



Article Deep Learning-Based Localization for UWB Systems

Doan Tan Anh Nguyen ¹, Han-Gyeol Lee ¹, Eui-Rim Jeong ², Han Lim Lee ¹, and Jingon Joung ^{1,*}

- ¹ School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, Korea; tananh0404@cau.ac.kr (D.T.A.N.); forener@cau.ac.kr (H.-G.L.); hanlimlee@cau.ac.kr (H.L.L.)
- ² Department of Information and Communication Engineering, Hanbat National University, Daejeon 305-719, Korea; erjeong@hanbat.ac.kr
- * Correspondence: jgjoung@cau.ac.kr

Received: 20 September 2020; Accepted: 14 October 2020; Published: 18 October 2020



Abstract: Localization has been extensively studied owing to its huge potential in various areas, such as Internet of Things, 5G, and unmanned aerial vehicle services. Its wide applications include home automation, advanced production automation, and unmanned vehicle control. In this study, we propose a novel localization method that utilizes convolutional neural network (CNN) and ultra-wideband (UWB) signals. A localization problem is converted to a regression problem with the proposed CNN, in which the ranging and positioning phases are integrated. By integrating the ranging and positioning phases, the proposed CNN estimates the location of UWB transmitter directly without any additional step. To integrate both phases of localization, a simple-yet efficient input image generation method is proposed. In the proposed input image generation method, three oversampled two-dimensional input images are generated from the three received UWB signals and they are provided to the designed CNN through the three channels, which are represented by red-, green-, and blue-color channels, respectively. The proposed CNN-based localization system then estimates the location of the UWB transmitter directly using the three-channel image as an input of the CNN. Simulation results verify that the proposed CNN-based localization method outperforms the traditional threshold-based and existing CNN-based methods. Also, it is observed that the proposed method performs well under an asymmetric environment, unlike the existing method.

Keywords: localization; deep learning; convolutional neural network (CNN); ultra-wideband (UWB) system

1. Introduction

Device location information is gaining significant attention owing to its usage in indoor/outdoor tracking and navigation services in both small-and large-scale areas. With the recent development of many Internet of Things applications, the use of localization systems by numerous services has become significantly challenging. Indoor locating systems heavily rely on wireless technologies, such as Wireless Fidelity (Wi-Fi), Bluetooth, radio frequency identification device (RFID), and ultra-wideband (UWB) [1]. The Wi-Fi technology is broadly available with great accuracy because it can be implemented at low costs. However, Wi-Fi systems can be significantly affected by noise. To resolve this issue, complex algorithms are required for localizing the Wi-Fi systems. Bluetooth is known for connecting fixed or moving devices within a certain personal space. It provides large throughput with a wide signal reception range while consuming low energy; however, real-time localization in Bluetooth is challenging owing to the significant received signal delay and noise. The RFID systems are similar to the Bluetooth systems in terms of wide signal reception range, low power consumption, and low localization accuracy. UWB technology has been primarily used in various indoor environments.

Its popularity in indoor localization can be attributed to its robustness against interference from other devices. Furthermore, UWB signals can penetrate various materials including walls, and they are robust against multipath effects. Thus, UWB systems can provide high accuracy. However, they have disadvantages, such as shorter range, higher cost, and more complex hardware requirements.

1.1. Related Work for UWB System Localization

Various studies have implemented UWB technology to a wide range of localization systems, e.g., (i) classical correlation-based method using time-of-arrival (ToA) and angle-of-arrival (AoA) [2–4], (ii) machine-learning-based algorithms [5], (iii) deep-learning-based methods using classification [6–8], and (iv) deep-learning-based method using regression [9–16].

In [2], the authors designed a joint ToA and AoA estimators for UWB indoor localization systems. Their experiment results demonstrated ranging and angular errors of approximately 10 cm and 1°, respectively, for a certain pulse. Because non-line-of-sight (NLoS) is a significant factor in UWB localization degradation, the authors in [3] developed an algorithm to mitigate the NLoS range error by introducing the Taylor series least-square method. The authors conducted two experiments in their study. In the first experiment, the radial location errors of the static testing points were measured. The errors of the moving target were measured in the second experiment. With the stationary testing points, almost all radial location errors were observed to be below 0.5 m, while the errors of the mobile target were often below 1 m. In [4], the authors demonstrated an impulse-based UWB localization system that utilized the time difference of arrival estimation and an error correction algorithm. The experiment was set in a $6.3 \times 5.9 \times 3$ m³ room, and from the experiment, positioning accuracy was achieved within 0.02 m.

Machine learning algorithms are powerful tools as a system approximator. They can obtain desirable outputs for proper input without thorough system analysis. Owing to this characteristic, machine learning algorithms have been integrated into numerous indoor localization systems. For example, some machine learning-based localization systems can convert a localization problem into a classification problem [5]. Here, the authors developed a localization system that utilized a multiclass support vector machine and UWB signals to identify the rooms where the UWB transmitter (Tx) was located. In their experiment, the proposed localization system achieved an accuracy of 95%. However, the most widely used machine learning algorithms are deep learning algorithms.

In some researches, a localization problem was solved by using a deep learning algorithm from a viewpoint of *classification* [6–8]. In [6], three different deep neural networks (DNNs) and one convolutional neural network (CNN) for localization using the received signal strength (RSS) and channel state information (CSI) were proposed. Their experimental results showed that the CNN-based localization system with CSI could achieve an accuracy of 99%. In [7], the authors introduced a fingerprinting method that combined an extreme learning machine scheme and autoencoder to classify the location of devices from the corresponding RSS. From the numerical results in a $40.4 \times 28.8 \text{ m}^2$ laboratory with 19 reference rooms, it was shown that the success rate of the classification is above 92%. In [8], a CNN-based localization system using the RSS indicator was proposed. The output of this system included one of the 74 reference points across the area where the position estimation was performed. In their experiment, the proposed localization system achieved an accuracy of 94.45% with an average localization error of as low as 1.44 m. These machine learning-based localization systems that solve a localization problem as a classification problem, however, have an inherent error because the localization based on classification estimates discrete values instead of continuous location values. Furthermore, as the area where the localization system operates increases, it must be divided into more subareas that serve as labels to reduce the discretization error. Such a large number of labels complicates the classification-based localization systems.

