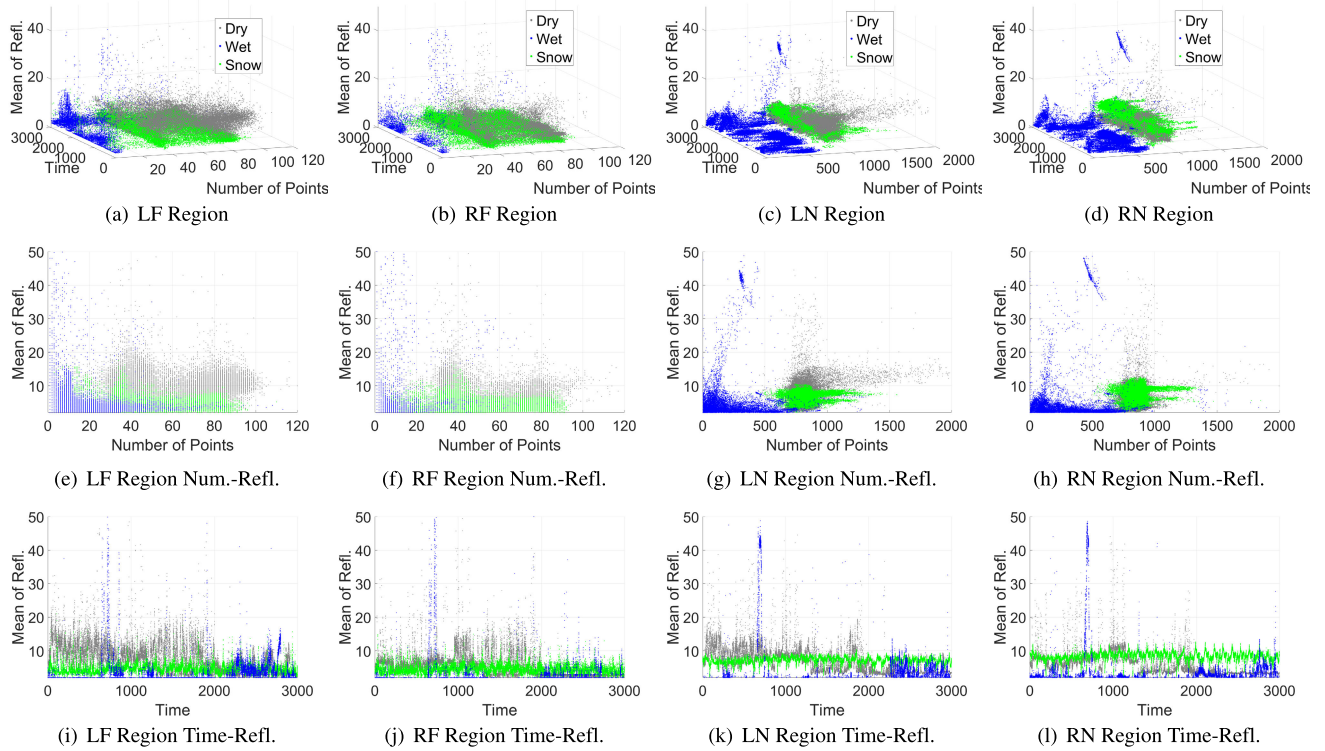
FIGURE 5. Feature of the LiDAR data with each road condition. Gray indicated the dry condition, blue is the wet condition, and green is the snow condition. (a) –(d) are 3-dimension feature data (time vs number of points vs mean of reflecitivty), (e)-(h) are 2-dimension feature data (number of points vs mean of reflectivity), and (i)-(j) are 2-dimension feature data (time vs mean of reflectivity).
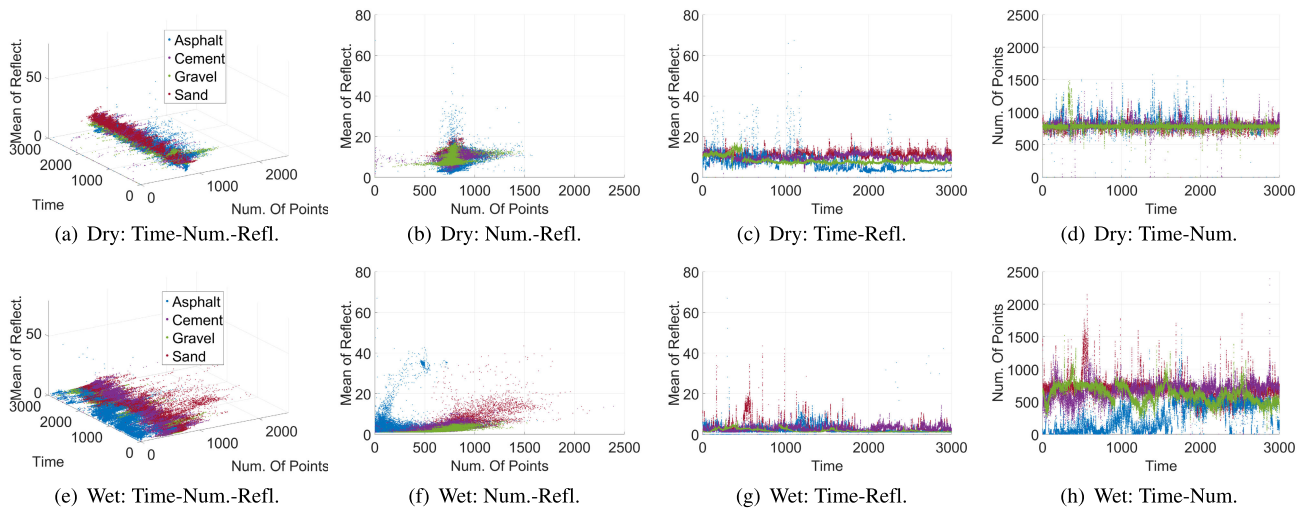

