



Article Teacher-Assistant Knowledge Distillation Based Indoor Positioning System

Aqilah Binti Mazlan¹, Yin Hoe Ng^{1,*} and Chee Keong Tan²

- ¹ Faculty of Engineering, Multimedia University, Cyberjaya 63100, Malaysia
- ² School of Information Technology, Monash University Malaysia, Subang Jaya 47500, Malaysia
- * Correspondence: yhng@mmu.edu.my

Abstract: Indoor positioning systems have been of great importance, especially for applications that require the precise location of objects and users. Convolutional neural network-based indoor positioning systems (IPS) have garnered much interest in recent years due to their ability to achieve high positioning accuracy and low positioning error, regardless of signal fluctuation. Nevertheless, a powerful CNN framework comes with a high computational cost. Hence, there will be difficulty in deploying such a system on a computationally restricted device. Knowledge distillation has been an excellent solution which allows smaller networks to imitate the performance of larger networks. However, problems such as degradation in the student's positioning performance, occur when a far more complex CNN is used to train a small CNN, because the small CNN does not have the ability to fully capture the knowledge that has been passed down. In this paper, we implemented the teacher-assistant framework to allow a simple CNN indoor positioning system to closely imitate a superior indoor positioning scheme. The framework involves transferring knowledge from a large pre-trained network to a small network by passing through an intermediate network. Based on our observation, the positioning error of a small network can be reduced to up to 38.79% by implementing the teacher-assistant knowledge distillation framework, while a typical knowledge distillation framework can only reduce the error to 30.18%.

Keywords: indoor positioning; received signal strength indicator; convolutional neural networks; knowledge distillation; teacher assistant

1. Introduction

Over the last decade, mobile technologies, such as smartphones, have become a necessity of everyday life, considering their applications are getting progressively diverse. Following the evolution of mobile devices, various software and programs are made available on these devices. Location-based services (LBS) are the center of much research because plenty of applications, including navigation, traffic updates, asset tracking and safety-related services, require the availability of geographical positioning data to add value to their services [1]. According to Statista [2], the number of mobile users using LBS in the United States is around 150 million. Existing LBS are mainly centered on the outdoor environment. However, indoor LBS have shown immense business potential, since the technology has been adapted to numerous systems, for example, a real-time locating system to guide visually impaired people [3] and a monitoring system to keep track of an isolated COVID-19 patient [4]. Based on a report available at the Research and Market [5] site, it is stated that the value of the global indoor positioning and navigation market is \$6.92 billion in 2020, and by 2025, its worth is expected to shoot up to \$23.6 billion.

For outdoor environments, the existence of Global Navigation Satellite Systems (GNSS), such as the global positioning systems (GPS), have long been relied upon thanks to their ability to maintain excellent performance [6]; these systems have been used widely for a long period. Unfortunately, it is difficult to implement such systems indoors because they



Citation: Mazlan, A.B.; Ng, Y.H.; Tan, C.K. Teacher-Assistant Knowledge Distillation Based Indoor Positioning System. *Sustainability* **2022**, *14*, 14652. https://doi.org/10.3390/su142114652

Academic Editors: Muhammad Syafrudin, Ganjar Alfian, Norma Latif Fitriyani and Muhammad Anshari

Received: 5 September 2022 Accepted: 2 November 2022 Published: 7 November 2022

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



Copyright: © 2022 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 (https:// creativecommons.org/licenses/by/ 4.0/). demand line-of-sight (LoS) between the satellites and the receiver [7]. Since an indoor space contains obstacles and external walls, which lead to poor coverage of satellite signals [6], GPS is unsuitable for indoor location-based services [7]. Therefore, several other techniques are much preferred for indoor settings.

Earlier on, indoor positioning systems (IPS) typically adopted geometry-based algorithms, such as triangulation [8] and trilateration [9], for which the location of the base station and LoS wireless signals are required for the system to perform effectively. Unfortunately, issues with this technique emerged, since information on the location of the base station was unavailable in some instances, and it is challenging to conduct LoS transmission in an indoor environment due to the presence of objects in that indoor environment. Hence, a different scheme is preferable in complex indoor environments, and in recent years, the fingerprint approach managed to bring about satisfactory localization accuracy, because even though the number of reference points in unit space affects this technique [10], it does not require the knowledge of the exact access point (AP) locations or the angle and distance measurements [11].