On the other hand, the machine learning-based localization algorithms can be transformed to a *regression* problem [9–16]. In [9], the authors proposed three different DNN-based localization systems using RSS. The output of these networks included the x-and y-coordinates of the estimated

location. Their simulation results showed that the best DNN-based localization method could achieve a root mean squared error (RMSE) of approximately 1 m. In [10], the authors designed a deep long-short-term-memory (LSTM) scheme that uses the RSS indicator to estimate the target location. In a 55×50 m² office, 1.75 m of an average localization error was achieved. In [11], the authors employed a recurrent neural network (RNN) for indoor localization by exploiting the RSS indicator and the trajectory information. Their on-site experiment showed that an average localization error is below 0.75 m. In [12], an LSTM network was proposed to predict the device's location based on the amplitude and phase of CSI, with 1.35 m of an average error. A deep CNN-based indoor localization method was proposed in [13] for a 5-GHz Wi-Fi system. The input of the CNN was generated from the CSI and AoA. In a 6×9 m² laboratory, 1.8 m of average localization error was achieved. On the other hand, in a 2.4×24 m² corridor, 2.4 m of average localization error was achieved. In [14], the authors proposed a CNN-based distance estimation algorithm between a Tx and a receiver (Rx) using the channel impulse response (CIR) of the UWB signals. In their experiment, the employed CNN-based distance estimation algorithm could achieve an RMSE of less than 1 m at a moderate signal-to-noise ratio (SNR). However, an additional positioning algorithm, such as data fusion [15], is required to obtain the position of Tx because the ranging method only estimated the distance between Tx and Rx. From UWB pulse's ToA, a distance was estimated, and an LSTM network estimated the location of Tx [16]. The simulation results showed that the error of the distance estimation hindered LSTM from being trained properly and that the localization error was approximately 7 m.

1.2. Our Contribution

In this study, we propose a novel CNN-based localization method that utilizes the UWB technology. The conventional CNN-based method mentioned in [14] estimates the distance between a Tx and an Rx then takes a positioning algorithm to estimate the location of Tx. The two sequential estimation steps, namely the distance and position estimation may cause an error propagation issue. On the other hand, the LSTM network-based localization method in [16] estimates Tx location directly; however, it requires an additional algorithm for estimation of ToA because it uses estimated distance based on ToA.

On the other hand, the proposed CNN-based localization method uses received UWB signals to estimate the Tx location directly. To this end, we design an input image generation method for the proposed method. In [14], the CNN-based ranging method uses a two-dimensional (2D) image with one channel as the input. Contrary to the method in [14], three oversampled 2D images are inputted through three channels to the proposed CNN to localize by integrating the ranging and positioning phases. The proposed CNN-based method with three oversampled 2D images solves the localization problem directly as a regression problem and yields the x-and y-coordinates of Tx as an output. By solving the localization problem as a regression problem, the discretization error can be reduced. Furthermore, the complexity of the CNN model is independent of the area size. The proposed CNN-based and CNN-based methods. Moreover, the threshold-based and conventional ToA-based and CNN-based methods. Moreover, the threshold-based and conventional CNN-based methods show higher localization errors under the asymmetric shape of the area where the localization is performed, e.g., a long and narrow corridor. However, the proposed method performs well irrespective of the shape of the area. The major contributions of this paper can be summarized as follows:

- The ranging and positioning phases are integrated, so that error propagation from ranging to positioning can be reduced.
- For the proposed localization method that integrates the ranging and positioning phase, a novel CNN structure is designed, which is trained through three 2D images.
- Compared to the conventional CNN-based localization method in [14] that requires three CNNs, only one CNN is required for the proposed method.

• The proposed method improves localization accuracy with robustness against the asymmetry of the area where the localization is performed.

The remainder of this paper is organized as follows. In Section 2, the localization problem, UWB system, channel models (CMs), area, and antenna setup are described. In Section 3, the conventional localization methods based on the ToA estimation and CNN for UWB systems are briefly reviewed; then, the proposed CNN-based localization method is explained. In Section 4, we compare the localization performance of the conventional localization systems and the proposed CNN-based localization systems and the proposed cNN-based localization systems and the proposed cNN-based localization system in terms of various parameters and then analyze the numerical simulation results. Finally, Section 5 presents the conclusion and future work.

2. System and Signal Models

In this section, we describe a localization system model, signal model, CMs, and the environments.

2.1. Localization System Model

In this study, a rectangular area is considered for the proposed localization system as shown in Figure 1. The bottom-left point of this rectangular area on the x-y plane is positioned at (0,0). The size of the area is $D_1 \times D_2$ where D_1 and D_2 are width and height of the area, respectively. Here, a symmetry factor γ is defined as the ratio of D_1 and D_2 i.e., $\gamma = D_1/D_2$, which represents the symmetry of the area. The three Rx's are positioned on boundary of the area as shown in Figure 1. The coordinates of Tx and Rx-*i* are (x, y) and (x_i, y_i) , respectively, where $i \in \{1, 2, 3\}$ is the index of Rx. The location of Tx is uniformly distributed inside the area, i.e., $x \sim \mathcal{U}[0, D_1]$, $y \sim \mathcal{U}[0, D_2]$, where $\mathcal{U}[a, b]$ denotes the continuous uniform distribution with interval [a, b]. The distance from the Tx to Rx-*i* is denoted as d_i , i.e., $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$.