FIGURE 6. Feature of the LiDAR data with each road type. (a) –(d) are the feature of the road type on dry road, and (e)-(h) are the feature of the road type on wet road.

require complex computation and a long calculation time. SCG with a step-size scaling mechanism was developed to overcome this problem. The basic idea of the SCG is to combine the model-trust region approach used in the Levenberg-Marquardt algorithm with the CGA. Thus, the SCG was used to optimize classification problems [19]. In DNN, the optimization is conducted by minimizing the error between the target data and the neural network's output. Let the error function in the training set can be written as

$$E_{[i]} = f_{[i]}(x_{[i]}; \theta_{[i]}) - T_{[i]},$$
$$i \in \{LF, RF, LN, RN\} \quad (1)$$

where i is the index of the networks for each region, $f_{[i]}(x_{[i]}; \theta_{[i]})$ is the output of the neural network, $T_{[i]}$, $x_{[i]}$, and
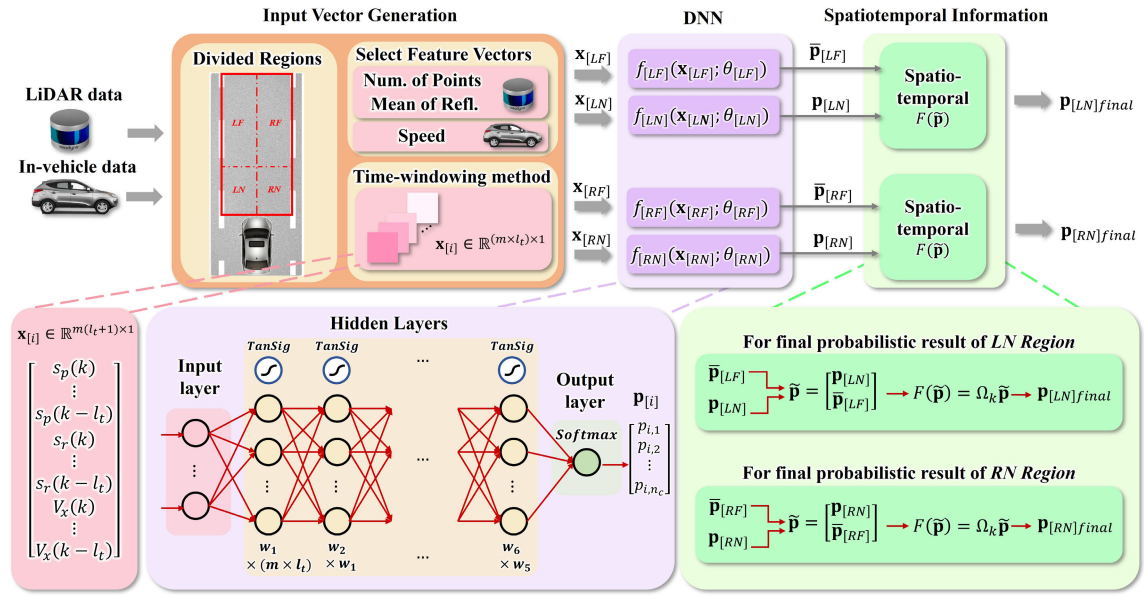
FIGURE 7. Overall structure of the proposed algorithm. The final result $p_{[LN]final}$ and $p_{[RN]final}$ is updated through the function $F(\tilde{p})$ by using the concatenated vector $\tilde{p}$ that stack the current result and past results over $k - l_S$ to k step. ($m = 3$, $l_t = 9$, $l_S = 5$, $n_c = 9$, $i \in \{LN, RN, LF, RF\}$).
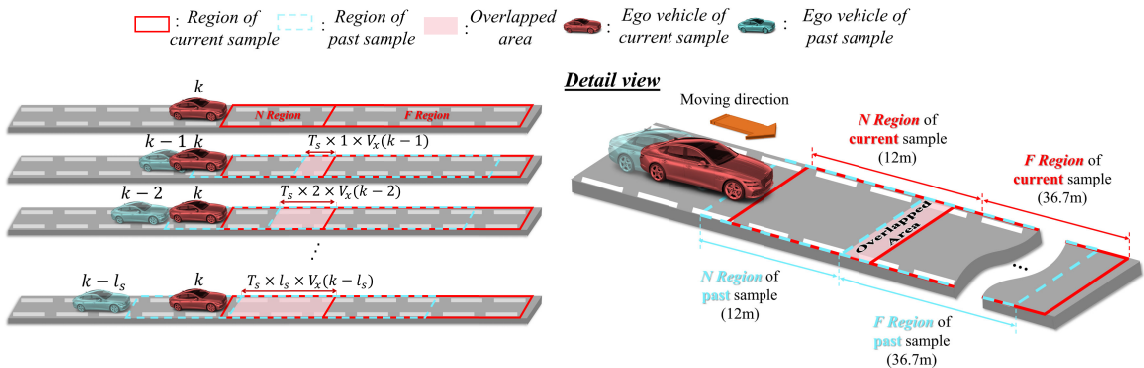
FIGURE 8. As the vehicle moves forward at the speed V_x, the yellow hatched region of the current sample time, and the blue hatched region of the past $k - 1$ sample time overlap by a distance of $T_s \times 1 \times V_x(k - 1)$ (Blue region). In the same way, the ratio of overlapping in the $k - 2$, $k - 3$, ..., $k - l_S$ sample time can be determined. The road condition of the yellow hatched region can be classified by using the spatiotemporal information with the past results of the blue hatched region. In this paper, we set the parameter as $l_S = 5$, $T_s = 0.1s$.

