IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 November 2022, accepted 2 December 2022, date of publication 5 December 2022,
date of current version 8 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226816

RESEARCH ARTICLE

Fi-Vi: Large-Area Indoor Localization Scheme Combining ML/DL-Based Wireless Fingerprinting and Visual Positioning

SANGWOO PARK[1], DONG HO KIM[2], (Senior Member, IEEE), AND CHEOLWOO YOU[1], (Member, IEEE)

[1]Department of Information and Communications Engineering, Myongji University, Yongin 17058, South Korea
[2]Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology, Seoul 139743, South Korea

Corresponding author: Cheolwoo You (cwyou@mju.ac.kr)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00994, Development of autonomous VR and AR content generation technology reflecting usage environment).

ABSTRACT Due to recent technological developments such as online navigation, augmented reality (AR), virtual reality (VR), and digital twins, and the high demand from users for various location-based services (LBS), research on location estimation techniques is being actively conducted. As a result, there is an increasing demand for effective localization technologies that can be used in places where the use of Global Positioning System (GPS) is limited, especially in indoor spaces with very large areas. In this paper, a new structure for an indoor localization system in which wireless fingerprinting and visual-based positioning are hierarchically combined—the so-called Fi-Vi system—is proposed. This scheme consists of two steps: fingerprint-based localization (FBL) followed by visual-based localization (VBL). In the first positioning step (i.e., the FBL stage), the entire area of a significantly broad range for localization is divided into multiple regions, the size and the number of which depend on the target accuracy of this step. Moreover, a machine-learning (ML) or deep-learning (DL) model trained on a Wi-Fi fingerprint radio map selects suitable candidate regions among these multiple regions. In the second positioning step (i.e., the VBL stage), the final location is precisely estimated through visual-based positioning based on the received information regarding the candidate regions. The FBL stage uses a sparse radio map (SRM) for fingerprinting, which can be constructed with relatively little effort and cost compared to radio maps used in conventional fingerprinting methods. As a result, it can be easily combined with existing visual-based positioning methods with almost negligible implementation complexity. Because of the hierarchical structure and SRM, the proposed scheme shows a significant performance improvement in terms of computational load and time required for indoor localization compared to the use of the existing visual-based indoor positioning method alone. In addition, it provides high accuracy and robustness even in a dynamically changing indoor wireless environment where conventional wireless fingerprinting methods show significant performance degradation. Finally, the performance analysis of the proposed scheme was performed using the UJIIndoorLoc dataset. Experiments and theoretical analysis have shown that when the estimation accuracy of the candidate region for the test dataset was achieved at about 99% through the FBL stage, the average computational amount of the VBL stage for the final position estimation was only about 16% of that in cases where the visual-based positioning method was used alone.

INDEX TERMS Indoor localization, wireless fingerprinting, machine learning, deep learning, radio map, visual-based localization.

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

I. INTRODUCTION

Recently, due to the spread of high-performance mobile communication terminals such as smartphones and tablet PCs and the development of communication technology, demand for various new services such as augmented reality (AR) and virtual reality (VR) services, digital care, and online navigation has significantly increased. In particular, demand for location-based services (LBS) utilizing users' location information via various communication terminals has grown rapidly. For extended reality (XR) services, which have recently received significant attention, virtual objects superimposed on real objects must be properly aligned to obtain a clear integrated image of the real objects within XR systems. To achieve this, a location tracking system that can accurately calculate a user's location and direction in their surrounding environment is needed [1].

Currently, LBS being actively used rely heavily on Global Positioning System (GPS)-based localization technology. GPS with a wide signal radius can provide stable service in an outdoor environment with line of sight (LOS). However, most indoor environments, such as buildings, tunnels, and underground, have shaded areas where GPS signals cannot be received or are very weak [2]. Therefore, there is an increasing need for new positioning techniques that can more accurately estimate users' locations in various indoor environments, and many related methods are being studied [3]. In particular, researchers are investigating indoor positioning methods that use various wireless signals such as Wi-Fi [4], [5], [6], Bluetooth [7], [8], [9], [10], radio frequency identification (RFID) [11], [12], [13], [14] magnetic fields [15], [16], ultra-wideband (UWB) [17], [18], and wireless sensor network (WSN) environments [19].

Through several studies, it has become widely known that the fingerprinting method using received signal strength indication (RSSI) for localization is an efficient alternative in indoor environments where GPS signals cannot penetrate. RSSI, which is a wireless signal that reaches a given device through an indoor structure, reflects various distortions caused by the indoor environment, so it can be efficiently used for indoor location estimation [20]. Wi-Fi is already widely distributed in indoor environments, requiring low infrastructure implementation costs to build an indoor positioning system. Furthermore, Wi-Fi-based fingerprinting can be applied without knowledge of the location and direction of the access point (AP). As a result, this method has become a promising method for indoor localization. Nevertheless, there are many limitations that make it difficult to apply fingerprinting methods. For instance, the fingerprint database of the space to be located must be precisely configured in advance [21]. In addition, as the range of positioning widens, additional hardware infrastructure installation is required, increasing implementation costs. Furthermore, properly deploying a finite number of APs is important in improving positioning accuracy, but it is a difficult task that requires significant time and manual work [22]. There is a trade-off between the size of a cell and the positioning accuracy—for example, where a set cell size is smaller than a positioning error range, the positioning accuracy decreases, and if the cell size is increased, the positioning accuracy increases but the range of positioning error may increase. This trade-off becomes a factor that makes fine positioning difficult when using the fingerprinting method.

RSSI decreases significantly when it passes through obstacles such as walls. In addition, the strength of the received signal varies depending on how it passes through spaces separated by obstacles. Considering such spatial features, RSSI will be effective in distinguishing indoor spaces. However, narrowing the gap between the reference points (RPs) increases the RSSI similarity between samples, making it difficult to predict the exact location of each RP. This is particularly noticeable in structures with fewer obstacles, such as corridors. It is important to establish appropriate RP intervals because localization at a finer interval without the installation of additional APs will cause a decrease in positioning accuracy [23]. It is difficult to ensure performance in environments with frequently moving obstacles or a large floating population using fingerprinting methods, and new data may need to be collected if there is a change in the channel environment such as AP change or power adjustment. Suning He et al. proposed a method to automatically update a fingerprint if there is a changed AP signal from the previously collected Wi-Fi fingerprint in a wide range of indoor environments. They reduced the average positioning error in environments with altered AP signals by about 20% through the proposed method [24].

Meanwhile, it is necessary to consider problems that may occur due to the configuration of a fingerprint consisting of a collection step that proceeds the offline phase and a positioning step that proceeds the online phase. Because each device may have different antenna and chip designs, even RSSI values measured at the same location can be different if the devices used for measurement are different. Xue, Jianzhe et al. showed that even if measured in the same place on the same device, an RSSI value can differ within 10 dB [25]. Due to this problem, if the device used in the offline phase and the device used in the online phase are different, the RSSI obtained at the same location may be different, which may degrade positioning accuracy.

As mentioned above, despite various studies, the theoretical limitations of the RSSI fingerprint-based methodology are open problems [26], and the results obtained depend heavily on the environment and devices used in the experiment [27], [28]. As well, most previous work has focused on improving the accuracy of the model according to specific data. Therefore, the consistency problems of positioning accuracy according to data must still be addressed. In addition, the collected fingerprints cannot reflect visual information in the space where the user is, so they cannot reflect additional information, such as the direction the user is looking. For these reasons, precise indoor positioning based on RSSI is difficult, and as the required service level increases, a method to fundamentally address the outlined problem is needed to achieve

greater positioning accuracy. To solve this problem, several studies have been conducted to complement the fingerprinting method in combination with additional sensors [29].

Another promising indoor positioning technique is a visual-based method. In particular, the visual-based indoor positioning method has a relatively low positioning error compared to the fingerprinting method and has been widely studied for its applications in the field of AR given the advantage of users being able to consider the visual information being acquired [30], [31]. However, the visual-based indoor positioning method requires greater computation and hardware complexity compared to the fingerprinting method, and there is the issue of positioning accuracy in a space composed of similar images being poor. Therefore, there are many problems to consider in order to apply this method to tasks requiring real-time processing or devices with low computation-processing capabilities.

In the visual-based indoor positioning method, since the position estimation performance varies depending on the completeness of the map, considerable technology and effort are required to generate a map with high completeness. In the case of visual simultaneous localization and mapping (vSLAM), one of the representative related technologies, there is distortion due to various causes between the visual map generated from the collected images and ground-truth. Occasionally, the generated visual maps do not correctly reflect the actual information, which can lead to discrepancies between the location estimated by vSLAM and the actual location. In contrast, fingerprints have somewhat different characteristics from visual maps because they are constructed using a radio map that reflects spatial information including the environmental factors of the collection site at the actual location.

In this paper, we propose so-called ''Fi-Vi,'' an indoor positioning system that hierarchically combines the wireless fingerprinting method and visual-based positioning method. The proposed scheme has robustness against changes in the wireless channel environment compared to conventional wireless fingerprinting methods, even in very large indoor spaces. Additionally, an advantage of the proposed scheme is its very fast processing speed compared to the conventional visual-based positioning method. The Fi-Vi system first positions a relatively wide range based on probabilities predicted via machine learning (ML) or deep learning (DL)- based fingerprinting methods, and then performs visual-based positioning with very few calculations based on the information obtained by the ML/DL-based fingerprinting method. As a result, the Fi-Vi system is capable of indoor localization with equal or better accuracy, requiring significantly less computation than the conventional visual-based positioning method alone. As well, a radio map can be used to correct the distortion that occurs during the stage of visual map generation.

The main contributions of this paper are threefold and are summarized below:

- In this paper, we propose a novel indoor positioning scheme that enables efficient positioning in a large area through a hierarchical combination of two kinds of positioning methods. Experiments and theoretical analysis confirmed that the proposed system can be implemented by adding a negligible level of implementation complexity to the existing visual-based positioning system, but it can achieve high positioning accuracy while significantly reducing computational complexity. In addition, an advantage obtained through the combination of the two methods is that it is possible to alleviate decreases in accuracy in a space with many similar shapes, which is a problem with visual-based indoor positioning.
- In the proposed system, ML/DL-based fingerprinting methods are employed to classify the entire positioning target space into multiple regions of relatively large size compared to conventional fingerprinting methods. These ML/DL-based classifiers learn a sparse radio map (SRM) that is constructed using very few RPs over a very wide area. Therefore, the proposed system is relatively free from accuracy degradation due to differences between RSSIs obtained at the same location in online and offline phases, labor-intensive tasks, and high costs required for precise positioning (e.g., adding many APs or setting appropriate RP intervals), which are typical problems of conventional fingerprinting methods. In addition, the proposed system specifically measures the region classified through the fingerprinting-based classifier using visual-based positioning, thereby reducing the additional effort or complex computations required for further learning or improvement to construct a fingerprinting model with high positioning accuracy.
- The proposed system architecture is relatively robust when there are environmental changes caused by various reasons (i.e., RSSI collector problems, changes in channel environment or time zone between online and offline phases, and changes in RSSI reception sensitivity due to differences in equipment used for measurement) compared to the conventional fingerprinting method.

The remainder of this paper is organized as follows: In Section II, some basic concepts of fingerprinting and visual-based positioning are presented, and the dataset used in the experiments is also briefly introduced. In Section III, we describe the structure of the proposed system in detail. In Section IV, the numerical results of the simulations are described. In Section V, we analyze the complexity according to the proposed evaluation metrics and discuss effective application methods according to the environment in which the proposed scheme is applied. Finally, we conclude this work with discussion on future work in Section VI.

II. PRELIMINARIES

In this section, we briefly introduce the typical indoor positioning method using fingerprints, sensor-based positioning methods, and related works. As well, we describe a

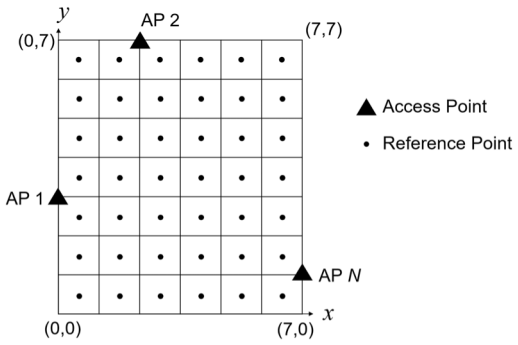

FIGURE 1. An example of fingerprinting.

dataset used in the performance analysis of the proposed scheme.

A. FINGERPRINTING

The fingerprinting method is based on empirical data. The localization steps of the fingerprinting method are generally divided into two stages. In the offline stage, the whole region to be localized is divided into virtual grid forms, and the center of each grid is set to RP; the RSSI for each RP is collected and stored in a database (DB). In the online stage, the most similar RP is selected as the location of the user/device by comparing the RSSI value with the DB stored in the offline stage [2].

Fig. 1 represents an example of the RSSI collection stage in a Wi-Fi–based fingerprinting method, where there are N APs in a specific space. Here, each coordinate is set by assuming any point in the virtual space as the origin (0,0) while assuming to have set the RPs at a uniform interval (here, 1 m). Each triangle represents an AP, and the small circles in the grid represent locations—i.e., the RPs—to measure the RSSI for the grid when a DB is constructed.

RSSI collections are performed in the following order: First, the coordinates of the RP and RSSI values collected from adjacent APs from the corresponding coordinates are recorded in the DB. The configured DB is referred to as a radio map, and assuming that there is a total of M RPs, the radio map may be expressed as follows:

$$RadioMap \Rightarrow \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} & \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_M & y_M \end{pmatrix} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} & \\ \vdots & \vdots & \ddots & \vdots & \\ r_{M,1} & r_{M,2} & \cdots & r_{M,N} & \end{bmatrix} \tag{1}$$

where $r_{i,j}$ denotes an RSSI value received from the j-th AP in the RP located in (x_i, y_i). This radio map will be used to determine the location of the user or device in the online phase.