In Figure 1, the localization aims to determine the unknown position of Tx, i.e., (x, y), based on a set of measurements at three Rx's, namely, Rx-1, Rx-2, and Rx-3. Usually, the basic localization techniques consist of two main steps:

- (i) *Ranging phase*: Selected measurements are performed to estimate the distance between transceivers.
- (ii) *Positioning phase*: The measurements are then processed to determine the position of the target.



Figure 1. System model with three receivers (Rx's) and one transmitter (Tx), in which the location of the Tx is to be estimated through the three Rx's.

In the ranging phase, each Rx estimates the distance to a Tx using the measurements extracted from the received signals, such as RSS, ToA, and AoA. In the localization for UWB systems, time-based measurements, such as ToA or time differences of arrival, are frequently used for ranging owing to the outstanding ability of UWB signals to resolve the multipath effects and penetrate obstacles [17].

of dense multipath components is significantly severe in UWB systems mainly operate. The effect nature of UWB channels [18]. Besides the multipath effect, the complicated statistical characteristics of UWB channels make the mathematical analysis for UWB localization systems considerably difficult. Here, the statistical model of small-scale fading is determined depending on the environments where the UWB system operates. Therefore, alternative amplitude distributions must be used [19], implying that advanced and adaptive signal processing techniques would be required according to the CMs.

In the positioning phase, the solution of the least-square problem [20] can be applied. However, owing to the presence of errors in the estimated distance, the least-square problem of localization becomes a nonlinear optimization that requires additional calculations and estimations, resulting in prohibitive complexity. Machine learning algorithms, in this context, does not require a positioning phase; they utilize the complicated statistical characteristics of UWB signals and CMs. A machine learning algorithm used in a localization system is trained to estimate the location of Tx, instead of the distance between Tx and Rx, and thus, the positioning phase could be omitted.

As mentioned above, the statistical channel properties of UWB systems are complicated. However, these complicated characteristics can be interpreted as the distinctiveness of each location. In [21], a UWB Tx was located at the center of the environment, and the power delay profiles (PDPs) of the UWB signal at several Rx's distributed over the area were measured. The changes in the small-scale fading effects corresponded to those in the PDP owing to small variations in the Rx position. In other words, the CIR at the fixed Rx varies according to the small position changes in the Tx. These variations could provide distinguishable patterns corresponding to certain Tx positions with the machine learning-based localization system for the UWB technology. This implies that machine learning-based localization for UWB systems estimates the position of Tx more accurately by utilizing the distinct CIR patterns.

2.2. Signal and Channel Models

UWB signals are used in indoor localization system owing to their robustness against the interferences as well as the multipath and NLoS effects. In this paper, the IEEE 802.15.4a CMs in [22] are considered for our localization system. Suppose that Tx sends UWB signals to Rx's. The received signal $r_i(t)$ of RX-*i* is then represented as follows [22]:

$$r_i(t) = h_i(t) * s(t) + n_i(t), \ i \in \{1, 2, 3\},$$
(1)

where $h_i(t)$ is the impulse response of channel between Tx and Rx-*i*, s(t) is the transmitted causal signal at time *t*, $n_i(t)$ is an additive white Gaussian noise at Rx-*i*, and * denotes the convolution operation. Here, the channels are modeled as follows [22]:

$$h_i(t) = \sum_{l=0}^{L_i - 1} \sum_{k=0}^{K_l - 1} a_{k,l} \exp(j\phi_{k,l}) \delta(t - T_l - \tau_{k,l}), \ i \in \{1, 2, 3\},$$
(2)

where L_i is the total number of clusters, K_l is the total number of multipath components of the *l*th cluster, $a_{k,l}$ shows the tap weight of the *k*th multipath component of the *l*th cluster, $\phi_{k,l}$ is the uniformly distributed phase of the *k*th multipath in the *l*th cluster, and $\delta(\cdot)$ denotes a delta function, T_l denotes the delay for the *l*th cluster, $\tau_{k,l}$ is the intra-cluster delay for the *k*th multipath component of the *l*th cluster. The CIR, $h_i(t)$, is modeled differently based on the environments where the measurements are performed. A more detailed description of each CM and the related channel parameters is specified in [22], in which the residential, office, suburban, industrial, and open outdoor channels are modeled

for the cases of line-of-sight (LoS) and NLoS. The CMs considered in this study are classified in Table 1. The signal, denoted by $y_i(t)$, after matched filtering at Rx-*i* is then represented as [23]:

$$y_i(t) = s(T_p - t) * r_i(t),$$
 (3)

where the T_p denotes the delay for causality of the signals.

Channel Model	Environment
CM1	Residential, LoS
CM2	Residential, NLoS
CM3	Office, LoS
CM4	Office, NLoS
CM5	suburban LoS
CM6	suburban NLoS
CM7	Industrial, LoS
CM8	Industrial, NLoS
CM9	Open Outdoor (Snow-covered, farm)

Table 1. Class of CMs [22].

3. Conventional and Proposed Localization Methods

In this section, we briefly introduce the conventional ToA-based localization and CNN-based localization methods. For convenience, the conventional and proposed CNN-based localization methods are referred to as CNN-based distance estimation (CNN-DE) and CNN-based location estimation (CNN-LE), respectively.

3.1. Conventional ToA-Based Localization Method

In [24], threshold-based ToA estimation was proposed. From $y_i(t)$ in (3), an Rx-*i* estimates the ToA when the received signal energy is greater than the predetermined threshold. To improve the accuracy of ToA estimation, the threshold is typically designed such that it is inversely proportional to the average received SNR. More details can be found in [24]. Subsequently, the Rx-*i* can obtain the estimated distance \hat{d}_i from Tx by multiplying the estimated ToA and speed of light, i.e., $\hat{d}_i = \hat{\Delta}_i \times c$, where $\hat{\Delta}_i$ is the estimated ToA at Rx-*i* and *c* is the speed of light. This ranging information obtained from the three Rx's is used to estimate the position of Tx. In this study, we use a least-square solution in the positioning phase of the ToA-based localization for comparison purposes.