$\theta_{[i]}$ are the target data, the selected feature vector inputs, and weight parameters for the $i \in \{LF, RF, LN, RN\}$, respectively. The loss function used for training the network is the mean square error (MSE) function, as follows:

$$L_{[i]}^n = \frac{1}{N_b} \sum_{l=(n-1)N_b+1}^{n \cdot N_b} E_{[i]}^{l\ T} \cdot E_{[i]}^l + \frac{\lambda}{2} ||\theta_{[i]}||_2^2 \quad (2)$$

where n and N_b are the index and size of the mini-batch, respectively. Here, we use the regularization strategy to prevent overfitting that adds the L^2 parameter norm penalty, with hyper-parameter λ, $\frac{\lambda}{2}||\theta_{[i]}||_2^2$ to MSE function [20]. Now, the SCG algorithm is utilized, and the optimal parameter of

the network is updated as follows:

$$\theta_{[i]}^n = \theta_{[i]}^{n-1} + \gamma_{[i]}^n \nabla_{[i]}^n L_{[i]}^n \quad (3)$$

where $\gamma_{[i]}^n$ is a step size, and $\nabla_{[i]}^n$ is a conjugate direction, respectively. The step size is calculated by the algorithm from [18]. The final optimal parameter can be obtained through the optimization process as

$$\theta_{[i]}^* = \arg\min_{\theta_{[i]}^n}(L_{[i]}^n(f_{[i]}(x_{[i]}; \theta_{[i]}), T_{[i]})). \quad (4)$$

B. SPATIOTEMPORAL FORMULATION

To generate the input vector, we define the concatenated vectors from $k - l_t$ to k steps $\bar{s}_p(k|k - l_t)$, $\bar{s}_r(k|k - l_t)$, and

$\bar{V}_x(k|k - l_t)$ such as

$$\bar{s}_p(k|k - l_t) = [s_p(k), s_p(k - 1), \ldots, s_p(k - l_t)]^T$$
$$\bar{s}_r(k|k - l_t) = [s_r(k), s_r(k - 1), \ldots, s_r(k - l_t)]^T$$
$$\bar{V}_x(k|k - l_t) = [V_x(k), V_x(k - 1), \ldots, V_x(k - l_t)]^T \quad (5)$$

where l_t is the stacking step for the time windowing method, s_p is the number of points, s_r is the mean of the reflectivity, and V_x is the longitudinal speed of ego-vehicle, respectively. Let $x_{[i]} \in \mathbb{R}^{m(l_t+1)\times 1}$ be the input vector for $i \in \{LF, RF, LN, RN\}$ where m is the number of feature. Then, we can present the input vector of the network as

$$x_{[i]} = \begin{bmatrix} \bar{s}_p(k|k - l_t) \\ \bar{s}_r(k|k - l_t) \\ \bar{V}_x(k|k - l_t) \end{bmatrix} \in \mathbb{R}^{m(l_t+1)\times 1}. \quad (6)$$

We want to estimate the probability that $P(y = N|x_{[i]})$ for each value of $N = 1, \ldots, n_c$ to utilize the spatiotemporal formulation where n_c is the number of classes and y is the label. In other words, we want to obtain the class label's probability distribution output, taking on each N different possible value. To this end, we added the *softmax* function, which normalizes the probability distribution to the exponentials, to the output layer of the network model [20]. Using this exponential property, the probability can be presented as

$$P(y = N|x_{[i]}) = \frac{\exp(x_{[i]}^T \theta_{[i]}^{(N)})}{\sum_{l=1}^{n_c} \exp(x_{[i]}^T \theta_{[i]}^{(l)})} \quad (7)$$

where $\theta_{[i]}^{(\cdot)}$ are the weighting vectors of the optimized network. Therefore, the network output has a n_c-dimensional vector, whose elements sum to 1, giving us n_c predicted probabilities. Let $p_{[i]} \in \mathbb{R}^{n_c \times 1}$ be the probability distribution output from the optimized neural network for $i \in \{LF, RF, LN, RN\}$ through the *softmax* layer, then it can be presented as

$$p_{[i]} = \begin{bmatrix} P(y = 1|x_{[i]}; \theta_{[i]}) \\ P(y = 2|x_{[i]}; \theta_{[i]}) \\ \vdots \\ P(y = n_c|x_{[i]}; \theta_{[i]}) \end{bmatrix}$$
$$= \frac{1}{\sum_{l=1}^{n_c} \exp(x_{[i]}^T \theta_{[i]}^{(l)})} \begin{bmatrix} \exp(x_{[i]}^T \theta_{[i]}^{(1)}) \\ \exp(x_{[i]}^T \theta_{[i]}^{(2)}) \\ \vdots \\ \exp(x_{[i]}^T \theta_{[i]}^{(n_c)}) \end{bmatrix}. \quad (8)$$

Here, we introduce concatenated vector $\tilde{p}(k) \in \mathbb{R}^{n_c(l_s+1)\times 1}$ with stacking step l_s by stacking the current probabilistic result of $\{LN\,Region, RN\,Region\}$ and past probabilistic result of $\{LF\,Region, RF\,Region\}$ over $k - l_s$ to k step as

$$\tilde{p}(k) = \begin{bmatrix} p_{[i]}(k) \\ \bar{p}_{[j]}(k - 1|k - l_s) \end{bmatrix} \quad (9)$$

where

$$\bar{p}_{[j]}(k - 1|k - l_s) = \begin{bmatrix} p_{[j]}(k - 1) \\ p_{[j]}(k - 2) \\ \vdots \\ p_{[j]}(k - l_s) \end{bmatrix} \in \mathbb{R}^{n_c l_s \times 1},$$
$$i \in \{LN, RN\}, j \in \{LF, RF\}.$$