A prevalent technology used for indoor positioning is Wi-Fi fingerprinting. The main reason Wi-Fi fingerprinting has become a great candidate for indoor positioning is the ubiquity of Wi-Fi access points to support wireless Internet connectivity [12]. In general, fingerprint-based IPS operates in two phases. The first phase, known as the offline phase, is performed by measuring the received signal strength indicator (RSSI) of the APs in the environment at known locations, which are regarded as the reference points (RPs). Signal collection from various RPs is conducted to build a fingerprint database. Subsequently, in the online phase, real-time RSSIs are measured by mobile devices at an unknown position. The device's location can be predicted by computing the similarity between the real-time RSSI and the set of fingerprints stored in the database by applying statistical modeling or machine learning algorithms [10].

It was observed that there had been a tremendous leap in performance for natural language processing [13], speech recognition [14] and image processing [15] using deep learning, and this has initiated the implementation of deep learning algorithms in an indoor positioning system. The convolutional neural network (CNN) has recently been a favourite choice because the convolution operation allows the positioning system to study the entire topology of an RSSI image to provide a high-accuracy positioning [16]. The hurdle of a CNN-based IPS is its high computational requirement, which causes difficulties in establishing the IPS model on inexpensive devices with limited resources. The architecture of the model affects the computational requirements. By substantially reducing the size of the CNN-IPS model, the computational need can be minimized at the expense of positioning accuracy.

The model compression can be applied to a CNN-based IPS as a strategy to overcome this issue. A common way to apply model compression is by using knowledge distillation (KD). However, it can be seen that with an increase in the complexity of the teacher model, a lightweight student model cannot perfectly imitate the teacher model's performance because it does not have sufficient capacity to mimic the high-complexity teacher model [17]. Additionally, the logits generated by the cumbersome teacher model are less soft since the certainty of associating the RSSI fingerprint images to their location classes is greater [17]. To overcome this issue, the teacher model may transfer its knowledge to another model with closer complexity. Following that, the model introduced can transfer the informative knowledge it had gained to the simple student model, as presented in [17]. The authors emphasized that adding an additional distillation step can bring finer students. By establishing multiple teacher assistants, they have also demonstrated that the best distillation path to enhance the learning experience occurs when the process passes through all intermediate teacher-assistants. Since such benefits can be observed from the teacher-assistant knowledge distillation (TAKD), the framework may boost the positioning accuracy of a simple CNN-IPS. However, there are still some unanswered problems concerning the possibility of TAKD successfully assisting the transfer of knowledge, mainly the relationship between

the RSSI and location classes from the teacher model to the student model, in order to reduce the positioning error of the system.

This work presents two CNN-based IPS with different complexity, taking RSSIs and converting them into a 2D fingerprint image. The more complex model has a higher number of convolutional layers, filters and hidden layers. Subsequently, several intermediate CNN algorithms are developed to introduce a TAKD framework in this work. The TAKD framework is able to distill the useful information from a teacher model to the student model through an intermediary model known as the teacher assistant model. The main contributions are as follows:

- Two CNN-based IPS with different complexities are generated to prove that the model's complexity will affect the performance of the positioning system. Then, knowledge distillation is performed so that the simple model can mimic the performance of the larger model.
- 2. Two CNN-based algorithms are developed as the teacher assistant model. Then, we proposed a TAKD framework to distill knowledge from the pre-trained teacher model to the teacher assistant model and lastly, to the student model. Ultimately, we investigate the benefit of employing the proposed technique by comparing the performance of IPSs that utilize the TAKD framework, the baseline KD framework and the performance of CNN-based IPS.

The remainder of the paper is structured as follows. Section 2 reviews the related work done on the indoor positioning system. Section 3 provides details regarding the existing work that will be used as a benchmark for this work. The proposed TAKD CNN-based IPS is presented in Section 4. In Section 5, a thorough comparison of the proposed and existing techniques has been made to emphasize the benefit of the proposed framework. Lastly, we conclude this work in Section 6.