3.2. Conventional CNN-DE Method

The CNN-DE method consists of the ranging phase based on [14] and the positioning phase based on the method in [15]. In the ranging phase of CNN-DE, a CNN at each Rx estimates the distance to a Tx using the one-channel image. Thus, three CNNs are required in CNN-DE. Since the three Rx's for the conventional CNN-DE method perform the identical procedure, without loss of generality, we describe the CNN-DE method focusing on one Rx-*i*.

To utilize a CNN, the received signals are transformed to a 2D image. The received signal $y_{i,q}(t)$ is recorded and sampled with the sampling frequency f_s at Rx-*i*. The Rx-*i* then generates a complex-valued vector $\mathbf{y}_{i,q} \in \mathbb{C}^{N \times 1}$, where the *n*th element is a sampled value of the received signal $y_{i,q}[n]$, $n = \{0, ..., N - 1\}$, and *N* is the number of samples per signal for a measurement of CIR. Here, *q* represents a training sample index where $q = \{1, ..., Q\}$ and *Q* is the number of training data for a CNN.

Because the CNN can only take real-valued inputs, $\mathbf{y}_{i,q}$ should be transformed into a real-valued vector $\mathbf{v}_{i,q} \in \mathbb{R}^{N \times 1}$, whose *n*th element is the normalized absolute value of $\mathbf{y}_{i,q}$, i.e.,

$$v_{i,q}[n] = \frac{|y_{i,q}[n]|}{\max_{m \in \{1,\dots,N\}} |y_{i,q}[m]|}, \ i \in \{1,2,3\}, \ q \in \{1,\dots,Q\}.$$
(4)

The normalized absolute valued signals, i.e., $\{\mathbf{v}_{i,q}\}$, with N = 3600 are shown at the left subfigure in Figure 2. The input $M \times (N/M)$ image matrix, $\mathbf{V}_{i,q}$, for training the CNN at Rx-*i* is then generated by reshaping this real-valued vector, $\mathbf{v}_{i,q}$, as follows:

$$\mathbf{V}_{i,q} = \max[\mathbf{v}_q]_M \in \mathbb{R}^{M \times (N/M)}, \ i \in \{1, 2, 3\}, \ q \in \{1, \dots, Q\},$$
(5)

where $mat[\cdot]_M$ refers to a matricization operation with a column-major order and its dimension M. This one-channel image is used as an input for the CNN-DE in the ranging phase.

The one-channel image is visualized as a monochrome image, as shown at the right subfigure in Figure 2, where M = 60. Here, the pixels with a large value are brighter while those with a low value have a darker value. Therefore, the large-amplitude multi-path components (MPC) and candidates of the first arriving MPC correspond to the bright and moderately shaded pixels of the first few pixels before the brightest pixel in each row, respectively. Considering that the two important parameters related to the distance based on the estimated ToA are the first arriving MPC and the number of local peaks before the global peak [25], it is evident that the patterns are closely related to the distance between the Tx and Rx appear on the one-channel image. The one-channel image and the true distance between the Tx and each Rx are then used in the training phase of CNN-DE.



Figure 2. Example of the input for CNN-DE training of Rx-*i*. Here, the input sequence vector $\mathbf{v}_{i,q}$ (the left subfigure) is generated from the received signals at Rx-*i*, and it has a length of 3600, i.e., N = 3600. One-channel input image matrix $\mathbf{V}_{i,q}$ (the right subfigure) is 60×60 , i.e., M = 60.

As stated, one CNN is required for each Rx for the CNN-DE. The structure of one CNN of three for CNN-DE is shown in Figure 3. The CNN consists of four convolutional layers for feature extraction and a fully connected layer for ranging followed by one regression layer. The output of CNN is the distance between a Tx and one of three Rx's.

In the positioning phase, the location of Tx is estimated based on the least-square solution of the localization problem using the estimated three distances obtained in the ranging phase. The closed-form of the estimated location of Tx for the case of three Rx's is represented as follows [15]:

$$\hat{\mathbf{x}} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{b}, \qquad (6)$$

where

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}, \ \mathbf{H} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix}, \ \mathbf{b} = \frac{1}{2} \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 - \hat{d}_2^2 + \hat{d}_1^2 \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 - \hat{d}_3^2 + \hat{d}_1^2 \end{bmatrix},$$
(7)

 (\hat{x}, \hat{y}) are the coordinates of the estimated location of Tx, and \hat{d}_i is the estimated distances between the Tx and Rx-*i*. With this localization method, CNN-DE estimates the location of the Tx with the three separately estimated distances indirectly.



Figure 3. Structure of CNNs for the CNN-DE method. Each Rx requires CNN, and thus, three CNNs are required for the CNN-DE method.

3.3. Proposed CNN-LE Method

In the proposed CNN-LE method, the position of a Tx is estimated directly without any positioning phase, unlike the CNN-DE. Therefore, only one CNN can be employed at any of Rx's or a data collection center. To integrate the positioning phase to one CNN of the CNN-LE, however, additional input image processing is required.