B. VISUAL SLAM

SLAM is a promising method for sensor-based indoor positioning. As such, SLAM is a technology that simultaneously localizes and maps the environment through built-in sensors, which are activated as a robot passes through an unknown environment [32]. SLAM performs mapping using a variety of sensors, including LiDAR, RGB-D, and optical camera sensors. Among them, optical camera sensors are smaller and cheaper; moreover, they can further utilize semantic information of images (e.g., people, objects, etc.) and thus are widely used in SLAM research. Furthermore, service providers benefit from the wide distribution of devices with optical cameras. However, vSLAMs using optical-camera-based sensors are sensitive to environmental and lighting changes, resulting in errors between mapping and positioning. As well, there is a problem in that the amount of computation that must be performed to find a location increases rapidly depending on the size of the whole positioning area and the number of feature points obtained from the visual information. Despite the growing demand for vSLAM, as yet, there is no perfect solution to this issue given ongoing changes in environmental conditions and various problems caused by the use of optical camera lenses. For instance, indoor spaces comprised of similar scenes, such as classrooms and hospitals, and indoor spaces with low illumination lower the positioning accuracy of vSLAM [33].

In addition, maps produced through vSLAM vary in location estimation performance depending on the completeness of the map [34]. However, the creation process of most vSLAM maps is based on the experience of the creator. Therefore, the estimation performance depends on the SLAM algorithm and the competence of the map creator. Specifically, when the amount of information collected for a specific area is insufficient, vSLAM-based location estimation for the corresponding area becomes inaccurate [35]. To address this problem, various studies have been conducted in which two or more types of sensors are applied together [36], [37], [38].

C. FINGERPRINT LOCALIZATION BASED ON MACHINE LEARNING AND DEEP LEARNING

Research has been actively conducted to interpret fingerprints by applying ML/DL techniques. For example, Mauro Brunato et al. conducted fingerprint-based indoor localization under a wireless local area network (WLAN), using support vector machine (SVM), weighted k-nearest neighbors (KNN), multi-layer perceptron (MLP), and Bayesian approach (BAY) algorithm, and discussed processing time and power consumption [39]. In [40], indoor positioning accuracy was estimated by applying SVM and logistic regression techniques. In [4], the authors considered the physical constraints of pairwise distance by selecting adjacent terminals; they applied KNN to improve positional accuracy compared to classical fingerprinting methods. In addition, the study in [41] was conducted to apply the RSSI of APs measured in multi-building–multi-floor buildings to deep neural networks (DNNs). The study in [42] was conducted to improve the accuracy of distorted fingerprints using a denoising autoencoder.

FIGURE 2. The three buildings used for data collection at Universitat Jaume I. (This figure was captured from Google Maps.)

Meanwhile, conventional fingerprinting methods have the computational complexity of O(MN), and as the number of fingerprints, M, constituting radio map and the number of APs, N, constituting radio map increase, the positioning space explosively increases. Some ML/DL models are known as highly effective methods of computational relaxation. In general, in a typical DNN model where learning is completed, the time complexity of a hidden layer with n inputs and H outputs has O(nH) regardless of N *or* M [39].

D. DATASET

Experimenting with and evaluating the proposed structure requires large Wi-Fi RSSI datasets collected across numerous buildings and floors. In this study, UJIIndoorLoc [43] was used as the dataset for the experiments. UJIIndoorLoc is a widely known Wi-Fi–based fingerprint dataset used in multi-building–multi-floor indoor positioning research. UJIIndoorLoc was published by Indoor Positioning and Indoor Navigation (IPIN) in 2014 and consists of data measured in three buildings—four or five floors of the Universitat Jaume I in the state of Valencia, Spain. Fig. 2 shows a view of Universitat Jaume I captured from Google Maps. The total area used for the measurements corresponds to almost 108,700 m^2. UJIIndoorLoc consists of RSSI values received from 520 wireless APs, including 19,938 items of training data and 1,111 items of test data. The training data includes RSSI (dBm) values collected from the RPs, altitude/longitude of the RPs, building numbers, floor numbers, room numbers, relative location information (in the room or corridor), device information used for measurement, height of persons, and measurement times. Additionally, the test data contains only information on RSSI (dBm) values collected from the RPs, altitude/longitude of the RPs, building numbers, floor numbers, device information used for measurement, and measurement times. Signal strength is expressed as a negative integer value from -104 dBm to 0. The signal strength of the AP not received from the RP is indicated as +100. UJIIndoorLoc was created by 20 users of 25 Android OS terminals, and the test data was collected three months after the training data was collected. A total of 25 types of devices were used to build the UJIIndoorLoc dataset. Of these, 17 types of devices and 11 types of devices were used to collect the training dataset and the test dataset, respectively. Since this data was collected in consideration of various environments, it can be understood to reflect the problems of other model devices that may occur in the fingerprinting method above described, as well as environmental changes in the collection and positioning steps. Therefore, the experimental results obtained using this data are suitable for estimating the performance of the proposed system reflecting the actual environment. In addition, it contains many samples for multi-building–multi-floor buildings over a large area where the spacing between RPs, making it more suitable for spatial positioning than accurate coordinates; thus, it is a suitable dataset for use in the indoor positioning experiment proposed in this paper.

TABLE 1. The main properties of UJIIndoorLoc dataset.

Features	Values
Total number of WAPs	520
Total number of buildings	3
Total number of floors	13
Total number of RPs	735
The number of user ID types	20
The number of phone ID types	25
Representation of missed RSSI values	+100 dBm

Table 1 describes some of the properties constituting UJIIndoorLoc. On average, only 3% of the RSSI values measured from 520 APs had meaningful signal strength. For sparse data, an autoencoder can effectively reduce data dimensionality while preserving the functional information needed to distinguish samples [41].

III. PROPOSED SCHEMES

In this section, a large-area indoor localization scheme in which Wi-Fi fingerprinting and visual-based positioning are hierarchically combined—the so-called Fi-Vi system—is presented. This Fi-Vi system consists of the following two stages: (1) fingerprint-based localization (FBL) stage using an SRM, and (2) visual-based localization (VBL) stage using a visual map. The main role of FBL is to reduce the search range of the visual map to be used by VBL with high reliability. As a result, the combined VBL and FBL approach requires a significantly reduced computational amount, has a faster processing speed, and improves position estimation performance compared the use of VBL alone. In particular, the Fi-Vi system uses an SRM that is constructed using very few RPs over a very wide area; an SRM is also easy to build and is considerably smaller in size than the radio map of

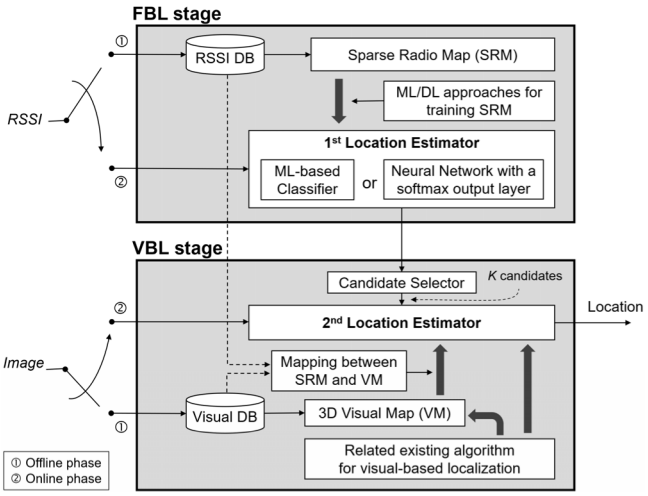

FIGURE 3. Generalized architecture of the proposed Fi-Vi system.

a conventional indoor positioning system using only Wi-Fi fingerprinting. In addition, when building a visual map during the visual-based indoor localization process, SRMs can be built simultaneously with little effort.

A. OVERALL SYSTEM ARCHITECHTURE

Fig. 3 shows the architecture of the Fi-Vi system, which can be divided into two parts—the FBL and VBL stages. In addition, as shown in this figure, both FBL and VBL have an offline phase and an online phase. In the offline phase, the collector gathers the wireless signals—RSSI values in the present work—from nearby Wi-Fi APs, as well as images/videos through various sensors while exploring an unknown space. FBL and VBL use the collected information to perform various tasks, such as the construction and learning of the SRM and 3D visual map. As an implementation example, a mobile device for building a visual map collects images/videos while traversing an entire space. At the same time, the mobile device intermittently collects RSSI values according to a predetermined rule (e.g., at every specific time interval or after every move of a certain distance), which is a highly trivial operation. The visual map corresponding to the collected RSSI is constructed according to sub-regions, which can be determined manually or automatically. In the online phase, FBL and VBL cooperate to perform indoor localization based on the Wi-Fi signal and image/video collected in real time by the user device at a specific location.

First, in the offline phase, FBL selects only the data necessary to build an SRM of the desired scale from the RSSI DB containing a variety of collected information. The selection criterion is determined by the sparseness of the SRM, which is directly related to the size of regions to be estimated by the FBL and the desired estimation accuracy. The characteristics of the selected data, especially undefined or meaningless feature values, are preprocessed to make them suitable for training the first location estimator. After this process, the first location estimator will have learned the characteristics of the constructed SRM. At this point, the output value of the first location estimator must have suitable characteristics to cooperate with VBL—to achieve this, various ML and DL techniques were applied in the present work. During the offline phase, VBL builds a 3D visual map from the collected images or videos and then performs matching between the constructed visual map and the SRM.

In the online phase, the mobile device measures images and RSSI values simultaneously for indoor positioning. Given the RSSI values from the nearby Wi-Fi APs measured by the mobile device at a specific location, the first location estimator of FBL outputs information on the candidate regions defined while offline; this information consists of probability values indicating that each candidate region is a correct answer. After receiving the output from the first location estimator, the candidate selector of the VBL selects K candidates from all candidates according to a predetermined rule and sends them to the second location estimator. Finally, the second location estimator determines the final location through several steps based on the information regarding the K candidates.

B. SPARSE RADIO MAP

As mentioned in Section I, in general RSSI-based fingerprinting methods, it is necessary to construct a complex and dense radio map by collecting RSSIs from a very large number of RPs arranged at very tight intervals to perform accurate positioning over a wide area. This is a major problem that makes it difficult to apply fingerprinting methods to a precise positioning system for a large area. In contrast, in the proposed Fi-Vi system, fingerprinting is not used for highly precise positioning; instead, it is used for coarse positioning performed by the first location estimator—i.e., selecting a small number of candidate regions used by the second location estimator that performs an accurate final localization. For example, the first location estimator classifies an entire area into buildings, floors, rooms, etc., helping the second location estimator to quickly estimate the exact final location with significantly reduced computational load. Therefore, there is no need to densely arrange RPs when constructing a radio map, and it is sufficient to use an SRM constructed only of RSSIs collected from very sparsely placed RPs.

Fig. 4 shows the difference between the conventional radio map and the SRM, where the building is the sixth floor of the fifth engineering building of the university to which the authors belong. In this figure, the distance between the RPs in the radio map for precise positioning is about 0.7 m. In this case, when positioning over a large area through fingerprinting, difficulties arise in both steps of data collection and positioning due to too many RPs. However, as shown in Fig. 4, the SRM requires a relatively small number of RPs compared to the radio map.

As described below in Section III–C, the proposed Fi-Vi system uses this SRM to train the first location estimator for the efficient operation of the overall system. We confirmed through some preliminary experiments that a simple classifier

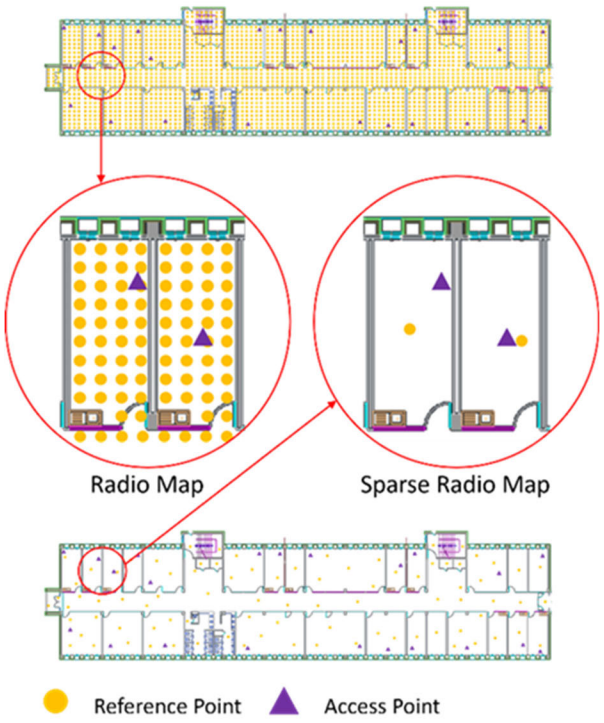

FIGURE 4. Comparison between the radio map (up/left) and sparse radio map (down/right).

trained on SRM, built on a very small amount of data collected without much effort, can localize the floors or rooms in buildings with high accuracy.

C. FBL STAGE UTILIZING DL/ML APPROACHES

In this subsection, a DL-based FBL method is proposed. This scheme is suitable to deliver a single candidate to the VBL, but sometimes it does not work efficiently when using information about multiple candidates. To solve this problem, ML-based FBL methods are also proposed.

As mentioned above, the proposed FBL methods first determine how many sub-regions to divide the entire area into to construct an SRM in the offline phase. These sub-regions become the candidate regions to be estimated by the FBL, and in this paper, it was assumed that there were N_c candidate regions. One RP refers to a specific location for measuring RSSI values. The candidate regions may have various sizes, and there may be one or more RPs that correspond to one candidate region. As a result, the RPs will be non-uniformly distributed. To construct the radio map sparsely, each selected candidate region was relatively wide. The following are examples of suitable candidate regions: each building, each floor of a building, each hallway or part of a hallway, each room, etc.

Based on the SRM characteristics outlined above, the proposed FBL methods can be differentiated from existing Wi-Fi fingerprinting methods that construct a radio map for a relatively fine and uniform area. On the other hand, if necessary,

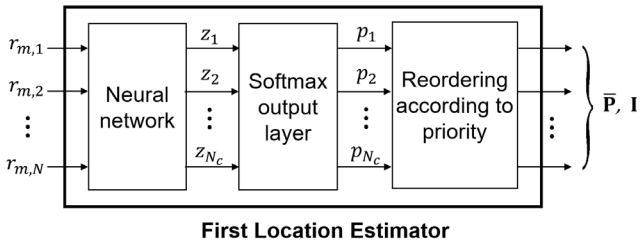

FIGURE 5. First location estimator based on a deep learning approach.

appropriate candidate regions may be determined through analysis after collecting data from arbitrary RPs.