Here, the spatiotemporal function with concatenated vector $F(\tilde{p}(k))$ is defined as follows:

$$F(\tilde{p}(k)) := (\Omega_k \otimes I_{n_c})\tilde{p}(k) \in \mathbb{R}^{n_c \times 1} \quad (10)$$

where $\otimes$ means the Kronecker product, $I_{n_c} \in \mathbb{R}^{n_c \times n_c}$ denotes $n_c \times n_c$ identity matrix and

$$\Omega_k = \begin{bmatrix} \alpha_k & \alpha_{k-1} & \ldots & \alpha_{k-l_s} \end{bmatrix} \in \mathbb{R}^{1 \times (l_s+1)},$$

with the time-varying weight constants α_l for $l \in \{k, k - 1, \ldots, k - l_s\}$ which are $\sum_{l=k}^{k-l_s} \alpha_l = 1$ corresponding to each probabilistic result. Network output results of past *LF region* and *RF region* are used to obtain road conditions and types of *LN region* and *RN region* in the current step.

In order to utilize the spatiotemporal information, it is necessary to set weight constants, α_k on the DNN output probabilities. To this end, we utilize the length information of the overlapped area, as shown in Fig. 8. Let the current time's vehicle speed be $V_x(k)$ m/s, and T_s is the sample rate of the LiDAR sensor (100 ms). When the vehicle moves forward at the longitudinal speed, V_x, the *LF Region* in step $k - 1$ overlaps with the *LN Region* in step k as the $T_s \times 1 \times V_x(k - 1)$ distance. In general formulation, the *LF Region* in step $k - l_s$ overlaps with the *LN Region* in step k as the $T_s \times l_s \times V_x(k - l_s)$ distance. Likewise, the *RF Region* and *RN Region* overlap in the same method. By utilizing this spatial information, it is possible to determine the road condition and type in the *LN Region* fused with the past probabilistic results of the *LF Region* and in the *RN Region* with the past probabilistic results of the *RF Region*. In (11), the weight matrix $\Omega_k(\cdot)$ dependent on V_x can be determined as follows:

$$\Omega_k(\bar{V}_x(k - 1|k - l_s))$$
$$= \frac{1}{\beta_1 + \sum_{l=2}^{l_s}(l \cdot T_s \cdot V_x(k - l))} \begin{bmatrix} \beta_1 \\ T_s V_x(k - 1) \\ 2T_s V_x(k - 2) \\ \vdots \\ l_s T_s V_x(k - l_s) \end{bmatrix}^T \quad (11)$$

Here, β_1 is the length of the *LN Region*(or *RN Region*), e.g., 12, and the term $1/(\beta_1 + \sum_{l=2}^{l_s}(l \cdot T_s \cdot V_x(k - l)))$ needs for normalization. Using this weight matrix, we can propose a spatial fusion method as follows:

$$p_{[i]final}(k) = F(\tilde{p}(k))i \in \{LN, RN\}. \quad (12)$$

The overall process of the proposed method can be presented as Algorithm 1.

Algorithm 1 Fusion Algorithm

Require: Network model $f_{[i]}$, Network input $x_{[i]}$, Longitudinal speed V_x, Sample time T_s, Number of validation dataset N_{val}, Number of stacking step l_s

Require: $i \in \{LN, RN, LF, RF\}$

1: **for** $k = 1, \ldots, N_{val}$ **do**
2:     Obtain network output $p_{[i]}(k)$ using Eq. (8)
3:     **if** $k = 1, \ldots, l_s$ **then**
4:         $p_{[i]final}(k) = p_{[i]}(k)$
5:     **else**
6:         **for** $j = k, \ldots, k - l_s$ **do**
7:             Obtain $\bar{p}(k)$ using Eq. (9)
8:         **end for**
9:         Obtain $V_x(k - l_s), \ldots, V_x(k - 1)$
10:       Calculate $\Omega_k(\bar{V}_x(k - 1 | k - l_s))$ using Eq. (11)
11:       Update final result $p_{[i]final}(k)$ using Eq. (12)
12:   **end if**
13: **end for**

FIGURE 9. Simple architecture of DEL, LSTM, TWM, KNN, and SVM.

IV. EXPERIMENT RESULTS

A. RESULT OF PROPOSED METHOD

To validate the proposed algorithm, we utilized the validation dataset of Table 1, which was not used for training. Datasets were collected from various places around the urban, rural, and mountain areas near Seoul city in Korea. Paved roads, such as asphalt and cement, were collected from the highway and local roads, including the proving ground of Korea Intelligent Automotive Parts Promotion Institute (KIAPI), and unpaved roads, such as gravel and sand, were collected in lane-free situations. Furthermore, additional data were acquired when it rained or snowed in each place to acquire data with different conditions on the road surfaces. In Table 2 and Table 3, the confusion matrices show the overall accuracy in bold at 98.0% and 98.6% for each region near the vehicle. In addition, an essential value in the road classification system is the proportion of risk situations. For example, suppose a snow road is misclassified as dry or wet. In these cases, fatal car accidents can occur because of delivering incorrect friction data between the tire and the road to the control system. Hence, the risk situation is one of the most crucial issues and shows low values of about 0.01% for each region.