As shown in the left subfigure of Figure 4, the normalized absolute values of the three received signals at Rx-1, Rx-2, and Rx-3, namely, $\mathbf{v}_{1,q}$, $\mathbf{v}_{2,q}$, and $\mathbf{v}_{3,q}$, can be represented by solid-red, dashed-green, and dotted-blue lines, respectively. Here, the length of each sequence is 14,400, i.e., four times the length of an input image of CNN-DE. Based on trilateration, which is the most fundamental positioning technique, three normalized absolute values are transformed to a three-channel RGB image as visualized at the right subfigure in Figure 4. The resolution of each channel of the three-channel image for CNN-LE is then four times that of an input image of CNN-DE. Here, the high-brightness vertical streaks hold a position in each channel closely related to the estimated distances, \hat{d}_i , between a Tx and Rx-*i*. Therefore, it can be expected that the patterns related to the positioning phase are included in the cross-correlation between the channels so that the CNN for the CNN-LE method can estimate the location of Tx directly. To this end, the three 2D image matrices, namely $\mathbf{V}_{1,q}$, $\mathbf{V}_{2,q}$, and $\mathbf{V}_{3,q}$, are combined to a three-channel 2D image input $\mathbf{V}_q \in \mathbb{R}^{M \times (N/M) \times 3}$. In the proposed CNN-LE method, this three-channel RGB image \mathbf{V}_q and the true location of the Tx are used in the training phase of CNN-LE to directly estimate the location of Tx. The estimated coordinates (\hat{x}_q , $\hat{\eta}_q$) of Tx are then the output of the CNN for CNN-LE.

The CNN structure used in CNN-LE is designed as shown in Figure 5. The designed CNN includes ten convolutional layers with 3 convolutional filters and stride one for all filters. The number of channels for the filters are 16, 16, 32, 32, 64, 64, 128, 128, 256, and 256, respectively. The size of the filter, the number of strides, and the number of channels are denoted sequentially in the parenthesis,

e.g., $(3 \times 3, 1, 16)$ for the first convolution layer. As an activation function, the rectified linear unit (ReLU) layers are followed by the first, third, fifth, seventh, and ninth convolution layers. For the second, fourth, sixth, eighth, and tenth convolution layers, the batch normalization layers are included between the convolution and ReLU layers. The max-pooling layers are after the second, fourth, sixth, and eighth ReLU layers for down-sampling the convolutional layer output. The first and second pooling layers have a 2 × 2 pool each with stride two, whereas the third and fourth pooling layers have a 3 × 3 pool each with stride three, which are denoted sequentially in the parenthesis, e.g., (2 × 2, 2) for the first max-pooling layer. After the last ReLU layer, i.e., the tenth ReLU layer, two fully connected layers with the 256 and 2 units (outputs), respectively, are included to generate two inputs of regression layers to estimates of *x* and *y*. A regression layer is located at the last of the network to compute the loss by comparing the ground truth of location, i.e., (x_q, y_q) , of the training data and the output of the fully connected layer 2, where the loss is defined as a half-mean-squared-error as follows [26]:

$$\log = \frac{1}{4} \sum_{q=1}^{Q} \left((x_q - \tilde{x}_q)^2 + (y_q - \tilde{y}_q)^2 \right), \tag{8}$$

where \tilde{x}_q and \tilde{y}_q are the network prediction from the *q*th training sample. The proposed CNN learns the weights of the network to minimize the loss in (8) after training with *Q* training samples. The algorithm of the proposed CNN-LE is summarized in Table 2 which is the localization procedure after the training CNN, and thus the training index *q* is omitted.



Figure 4. Example of the input for CNN-LE training. Here, the input sequence vectors $\mathbf{v}_{1,q}$, $\mathbf{v}_{2,q'}$, and $\mathbf{v}_{3,q}$ shown in (**a**) are generated from the received signals of Rx-1, Rx-2, and Rx-3, respectively. Each sequence has a length of 14,400, i.e., N = 14,400. (**b**) Three-channel input images, namely, $\mathbf{V}_{1,q} \in \mathbb{R}^{120 \times 120}$, $\mathbf{V}_{2,q} \in \mathbb{R}^{120 \times 120}$, and $\mathbf{V}_{3,q} \in \mathbb{R}^{120 \times 120}$, are the input of the proposed CNN-LE, where M = 120.

Table 2. Procedure of the proposed CNN-LE localization (excluding the training procedure and the training index *q* is omitted).

Step	Procedure
1	Collect the received signals $y_1(t)$, $y_2(t)$, and $y_3(t)$ in (3) from Rx's 1, 2, and 3, respectively.
2	Generate the normalized and oversampled absolute sequence of the received signals, i.e., $v_1[n]$, $v_2[n]$, and $v_3[n]$, from (4).
3	Generate three 2D images, i.e., V_1 , V_2 , and V_3 , from (5).
4	Generate the input of CNN-LE, i.e., $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3] \in \mathbb{R}^{M \times (N/M) \times 3}$.
5	Provide V to CNN-LE shown in Figure 5.
6	Obtain the estimation (\hat{x}, \hat{y}) of the location (x, y) from the CNN-LE output.



Figure 5. CNN model for the CNN-LE method. Only one CNN is required for the proposed CNN-LE method. The CNN can be implemented at any of three Rx's.

4. Simulation Results

In this section, the localization performance of the propsed CNN-LE is evaluated by using MATLAB R2020a. In simulation, 40,000 input signals were generated to simulate ToA-based, CNN-DE, and CNN-LE methods. Three-quarter input signals were used for training and one-quarter signals were set for cross-validation in both conventional and proposed CNN-based localization methods. The sampling frequency f_s was fixed at 24 GHz. The parameters of CMs listed in Table 1 follow them in [22]. The area setup for the simulation is illustrated in Figure 1 with three Rx's placed at (0,0), $(\frac{D_1}{2}, D_2)$, and $(D_1, 0)$ respectively. For the input size, N and M were set to 3600 and 60, respectively, implying that the input image sizes for CNN-DE and CNN-LE were 60×60 and 120×120 pixels, respectively.

The differences in the CNN for CNN-DE and CNN-LE are listed in Table 3, and the training options for each CNN-based localization are presented in Table 4, where SGDM denotes the stochastic gradient descent with momentum algorithm. Throughout the extensive simulations, the number of training samples and training options are selected to ensure the optimal result. The localization RMSE is used as a performance metric that is calculated as follows:

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \left[(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2 \right]},$$
(9)

where *T* is the number of test samples, (\hat{x}_t, \hat{y}_t) and (x_t, y_t) are the estimated and true coordinates of Tx location of the *t*th test sample in meters, respectively.