1) DL-BASED FBL

In this sub-section, the implementation of the FBL using a deep learning approach is described

During the offline phase, when there are N Wi-Fi APs in the entire area, the device collects the RSSI vectors defined as follows:

$$\boldsymbol{r}_m = [r_{m,1}, r_{m,2}, \cdots, r_{m,N}], \tag{2}$$

where $r_{m,i}$ is the RSSI value measured from the ith AP at the m-th RP.

RSSI vectors can be generated by various devices and by multiple people. If an automatic device is used to construct a visual map, RSSI vectors can be generated at the same time as an image/video is acquired by the device. As well, if multiple measurements are performed at the mth RP, several different versions of $\mathbf{r}_m$ are generated. In this paper, when indicating $\mathbf{r}_m$, the indices corresponding to the measured device, person, time, and version are omitted for convenience of explanation.

The entire area $\boldsymbol{\mathcal{S}}$ for indoor positioning can be expressed as follows:

$$\boldsymbol{\mathcal{S}} = \left\{\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{N_c}\right\}, \tag{3}$$

where $\mathcal{S}_i$ means the ith candidate region. The indices of the candidate regions are encoded for use in training, with one-hot encoding used in this paper. For example, [0, 0, 1, 0, 0, …, 0, 0] represents $\mathcal{S}_3$.

Dataset D_{in} to be used for learning consists of the following data vectors:

$$\boldsymbol{d}_m = \left[\boldsymbol{r}_m, \boldsymbol{v}_{\mathbf{r}_m}\right], \tag{4}$$

where $\mathbf{v}_{\mathbf{r}_m}$ is the one-hot vector for the candidate region corresponding to $\mathbf{r}_m$. When training the first location estimator, the desired output for the input $\mathbf{r}_m$ becomes $\mathbf{v}_{\mathbf{r}_m}$.

Fig. 5 shows the structure of the first location estimator when the DL technique is applied. A neural network is connected to a dense output layer with N_c neurons (one per class) using a softmax activation function to output the estimated probabilities of the candidate regions. Various neural networks can be used for this scheme.

In this paper, we assumed that the kth neuron of the softmax output layer outputs the probability, p_k, that the RSSI

vector given as an input belongs to the kth candidate region, $\mathcal{S}_k$, which is calculated by the following:

$$p_k = \sigma(\mathbf{z})_k = \frac{e^{z_k}}{\sum_{i=1}^{N_c} e^{z_i}} \quad for \;\; k = 1, \cdots, N_c, \tag{5}$$

where $\mathbf{z} = (z_1, \ldots, z_{N_c}) \in \mathbb{R}^{N_c}$ is the output vector of the neural network. Thus, the output vector of the softmax output layer becomes

$$\boldsymbol{p} = \left[p_1, p_2, \cdots, p_{N_c}\right], \tag{6}$$

In the online phase, the first location estimator receives as an input the RSSI vector $\mathbf{r} = [r_1, r_2, \cdots, r_N]$ measured by the user device for positioning at any time, and then outputs $\mathbf{p}$, which represents the estimated probabilities that the current location will belong to the candidate regions. Next, $\mathbf{p}$ is rearranged according to a predetermined priority. Assuming that the rearrangement is simply performed based on the magnitude of the probability value, the reordered vector $\bar{\mathbf{p}}$ can be expressed as follows:

$$\begin{aligned} \bar{\boldsymbol{p}} &= [\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_{Nc}] \\ &= \left[p_{I_1}, p_{I_2}, \cdots, p_{I_{N_c}}\right] \end{aligned} \tag{7}$$

using the index vector $\mathbf{I} = \left[I_1, I_2, \cdots, I_{N_c}\right]$ that satisfies the following inequality:

$$p_{I_1} \geq p_{I_2} \geq \cdots \geq p_{I_{N_c}}, \tag{8}$$

where $I_i \in \{1, 2, \ldots, N_c$.

2) ML-BASED FBL

In general, DL-based localization approaches tend to learn to maximize the probability for one optimal candidate region and to have very small probabilities for other candidate regions. Additionally, the probability for the optimal candidate region generated by the first location estimator is not directly proportional to the accuracy of the actual position estimation. As a result, even if several candidate regions are used in the VBL stage, the additional effects obtained through this are often insufficient. This fact is addressed later in the simulation results presented in Section IV.

In addition, in the case of the SRM, the size of the candidate regions and the intervals between the RPs are non-uniform, each candidate region has different adjacent APs, and the received RSSI values also change over time. Nevertheless, the number of data samples used for training is relatively small. Naturally, it is difficult to train the first location estimator to have very high estimation accuracy for all candidate regions. As a result, the achievable accuracy of the first location estimator is limited, and this limit affects the final accuracy of the whole system.

Through various experiments, we found that increasing the sum of the estimation accuracies of several important candidate regions (i.e., K candidate regions) was more suitable for the final performance of the proposed system than increasing the estimation accuracy of one optimal candidate region. We also found that several ML-based approaches can provide meaningful estimation probabilities to several candidate regions.

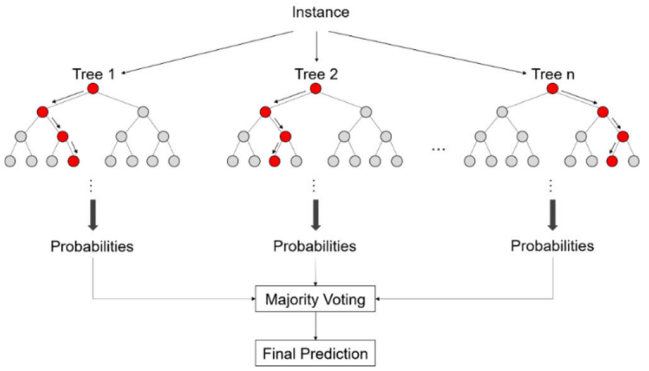

FIGURE 6. General structure of decision tree–based ensemble models.

In this paper, ensemble models [44] based on a decision tree are employed as an ML technique for FBL. Fig. 6 shows the general structure of the decision tree–based ensemble model. The decision tree algorithm has an intuitive classification criterion and has little effect on data scaling or normalization. However, in general, the decision tree algorithm is prone to overfitting by obsessing over exceptions, resulting in poor prediction performance in real data. However, the ensemble model mitigates this problem by maximizing the bias-variance trade-off by combining multiple classifiers to learn various situations. Classification of the ensemble model is performed by combining the predictions of several classifiers used for ensemble learning. The final predicted value of the ensemble model is obtained through a majority vote on the output values of all classifiers.

Representative tree–based ensemble models include Extra trees [45] and LightGBM [46]. The Extra trees model is a variant of the random forest (RF) model but uses all original data differently than RF, which constitutes a dataset for each classifier to learn through bootstrap. Here, bootstrapping means sampling a dataset of the same size as the original data by overlapping a portion of the entire data. Due to this point, the bias of the Extra trees model can be lower than that of RF. In addition, variance can be reduced by randomly selecting a split point. Furthermore, compared to RF, the Extra trees model requires only about 36% of the computing time on average.

LightGBM is a variant of the gradient boosting decision tree (GBDT) method. This is an algorithm using gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) to solve the trade-off between accuracy and efficiency that occurs when the feature dimension is high and the data size is large—a known problem of the existing GBDT. When dealing with a large dataset, GOSS can derive a more accurate estimate of information gain by keeping instances of data with gradients larger than a predefined threshold (i.e., less-trained data instances) and dropping instances of well-trained data at random. When dealing with

a large number of features, the EFB algorithm is a method of bundling exclusive features into a much smaller number of dense features. This makes it possible to effectively avoid unnecessary calculations for features with a value of zero. Through the above features, LightGBM provides better performance than XGBoost in terms of computation speed and memory usage.

An important feature that the proposed Fi-Vi system should have is high performance improvement through low resource consumption. Because of the above advantages, in this paper, Extra trees and LightGBM were used as tree-based ensemble models to be applied to ML-based FBL; in some cases, they showed better performance than DL-based FBL. The structure and operation method of the ML-based FBL are the same as those of the DL-based FBL, except that the neural network with the softmax output layer in the first location estimator is replaced with a decision tree–based ensemble model.

D. VBL STAGE

In the offline phase, a mobile device for building a visual map collects images/videos while traversing the entire space, and then VBL builds a 3D visual map using a visual DB composed of the collected data. The method used to construct the visual map has little effect on the overall implementation of the Fi-Vi system, so it is assumed without loss of generality that a relevant existing method is used.

The visual map corresponding to the collected RSSI is constructed according to the sub-regions, which can be manually predetermined by the collector or automatically configured based on the criteria. For example, when a predetermined time has elapsed after the start of the search or the collector has entered a place identified as a new area, the RSSI and visual map collected up to that point are mapped to the corresponding area (i.e., sub-region). Next, utilizing the information mapped to the sub-regions, VBL performs matching according to a predefined rule between the constructed visual map and the SRM. Through this, the entire visual map is divided into N_c sub-visual maps (sub-VMs), and then each sub-VM is matched one-to-one with one of the N_c candidate regions defined in the FBL stage. In this case, the N_c sub-VMs can partially overlap one another. For example, when the unit of the candidate regions classified through the FBL stage is set to room level, the corresponding sub-VM may also be set to a unit of the same level, as shown in Case A of Fig. 7. Alternatively, each sub-VM can be set as a bundle of candidate regions classified through the FBL stage. That is, one sub-VM can be configured by grouping several adjacent rooms as shown in Case B of Fig. 7.

In the online phase, VBL receives one or more image/video captured by the user device for indoor localization and at the same time receives information about the candidate regions from the FBL—i.e., $\bar{\mathbf{P}}$ and $\mathbf{I}$. Then, the candidate selector chooses K candidates based on a predefined rule. In this paper, it is assumed that selection is made in the order of highest probability. In this case, the following two vectors are

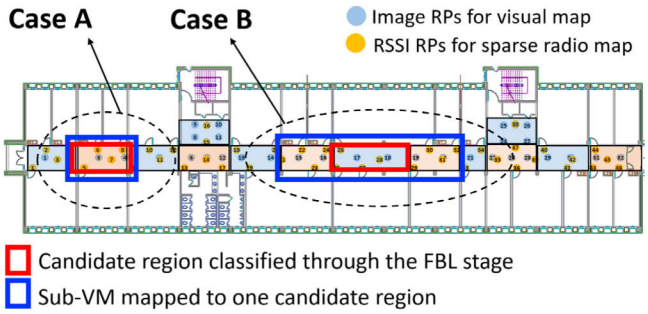

FIGURE 7. Examples of sub-VMs mapped with candidate regions.

passed to the second location estimator:

$$\bar{\boldsymbol{P}}_K = [\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_K], \tag{9}$$

$$\boldsymbol{I}_K = [I_1, I_2, \cdots, I_K] \tag{10}$$

where $1 \leq K \leq N_c$.

After determining the search priority for the K candidates, the second location estimator performs final location estimation through several steps based on the ranking. If it is assumed that each step is performed in the order of the highest probability for one candidate, the i-th step is performed based on the sub-VM corresponding to $\mathcal{S}_{I_i}$, which is the candidate region with the ith highest probability (i.e., $\bar{p}_i$) among the K candidate regions.

If the location estimation fails in a given step, the next step is performed. If the exact location still cannot be estimated after performing the steps for K candidate regions, location estimation is performed again on the remaining sub-VMs, excepting the K sub-VMs. If the VBL fails to localize after all steps, the system returns a failed result. Algorithm 1 is a flow showing the operation of the VBL in the online phase described above.

Algorithm 1 Pseudo-Code of the VBL Stage Process

Input : Given image (*Img*), K, $\bar{\mathbf{P}}$, $\mathbf{I}_K$, $\mathcal{S}$
Output : Location τ
$i \leftarrow 1$
for each step t **do**
  Compare *Img* with data belonging to the sub-VM corresponding to $\mathcal{S}_{I_i}$, which is the candidate region with $\bar{p}_i$.
  Predict τ
  if matching result satisfies the predefined criterion
    return τ, localization success
  else if $i < K$
    $i++$, **continue**
  else
    Compare *Img* with data belonging to all remaining sub-VMs.
    Predict τ
    if matching result satisfies the predefined criterion
      return τ, localization success
      else localization failure, **break**
    //end if
  //end if
//end for
VBL stage **end**

Note that each step can use several candidates together to perform localization. Since the visual-based localization method to be used in the steps is also independent of the proposed schemes, we assume that search algorithms used in existing visual-based indoor positioning methods are utilized in this paper. This assumption includes the process of performing an appropriate image-preprocessing method according to the used search algorithm for the VBL stage.

IV. NUMERICAL RESULTS

In this section, we present the numerical results of the proposed system using FBL with various ML/DL techniques.

A. SIMULATION ENVIRONMENT

For sparse radio mapping, the UJIIndoorLoc dataset described in Section II was used. Our experimental environment for FBL simulation was a Windows server with an Intel i9-10920x 3.60GHz CPU and 64 GB of memory; the image-matching simulation environment was a Windows server with an AMD Ryzen 7 3700X 8-Core Processor 3.59 GHz CPU, 32 GB of memory, and a GeForce GTX 1660 Super GPU. The ML/DL models were trained on Keras 2.5.1 (with Tensorflow 2.5) using Python 3.8.5.

As described above, the candidate region estimated in the FBL stage may have various sizes, and in the case of the UJIIndoorLoc dataset, a building, floor, room, or the like may be a candidate region. In the proposed system, the main purpose of the FBL stage is to select candidate regions that can reduce the computational complexity of the VBL stage and the error of the final location. In this case, to effectively reduce the computational complexity required for the VBL stage, it may be considered that the size of the candidate regions classified by the FBL stage should have a relatively narrow range. Moreover, we have already created a first location estimator model with 85% training accuracy in room-level classification, which is the smallest unit used in this model, for sparse radio mapping through experiments. However, based on several reasons described below, we judged that it is more appropriate to use a floor rather than a room as the classification unit of the candidate regions to be estimated in the FBL stage. That is, in this paper, we use a floor-level classification model for the first location estimator.