2. Related Works

Numerous works have been done on the fingerprint-based indoor positioning system, and a few methods described in this paper are summarized in Table 1. The classical approach to estimating a user's location is by implementing the K-nearest neighbors (k-NN). The first k-NN-based IPS was introduced through the development of RADAR [18]. The system utilizes radio-frequency wireless networks to locate and track users inside of a building. Since then, several other works [19,20] have appeared to develop IPS based on k-NN. The IPS in [19] implements 1-NN, whereby the number of nearest neighbors is set to 1 using the UJIIndoorLoc dataset, which covers a real multi-building and multi-floor scenario. From there, they obtained an average positioning error of 7.9 m with 89.92% success rate, which has become the standard for other IPS. Other machine learning algorithms, such as decision tree [21] and random forest [22], were introduced to enhance the performance of IPS. Through [21], it is found that the computational efficiency of the decision tree-based IPS is superior to that of a simple 1-NN algorithm. However, in terms of accuracy, the results of 1-NN were similar to or better than that of the decision tree-based IPS.

Although machine learning provides satisfactory localization output, it requires timeconsuming parameter tuning, especially for larger buildings which contain numerous data. Additionally, it is deprived of the ability to thoroughly learn reliable features from the training data used to map out the RSSIs to their locations. If a higher training accuracy is needed, the algorithm must be able to process the training data extensively. Deep learning algorithms are known to be able to extract complex structure from data, as they are designed based on the hierarchical structure of the human brain [23]. Thus, many works have adopted deep learning algorithms for location prediction. The works in [24–26] incorporated deep neural networks (DNN) to counter feature extraction issues. In [24], the authors designed a DNN structure pre-trained by Stacked Denoising Autoencoder (SDA) for indoor positioning with Wi-Fi fingerprinting to tackle the fluctuating wireless signals. Hidden Markov Model (HMM) was used to improve the accuracy of the DNN model. The work in [25] also devised a DNN-based positioning scheme with Wi-Fi fingerprints. The main aim of the scheme was to achieve accurate indoor positioning with a reduced workforce.

Table 1. Summary of studies on indoor positioning system.

Work	Methods	Motive	Input Variable (s)	Output	Findings	Limitation
[18]	kNN algorithm, with a Euclidean distance similarity metric	Build a RF based locating and tracking system	WiFi RSS	Estimated position	The first WiFi fingerprinting system RADAR. Median localization error of 2.94 m.	Poor positioning performance in complex indoor environment due to its inability to fully learn reliable feature from training data.
[19]	1-NN algorithm, with a Euclidean distance similarity metric	Create database for multi-floor and multi-building localization comparison	520 WiFi RSS	Estimated floor, estimated building and estimated position (<i>x</i> , <i>y</i>)	The positioning system achieves a mean error of 7.9 m and a success rate of 89.92%.	Poor positioning performance
[21]	Decision tree	Build an efficient positioning technique using fingerprint method	WiFi RSS	Estimated position	Achieved a higher computational efficiency than 1-NN	Poor positioning performance
[24]	DNN pretrained by SDA + HMM	Build an indoor/outdoor wireless positioning	163 WiFi RSS	Estimated grid	RMSE of 0.39 m for indoor environment.	High computational complexity
[25]	DNN(SAE)	Build a multi-building and multi-floor classification	520 WiFi RSS	Estimated building and floor	Classification accuracy of 92%	The approach only consider building and floor estimation, therefore it does not estimate any specific coordinate.
[26]	DNN(SAE+ feed-forward multi-label classifier)	Build a scalable deep learning architecture for multi-building and multi-floor	520 WiFi RSS	Estimated floor, estimated building and estimated position (<i>x</i> , <i>y</i>)	Positioning error of 9.29 m and the building and floor success rate is 99.82% and 91.27% respectively.	High computational complexity
[27]	CNN	Extract images out of fingerprint to train CNN	WiFi RSS	Fine-grain location	Lowest mean error compared to DNN, SVR and KNN Average localization error of less than 2 m.	High computational complexity
[28]	1D-CNN +SAE	Build a deep-learning model for multi-building and multi-floor indoor localization	520 WiFi RSS	Estimated floor, estimated building and estimated position (<i>x</i> , <i>y</i>)	Achieved highest floor success rate compared to other technique Success rate of building and floor localization is 100% and 95% respectively. Positioning errors of 7.6 m	High computational complexity
[29]	KD-CNN	Improve the performance of a simple CNN-IPS model	14 BLE RSS	Location class (x, y, z)	The KD-CNN-IPS achieve better accuracy and average error than CNN-IPS. Positioning error as low as 1.5 m is achieved.	Poor positioning performance when the complexity gap between the teacher and student models is large.