B. ABLATION STUDY

We conducted an ablation study to explain the reason for considering the vehicle speed in the feature vector. Since this paper aims to classify road surfaces while the vehicle moves at speeds within a certain range, it is necessary to use the selected feature from LiDAR considering speed to learn the classifier. Therefore, we evaluated the performance of the classifier for both cases. The performance of the classifier of the proposed method with the network input $x_{[i]} = [\bar{s}_p(k|k - l_t)\,\bar{s}_r(k|k - l_t)\,\bar{V}_x(k|k - l_t)]^T$ has an accuracy of 98.0% and 98.6% for LN region and RN region, respectively. However,
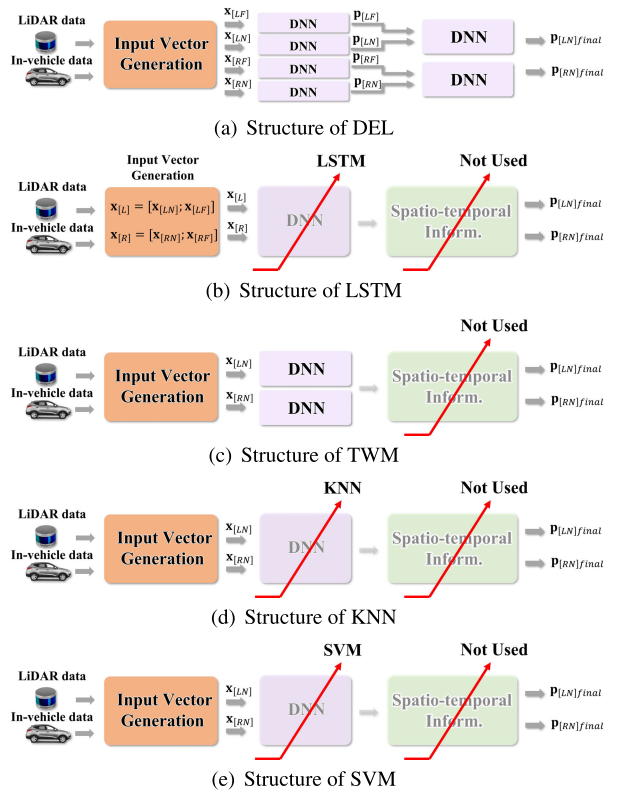
the performance of the classifier in a method that the vehicle speed is not selected as a feature vector, and the network input is $x_{[i]} = [\bar{s}_p(k|k - l_t)\,\bar{s}_r(k|k - l_t)]^T$ drops to 82.2% and 81.4% accuracy for LN region and RN region, respectively as shown in Table 4 and 5. In other words, to utilize spatiotemporal information, it is necessary to consider the vehicle's speed since the LiDAR feature is affected by vehicle motion during the vehicle moves. More accurate road surface classifiers can be conducted using spatiotemporal information through classifiers that take into account the distribution of features of LiDAR according to vehicle speed.

C. COMPARATIVE STUDY

While numerous papers utilize LiDAR to classify road conditions [14], [21], our proposed approach involves a simultaneous assessment of both the road condition and type, as well as evaluating the road ahead during vehicle operation for AEB. As a result, comparing our approach with the papers mentioned above becomes inherently ambiguous. Therefore, we compared our proposed method and the conventional approaches, utilizing the feature vector we generated. There are five other algorithms: deep ensemble learning (DEL) [22], long-short term memory (LSTM) [23], using only the time windowing method (TWM), K-nearest neighbor (KNN) [24], and support vector machine (SVM) [25]. A schematic structure of the five algorithms is shown in Fig. 9. DEL uses spatiotemporal information similar to the proposed

TABLE 2. The confusion matrix of validation data (LN Region).

Prediction			Actual								Snow	Total	Precision (%)
			Dry				Wet						
			Asphalt	Cement	Gravel	Sand	Asphalt	Cement	Gravel	Sand			
	Dry	Asphalt	6,906	4	0	24	0	4	0	0	8	6,946	99.4
		Cement	6	6,673	7	455	0	1	0	0	10	7,152	93.3
		Gravel	3	8	6,980	2	0	1	0	19	21	7,034	99.2
		Sand	35	247	7	6,516	0	1	0	0	0	6,806	95.7
	Wet	Asphalt	0	0	0	0	6,996	5	0	1	0	7,002	99.9
		Cement	16	2	0	0	4	6,919	1	8	4	6,954	99.5
		Gravel	0	0	0	0	0	6	6,949	119	0	7,074	98.2
		Sand	0	2	0	1	0	32	50	6,848	8	6,941	98.7
	Snow		34	64	6	2	0	31	0	5	6,949	7,091	98.0
Total			7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	63,000	
Recall (%)			98.7	95.3	99.7	93.1	99.9	98.8	99.3	97.8	99.3	**Accuracy (%): 98.0**	

TABLE 3. The confusion matrix of validation data (RN Region).

Prediction			Actual								Snow	Total	Precision (%)
			Dry				Wet						
			Asphalt	Cement	Gravel	Sand	Asphalt	Cement	Gravel	Sand			
	Dry	Asphalt	6,977	4	0	10	0	7	0	1	9	7,008	99.6
		Cement	0	6,761	4	216	0	1	0	0	60	7,042	96.0
		Gravel	0	12	6,981	2	0	0	0	20	38	7,053	99.0
		Sand	17	153	9	6,772	0	1	0	0	7	6,959	97.3
	Wet	Asphalt	1	0	0	0	6,994	2	0	1	0	6,998	99.9
		Cement	1	0	0	0	6	6,960	2	5	3	6,977	99.8
		Gravel	0	0	0	0	0	3	6,944	109	1	7,057	98.4
		Sand	0	2	0	1	0	32	50	6,848	8	6,941	98.7
	Snow		4	70	4	0	0	3	0	4	6,865	6,950	98.8
Total			7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	63,000	
Recall (%)			99.7	96.6	99.7	96.7	99.9	99.4	99.2	98.0	98.0	**Accuracy (%): 98.6**	

TABLE 4. The confusion matrix of validation data (LN Region w/o velocity feature).