First, in general, in the case of each floor present in a building, the area is relatively the same, while the composition shape is diverse. For example, in a building with two floors, one floor may consist of three large classrooms, and the other floor may consist of ten small classrooms. In this case, the area of each room is not constant, so there may be a difference in the number of RPs in each room. If the sparsity between the RPs present in the radio map is very large, a small room may not have any RPs, which can lead to a significant imbalance in the positioning accuracy for each room. If the room level is to be estimated, the dataset with reduced sparseness in the radio map should be collected again. From this point of view, the sparseness of the UJIIndoorLoc dataset used in the experiment can be estimated for the room level, but it is more appropriate to estimate at the floor level. Second, it is difficult to evaluate room-level estimation accuracy using UJIIndoorLoc. The test dataset only provides information about the buildings and floors in which the values were collected and not about the rooms in which the values were collected. For this reason, other studies have attempted to verify the prediction accuracy in rooms by dividing the training dataset. However, using training data as verification data makes it difficult to evaluate performance considering various environmental changes, such as changes in RSSI data collectors, changes in channel environment and time between online and offline phases, and differences in RSSI reception sensitivity from changes in equipment used for measurement. For reference, there are a total of 13 floors included in the UJIIndoorLoc dataset, and thus $N_c = 13$.

TABLE 2. Floor-level accuracy according to various settings for uncollected rssi values.

Converted $RSSI_{missed}$	Training Dataset Accuracy		Test Dataset Accuracy
	Training Accuracy	Validation Accuracy	
100dBm (default)	99.07%	97.15%	79.66%
-50dBm	99.34%	96.50%	75.61%
-100dBm	99.51%	99.45%	92.44%
-110dBme	99.52%	99.55%	93.16%
-500dBm	98.91%	97.70%	84.25%
-1000dBme	98.85%	97.70%	84.34%

B. PREPROCESSING

The data to be used in the ML/DL model requires appropriate preprocessing according to the characteristics of the data before it is used for learning.

1) REPRESENTATION OF MISSING VALUES

As mentioned above, the RSSI input range of UJIIndoorLoc ranges from −104 dBm to 0 dBm, and the RSSI value $RSSI_{missed}$ of the uncollected AP is indicated as +100. This causes nonlinearity in the data. That is, since the larger RSSI value does not mean better reception strength, $RSSI_{missed}$ should be changed to an appropriate value to solve this problem.

Table 2 shows the results of the floor-level accuracy measured after learning using the DNN (DNN-SAE) structure applied with a stacked autoencoder (SAE) after converting $RSSI_{missed}$ to various values. Here, validation accuracy refers to the accuracy of a model that is trained excepting some learning datasets and then validated with the excluded data to verify the generalization performance of the model. Experimental results have shown that no matter what threshold is applied to $RSSI_{missed}$, the learning of a given training dataset is performed well; thus, there is no significant change in training and validation accuracy, but there is a significant difference in the accuracy of the test dataset. In other words,

TABLE 3. Floor-level accuracy according to scaling methods.

Method	Training Dataset Accuracy		Test Dataset Accuracy
	Training Accuracy	Validation Accuracy	
dBm-scale RSSI	99.52%	99.55%	93.16%
power-scale RSSI	98.53%	98.90%	96.04%

values greater than −100 dBm, which provide false information about the RSSI value, are not suitable for use to the threshold. While a value that is too small is semantically appropriate, the valid input range is too large, which reduces the model's accuracy. This problem appears the same even after conversion to power scale data, which is described later. Therefore, the value of $RSSI_{missed}$ should be appropriately selected within a range that does not impair the linearity of the data. In this paper, the value of $RSSI_{missed}$ was set to −110 dBm, which is smaller than the smallest measurement, −104 dBm.

2) FEATURE SCALING AND DATA NORMALIZATION

Some ML/DL algorithms are highly sensitive depending on how raw data is modified. For the model to train reliably, it is necessary to apply an appropriate scaling method according to the features used in the model training. The features of this paper include RSSI collected from each AP. In [41], the authors showed that in the floor-level classification model using UJIIndoorLoc, when each feature was normalized to follow $N(0,1)$, scaling for each RSSI vector was about 6% more accurate than scaling for each AP. Note that $N(0,1)$ indicates the standard normal distribution.

Additionally, the unit of RSSI can affect accuracy [47], [48]. In the case of the dBm scale, the performance of the model may be degraded due to nonlinearity according to the increase or decrease of RSSI. Using a power-scale RSSI is an effective way to use RSSI data.

The dBm-scale RSSI is converted to the power-scale RSSI as follows:

$$RSSI_{(\mathrm{mW})} = 10^{RSSI_{(\mathrm{dBm})}/10} \cdot 1\mathrm{mW}. \tag{11}$$

The power-scale RSSI represents the RSSI value as it is and tends to preserve the linearity of the data. Table 3 shows the experimental results by changing the input to dBm-scale RSSI and power-scale RSSI, respectively, in the model used in the previous experiment. At this time, when experimenting with the power-scale RSSI method, $RSSI_{missed}$ was converted to a power scale and replaced with a value of "0". Experimental results have shown that converting RSSI to a power scale reduces training accuracy by about 1% but improves performance by about 2.88% in terms of test dataset accuracy.

All models used in this paper underwent the following preprocessing based on previous results: a) the conversion of RSSI dBm scale to power scale; b) the replacement of the converted "$RSSI_{missed}$" value with "0"; and c) the normalization of each RSSI vector.

TABLE 4. Floor-level classification performance of various ML/DL models applied to FBL Stage.

Classifier Model	Training Dataset Accuracy		Test Dataset Accuracy	Processing Time (msec)
	Training Accuracy	Validation Accuracy		
DNN-SAE	98.53%	98.90%	96.04%	2.01
1D-CNN	99.81%	99.44%	92.62%	2.71
2D-CNN	98.02%	97.22%	93.34%	5.14
2D-CNN-Conv. Autoencoder	98.48%	97.78%	93.16%	2.48
LightGBM	99.79%	99.47%	93.79%	0.91
Extra trees	99.80%	99.65%	95.13%	0.88
CNNLoc [49]	-	-	96.03%	-
cf. Processing time of k-NN brute force image matcher = 15.73 msec				

C. PERFORMANCE EVALUATION FOR ML/DL MODELS

To select an appropriate ML/DL model to be used for the classification during the FBL stage, the preprocessed data was trained using various ML/DL-based techniques to construct a floor-level classification model and validate its performance. Table 4 presents the experimental results, where the processing time represents the average time required to inference approximately one input sample for each trained classification model. The average time is indicated for relative comparison with the computational complexity for the inference of each model.

In terms of training accuracy, 1D-CNN, Extra trees, and LightGBM showed high accuracy of about 99.8%, with the Extra-trees model showing high accuracy in terms of validation. Except for the above three models, the average accuracy of the training dataset of the other classification models was about 1.46% lower than that of the previous three models. Regarding test dataset accuracy, the DNN-SAE model achieved an accuracy of 96.04%, which was about 1% higher than for the Extra trees with the second highest accuracy. This is about 2.8% higher than the average accuracy of 93.22% for all candidate models excluding Extra trees. Although the DNN-SAE model had a shorter processing time than other DL-based models, it required a processing time of an average of about 2.25-times longer than the ML-based models. The tree-based ensemble model, Extra trees, required the least processing time, and the LightGBM model required the next lowest processing time. To briefly demonstrate the feasibility of the proposed system, we assigned six RPs to one floor of a building at the authors' university and collected 188 images from these RPs. We then conducted an experiment to apply an image-matching algorithm using images associated with RSSI values collected at each RP. For comparison with the average image matching time, the processing time of the *k-NN*-brute force (*k-NN-BF*) image matcher is separately described at the bottom of Table 4. The processing time of the *k-NN-BF* image match was calculated from the average

TABLE 5. Main parameters of the selected models.

DNN-SAE		Light GBM		Extra trees	
Parameter	Values	Parameter	Values	Parameter	Values
Batch size	256	Alpha	0.15	Bootstrap	False
Optimizer	Adam	Max depth	-1	Max depth	None
SAE neurons	256-128-64-128-256	Minimum split gain	0.2	Max samples	None
Learning rate	0.001	Learning rate	0.1	Split criterion	gini
Dropout rate	0.0	No. of leaves	90	Max leaf nodes	None
SAE loss function	MSE	No. of estimators	160	No. of estimators	100
SAE activation function	ReLU	Minimum child weight	0.001	Minimum samples leaf	1
DNN loss function	Categorical cross entropy	Minimum child samples	76	Minimum samples split	2
DNN activation function	ReLU	Feature fraction	1.0	Minimum impurity decrease	0.0
Output layer activation function	Softmax	Lambda	4	Cost complex pruning	0.0

retrieving time of the *k-NN-BF* image matcher ($k = 2$) after extracting the key points and descriptors of the image using the ORB [49] algorithm widely used in vSLAM. The processing time increased linearly as the number of photos increased. For reference, as shown in Table 4, a recent study using the UJIIndoorLoc dataset showed a floor-level accuracy of about 96.03%, which was achieved using a CNN-based indoor localization system with Wi-Fi fingerprints for multi-building and multi-floor localization [48].

Based on the results obtained through the experiments, the DNN-SAE with the highest test dataset accuracy was selected as a model to be applied to the first location estimator in the FBL stage for the performance evaluation of the proposed method. LightGBM and Extra trees with low processing time and high training accuracy were also selected. The main parameters of the selected models are shown in Table 5.

D. SIMULATION RESULTS

1) FBL ACCURACY ACCORDING TO THE NUMBER OF CANDIDATES, *K*

Fig. 8 shows the floor-level estimation accuracy according to the number of candidates, K, for each model used in the experiments. For example, $K = 3$ indicates the accuracy of whether the first three candidate regions listed contain the correct areas when the model's prediction results are listed in order of the most likely candidate regions. Fig. 8(a) shows the floor-level estimation accuracy for the training dataset. Here, $K = 1$ has the highest accuracy of Extra trees at about 99.8%, followed by LightGBM at 99.79%. DNN-SAE was the lowest with an accuracy of about 98.53%. When $K = 2$, the accuracy improved the most for all models. In particular, the accuracy of DNN-SAE increased

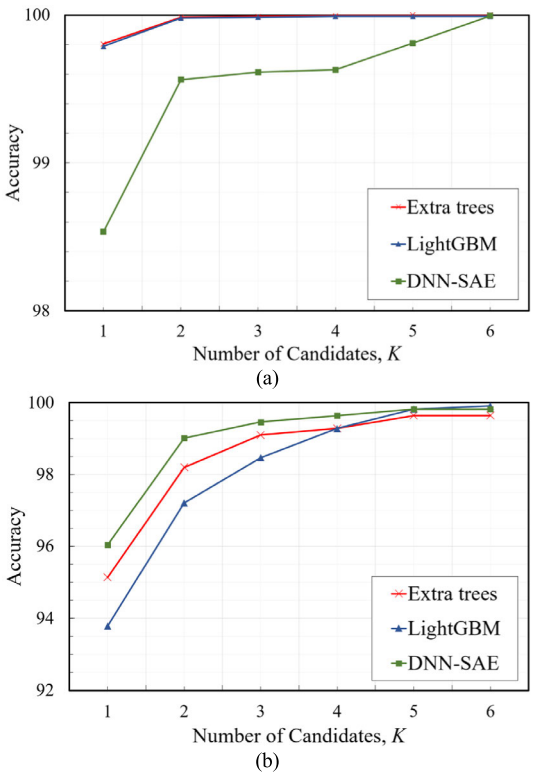

FIGURE 8. Floor-level estimation accuracy of the selected ML/DL models according to the number of candidates, *K*: (a) Training dataset accuracy. (b) Test dataset accuracy.

significantly by about 1.03% compared to the other two models, which improved only by about 0.19% each. Thereafter, as K increased, the improvement in accuracy slowed down. In particular, for LightGBM, the accuracy hardly improved when K was 3 to 6, and the accuracy of DNN-SAE exceeded that of LightGBM. This means that for training datasets, continuing to increase the number of candidates, K, in LightGBM did not help to locate the correct candidate region, and the K regions selected according to the prediction probabilities do not have meaningful information to find the right area. Extra trees achieved 100% accuracy when $K = 5$, and LightGBM and DNN-SAE achieved 100% accuracy when $K = 11$ and $K = 13$, respectively. Fig. 8(b) shows the floor-level estimation accuracy according to K for the test dataset of each model. When $K = 1$, the accuracy of the DNN-SAE model was 96.04%, which was the highest among the three models. However, as K increased, the performance of the other two models improved significantly. When $K = 4$, the accuracy of the LightGBM model was about 99.28%, which was the same as that of the Extra trees, and at $K = 5$, it was about 99.82%, which was the same as that of the DNN-SAE model. When $K = 4$, the accuracy of the LightGBM model was about 99.28%, which was the same as the Extra trees, and when $K = 5$, it was about 99.82%, which was the same as the DNN-SAE model. When $K = 6$, the accuracy of the LightGBM model was about 99.91%, exceeding the accuracy of the other two models. This shows that as K increases,

the performance improvement of each model differs; accordingly, a model with high initial accuracy can exhibit the same or lower accuracy as a model with low initial accuracy.

In addition, when K has a relatively small value, the accuracy improves significantly as K increases. However, if K is greater than a certain value, this increment is greatly reduced. This means that the prediction probability of the model contains information about the correct area, from which we can see that selecting multiple candidate regions in the order of prediction probability from largest to smallest has a significant effect on determining the correct area.

2) ROBUSTNESS UNDER DYNAMIC ENVIRONMENTS

For a variety of reasons, unexpected errors in real-world situations cause the accuracy of trained models to be degraded. Furthermore, training models to have high accuracy requires a lot of time and resources, and sometimes these models may exhibit lower-than-expected accuracy for unlearned data. That is, it is difficult to train a model that ensures the accurate prediction of all unlearned data. Therefore, several studies using fingerprints were conducted assuming a static environment. However, it is difficult to ensure the performance of those results in a dynamic environment because wireless signals are sensitive to changes in the measurement environment, and the RSSI values measured at the same location also depend on changes in the surrounding environment [50]. For the indoor positioning system to exert a significant effect in an actual environment, robustness to dynamic environment changes is required. To verify that the Fi-Vi system is robust even in dynamic environments, we have conducted experiments on noised datasets that virtually reflect irregular errors that may occur in the real world. Noised datasets were created by adding pseudo-noise to each RSSI value of the training or test dataset as follows:

$$\tilde{r}_{m,i} = r_{m,i} + \delta_{m,i}, \tag{12}$$

where $\delta_{m,i}$ is a value generated according $\delta \sim \alpha \cdot 0.1 RSSI_{avg} \cdot N(0, 1)$ and $RSSI_{avg}$ is the mean of the whole collected RSSI. If there was an abnormal case in which $\tilde{r}_{m,i}$ became smaller than $RSSI_{missed}$ due to the addition of noise, the corresponding value was corrected to 0. This experiment was performed to evaluate the estimation accuracy of the pseudo-noised datasets shown by models that have learned via training data.