The authors in [27] stated that IPS, which utilizes specific deep learning algorithms, produces low positioning accuracy caused by the instability of RSS due to the fading and

shadowing effect. Due to the fast pace of technological advancement, a more accurate IPS is desired; hence, several works [27,28] have started employing convolutional neural network (CNN) algorithms in IPS. The empirical results acquired by [16] proved that CNN has superior positioning capability, as their proposed CNN framework shows a lower positioning error than DNN-based IPS. In order to enable CNN-based positioning to be smoothly deployed in mobile applications, ref. [29] introduced knowledge distillation into the CNN-based positioning, resulting in KD-CNN-IPS, and demonstrated the relevance of the proposed framework. When a lightweight student model learns from a larger model, it can improve its positioning performance and provide a better execution time than the larger model. Despite the adequate performance, the architecture of the complex teacher and student models do not differ significantly. Therefore, it is understandable why the student model was able to train well under the supervision of the teacher model. However, in cases where the positioning system takes in readings from a much higher number of APs, the complexity of the CNN algorithm would have to rise. As a result, a larger complexity gap between the teacher and the student model is introduced. When this happens, the student model might not be able to learn competently from the teacher model, and we could anticipate a possibly inferior positioning performance. Hence, a more suitable framework must be studied to address this.

3. Existing Methods

For this work, we benchmarked our proposed method against a basic CNN-based IPS and the recent (KD-CNN-IPS) [29] framework. Thus, a detailed explanation regarding the architecture of the CNN-based IPS is presented in Section 3.1 and the working principle of the KD-CNN-IPS is provided in Section 3.2.

3.1. CNN-Based IPS

The core idea of CNN-based IPS is to map N samples of the input RSSIs $\{\mathbf{r}^n | n = 1, 2, \dots, N\}$ to their corresponding location labels $\{\mathbf{y}^n | n = 1, 2, \dots, N\}$. Consider an IPS with M location classes and g_m samples in the mth location class. The total number of samples in the dataset *N* can be expressed as $N = \sum_{m=1}^{M} g_m$. The CNN-IPS architecture used in this work mainly comprises an input lease of in this work mainly comprises an input layer and several convolutional layers, followed by a max pooling layer, a flattened layer and a dense layer. For the purpose of providing better visualization of the CNN-IPS architecture, Figure 1, which illustrates the architecture of the complex model, is presented. Initially, the CNN-IPS takes in a series of one-dimensional (1D) RSSI vectors $\mathbf{r}^n = [r_1^n, r_2^n, \cdots, r_K^n]$. The element of the vector is represented by r_k^n where $k \in [1, K]$ and K is the total number of APs available. After obtaining the 1D RSSI vectors, the vectors will be transformed into a square fingerprint image X^n of size $Q_1 \times Q_1$. To ensure that it is possible to reshape the vector into a square fingerprint image, the number of elements in r^n needs to be the square of an integer. Therefore, if $K \neq c^2$, where *c* is an integer, r^n will be zero padded before the image is processed by the convolutional layers, and subsequently, the dense layer. At the final layer of the dense network, a softmax activation function is implemented to calculate the probability for each of the location classes. The activation function is represented by the following equation:

$$f_{\text{softmax}}(x_j) = \frac{e^{x_j}}{\sum\limits_{l=1}^{L} e^{x_l}}$$
(1)

where x_j is the logit of the *j*-th neuron, l = 1, 2, ..., L and *L* is the total number of neurons in the dense layer. In this case, *L* equals *M*, since the number of neurons at the last fully connected layer will be identical to the number of location classes. Hence, a predicted output vector $\hat{y}^n = \begin{bmatrix} \hat{y}_1^n & \hat{y}_2^n & \cdots & \hat{y}_M^n \end{bmatrix}$ of size $1 \times M$ is generated.





Figure 1. Architecture of the CNN-IPS teacher model.