Although localization for UWB systems is more suitable for indoor environments owing to the robustness against the multipath effect and NLoS by the blockage, we perform simulations for both indoor and outdoor CMs to check whether the proposed CNN-based localization could estimate the accurate location of Tx regardless of CMs.

Three main experiments are conducted. Each experiment examines the effect of the SNR, area size, and asymmetry of the area on the localization accuracy. In each experiment, we compare the localization accuracy of the estimated ToA-based localization, CNN-DE, and CNN-LE. We also

analyze the results based on certain channel parameters [25] to specify the suitable environments and requirements for the localization system. Subsequently, we examine whether the CNN of CNN-LE that was trained for a specific CM can work well in other CM to determine the possibility of *transfer learning* to reduce the learning time, which is a disadvantage of the proposed method. The CNN of CNN-LE that is trained with the CM# data will be referred to as CM#-NET with $\# \in \{1, ..., 9\}$.

Parameters	CNN-DE	CNN-LE
Number of input layers	1	1
Number of convolutional layers	4	10
Number of batchnorm layers	4	5
Number of ReLU layers	4	10
Number of max-pooling layers	3	4
Number of fully connected layers	1	2
Number of regression layers	1	1
Number of total layers	18	33

Table 3. CNN model parameters.

Table 4. CNN training options and time complexity results.

Options	CNN-DE	CNN-LE
Solver	SGDM	SGDM
Initial learn rate	0.001	0.02
Gradient threshold	infinite	1
Learn rate drop period	1	2
Learn rate drop factor	0.9	0.9
Shuffle	every epoch	every epoch
Max epoch	30	30
Mini batch size	200	200
Validation frequency (iterations)	50	100
Number of epochs to converge	19	29
Training time	2 min	16 min
Localization time	2.2 s	6.1 s

Using a computer with 3.6-GHz CPU and 32-GB RAM, the proposed CNN-LE required eight and three times longer training and localization times, respectively, compared to a conventional CNN-DE method. The time complexity increase of the proposed method comes from the fact that the training image and CNN structure of the proposed CNN-LE is relatively greater than that of the conventional CNN-DE method.

4.1. Performance with Respect to SNR

In the simulations with respect to the SNR, the area size is fixed to $20 \times 20 \text{ m}^2$. Figure 6 shows the localization accuracy i.e., RMSE, comparison for the ToA-based localization, CNN-DE, and proposed CNN-LE with respect to the SNR. For all CMs and SNRs, the proposed CNN-LE outperforms other methods. It achieves the RMSE below 1 m in CM3 (Office LoS environment) and CM7 (Industrial LoS environment) when SNR is greater than 10 dB, which is the required performance for highly accurate indoor localization systems. Generally, the RMSE decreases as the SNR increases for all methods and channels. This is because the signals with higher SNR are significantly robust against the additive white Gaussian noise and path loss, which clarifies the patterns related to the distance to ensure that the CNN can learn these patterns efficiently. For the LoS CMs (i.e., CM1, CM3, and CM7) and NLoS CMs (i.e., CM2, CM4, and CM8), a slight performance gap is observed, except the suburban environments (i.e., CM5 and CM6). These general results over the LoS and NLoS CMs can be attributed to the good

penetration property of UWB signals. The most significant difference between CM5 and CM6 is the average energy of the first arrived MPC [25]. As mentioned in Section 3.2, one of the important channel parameters for ranging is the first arrived MPC, specifically the average energy of the first arrived MPC. The severe decrease in the first arrived MPC introduced by NLoS is shown in CM6, resulting in severe degradation of localization accuracy in the suburban NLoS channels.



Figure 6. RMSE (y-axis) performance across SNR (x-axis). CM# data is used for both training and test.

For the outdoor and indoor CMs, the increase in improvement by the SNR is more evident in the indoor CMs than the outdoor CMs except for CM1 and CM2, which are the residential CMs. The significant difference between the residential and other indoor CMs lies in the number of local peaks. The average channels of the input image for CM1 and CM3 are represented in Figure 7. As shown in Figure 7, the local peaks of CM1 corresponding to bright streaks are distributed over the input image, resulting in complicated patterns. In contrast, clear patterns appeared in CM3, which can be utilized by CNN for efficient operations. This result of severe multipath effect implies that additional input image preprocessing, such as the peak-detection algorithm [27], can be required for residential environments. Moreover, some remarkable improvements were observed. For examples, there is approximately 3.8 m average RMSE improvement below 10 dB SNR; 1.05 m average RMSE improvement between 10 dB and 20 dB SNRs in CNN-LE method for indoor CMs. Thus, we can use 10 dB as the target SNR for the implementation of CNN-LE.

The most remarkable result is that a CNN trained for a specific indoor CM can perform well in other indoor CMs, and a similar tendency could be observed for outdoor CMs. For example, CM3-NET performs well in other indoor CMs; CM6-NET and CM9-NET exhibit good performance when their CMs are interchanged as shown in Figure 8. From these results, we can surmise that transfer learning can be employed to reduce the learning time.



Figure 7. Average input images with emphasized contrast through a monochrome image. (**a**) CM1. (**b**) CM3.



Figure 8. RMSE (*y*-axis) across SNR (*x*-axis). CM#-NET is a CNN that is trained by using data generated under CM# model. CM#-NET is tested with various CMs.

4.2. Performance with Respect to Map Size

Considering that the localization methods for the UWB system mainly operate in indoor environments, 30 dB SNR in this simulation is reasonable. Figure 9 illustrates the RMSE of the compared systems, namely ToA-based, CNN-DE, and the proposed CNN-LE method. Generally, the RMSE increases as the size of the area increases for all methods and CMs. The increase in the RMSE is expected because a larger average distance between Tx and each Rx results in a larger path loss. Correspondingly, increasing the area of localization yields a similar effect as decreasing the SNR.