Fig. 9 shows the floor-level estimation accuracy of the noised datasets according to K. Here, Fig. 9(a), Fig. 9(b), and Fig. 9(c) are the experimental results of the pseudo-noise added to the training dataset (pseudo-noised training dataset), and Fig. 9(d) is the experimental result of the pseudo-noise added to the test dataset (pseudo-noised test dataset) for additional comparison. Fig. 9(a) shows the experimental results assuming that there is a general environmental change ($\alpha = 1$) between the time when training data is collected and localization is requested. Here, when $K = 1$, the dotted arrow indicates the accuracy-decrease width when pseudo-noise is added compared to the accuracy of the model according to the original data. When $K = 1$, the accuracy of the Extra trees model was about 92.62%, which decreased by nearly 7.2%

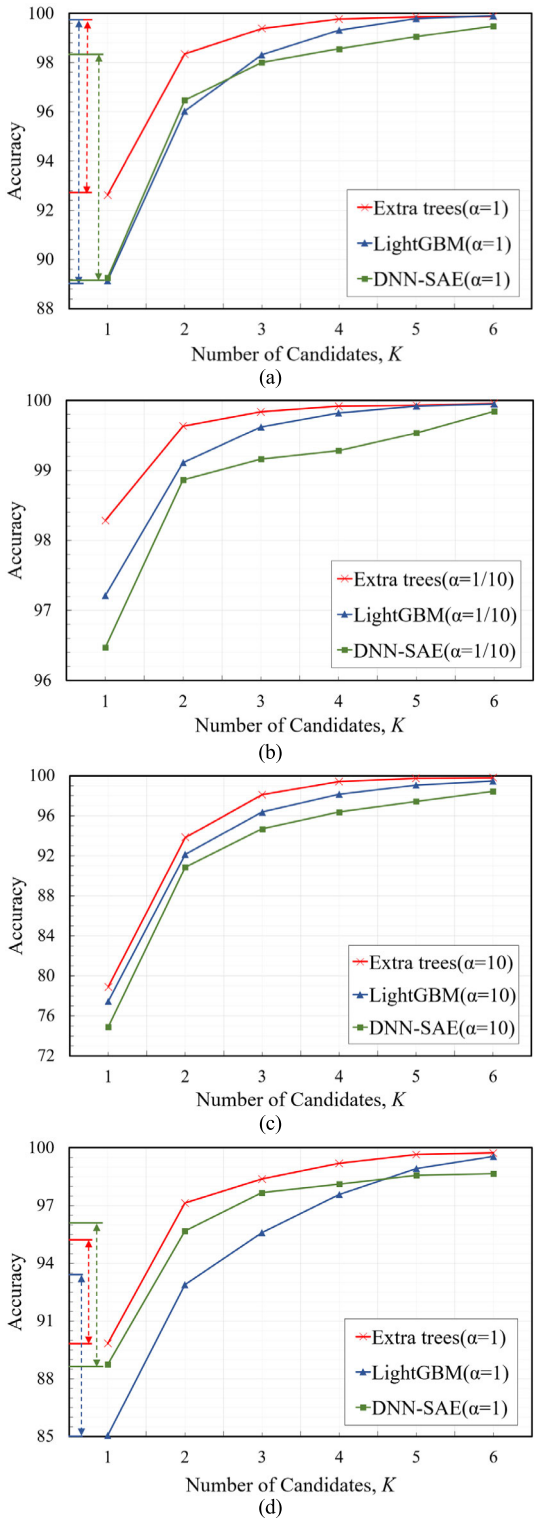

FIGURE 9. Floor-level estimation accuracy of the selected ML/DL models according to the amount of noise: Pseudo-noised training dataset accuracy when (a) $\alpha \mathcal{D}$ 1, (b) $\alpha \mathcal{D}$ 1/10 and (c) $\alpha \mathcal{D}$ 10. (d) Pseudo-noised test dataset accuracy when $\alpha \mathcal{D}$ 1.

compared to the accuracy of the training data. However, when $K = 4$, the accuracy of the Extra trees model is recovered to the accuracy of the training data when $K = 1$. As well, when $K = 1$, the accuracy of the LightGBM model decreased the most by about 10.65% compared to the accuracy of the training data and showed lower accuracy than the other two models. However, when K = 2, the accuracy of LightGBM exceeded that of DNN-SAE, and again at K = 6, the accuracy of the Extra trees exceeded that of the other two models. Fig. 9(b) and Fig. 9(c) show experimental results assuming relatively weak environmental changes ($\alpha = 0.1$) and strong environmental changes ($\alpha = 10$) compared to Fig. 9(a), respectively. As seen in these figures, DNN-SAE showed the highest accuracy in terms of test data, but the decrease in accuracy was most pronounced when pseudo-noise was included; furthermore, the decrease in accuracy was relatively low in the case of Extra trees.

This phenomenon was also evident in the comparison between Fig. 8(b) and Fig. 9(d). In Fig. 9(d), the dotted arrow indicates the decrease in accuracy of each model on the pseudo-noised test dataset compared to the accuracy of each model on the test dataset when $K = 1$. Also, when $K = 1$, the accuracy of the DNN-SAE model was the highest, as shown in Fig. 8(b), but the decrease in accuracy was the largest as indicated in Fig. 9(d). Similarly, in the case of Extra trees, the decrease in accuracy due to pseudo-noise was the lowest. In conclusion, through the above experiments, we confirmed that the Extra trees model was consistently robust to dynamic environments. Also, in all three models, the accuracy rose steeply and then slowed down as K increased. This indicates that the proposed system is robust to a dynamic environment, and it is possible to effectively reduce the computational load of the vision-based positioning method by searching the selected candidate regions according to the prediction probability of the model.

Algorithm 2 Pseudo-Code for Calculating μ

Input: N_s input samples (RSSI vectors), the one-hot encoded desired output $\mathbf{q}_i$ corresponding to i-th input sample, defined by

$$\mathbf{q}_i = [q_{i,1}, q_{i,2}, \ldots, q_{i,N_c}] \tag{13}$$

Output: μ
Initialize $N_{Total} = 0$
for $i = 1$: N_s **do**
  Input i-th RSSI vector into the trained first location estimator
  Find the index vector $\mathbf{I}$ based on $\mathbf{P}$
  Calculating the reordered desired output vector $\bar{\mathbf{q}}_i$ by using $\mathbf{I}$, which is defined as :

$$\bar{\mathbf{q}}_i = [\bar{q}_{i,1}, \bar{q}_{i,2}, \ldots, \bar{q}_{i,N_c}] = [q_{i,I_1}, q_{i,I_2}, \ldots, q_{i,I_{N_c}}] \tag{14}$$

  Initialize $j = 1$
  while($\bar{q}_j == 0$)
    $j++$;
  $N_{Total} += j$
  //end for
  $\mu = N_{Total} / N_s$
  return μ

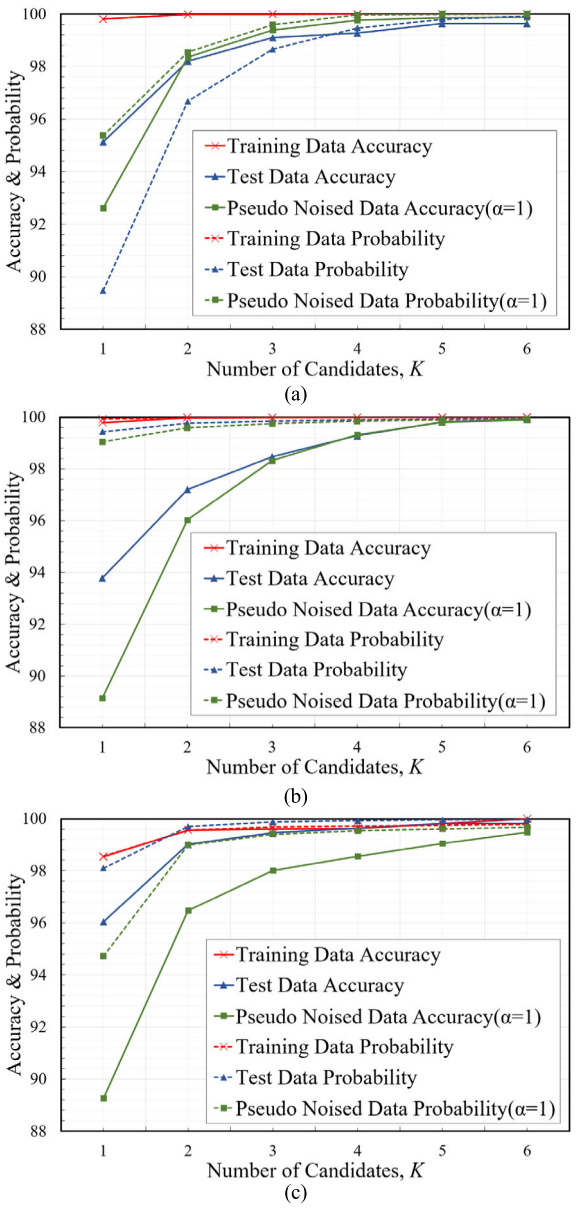

FIGURE 10. Comparison of accuracy and probability according to the number of candidates, K: (a) Extra trees. (b) LightGBM. (c) DNN-SAE.

3) COMPARISON OF ACCURACY AND PROBABILITY

In Fig. 10, the solid lines indicate the estimation accuracy for each model, and the dotted lines indicate the sum of prediction probabilities from the candidate region predicted by the corresponding model with the highest probability to the candidate region predicted with the K-th highest probability (e.g., sum of softmax layer outputs of K candidate regions in DNN-SAE). As seen in Fig. 10, it is difficult to define a certain proportional relationship between the accuracy of each model according to the experiment and prediction probability. Specifically, as shown in Fig. 10(b), the prediction probabilities of the remaining candidate regions except for the candidate region predicted with the highest probability were highly insignificant; thus, comparing the values with

each other may seem meaningless. Nevertheless, choosing the candidate regions in order of highest prediction probability significantly improved accuracy. Intuitively, when a prediction is wrong, we consider the next most likely case as the new answer. This intuitively suggests that candidate regions selected in order of higher predictive probability are more likely to be target regions than randomly selected candidate regions. We considered this to be an applicable concept for well-trained ML/DL models. As we expected, the experimental results showed that selecting candidate regions in highly predictive probability order was a better way to inference target regions than randomly selecting candidate regions.

Through previous experiments, we confirmed that selecting the K candidate region based on the model's predictive probability instead of consuming significant resources or time to increase the model's accuracy can improve the model's accuracy without further modification or training. In general, however, improvements in accuracy decrease as K increases, and computations for the VBL stage increase. Therefore, to build an efficient Fi-Vi architecture, some new metrics are needed to select and evaluate the appropriate number of candidates, which is described in Section V–A.

V. ANALYSIS OF COMPLEXITY AND PERFORMANCE

There are various factors to consider in an indoor positioning system (IPS). Among them, how accurate the location can be estimated is the most important requirement. For instance, in a typical indoor positioning system, accuracy is evaluated by the average Euclidean distance between the estimated position and the actual position. In this paper, the proposed system consists of a combination of two independent positioning systems. The two positioning systems operate hierarchically, but the accuracy of each positioning system affects the others. Furthermore, the two positioning systems derive their results using different kinds of methods—classification and regression, respectively. Therefore, it may not be appropriate to evaluate the performance of the proposed system using only the accuracy of each positioning system. If unsuitable system variables are selected, the system complexity of the proposed scheme may be higher in some situations than in the case where the visual-based positioning method is used alone. Therefore, in this section, we analyze performance and complexity by presenting relevant evaluation metrics and conditions for the proposed scheme to operate efficiently.

A. THE NUMBER OF CANDIDATE REGIONS, K

In general, as K increases, the classification accuracy of the FBL stage increases, but the classification resolution of the FBL stage decreases, and as a result, the computational amount of the VBL stage increases. In order to effectively reduce the overall complexity of the proposed system and improve its performance, it is important to select the appropriate K in comparison with the conventional visual-based positioning method used alone. As shown in Section IV–D, using a model with high accuracy for $K = 1$ does not guarantee a low amount of computation as K increases, since having a high predictive probability for the correct area means having a very low interest for candidate regions that are not predicted to be correct. Therefore, it is necessary to consider the following five metrics when selecting K.

1) AVERAGE OVER THE MINIMUM VALUE OF K NEEDED TO INCLUDE THE CORRECT REGION, μ

If the proposed system works ideally for a given input sample, the candidate selector chooses only the minimum number (K_{opt} in this paper) of candidate regions necessary to capture the correct position. The average of the K_{opt} values obtained for all input samples, μ, corresponds to a statistical value of the number of candidate regions that must be selected on average to ensure the correct region is captured. That is, μ can be helpful in identifying the appropriate minimum value of K and can also be used to estimate the complexity of the proposed system. Algorithm 2 is an example of a method for calculating μ using the output of the first location estimator performed N_s times on N_s RSSI vectors. Here, $\mathbf{q}_i$ in (13) is a one-hot encoded desired output vector in which only an element corresponding to an exact region is 1 and the others are 0. The μ thus obtained can be used directly to determine the minimum number of candidate regions of the proposed system. For example, the rounded value of μ can be used as μ.

Using the reordered desired output vector $\bar{\mathbf{q}}_i$ given in (2), a vector $\boldsymbol{\rho}$ consisting of probabilities that each candidate region selected by the first location estimator in the order of highest probability is an exact region can be expressed as

$$\begin{aligned}\boldsymbol{\rho} &= [\rho_1, \rho_2, \cdots, \rho_{N_c}] \\ &= \frac{1}{N_s}\boldsymbol{h}\end{aligned} \tag{15}$$

where the j-th entry of the $1 \times N_c$ vector $\mathbf{h}$ is defined as

$$h_j = \sum_{i=1}^{N_s} \bar{q}_{i,j} \tag{16}$$

which corresponds to the number of times the first location estimator estimates that the correct region of the input sample has the j-th highest probability among the candidate regions.