During training, loss functions are crucial for model optimization, and in theory, a perfect model would have a loss of 0. For classification models, the cross-entropy cost function is typically used for training, and it can be computed using the following:

$$L_{\text{CE}}(\boldsymbol{z}^{n}, \boldsymbol{y}^{n}) = H(f_{\text{softmax}}(\boldsymbol{z}^{n}), (\boldsymbol{y}^{n}))$$

$$= -\sum_{k=1}^{M} f_{\text{softmax}}(\boldsymbol{z}^{n}_{k}) \log(\boldsymbol{y}^{n}_{k})$$
(2)

where $H(\psi,\xi) = -\sum_{k=1}^{M} \psi_k \log(\xi_k)$ represents the cross-entropy loss function and $z^n = \begin{bmatrix} z_1^n & z_2^n & \cdots & z_M^n \end{bmatrix}$ is the logits vector.

3.2. Knowledge Distillation CNN-IPS (KD-CNN-IPS)

In order to comprehend the working principle of our proposed teacher-assistant knowledge distillation-based indoor positioning system (TAKD-CNN-IPS), it is crucial to understand the concept and operational steps of knowledge distillation, particularly response-based knowledge distillation. As mentioned in Section 2, knowledge distillation is implemented to enable a lightweight model to achieve an almost identical output to a cumbersome model by training the smaller model (student model) under the supervision of a larger, pre-trained model (teacher model) [30,31]. Soft labels generated by the teacher model are used in the training of the student model. During the training of the teacher model, the fingerprint images are mapped out to a vector of logits $z = [z_1 \ z_2 \ \cdots \ z_M]$ produced by the final dense layer. For the teacher model to generate soft labels, the temperature-scaled softmax activation function is applied on the logits,

$$f_{TS-\text{softmax}}(z_i) = \frac{e^{z_i}}{\sum\limits_{j=1}^{M} e^{\frac{z_j}{T}}}$$
(3)

where $T \ge 1$ denotes the temperature parameter.

For knowledge distillation to accomplish its task during the training of the student model, the loss function of the student model L^{S-KD} can be calculated using the following equation [30], so that the outputs of the student model match the soft targets of the teacher model.

$$L^{\rm S-KD} = \alpha L_{\rm CE}(\boldsymbol{z}_s^n, \boldsymbol{y}^n) + \beta L_{\rm KD}(\boldsymbol{z}_s^n, \boldsymbol{z}_t^n) \tag{4}$$

Unlike the CNN-based IPS, the loss function used to train the student model consists of both the student loss function $L_{CE}(z_s^n, y^n)$ and the distillation loss function $L_{KD}(z_s^n, z_t^n)$, where z_s^n and z_t^n represent the output logit vectors of the teacher and student model for the *n*th input sample, respectively. The student loss is the cross-entropy loss between the student predictions and the ground truth. The weight of the student loss function is α , where $\alpha \in [0, 1]$. On the other hand, $\beta = 1 - \alpha$ signifies the weight of the distillation loss. Mathematically, the distillation loss function can be formulated as:

$$L_{\rm KD}(\boldsymbol{z}_s^n, \boldsymbol{z}_t^n) = T^2 D_{\rm KL}(f_{\rm TS-Softmax}(\boldsymbol{z}_s^n), f_{\rm TS-Softmax}(\boldsymbol{z}_t^n))$$
(5)

where $D_{\text{KL}}(\psi, \xi)$ is the Kulback–Leiver divergence, which can be expressed as:

$$D_{\mathrm{KL}}(\psi,\xi) = \sum_{k=1}^{M} \psi_k \log(\psi_k/\xi_k)$$
(6)