Figure 10 shows the RMSE of each CM#-NET when a CM#-NET is applied to each CM test data with respect to the area size. Similar to the results in Section 4.1, the CM#-NETs trained in indoor and outdoor CMs work well in both environments. Especially, CM3-NET exhibits good localization performance for other indoor CMs. Therefore, we can start training the CNN using CM3-NET with the CM3 training data and utilize transfer learning to train the CNNs of other indoor CMs.



Figure 9. RMSE (*y*-axis) across the area size (*x*-axis, $D_1 = D_2$). CM# data is used for both training and test.



Figure 10. RMSE (*y*-axis) across the area size (*x*-axis, $D_1 = D_2$). CM#-NET is a CNN that is trained by using data generated under CM#. The CM#-NET is tested with various CMs.

4.3. Performance with Respect to Asymmetry of the area

In Figure 11, RMSEs are evaluated when the ratio of D_2 to D_1 , i.e., γ , varies while the SNR and area size are fixed at 30 dB and 225 m². For the given area size, the simulations are performed for three different γ values 0.36, 0.64, and 1 for all CMs. The RMSEs across all CMs are represented in Figure 11. The most noticeable observation from this result is as follows: As the symmetry of the area decreases, the localization performance of the CNN-DE degrades, whereas that of the proposed CNN-LE is almost stable. From these results, we see that CNN-DE is influenced by the shape of an

area: therefore, the CNN-DE could not work properly in some asymmetric environments, such as a corridor and container. On the other hand, we can surmise that the proposed CNN-LE operates properly irrespective of the shape of the area. Instead, the dominant factor in the localization accuracy of the proposed CNN-LE is the area size.



Figure 11. RMSE (*y*-axis) across the γ (*x*-axis) for an area of 225 m². CM# data is used for both training and test.

Figure 12 shows the RMSEs of each CM#-NET when a CM#-NET is applied to each CM test data with respect to γ for each area size. Similar to the results presented in Section 4.1, the CM#-NETs trained in indoor and outdoor CMs work well in other indoor and outdoor models, respectively. Therefore, a reduction in the learning time can be expected by implementing a transfer learning method.



Figure 12. RMSE (*y*-axis) across γ (*x*-axis) on an area of 225 m². CM#-NET is a CNN that is trained by using data generated under CM#. The CM#-NET is tested with various CMs.

5. Conclusions

In this study, we introduced a novel CNN-based localization method for UWB systems. Typical ranging and positioning phases for the localization were integrated and performed by the proposed CNN simultaneously. To train the CNN, RGB image data was constructed from three received signals. Localization performance was evaluated in terms of RMSE over various SNR conditions and CMs. From the rigorous numerical comparison, it was verified that the proposed CNN-based localization method outperforms the traditional ToA-based localization method and the recent CNN-based localization method that performs only a ranging phase with a separate positioning phase. Compared to the existing methods, the proposed CNN-based localization method achieved significant performance improvement, especially under the indoor CMs, where the UWB systems mainly operate. Furthermore, the proposed CNN-based localization was robust against the shape of the area where the target transmitter locates. The benefit of the proposed CNN-based localization can be achieved at the cost of the computational complexity for the training with oversampled data. There are various interesting future research topics, e.g., (i) reducing the computational complexity of the proposed CNN-based localization in three-dimensional space.

Author Contributions: Conceptualization, J.J. and E.-R.J.; simulation, D.T.A.N. and E.-R.J.; writing—original draft preparation, D.T.A.N. and H.-G.L.; writing—review and editing, H.L.L. and J.J.; supervision, J.J.; funding acquisition, H.L.L. and J.J. All authors have read and agreed to the published version of the manuscript.

Funding: The publication of this work was sponsored in part by the Chung-Ang University Young Scientist Scholarship in 2019 funded by Chung-Ang University, Seoul, Korea, and in part by the National Research Foundation of Korea (NRF) grant (2018R1A4A1023826) funded by the Korean government (MSIT).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AoA	Angle-of-Arrival
CIR	Channel Impulse Response
СМ	Channel Model
CM-NET	Channel Model Network
CNN	Convolutional Neural Network
CNN-DE	Convolutional Neural Network-based Distance Estimation
CNN-LE	Convolutional Neural Network-based Location Estimation
CSI	Channel State Information
DNN	Deep Neural Network
LoS	Line-of-Sight
LSTM	Long-Short-Term-Memory
MPC	Multi-Path Components
NLoS	Non-Line-of-Sight
PDP	Power Delay Profiles
ReLU	Rectified Linear Unit
RFID	Radio Frequency Identification Device
RGB	Red-Green-Blue
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSS	Received Signal Strength
SGDM	Stochastic Gradient Descent with Momentum
SNR	Signal-to-Noise Ratio
ТоА	Time-of-Arrival
UWB	Ultra Wideband
Wi-Fi	Wireless Fidelity