2) PROBABILITY-BASED PREDICTION EFFICIENCY, η

Various problems such as data imbalance, over-fitting issues, and differences in the collection environment between training data and test data can degrade the performance of models applied during the FBL stage. Accordingly, when the area predicted by the first location estimator is not the target area, there may be no significant difference in accuracy between selecting additional candidate regions based on the prediction probability of the first location estimator and selecting them randomly. In severe cases, better accuracy may be shown when randomly selecting candidate regions, and thus analysis must be performed to solve this problem.

When the first location estimator selects $i (1 < i < N_c)$ in the order in which the prediction probability is highest for localization N_s times, the probability $P_{prob\_based_i}$ that

$[\mathcal{S}_{I_1}, \mathcal{S}_{I_2}, \cdots, \mathcal{S}_{I_i}]$ contains the target area is defined as

$$P_{prob\_based_i} = \sum_{j=1}^{i} \rho_j. \tag{17}$$

Using (17), the probability-based prediction efficiency, η, of the proposed method, which selects candidate regions to be searched based on a comparison probability when randomly selecting them, can be defined as follows:

$$\eta = \frac{P_{prob\_based_i}}{P_{random_i}} \tag{18}$$

where P_{random_i} denotes the probability that, when selecting i candidate regions, the target area will be included in the area predicted by the first location estimator with the highest probability and randomly selected $i-1$ candidate regions from the remaining $N_c - 1$ candidate regions, as follows:

$$P_{random_i} = \rho_1 + (1-\rho_1) \cdot \left(\frac{i-1}{N_c - 1} \right). \tag{19}$$

The proposed scheme is meaningful when the prediction probability–based candidate region selection method of the model has a greater performance gain than when the candidate region is randomly selected. That is, the inequality $\eta > 1$ must be satisfied.

3) CANDIDATE SELECTION EFFICIENCY, φ

For the selection of one additional candidate region, we can consider how much more gain in accuracy occurs when selecting based on the prediction probability of the first location estimator compared to when selecting randomly. This gain, φ, can be defined as

$$\varphi = \frac{P_{prob\_based_i} - P_{prob\_based_{i-1}}}{P_{random_i} - P_{random_{i-1}}}. \tag{20}$$

Note that (20) satisfies $0 \leq \varphi \leq N_c - 1$.

4) PROBABILITY-BASED ACCURACY EFFICIENCY, υ

Considering φ when selecting the number of candidate regions, υ can be used as an auxiliary metric defined as

$$\upsilon = \frac{P_{prob\_based_i}}{P_{prob\_based_{i-1}} + P_{i-th\_rand}}, \tag{21}$$

where

$$P_{i-th\_rand} = \frac{\sum_{j=i}^{N_c} \rho_j}{N_c - (i-1)}. \tag{22}$$

When selecting the i-th candidate region after selecting i-1 candidate regions based on the probability, υ represents the accuracy ratio between the case of selection based on the prediction probability and the case of selecting randomly from the remaining $N_c - (i-1)$ candidate regions. If φ is less than 1, but υ is greater than 1, we know that selecting candidate regions based on probability is still more accurate than randomly selecting candidate regions—even if the increase in accuracy obtained with increasing K is small. Here, it is preferable that υ for efficiently selecting the number of candidate regions has a value greater than 1.

5) AVERAGE VALUE OF THE NUMBER OF CANDIDATE REGIONS REQUIRED WHEN $\mathcal{S}_{I_1}$ IS NOT THE CORRECT REGION, μ_C

In the proposed system, the advantage of using the candidate region obtained from the first location estimator for visual-based positioning is not only reducing the average computational amount of the entire system, but also alleviating the problem of positioning accuracy degradation due to similar visual positions. However, classifiers are generally trained to achieve high accuracy, and the number of meaningful candidate regions determined through predictive probabilities obtained through the first location estimator in this process is not considered to be significant in the selection process of μ. To reflect this, when the candidate region with the highest prediction probability of the primary position estimator is not the target region, the average number of the candidate regions including the target region, μ_C, is calculated as follows:

$$\mu_c = \sum_{i=2}^{N_c} \frac{h_i \cdot i}{(N_s - h_1)}. \tag{23}$$

The proposed system will operate under various scenarios. In order for the proposed system to operate efficiently, it is desirable to determine K by considering all or parts of η, μ, μ_C, φ, and υ according to on a given situation.

B. COMPUTATIONAL COMPLEXITY

In this paper, we assume that vSLAM using conventional optical cameras is used as a visual-based positioning method. The computational complexity required to perform a single indoor localization using vSLAM is expressed as follows:

$$C_\forall = C_{\mathcal{S}_1} + C_{\mathcal{S}_2} + \cdots + C_{\mathcal{S}_{N_c}} \tag{24}$$

where $C_{\mathcal{S}_i}$ represents the calculation amount required for the visual-based positioning system for the candidate region $\mathcal{S}_i$. For the convenience of explanation, it is assumed that $C_{S_i} = C_\forall / N_c$. As mentioned above, the proposed method is an indoor positioning system in a large area. In general, since each floor of a building is of a similar size, the above assumption does not impair generality. To evaluate the performance of the proposed system, we assume that there are no errors in vSLAM operation.

In terms of complexity, it is desirable that the average computational load C_{Fi_Vi} required to execute a single positioning of the proposed Fi-Vi system is designed to be at least equal to or less than the complexity $C_\forall$ of the positioning system using only the existing visual-based method.

C_{Fi_Vi} is expressed as the sum of the average computation amount C_{FBL} of the FBL stage and the average computation amount C_{VBL} of the VBL stage. Since it was confirmed through the experiment in Section IV–C that C_{FBL} is very small compared to C_{VBL}, the effect of C_{FBL} is negligible, and therefore C_{Fi_Vi} is greatly affected by C_{VBL}. That is, it is preferable that C_{Fi_Vi} is designed to satisfy the following equation for a large area:

$$C_\forall \geq C_{Fi\_Vi} = C_{VBL} + C_{FBL} \approx C_{VBL} + \varepsilon \tag{25}$$

where ε indicates a very small positive real number. Therefore, achieving a small C_{Fi_Vi} depends on how effectively C_{VBL} can be reduced through the efficient design of the FBL stage.

C_{VBL} is calculated as $C_{VBL} = C_m/N_s$, where C_m is the sum of the average computational amount of all samples calculated as follows:

$$C_m = \sum_{i=1}^{N_s} \bar{S}_i, \tag{26}$$

where $\bar{S}_i$ is the expected value of the amount of computation required for the VBL stage of the i-th sample. This can be written as follows:

$$\bar{S}_i = E\left[\bar{S}_i(n_{step})\right] \quad for\ 1 \le n_{step} \le N_c, \tag{27}$$

where $\bar{S}_i\left(n_{step}\right)$ means the average calculation amount of the n_{step}th step of the VBL that operates through the predefined N_{step} step. This value depends on the accuracy of the FBL stage, K, and the search order and the method for candidate regions in the VBL stage.

1) CASE A: $K = 1$

[Scenario 1] This is the case where the accuracy of the first location estimator is 100%, which is the most ideal case. In this case, the second location estimator performs final positioning on the candidate region $\mathcal{S}_{I_1}$. At this time, $\bar{S}_i$ is determined as follows:

$$\overline{S}_i = \frac{C_\forall}{N_c}. \tag{28}$$

[Scenario 2] This scenario is a case where the prediction accuracy of the first location estimator is high and the prediction efficiency η has a relatively low value as K increases. In this case, increasing the number of candidates may not significantly contribute to lowering the expected computational. Therefore, if secondary positioning ($n_{step} = 1$) is performed on $\mathcal{S}_{I_1}$ and the accurate position estimation within the candidate region fails, secondary positioning ($n_{step} = 2$) is performed on $[\mathcal{S}_{I_2}, \mathcal{S}_{I_3}, \cdots, \mathcal{S}_{I_{N_c}}]$, and $\bar{S}_i$ is as follows:

$$\overline{S}_i = \frac{C_\forall}{N_c} \cdot \rho_1 + C_\forall(1 - \rho_1) \tag{29}$$

2) CASE B: $1 < K < N_c$

In general, as N_c increases for a given entire area **S**, the prediction accuracy of the classifier decreases. However, through previous experiments, it was confirmed that the prediction probability of the area not selected as the target area also contains meaningful information in predicting the target area. To increase the prediction accuracy for the target area, $\bar{S}_i$ can be obtained for each of the following two scenarios for the entire K candidate regions selected based on the model's prediction probability.

[Scenario 3] This is the case where $\rho_{I_1}, \ldots, \rho_{I_K}$ are all meaningful to some extent. In this case, to reduce the computational load of the VBL stage, it may be effective to first perform VBL ($n_{step} = 1$) in the K candidate regions and then perform VBL ($n_{step} = 2$) in all remaining candidate regions when the location estimation fails. This is an extended form of Scenario 2, where $\bar{S}_i$ is as follows:

$$\overline{S}_i = K \cdot \frac{C_\forall}{N_c} \cdot \sum_{i=1}^{K} \rho_i + C_\forall \left(1 - \sum_{i=1}^{K} \rho_i\right). \tag{30}$$

[Scenario 4] This is the case where the difference in the values of $\rho_{I_1}, \ldots, \rho_{I_K}$ is relatively large. In this case, sequentially searching for $[\mathcal{S}_{I_1}, \mathcal{S}_{I_2}, \cdots, \mathcal{S}_{I_K}]$ may be effective in reducing the amount of calculation necessary. That is, the VBL stage is sequentially performed on the K candidate regions selected in the order in of high probability throughout the FBL stage. If all position estimation at $[\mathcal{S}_{I_1}, \mathcal{S}_{I_2}, \cdots, \mathcal{S}_{I_K}]$ fails, the VBL stage is collectively performed on $[\mathcal{S}_{I_{K+1}}, \mathcal{S}_{I_{K+2}}, \cdots, \mathcal{S}_{I_{N_c}}]$, where $\bar{S}_i$ is as follows:

$$\bar{S}_i = \frac{C_\forall}{N_c} \cdot \sum_{j=1}^{K} \left(j \cdot \sum_{i=1}^{j} \rho_i\right) + C_\forall \cdot \left(1 - \sum_{i=1}^{K} \rho_i\right). \tag{31}$$

3) CASE C: $K = N_c$

[Scenario 5] This is the case where the difference in the values of $\rho_{I_1}, \ldots, \rho_{I_{N_c}}$ is small. In this case, sequentially performing the VBL stage for all candidate regions $[\mathrm{S}_{I_1}, \mathrm{S}_{I_2}, \cdots, \mathrm{S}_{I_{N_c}}]$ may be effective to reduce the amount of calculation needed for positioning. This is an extended form of Scenario 4, where the FBL stage behaves like a kind of scheduler that allows all sub-VMs in the VBL to be explored in a sequence, where $\bar{S}_i$ is as follows:

$$\bar{S}_i = \frac{C_\forall}{N_c} \cdot \sum_{j=1}^{K} \left(j \cdot \sum_{i=1}^{j} \rho_i\right). \tag{32}$$

C. PROCESSING TIME

The processing time of a location estimation system is an important issue in providing real-time services. Assuming that the average processing time when using only the existing visual-based indoor positioning is $T_\forall$, the average processing time of the proposed system, T_{Fi_Vi}, must satisfy the following inequality:

$$T_\forall \ge T_{Fi\_Vi} = T_p + T_{DB} + T_{tr} \tag{33}$$

where T_p is the average processing time of the first location estimator and the second location estimator, T_{DB} is the average processing time it takes to set up the sub-VM corresponding to the result of the FBL stage, and T_{tr} is the time required for data transfer when offloading between the localization device and the server is performed. As shown in the Section IV–C, we confirmed that the value of T_p is greatly affected by the processing time of the second location estimator compared to the first location estimator. T_{DB} and T_{tr} depend on the implementation environment.

D. ANALYSIS OF EXPERIMENTAL RESULTS BASED ON EVALUATION METRICS

In this sub-section, we analyze the experimental results using the various metrics described above.

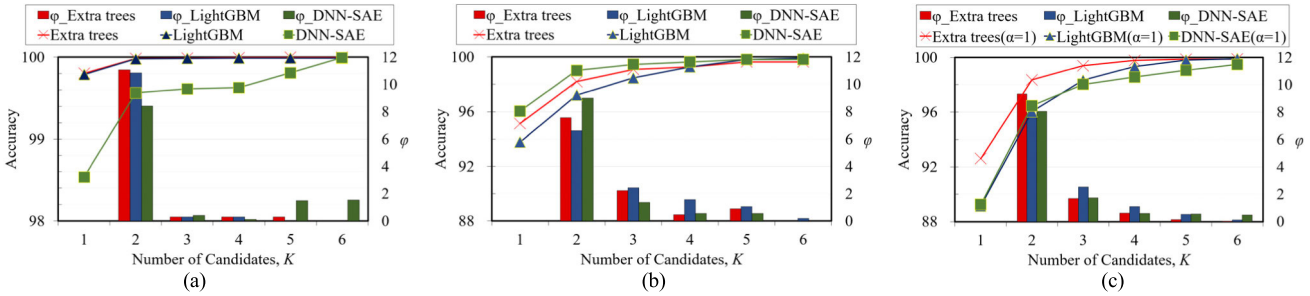

FIGURE 11. Accuracy and φ according to K: (a) Training dataset. (b) Test dataset. (c) Pseudo-noised dataset.

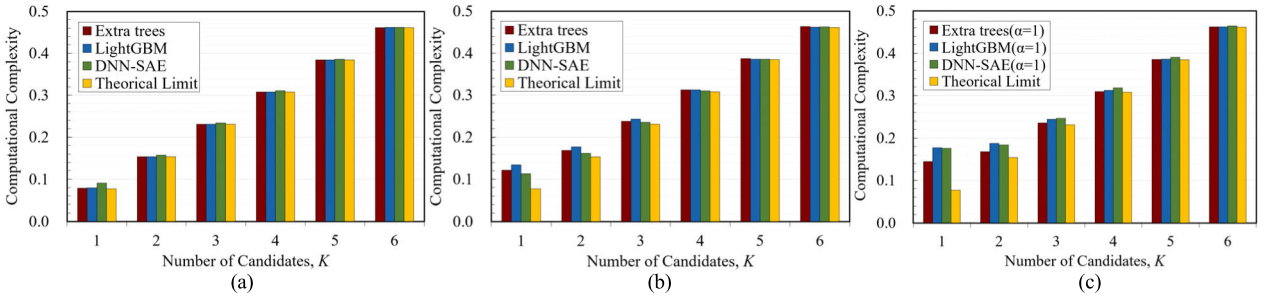

FIGURE 12. Computational complexity according to K in the case of Scenario 3: (a) Training dataset. (b) Test dataset. (c) Pseudo-noised dataset.