Prediction			Actual								Snow	Total	Precision (%)
			Dry				Wet						
			Asphalt	Cement	Gravel	Sand	Asphalt	Cement	Gravel	Sand			
	Dry	Asphalt	4,793	54	8	6	0	159	37	174	222	5,453	87.9
		Cement	661	5,271	67	772	1	1	0	9	45	6,827	77.2
		Gravel	208	204	6,568	0	1	33	0	9	888	7,911	83.0
		Sand	226	1,431	219	6,222	0	6	0	4	4	8,112	76.7
	Wet	Asphalt	2	2	1	0	6,134	82	150	191	1	6,563	93.5
		Cement	156	11	7	0	49	5,901	14	790	54	6,982	84.5
		Gravel	0	0	0	0	766	122	6,402	1043	1	8,334	76.8
		Sand	20	1	3	0	49	563	391	4,690	8	5,725	81.9
	Snow		934	26	127	0	0	133	6	90	5,777	7,093	81.4
Total			7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	63,000	
Recall (%)			68.5	75.3	93.8	88.9	87.6	84.3	91.5	67.0	82.5	**Accuracy (%): 82.2**	

TABLE 5. The confusion matrix of validation data (RN Region w/o velocity feature).

Prediction			Actual								Snow	Total	Precision (%)
			Dry				Wet						
			Asphalt	Cement	Gravel	Sand	Asphalt	Cement	Gravel	Sand			
	Dry	Asphalt	5,374	36	4	0	1	57	140	122	254	5,988	89.7
		Cement	293	4,927	117	157	0	0	0	8	101	5,603	87.9
		Gravel	363	579	6,568	0	0	12	0	27	895	8,444	77.8
		Sand	157	1,428	209	6,843	0	0	0	6	55	8,698	78.7
	Wet	Asphalt	1	1	0	0	6,367	309	379	163	1	7,221	88.2
		Cement	98	2	2	0	72	5,266	100	1,339	46	6,925	76.0
		Gravel	1	0	0	0	528	366	5,942	912	0	7,749	76.7
		Sand	40	0	9	0	32	959	429	4,333	16	5,818	74.5
	Snow		673	27	91	0	0	31	10	90	5,632	6,554	85.9
Total			7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	63,000	
Recall (%)			76.8	70.4	93.8	97.8	91.0	75.2	84.9	61.9	80.5	**Accuracy (%): 81.4**	

algorithm. However, there is a difference in which the parallel structure of DNN is used for data fusion, as shown in Fig 9 (a). The first DNN structure used for DEL is the same as the proposed method. The second DNN consists of three hidden layers, and the number of neurons in each layer is set to 100, 80, and 80, respectively. Both DNNs

TABLE 6. Accuracy of six methods used in comparative study(%).

|	Proposed	DEL	LSTM	TWM	KNN	SVM
LN	**98.0**	94.8	87.3	86.6	85.7	76.9
RN	**98.6**	94.6	87.0	83.6	83.5	79.5
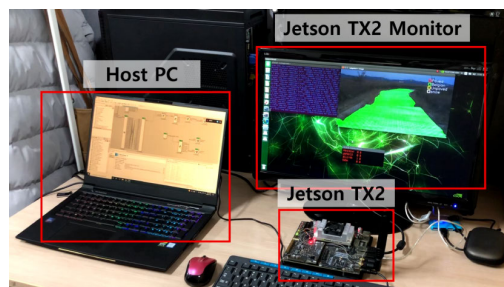

(a) Experimental environment
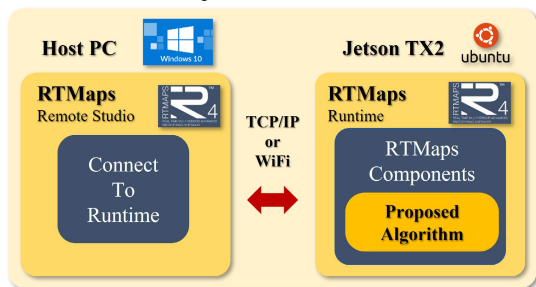
(b) Overall structure of real time operation using jetson TX2 board

FIGURE 10. Experimental environment and overall structure of real time operation.

used non-linear functions, e.g., tangent-sigmoid functions, as transfer functions. LSTM is useful for learning time-sequence data. In this comparative study, the number of hidden units is 200 in LSTM. Each unit's state and gate activation functions are set to tangent-hyperbolic and sigmoid functions, respectively. The structure of LSTM is shown in Fig 9 (b). TWM excludes the spatiotemporal fusion method from the proposed method, as shown in Fig. 9 (c). The structure of the DNN used here is the same as the proposed method. KNN and SVM techniques have been utilized to classify the condition and type of road surface using LiDAR. Therefore, we compared using KNN and SVM, except for DNN and spatiotemporal fusion method in the proposed method as shown in Fig. 9 (d) and (e) to validate our algorithm. In Table 6, we compared the accuracy of SVM, KNN, TWM, DEL, and LSTM with the accuracy of the proposed method. It can be seen that the accuracy of the proposed method showed the highest values in both regions. Here, we can see that the DEL method performs similarly to the proposed method because it contains spatiotemporal information. However, the proposed method can be analyzed for better performance for moving ego vehicles since it uses a fusion method dependent on the ego vehicle's speed.
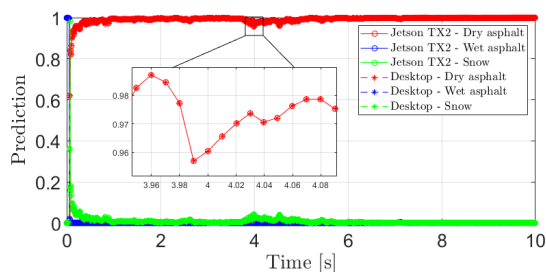
(a) Classification probability of dry asphalt with Jetson TX2
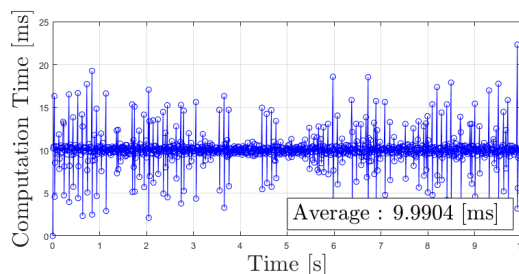
(b) Computation time

FIGURE 11. Experimental results using Jetson TX2.