References

- 1. Zafari, F.; Gkelias, A.; Leung, K.K. A Survey of Indoor Localization Systems and Technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599. [CrossRef]
- 2. Taponecco, L.; D'Amico, A.A.; Mengali, U. Joint TOA and AOA Estimation for UWB Localization Applications. *IEEE Trans. Wirel. Commun.* 2011, *10*, 2207–2217. [CrossRef]
- 3. Yu, K.; Wen, K.; Li, Y.; Zhang, S.; Zhang, K. A Novel NLOS Mitigation Algorithm for UWB Localization in Harsh Indoor Environment. *IEEE Trans. Veh. Technol.* **2019**, *68*, 686–699. [CrossRef]
- 4. Zwirello, L.; Schipper, T.; Harter, M.; Zwick, T. UWB Localization System for Indoor Applications: Concept, Realization and Analysis. *J. Electr. Comput. Eng.* **2012**, 2012, 849638. [CrossRef]
- Rana, S.P.; Dey, M.; Siddiqui, H.U.; Tiberi, G.; Ghavami, M.; Dudley, S. UWB localization employing supervised learning method. In Proceedings of the IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), Salamanca, Spain, 12–15 September 2017; pp. 1–5.
- 6. Hsieh, C.-H.; Chen, J.-Y.; Nien, B.-H. Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information. *IEEE Access* **2019**, *7*, 33256–33267. [CrossRef]
- 7. Khatab, Z.E.; Hajihoseini, A.; Ghorashi, S.A. A Fingerprint Method for Indoor Localization Using Autoencoder Based Deep Extreme Learning Machine. *IEEE Sens. Lett.* **2018**, *2*, 1–4. [CrossRef]
- 8. Sinha, R.S.; Hwang, S.-H. Comparison of CNN Applications for RSSI-Based Fingerprint Indoor Localization. *Electronics* **2019**, *8*, 989. [CrossRef]
- Flix, G.; Siller, M.; Ivarez, E.N. A fingerprinting indoor localization algorithm based deep learning. In Proceedings of the International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 1006–1011.
- 10. Chen, Z.; Zou, H.; Yang, J.; Jiang, H.; Xie, L. WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Syst. J.* **2019**, *14*, 3001–3010. [CrossRef]
- 11. Hoang, M.T.; Yuen, B.; Dong, X.; Lu, T.; Westendorp, R.; Reddy, K. Recurrent Neural Networks for Accurate RSSI Indoor Localization. *IEEE Internet Things J.* **2019**, *6*, 10639–10651. [CrossRef]
- 12. Zhang, Y.; Qu, C.; Wang, Y. An Indoor Positioning Method Based on CSI by Using Features Optimization Mechanism with LSTM. *IEEE Sens. J.* **2020**, *20*, 4868–4878. [CrossRef]
- 13. Wang, X.; Wang, X.; Mao, S. Deep Convolutional Neural Networks for Indoor Localization with CSI Images. *IEEE Trans. Netw. Sci. Eng.* **2018**, *7*, 316–217. [CrossRef]
- 14. Joung, J.; Jung, S.; Chung, S.; Jeong, E.-R. CNN-Based Tx–Rx Distance Estimation for UWB System Localisation. *Electr. Lett.* **2019**, *55*, 938–940. [CrossRef]
- 15. Sayed, A.H.; Tarighat, A.; Khajehnouri, N. Network-Based Wireless Location: Challenges Faced in Developing Techniques for Accurate Wireless Location Information. *IEEE Signal Process. Mag.* 2005, 22, 24–40. [CrossRef]
- Poulose, A.; Han, D.S. UWB Indoor Localization Using Deep Learning LSTM Networks. *Appl. Sci.* 2020, 10, 6290. [CrossRef]
- 17. Dardari, D.; Conti, A.; Ferner, U.; Giorgetti, A.; Win, M.Z. Ranging with Ultrawide Bandwidth Signals in Multipath Environments. *IEEE Proc.* **2009**, *97*, 404–426. [CrossRef]
- Guvenc, I.; Sahinoglu, Z. Threshold-based TOA estimation for impulse radio UWB systems. In Proceedings of the International Conference on Ultra-Wideband (ICUWB), Zurich, Switzerland, 5–8 September 2005; pp. 420–425.
- 19. Molisch, A.F. Ultrawideband Propagation Channels-Theory, Measurement, and Modeling. *IEEE Trans. Veh. Technol.* **2005**, *54*, 1528–1545. [CrossRef]
- 20. Guvenc, I.; Sinan G.; Sahinoglu, Z. Fundamental Limits and Improved Algorithms for Linear Least-Squares Wireless Position Estimation. *Wirel. Commun. Mob. Comput.* **2012**, *12*, 1037–1052. [CrossRef]
- 21. Cassioli, D.; Win, M.Z.; Molisch, A.F. The Ultra-Wide Bandwidth Indoor Channel From Statistical Model to Simulations. *IEEE J. Sel. Areas Commun.* **2002**, *20*, 1247–1257. [CrossRef]
- 22. Molisch, A.F.; Balakrishnan, K.; Cassioli, D.; Chong, C.-C.; Emami, S.; Fort, A.; Karedal, J.; Kunisch, J.; Schantz, H.; Schuster, U.; et al. *IEEE 802.15.4a Channel Model-Final Report*; Tech. Rep., Doc. IEEE 802.1504-0062-02-004a; IEEE: Piscataway, NJ, USA, 2005.
- 23. Turin, G. An introduction to matched filters. IRE Trans. Inf. Theory 1960, 6, 311-329. [CrossRef]
- 24. Liu, W.; Ding, H.; Huang, X.; Liu, Z. TOA Estimation in IR UWB Ranging with Energy Detection Receiver Using Received Signal Characteristics. *IEEE Commun. Lett.* **2012**, *16*, 738–741. [CrossRef]

- 25. Celebi, H.; Guvenc, I.; Arslan, H. On the statistics of channel models for UWB ranging. In Proceedings of the IEEE Sarnoff Symposium, Princeton, NJ, USA, 27–28 March 2006; pp. 1–4.
- 26. Create a Regression Output Layer—MATLAB Regression Layer. Available online: https://www.mathworks. com/help/deeplearning/ref/regressionlayer.html (accessed on 28 September 2020).
- 27. Flasi, C.; Dardari, D.; Mucchi, L.; Win, M.Z. Range estimation in UWB realistic environments. In Proceedings of the IEEE International Conference on Communications (ICC), Istanbul, Turkey, 11–15 June 2006; pp. 5692–5697.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).