1) ACCURACY AND COMPUTATIONAL COMPLEXITY

Fig. 11 and Fig. 12 show the experimental results according to Extra trees, LightGBM, and DNN-SAE for the training dataset, test dataset, and pseudo-noised dataset, respectively, in the performance metrics proposed above for system complexity evaluation. The experiment assumed Scenario 3. The line graph in Fig. 11 shows the accuracy of each model, and the bar graph in Fig. 11 shows φ according to K. As indicated in Fig. 11(a), the graph already shows very high accuracy in all three models for the training dataset at $K = 1$. In particular, the Extra trees and LightGBM achieved almost 100% accuracy at $K = 2$, and φ also had a value close to $N_c - 1$. However, DNN-SAE had the lowest accuracy among the three models at $K = 1$, converging on the accuracy of the other two models at $K = 6$. In addition, as shown in Fig. 11(b), the distribution of φ according to K confirmed that the accuracy of LightGBM linearly improves as K increases; the experimental results confirmed that the accuracy of LightGBM increases most gently. LightGBM achieved the highest accuracy among the three. Fig. 11(c) shows the accuracy of each model and the distribution of φ for the pseudo-noised dataset according to K, which shows that the accuracy of Extra trees at $K = 1$ is the highest at 92.62%, especially at $K = 3$, resulting in a significant improvement in accuracy with a small K number. Fig. 11(a)–(c) show an increase in accuracy over φ. As φ approaches $N_c - 1$, the accuracy increases significantly as K increases, and when φ is less than 1, it can be seen that the accuracy increase greatly slows down. This shows that when a candidate region is selected based on the predicted probability at $\varphi > 1$, the accuracy increase is greater than when a candidate region is randomly selected, and when $\varphi < 1$, the accuracy increase is less when selecting a candidate region based on the predicted probability at $\varphi < 1$.

TABLE 6. μ and μ_C for Extra trees, LightGBM, and DNN-SAE.

	Extra trees		LightGBM		DNN-SAE	
	μ	μ_C	μ	μ_C	μ	μ_C
Training data	1.00	2.15	1.00	2.40	1.03	2.97
Test data	1.10	3.00	1.12	2.93	1.06	2.59
Pseudo-noised data (α=1)	1.10	2.41	1.16	2.62	1.20	2.88

Fig. 12 shows the computational complexity of the entire system as K increases in Scenario 3, assuming $C_\forall$ as 1. The yellow bar graph represents the C_{Limit}—the lower limit of computational complexity that the entire system can achieve according to K. In Scenario 3, as K increases, the C_{Limit} increases by $C_\forall/N_c$ and converges on the C_{Limit} at specific K in the experimental results according to the three models. The increase in the number of candidate regions helps alleviate the error of the VBL stage for areas consisting of visually similar spaces, so although the gain in terms of the computational load may be small as K increases, the number of appropriate candidate regions, K, helps improve the overall system accuracy. Therefore, to optimize the proposed system, it is necessary to select the appropriate number of candidate regions, K, taking into account the trade-off between accuracy and computational complexity.

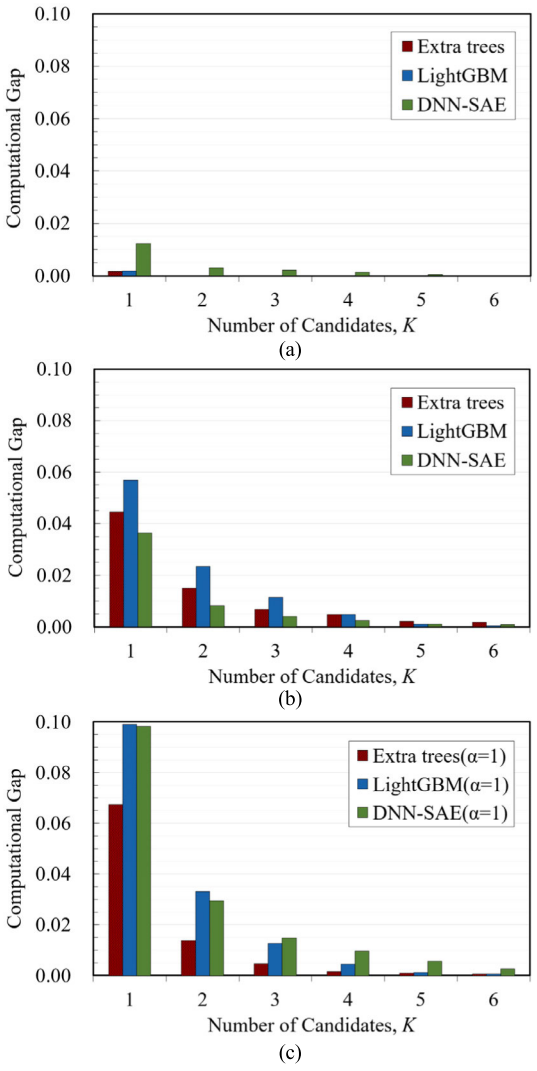

FIGURE 13. Computational gap according to K: (a) Training dataset. (b) Test dataset. (c) Pseudo-noised dataset.

Table 6 shows the results of calculating μ and μ_C when Extra trees, LightGBM, and DNN-SAE are used for the training dataset, test dataset, and pseudo-noised dataset, respectively, from the previously proposed performance metrics. When K was selected by rounding the obtained μ and μ_C values, the smallest calculation amount was required when $K = \mu$ in all three models for each dataset in Fig. 12, and there was a trend of converging on C_{Limit} at $K = \mu_C$. After this, it is clear that both the increase in accuracy and the decrease in calculation cost slow significantly. Therefore, it is meaningful to estimate the lower and upper values of K using μ and μ_C.

2) COMPUTATIONAL GAP

Assuming $C_\forall$ as 1, $C_m - C_{Limit}$, which is the difference between the average computational amount of all samples and the lower limit of computational complexity that the entire system can achieve, is defined in this paper as a computational gap. Fig. 13 shows the corresponding values according to K for each model in Scenario 3. A large calculation gap means that the information obtained (i.e., the increase in accuracy for the target area) is large compared to the amount of calculation that increases as K increases, which can be confirmed by comparing Fig. 12 and Fig. 13. This is particularly emphatic in the pseudo-noised dataset of Fig. 13(a)–(c), where the computational complexity increase is insignificant as K increases from 1 to 2 while the computational gap decreases significantly. After this, the computational gap according to the increase or decrease of K is greatly reduced, which means that as the number of candidate regions increases, the rate of increase in prediction accuracy of the candidate region relative to the increasing calculation amount decreases.

E. SUMMARY AND DISCUSSION

Table 7 summarizes the experimental results of Scenario 3 based on the amount of computation and level of accuracy. In Table 7, *gap* refers to the computational gap and *acc* refers to the accuracy of the FBL stage according to K. Since the computational gap has a small value of 1 or less, it is expressed by multiplying the computational gap by 100 for the convenience of expression, and all results except *gap* were rounded to the third decimal place. $R_{C_\forall}$ means the ratio of the average amount of computation of the proposed system to $C_\forall$ according to K. The values in bold in Table 7 are $R_{C_\forall}$, *acc*, φ, and *gap* according to K when each model achieves accuracy of 99% or higher.

If the accuracy of the FBL stage is required to be greater than or equal to 99% in an environment where test data is collected, the use of DNN-SAE is considered, as it requires a low amount of computation and provides high accuracy for the test data. On the other hand, when considering a dynamic environment, the Extra trees showed better results. As mentioned above, an appropriate model can be selected and used according to the requirements of the system to be applied.

As seen in Table 7, there is a trade-off between the necessary amount of calculation and the accuracy. Among them, the Extra trees showed a low amount of necessary computation and high accuracy for all types kinds of data used in the experiment. In particular, as confirmed in Fig. 8 and Fig. 9, the Extra trees showed a robust performance in all cases, including pseudo-noise. In addition, even with a relatively small number of candidate regions, such as $K = 2$ or $K = 3$, high accuracy of 98% or higher and a low computational gap were shown. As a result of the experiment, the amount of computation of the entire system with Extra trees applied required only 7.85% for training data, 12.14% for test data, and 14.36% for pseudo-noised data compared to conventional existing visual-based positioning systems. The accuracy of the FBL stage was 99.80%, 95.14%, and 92.62%, respectively. When selecting the optimal number of candidate regions in the proposed system, K, which minimizes the overall amount of computation, regardless of the accuracy of the FBL stage, may be considered K optimized for the

TABLE 7. Summary of simulation results in the case of scenario 3.

	K	Training dataset				Test dataset				Pseudo-noised dataset ($\alpha = 1$)			
		$R_{C_\forall}$	*acc*	φ	*gap*	$R_{C_\forall}$	*acc*	φ	*gap*	$R_{C_\forall}$	*acc*	φ	*gap*
Extra trees	1	**0.08**	**99.80**	**-**	**0.17**	0.12	95.14	-	4.46	0.14	92.62	-	6.74
	2	0.15	99.98	11.08	0.01	0.17	98.20	7.56	1.50	0.17	98.35	9.32	1.38
	3	0.23	99.99	0.31	0.01	**0.24**	**99.10**	**2.22**	**0.68**	**0.24**	**99.39**	**1.69**	**0.46**
	4	0.31	99.99	0.31	0.00	0.31	99.28	0.44	0.48	0.31	99.77	0.62	0.15
	5	0.38	100.0	0.31	0.00	0.39	99.64	0.89	0.21	0.39	99.86	0.15	0.08
	6	0.46	100.0	0.00	0.00	0.46	99.64	0.00	0.18	0.46	99.88	0.03	0.06
Light GBM	1	**0.08**	**99.79**	**-**	**0.18**	0.13	93.79	-	5.70	0.18	89.15		9.91
	2	0.15	99.98	10.86	0.01	0.18	97.21	6.61	2.33	0.19	96.03	7.61	3.30
	3	0.23	99.98	0.29	0.01	0.24	98.47	2.43	1.15	0.24	98.32	2.53	1.26
	4	0.31	99.99	0.29	0.00	**0.31**	**99.28**	**1.57**	**0.48**	**0.31**	**99.32**	**1.10**	**0.46**
	5	0.38	99.99	0.00	0.00	0.39	99.82	1.04	0.11	0.39	99.79	0.53	0.12
	6	0.46	99.99	0.00	0.00	0.46	99.91	0.17	0.05	0.46	99.90	0.12	0.05
DNN-SAE	1	0.09	98.53	-	1.24	0.11	96.04	-	3.63	0.18	89.26	-	9.82
	2	**0.16**	**99.56**	**8.42**	**0.30**	**0.16**	**99.01**	**9.00**	**0.83**	0.18	96.46	8.05	2.95
	3	0.23	99.61	0.41	0.21	0.23	99.46	1.36	0.41	0.25	98.01	1.73	1.49
	4	0.31	99.63	0.12	0.14	0.31	99.64	0.55	0.24	0.32	98.55	0.61	0.97
	5	0.39	99.81	1.48	0.04	0.39	99.82	0.55	0.11	**0.39**	**99.05**	**0.55**	**0.56**
	6	0.46	99.99	1.52	0.00	0.46	99.82	0.00	0.09	0.46	99.48	0.48	0.26
Average of bolded values		0.107	99.37	-	0.217	0.237	99.13	-	0.663	0.313	99.25	-	0.493

entire system. However, there is scope for degrading accuracy improvement in visual-based positioning, which is one of the main benefits of the proposed system, if K is selected considering only the computational cost without considering the accuracy of the FBL stage. In LightGBM and DNN-SAE in Table 7, looking at $R_{C_\forall}$ and *acc* in pseudo-noised data, respectively, at $K = 1$ and $K = 2$, although $R_{C_\forall}$ shows a very small difference, *acc* shows a performance gain of 6.88% and 7.2%, respectively. Therefore, when considering K to optimize the entire system, $R_{C_\forall}$ and *acc*, i.e., *gap* and φ, should be considered together.

The results summarized in Table 7 show that selecting K based on the prediction probability when $\varphi \geq 1$ is still a better method than randomly selecting candidate regions. Furthermore, this indicates that even if the increment of $C_\forall$ approaches $C_\forall / N_c$ as K increases, φ is greater than 1, and increasing K at $gap \geq 0$ still ensures a lower amount of required computation compared to conventional visual-based positioning methods and achieves higher accuracy in the FBL stage. If the enter area, $\boldsymbol{\mathcal{S}}$, is composed of visually distinctive characteristic spaces, sacrificing the computation to ensure high accuracy in the FBL stage will not be considered important. However, considering the application of the proposed system to a visually similarly configured space, it can be crucial to obtain high accuracy in the FBL stage even if the computational load of the entire system is increased. In this case, the unit of the candidate region classified through the FBL stage should be considered, as should the unit of space classified through the FBL stage. For example, when classifying by floor level during the FBL stage, it can be assumed that the FBL stage achieves high accuracy by selecting a large number of floors as candidate regions in visually similarly constructed spaces. At this time, it cannot be guaranteed that the accuracy of the VBL stage will be improved because there may be many visually similarly constructed spaces within the pre-classified regions during the FBL stage.

Therefore, to optimize the proposed system, it is necessary to consider the shape or characteristics of the applied space. As a result, even when the entire area, $\boldsymbol{\mathcal{S}}$, has the same area, the size of each candidate region may have to be changed according to the configuration or shape of the space, and the number N_c of candidate regions classifying the entire area suitable thereto will also be changed. For this purpose, the number of appropriate RPs that must be collected to configure a SRM should also be adjusted.

In this paper, our main aim was to validate the proposed Fi-Vi system via public data. Unfortunately, since there is no visual map of the space where the UJIIndoorLoc dataset was collected, we had no choice but to indirectly compare the performance improvement in terms of computational load and the accuracy of the proposed system. Therefore, we constructed new datasets and built their corresponding visual maps to further prove the feasibility of the proposed Fi-Vi system. Data were collected over six floors of two buildings at the authors' university, with Fig. 7 showing one floor of a building for which data was collected. In this process,

TABLE 8. Performance of the Fi-Vi system on the newly constructed datasets for feasibility test.