D. REAL TIME PERFORMANCE WITH NVIDA JETSON TX2

All the performances reported in the previous section were the outcome of a desktop personnel computer. To see the real time application of the proposed method, we executed the proposed algorithm by porting it into a Jetson TX2 board from NVIDIA. In the actual AEB application, sensor data fusion and risk assessment algorithms for potential collision candidates can also significantly impact the task computation time. However, it was deemed essential to check how much real-time resources the method proposed in this paper occupies in the total execution time. Therefore, we confirmed the performance assuming a driving condition with no forward vehicles or pedestrians. The experimental environment is shown in Fig. 10 (a), and the overall structure of the experiment is shown in Fig. 10 (b). The Host PC was used only to provide an interface for RTMaps and was not involved in running the algorithm. The Host PC and the Jetson TX2 are communicated over TCP/IP protocol. The CPU of the Host PC is an Intel i7-8700k six-core, and the CPU of the Jetson TX2 is Quad ARM A57 MPCore.

We compared the results calculated in the desktop environment with those calculated in real time in the Jetson TX2. We showed an implementation example for dry asphalt, wet asphalt, and snow classes to see whether our algorithm runs in real time. Figure 11 (a) compares the results in desktop and Jetson TX2 for data with a ground truth of dry asphalt. The circle marker is the result of Jetson TX2, and the star marker is the result of the desktop. In the enlarged part, it can be seen that the results in the desktop and Jetson TX2 are almost the same. In addition, we considered the computation time essential for the proposed algorithm to operate in real time, so we calculated the time when the algorithm was implemented in Jetson TX2. The computation time is shown in Fig. 11 (b), and its average is about 10 ms.

V. CONCLUSION

In this paper, we proposed a classification method for road surface conditions and types using LiDAR for AEB systems. We divided the front road area into four subregions to utilize spatial information. The reflectivity, the number of point clouds, and vehicle speed were selected as the feature vector from the LiDAR and in-vehicle sensors. Then, the DNN was used to classify road surface conditions and types for each subregion. The outputs of the DNN were used for spatiotemporal formulation to make the final classification results of the subregions near the ego vehicle. To validate the proposed algorithm, we compared it with five other methods. The accuracy of the subregions near the ego vehicle using the proposed method was the highest. In addition, we implemented the proposed algorithm into the Jetson TX2 board to show that our method can be applied in real time. The average computation time was about 10 ms, and the classification probability results were similar to desktop results. Therefore, the proposed method is expected to improve the safety of automated driving vehicles and be applicable in the real world. In the current study, the LiDAR and in-vehicle sensors were fused to classify road surfaces, but in future studies, environmental sensors such as acoustic sensors and cameras are used together to design a more accurate road surface classification system.

REFERENCES

[1] C. G. Wallman and H. Åström, "Friction measurement methods and the correlation between road friction and traffic safety: A literature review," Swedish Nat. Road Transp. Res. Inst. (VTI), Linköping, Sweden, Tech. Rep. 911A, 2001.

[2] C. A. Brooks and K. Iagnemma, "Vibration-based terrain classification for planetary exploration rovers," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1185–1191, Dec. 2005.

[3] J. H. Yang, D. J. Kim, W. Y. Choi, and C. C. Chung, "SAFE-STOP system: Tactical intention awareness based emergency collision avoidance for malicious cut-in of surrounding vehicle," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Indianapolis, IN, USA, Sep. 2021, pp. 1533–1540.

[4] J. C. Aguilar, J. C. Carrillo, A. G. Fernández, and E. Carabias Acosta, "Robust road condition detection system using in-vehicle standard sensors," *Sensors*, vol. 15, no. 12, pp. 32056–32078, Dec. 2015.

[5] I. Abdic, L. Fridman, D. E. Brown, W. Angell, B. Reimer, E. Marchi, and B. Schuller, "Detecting road surface wetness from audio: A deep learning approach," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Cancun, Mexico, Dec. 2016, pp. 3458–3463.

[6] L. Cheng, X. Zhang, and J. Shen, "Road surface condition classification using deep learning," *J. Vis. Commun. Image Represent.*, vol. 64, Oct. 2019, Art. no. 102638.

[7] M. Nolte, N. Kister, and M. Maurer, "Assessment of deep convolutional neural networks for road surface classification," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 381–386.

[8] S. Marianingsih and F. Utaminingrum, "Comparison of support vector machine classifier and Naïve Bayes classifier on road surface type classification," in *Proc. Int. Conf. Sustain. Inf. Eng. Technol. (SIET)*, Malang, Indonesia, Nov. 2018, pp. 48–53.

[9] P. Wang, "Research on comparison of LiDAR and camera in autonomous driving," *J. Phys., Conf.*, vol. 2093, no. 1, Nov. 2021, Art. no. 012032.

[10] M. Aki, T. Rojanaarpa, K. Nakano, Y. Suda, N. Takasuka, T. Isogai, and T. Kawai, "Road surface recognition using laser radar for automatic platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2800–2810, Oct. 2016.