4. Teacher-Assistant Knowledge Distillation Based Indoor Positioning System

A lightweight CNN-IPS model is not able to provide adequate positioning performance. In order to improve the accuracy of a model, more layers and parameters are often introduced. This implementation would result in a more complex model. Unfortunately, a complex model is unsuitable for deploying in computationally restricted devices. It has been established that knowledge distillation is able to improve the accuracy of the lightweight model by allowing the model to train by using both the ground truth label and the complex model's softened output. However, due to the low capacity of the student model, not much performance gain can be observed by the student model. Based on what has been presented, an introduction to the teacher-assistant model might be able to rectify this problem. It is essential to ensure that the size and capacity of the teacher-assistant model are somewhere in the middle of the teacher and student model so that the introduced model has enough capacity to learn from the teacher and while not being too big for the student model to learn from. In order to evaluate the effectiveness of the proposed scheme, this work adopted the Alcala Tutorial 2017 dataset [32], which contains RSSI information collected at the corridor of the School of Engineering of the University of Alcala. The dataset allows a comprehensive investigation to be conducted, as the performance of the proposed scheme is tested against the performance of the simple CNN-IPS model and the CNN-IPS model, which only use the baseline knowledge distillation scheme, as well as the complex CNN-IPS model. The simulation results should prove that the proposed scheme's execution time is similar to that of the simple CNN-IPS model while also having a better positioning performance than the simple CNN-IPS and the KD-CNN-IPS.

Understanding the KD-CNN-IPS framework in Section 3.2 makes it easier to grasp the fundamental concept of the proposed TAKD-CNN-IPS since the training process is similar to a conventional knowledge distillation. The development of the proposed indoor positioning system was actually inspired by the work presented in [17]. Figure 2 depicts the general block diagram for the proposed TAKD-CNN-IPS. As can be seen from Figure 2, extra intermediate networks are designed to act as teacher-assistant networks. The role of the teacher-assistant networks is to bridge the gap between the teacher network and the student network since the complexity level of the introduced network is in between the complexity of the teacher and the student network.



Figure 2. Block diagram of the proposed TAKD-CNN-IPS.

The initial training step of the TAKD-CNN-IPS is similar to those of the CNN-IPS and KD-CNN-IPS, where the 1D RSSI input vector must first be transformed into a 2D fingerprint image. Then, the teacher network will produce soft labels by using Equation (3). Unlike the KD-CNN-IPS, knowledge from the teacher model will first be transferred to the teacher-assistant model. Therefore, the teacher-assistant model will be trained using the following loss function:

$$L^{\text{TAKD}} = \alpha L_{\text{CE}}(\boldsymbol{z}_{ta}^{n}, \boldsymbol{y}^{n}) + \beta L_{\text{KD}}(\boldsymbol{z}_{ta}^{n}, \boldsymbol{z}_{t}^{n})$$
(7)

where $L_{CE}(z_{ta}^{n}, y^{n})$ represents the cross-entropy loss between the teacher assistant prediction and the ground truth while $L_{KD}(z_{ta}^{n}, z_{t}^{n})$ is the distillation loss. Note that the distillation loss is computed using the logits of the teacher-assistant model z_{ta}^{n} and the logits of the teacher model z_{t}^{n} . In the case of TAKD-CNN-IPS with U teacher-assistant networks, (7) will be applied to train the first teacher-assistant model and then the subsequent teacher-assistant models will adopt the following loss function:

$$L^{\text{TAKD}} = \alpha L_{\text{CE}}(\boldsymbol{z}_{ta_{u}}^{n}, \boldsymbol{y}^{n}) + \beta L_{\text{KD}}(\boldsymbol{z}_{ta_{u}}^{n}, \boldsymbol{z}_{ta_{u-1}}^{n})$$
(8)

where u = 1, 2, ..., U, $L_{CE}(z_{ta_u}^n, y^n)$ and $L_{KD}(z_{ta_u}^n, z_{ta_{u-1}}^n)$ represent the *u*th teacher-assistant loss and distillation loss, respectively. Unlike the first teacher-assistant model, the logits required to compute the distillation loss for the intermediate teacher assistant are calculated using the logits of the *u*th teacher assistant model $z_{ta_u}^n$ and the logits of the (*u*-1) th teacher assistant model $z_{ta_{u-1}}^n$.

Once the training process for the last teacher-assistant model (*U*th teacher-assistant model) completes, the student model will finally be trained by the *U*th teacher-assistant model using the following loss function:

$$L^{\text{TAKD}} = \alpha L_{\text{CE}}(\boldsymbol{z}_s^n, \boldsymbol{y}^n) + \beta L_{\text{KD}}(\boldsymbol{z}_s^n, \boldsymbol{z}_{ta_{ll}}^n)$$
(9)

where z_s^n and $z_{ta_u}^n$ signify the logits of the student and the loss function *U*th teacherassistant model, respectively. Algorithm 1 summarizes the overall localization process of the proposed TAKD-CNN-IPS.