	Accuracy of 1st location estimator			Accuracy of Fi-Vi	Mapping between candidate regions and sub-VMs	Ratio of the Sub-VMs used for positioning to the entire visual map	Computational load / Calculation time
	Training (995)	Validation (332)	Test (332)				
Only visual	-	-	-	100%	-	100%	100% / 12.74 sec
C1	97.89%	82.23%	85.84%	100%	Case A in Fig. 7	15.37%	14.12% / 1.80 sec
					Case B in Fig. 7	5.40%	4.94% / 0.63 sec
C2	99.99%	16.87%	98.79%	100%	Only estimated floor	18.06%	16.64% / 2.12 sec
					All floors in the building with the estimated region	17.51%	16.09% / 2.05 sec

reference points for collecting images to build a visual map (image RPs for visual map in Fig. 7) and reference points for collecting RSSI values to build an SRM (RSSI RPs for sparse wireless map in Fig. 7) were separately defined for the generalizability of the experiment. The number of image RPs is 196, and the number of images collected here is 12,390. The total number of RSSI RPs is 244, and the number of APs from which RSSI values were collected is 817. The collected RSSI values and images are assigned a building ID, floor ID, room ID, and RP ID according to the location in which they were collected and stored in the RSSI DB and Visual DB, respectively. This facilitates the mapping between candidate regions in the FBL stage and sub-VMs in the VBL stage. If the collector moves floors or moves to adjacent buildings, the collection manager simply needs to specify the floor or building number of the table in the DB in which the RSSI and images collected through the collector will be stored. This is not a difficult task.

Table 8 shows the performance of the proposed Fi-Vi system on the newly constructed datasets, where C1 and C2 correspond to the cases using the room level (one or two rooms per each candidate region) and the floor level as the unit of the candidate regions classified through the FBL stage, respectively. ''Only visual'' can be considered to be a case wherein an existing visual-based indoor positioning method is used alone. The number of candidates is one (i.e., $K = 1$). For image matching, the VGG-16 CNN model [51] was applied for feature extraction, and the similarity between features was measured using the Euclidean distance. The collected images were resized to 224×224 for VGG-16. In Case B of C1, the computational load required for the Fi-Vi system was only 4.94% compared to the case of using the visual-based localization technique alone. These results proved that the proposed Fi-Vi system could work well in real indoor environments.

VI. CONCLUSION

In this paper, we propose a large-area indoor localization scheme that can be efficiently used in unreliable GPS environments. The proposed method complements the shortcomings of existing methods by hierarchically combining ML/DL-based wireless fingerprinting and visual-based positioning techniques to improve positioning accuracy and performance. At the same time, the proposed system demonstrates lower computational complexity and faster processing time compared to the use of conventional visual-based indoor positioning systems alone. Since the wireless fingerprinting part of this hybrid scheme uses a SRM, it has a very low implementation complexity compared to the conventional fingerprinting method.

In this paper, we have demonstrated through experiments that the proposed method can effectively reduce the computational complexity of the conventional visual-based indoor positioning and that the fingerprinting method can effectively overcome problems due to various causes (e.g., differences in RSSI reception sensitivity due to problems caused by channel environment or time changes between online and offline stages and differences in equipment used for measurement). Furthermore, the experimental results show that the candidate regions selected based on the predictive probabilities of the ML/DL model can improve accuracy without further modification of or learning in the model and significantly improve performance over random selection. On the other hand, quantitative metrics for accuracy and computation were presented, respectively, and the performance of the proposed system was evaluated in detail in terms of accuracy and computation. These metrics provide useful guidelines for determining how the appropriate number of candidates K and VBL behave under various scenarios in which the proposed scheme can be used. Finally, the proposed method in this study is not limited to Wi-Fi–based fingerprinting but can be extended to various fingerprints, such as geomagnetic fingerprints; furthermore, it can also be effectively applied to multi-sensor-based indoor positioning systems that require a large amount of computation. In future studies, images and RSSI for large indoor areas will be collected simultaneously to construct new datasets suitable for the performance evaluation and structural improvement of the proposed system. In doing so, we will examine the differences between the actual environment and the theoretical results more closely and verify the effectiveness of the proposed method compared to the existing methods applied to buildings with many visually similar structures. Specifically, additional research is needed to determine the suitable sparseness of the radio map for the estimated area level (building, floor, room, etc.) of the first location estimator and the appropriate ML/DL model

selection methodology for learning the sparseness, as well as to extract the sub-dataset for the determined sparseness from the original dataset. Furthermore, the images that make up the space can change over time. In addition, when the device performing the localization and the environment in which the pre-generated visual map is configured (e.g., change of illuminance, movement of an object, etc.) are different, a position error may occur. Therefore, it is necessary to evaluate the robustness of the algorithms applied to the VBL stage under such circumstances. We also plan to study the trade-off between the performance and complexity of the entire system according to the determined estimated area level.

REFERENCES

[1] W. Wu, L. Ma, and B. Wang, "A novel monocular SLAM algorithm for high real-time based on Kalman filter," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 1667–1672.

[2] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[3] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, 1st Quart., 2009.

[4] L. Chen, K. Yang, and X. Wang, "Robust cooperative Wi-Fi fingerprint-based indoor localization," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1406–1417, Dec. 2016.

[5] Z. Zhong, Z. Tang, X. Li, T. Yuan, Y. Yang, M. Wei, Y. Zhang, R. Sheng, N. Grant, C. Ling, X. Huan, K. S. Kim, and S. Lee, "XJTLUIndoorLoc: A new fingerprinting database for indoor localization and trajectory estimation based on Wi-Fi RSS and geomagnetic field," in *Proc. 6th Int. Symp. Comput. Netw. Workshops (CANDARW)*, Nov. 2018, pp. 228–234.

[6] Q. Pu, M. Zhou, F. Zhang, and Z. Tian, "Group power constraint based Wi-Fi access point optimization for indoor positioning," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 5, pp. 1951–1972, May 2018.

[7] S. Liu, Y. Jiang, and A. Striegel, "Face-to-face proximity estimationusing Bluetooth on smartphones," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 811–823, Apr. 2014.

[8] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does BTLE measure up against WiFi? A comparison of indoor location performance," in *Proc. 20th Eur. Wireless Conf.*, 2014, pp. 1–6.

[9] D. Sun, E. Wei, Z. Ma, C. Wu, and S. Xu, "Optimized CNNs to indoor localization through BLE sensors using improved PSO," *Sensors*, vol. 21, no. 6, p. 1995, Mar. 2021.

[10] Y. Yao, Q. Bao, Q. Han, R. Yao, X. Xu, and J. Yan, "BtPDR: Bluetooth and PDR-based indoor fusion localization using smartphones," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 8, pp. 3657–3682, Aug. 2018.

[11] M. Liu, H. Wang, Y. Yang, Y. Zhang, L. Ma, and N. Wang, "RFID 3-D indoor localization for tag and tag-free target based on interference," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 3718–3732, Oct. 2019.

[12] L. M. Ni, Y. Liu, Y. Cho Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2003, pp. 407–415.

[13] J. Wang and D. Katabi, "Dude, where's my card?: RFID positioning that works with multipath and non-line of sight," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, Aug. 2013, pp. 51–62.

[14] A. Athalye, V. Savic, M. Bolic, and P. M. Djuric, "Novel semi-passive RFID system for indoor localization," *IEEE Sensors J.*, vol. 13, no. 2, pp. 528–537, Feb. 2013.

[15] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2011, pp. 141–154.

[16] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lv, "MaLoc: A practical magnetic fingerprinting approach to indoor localization using smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2014, pp. 243–253.

[17] S. Venkatesh and R. M. Buehrer, "Non-line-of-sight identification in ultra-wideband systems based on received signal statistics," *IET Microw., Antennas Propag.*, vol. 1, no. 6, pp. 1120–1130, 2007.

[18] P. Krapež, M. Vidmar, and M. Munih, "Distance measurements in UWB-radio localization systems corrected with a feedforward neural network model," *Sensors*, vol. 21, no. 7, p. 2294, Mar. 2021.

[19] L. Gogolak, S. Pletl, and D. Kukolj, "Indoor fingerprint localization in WSN environment based on neural network," in *Proc. IEEE 9th Int. Symp. Intell. Syst. Informat.*, Sep. 2011, pp. 293–296.

[20] M. Zhou, Y. Tang, Z. Tian, L. Xie, and W. Nie, "Robust neighborhood graphing for semi-supervised indoor localization with light-loaded location fingerprinting," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3378–3387, Oct. 2017.

[21] S. Subedi and J.-Y. Pyun, "Practical fingerprinting localization for indoor positioning system by using beacons," *J. Sensors*, vol. 2017, pp. 1–16, 2017.

[22] M. Ficco, C. Esposito, and A. Napolitano, "Calibrating indoor positioning systems with low efforts," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 737–751, Apr. 2014.

[23] V. Moghtadaiee and A. G. Dempster, "Design protocol and performance analysis of indoor fingerprinting positioning systems," *Phys. Commun.*, vol. 13, pp. 17–30, Dec. 2014.

[24] S. He, W. Lin, and S.-H. G. Chan, "Indoor localization and automatic fingerprint update with altered AP signals," *IEEE Trans. Mobile Comput.*, vol. 16, no. 7, pp. 1897–1910, Jul. 2017.

[25] J. Xue, J. Liu, M. Sheng, Y. Shi, and J. Li, "A WiFi fingerprint based high-adaptability indoor localization via machine learning," *China Commun.*, vol. 17, no. 7, pp. 247–259, Jul. 2020.

[26] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of RSS fingerprinting based indoor localization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2479–2487.

[27] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: A comparative study," in *Proc. 1st Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw., IEEE SECON*, Oct. 2004, pp. 406–414.

[28] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *Proc. IEEE INFOCOM*, vol. 2, Mar. 2004, pp. 1012–1022.

[29] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 1st Quart., 2015.

[30] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular SLAM for augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Sep. 2016, pp. 1–10.

[31] J.-C. Piao and S.-D. Kim, "Real-time visual–inertial SLAM based on adaptive keyframe selection for mobile AR applications," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2827–2836, Nov. 2019.

[32] H. Durrant-Whyte, D. Rye, and E. Nebot, "Localization of autonomous guided vehicles," in *Proc. Robot. Res.*, 1996, pp. 613–625.

[33] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.

[34] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[35] H. Sol, J. W. Kam, and S. S. Hwang, "An evaluation system to determine the completeness of a space map obtained by visual SLAM," *J. Korea Multimedia Soc.*, vol. 22, no. 4, pp. 417–423, 2019.

[36] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[37] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[38] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4503–4508.

[39] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Comput. Netw.*, vol. 47, no. 6, pp. 825–845, Apr. 2005.

[40] P. Sthapit, H.-S. Gang, and J.-Y. Pyun, "Bluetooth based indoor positioning using machine learning algorithms," in *Proc. IEEE Int. Conf. Consum. Electron. Asia (ICCE-Asia)*, Jun. 2018, pp. 206–212.

[41] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with WiFi fingerprints using deep learning," in *Proc. Int. Conf. Automat.* Cham, Switzerland: Springer, 2017, pp. 575–584.

[42] R. Wang, Z. Li, H. Luo, F. Zhao, W. Shao, and Q. Wang, "A robust Wi-Fi fingerprint positioning algorithm using stacked denoising autoencoder and multi-layer perceptron," *Remote Sens.*, vol. 11, no. 11, p. 1293, May 2019.
[43] J. Torres-Sospedra, R. Montoliu, A. Martinez-Uso, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2014, pp. 261–270.
[44] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer, 2000, pp. 1–15.
[45] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Mar. 2006.
[46] K. Guolin, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–9.
[47] J. Torres-Sospedra, R. Montoliu, S. Trilles, Ó. Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems," *Exp. Syst. Appl.*, vol. 42, no. 23, pp. 9263–9278, Dec. 2015.
[48] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with WiFi fingerprinting," *IEEE Access*, vol. 7, pp. 110698–110709, 2019, doi: 10.1109/ACCESS.2019.2933921.
[49] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[50] Y. Kim, H. Shin, Y. Chon, and H. Cha, "Smartphone-based Wi-Fi tracking system exploiting the RSS peak to overcome the RSS variance problem," *Pervasive Mobile Comput.*, vol. 9, no. 3, pp. 406–420, Jun. 2013.
[51] P. Wang, Q. Hu, Z. Fang, C. Zhao, and J. Cheng, "DeepSearch: A fast image search framework for mobile devices," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1, pp. 1–22, Jan. 2018.

SANGWOO PARK was born in Yesan, South Chungcheong, South Korea, in 1994. He received the B.S. and M.S. degrees in information and communications engineering from Myongji University, Yongin, South Korea, in 2022. From 2017 to 2019, he was a Military Officer at the Republic of Korea Army. He holds a patent for indoor positioning using wireless signals. His research interests include efficient indoor positioning, pattern recognition methods and applications, and next-generation mobile communication using artificial intelligence techniques.

DONG HO KIM (Senior Member, IEEE) received the B.S. degree from Yonsei University, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1999 and 2004, respectively. From 2004 to 2006, he was a Senior Member of Technical Staff at the 4G Wireless Technology Laboratory, Samsung Advanced Institute of Technology (SAIT). From 2004 to 2006, he was a Senior Researcher at the Mobile Communications Research Institute, Samsung Electronics. Since 2007, he has been a Professor with the Seoul National University of Science and Technology. His current research interests include the design of immersive media (VR/AR) and 5G/6G communication systems.

CHEOLWOO YOU (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronics engineering from Yonsei University, Seoul, South Korea, in 1993, 1995, and 1999, respectively. From January 1999 to April 2003, he was a Senior Research Engineer with the LG Electronics, Gyeonggi, South Korea. From 2003 to 2004, he was a Senior Research Engineer at the EoNex, Songnam, South Korea. From August 2004 to July 2006, he was with the Samsung Electronics, Suwon, South Korea. Since September 2006, he has been with the Department of Information and Communications Engineering, Myongji University, Gyeonggi, Yongin, South Korea. His research interests include next generation communication systems, artificial intelligence, air I/F technologies in the international standards, communication theory, signal processing, 5G/6G communication systems, machine and deep learning, AR/VR, the IoE/AIoT, and V2X.