[11] S. Wang, S. Kodagoda, and R. Ranasinghe, "Road terrain type classification based on laser measurement," in *Proc. Australas. Conf. Robot. Autom.*, Wellington, New Zealand, 2012, pp. 1–13. [Online]. Available: https://opus.lib.uts.edu.au/handle/10453/31691

[12] K. Walas and M. Nowicki, "Terrain classification using laser range finder," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, USA, Sep. 2014, pp. 5003–5009.

[13] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork, "Weather influence and classification with automotive LiDAR sensors," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Paris, France, Jun. 2019, pp. 1527–1534.

[14] G. Sebastian, T. Vattem, L. Lukic, C. Burgy, and T. Schumann, "RangeWeatherNet for LiDAR-only weather and road condition classification," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Nagoya, Japan, Jul. 2021, pp. 777–784.

[15] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "LiDAR point clouds correction acquired from a moving car based on CAN-bus data," 2017, *arXiv:1706.05886*.

[16] (2023). *European New Car Assessment Programme Test Protocol— AEB Carto-Car System, Version 4.2, Euro NCAP*. [Online]. Available: https://www.euroncap.com/en/for-engineers/protocols/safety-assist/

[17] D. J. Kim, J. S. Kim, S.-H. Lee, and C. C. Chung, "A comparative study of estimating road surface condition using support vector machine and deep neural networ," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Auckland, New Zealand, Oct. 2019, pp. 1066–1071.

[18] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, no. 4, pp. 525–533, 1993.

[19] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox User's Guide*. Natick, MA, USA: MathWorks, 2016.

[20] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[21] S. Jana, A. L. Middya, and S. Roy, "Participatory sensing based urban road condition classification using transfer learning," in *Mobile Networks and Applications*. Berlin, Germany: Springer, 2023, doi: 10.1007/s11036-023-02118-6.

[22] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer, 2000, pp. 1–15.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[24] H. Samet, "K-nearest neighbor finding using MaxNearestDist," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 243–252, Feb. 2008.

[25] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul. /Aug. 1998.

JU WON SEO (Graduate Student Member, IEEE) received the B.S. degree in mechanical and control engineering from Handong University, Pohang, South Korea, in 2019. She is currently pursuing the Ph.D. degree in electrical engineering with the System and Control Laboratory, College of Engineering, Hanyang University, Seoul, South Korea. Her main research interests include autonomous driving vehicles, path planning, sensor fusion, robust control, optimal control, and artificial intelligence. She is a Student Member of the Korean Society of Automotive Engineers (KSAE) and the Institute of Control, Robotics, and Systems (ICROS).

JIN SUNG KIM (Graduate Student Member, IEEE) received the B.S. degree in electronic engineering from Kookmin University, Seoul, South Korea, in 2014, and the M.S. degree in electrical engineering from Hanyang University, Seoul, in 2019, where he is currently pursuing the Ph.D. degree in electrical engineering. His main research interests include data-driven control, automated vehicles, optimal control, robust control, and artificial intelligence. He is a member of the IEEE Control System Society (CSS), the Society of Automotive Engineers (SAE), the Korean Society of Automotive Engineers (KSAE), and the Institute of Control, Robotics and Systems (ICROS).

JIN HO YANG (Graduate Student Member, IEEE) received the B.S. degree in electronics and communication engineering from Hanyang University, Ansan, South Korea, in 2017, and the M.S. degree in electrical engineering from Hanyang University, Seoul, South Korea, in 2020, where he is currently pursuing the Ph.D. degree in electrical engineering with the Systems and Control Laboratory.

Since 2020, he has been with Hyundai Mobis Technical Center, Yongin-si, South Korea, where he is a Research Engineer responsible for developing the ADAS, such as the autonomous emergency braking systems. His research interests include optimal estimation and control theory, machine learning, autonomous driving, and vehicle active safety. He is a member of the IEEE Intelligent Transportation Systems Society (ITSS), IEEE Robotics and Automation Society (RAS), the Korean Society of Automotive Engineers (KSAE), the Institute of Control, Robotics, and Systems (ICROS), and Korean Institute of Communications and Information Sciences (KICS).

CHUNG CHOO CHUNG (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, South Korea, and the Ph.D. degree in electrical and computer engineering from the University of Southern California, Los Angeles, CA, USA, in 1993.

From 1994 to 1997, he was with the Samsung Advanced Institute of Technology, South Korea. In 1997, he joined as a Faculty Member of Hanyang University, Seoul.

Dr. Chung is a member of the National Academy of Engineering of Korea (NAEK). He was the Program Co-Chair of ICCAS-SICE 2009, Fukuoka, Japan, and the 2015 IEEE Intelligent Vehicles Symposium, COEX, South Korea, the Organizing Chair of the International Conference on Control, Automation, and Systems (ICCAS), KINTEX, South Korea, in 2011, and the General Chair of ICCAS 2019, Jeju, South Korea, and CDC 2020, Jeju. He was an Associate Editor of the *Asian Journal of Control*, from 2000 to 2002, and a (founding) Editor of the *International Journal of Control, Automation, and Systems*, from 2003 to 2005. He also served as an Associate Editor for various international conferences, such as the IEEE Conference on Decision and Control (CDC), the American Control Conferences, the IEEE Intelligent Vehicles Symposium, and the Intelligent Transportation Systems Conference, and a Guest Editor for a Special Issue on advanced servo control for emerging data storage systems published by the IEEE TRANSACTIONS ON CONTROL SYSTEM TECHNOLOGY, in 2012, and the *IEEE Intelligent Transportation Systems Magazine*, in 2017. He is currently an Associate Editor of the *Mechatronics* (IFAC). He was the President of the Institute of Control, Robotics, and Systems, South Korea, in 2019.

● ● ●