Algorithm 1: Teacher-assistant knowledge distillation

Input: $r^n = \{r_k^n, n \in [1, N], k \in [1, K]\}$: The RSS value of the *n*th sample from the *k*th AP *N*: Total number of samples K: Total number of APs Output: Location class if $K \neq c^2$ where c^2 is an integer then 1 : 2 r^n is padded with 0. 3 end if : 4 r^n is transformed into $X^n =$ 5 for $n \leq N$ do Train the teacher model Employ X^n as the input of the teacher model. 6 7 : Apply (3) to calculate the soft labels ρ_T for the teacher network. Train the teacher-assistant model 8 : for $u \leq U$ do 9 Employ X^n as the input of the teacher-assistant model. 10 Apply (1) to generate the *u*th teacher assistant's hard predictions. Apply (3) to generate the *u*th teacher assistant's soft prediction ρ_{TA_u} . 11 12 Execute (2) to calculate teacher-assistant cross-entropy loss. 13 if u = 1 then 14 Execute (5) to calculate distillation loss. 15 Apply loss function from (7) to train the teacher-assistant model. if u > 1 then 16 17 Execute (5) to calculate distillation loss. 18 Apply loss function from (8) to train the teacher-assistant model. 19 end if end for Train the student model Apply X^n as the input of the student model. 20 21 22 23 24 Apply (1) to generate student's hard predictions. Apply (3) to generate student's soft prediction ρ_S . Execute (2) to calculate student cross-entropy loss. Execute (5) to calculate distillation loss. 25 Apply loss function from (9) to train the student model. 26 end

5. Results and Analysis

In this section, the proposed scheme's performance is thoroughly assessed and compared with the performance of the teacher, student and the baseline knowledge distillation scheme. To examine the proposed technique's applicability, this work utilizes the publicly available Alcala Tutorial 2017 dataset [32], which was collected in the School of Engineering of the University of Alcala when they organized the 2017 Fingerprinting-based Indoor Positioning tutorial. This dataset comprises RSSI readings from a single floor in one building. The number of reference points available in the Alcala Tutorial 2017 dataset is 110, which we labeled as coordinate, while RSSI measurements of every reference point in the database were taken from 162 access points.

5.1. Simulation Setting

The simulations were carried out using Python 3.7.12 and we adopted the Keras framework to model various deep learning-based IPS schemes considered. During the simulations, the 1512 sample points present in the database were split into 90% testing data and 10% training data. As for the networks, four CNN-IPS models with different numbers of convolutional layers, ranging from one to eight, were created. The configurations of all the models considered in this work are shown in Table 2. Referring to Table 2, the size of

the model indicates the number of convolutional layers in that particular model. The model with the highest complexity and most convolutional layers is appointed as the teacher model and is known as CNN-IPS (TM) in this work. Additionally, a student model was designed with the lowest convolutional number, termed CNN-IPS (SM). Other than that, a few model paths were generated so that the performance of the proposed technique, as well as the baseline technique, could be analyzed. We included the KD-CNN-IPS scheme to ensure that the proposed technique performs better than the baseline technique. Since the proposed approach incorporates teacher-assistant networks, we also consider three different models based on the proposed scheme, which are abbreviated as TAKD-CNN-IPS (M1), TAKD-CNN-IPS (M2) and TAKD-CNN-IPS (M3). Table 3 describes the model paths and the hyperparameter of each technique. It is important to note that, unlike the other two TAKD-CNN-IPS techniques, TAKD-CNN-IPS (M3) has two teacher-assistants, and it is trained from the model with the highest complexity to the lowest complexity.

CNN Model	Settings		
Size 8	No of convolutional layers: 8 Filter size: 2×2 No of filters: 32, 32, 32, 32, 128, 128, 128, 128 Activation function after convolutional layers: ReLU No of max pooling layers: 4 Kernel size: 2×2 Strides: 1×1 Fully connected layers: 10368 nodes Hidden layer: 110 nodes Output: 110 nodes		
Size 6	No of convolutional layers: 6 Filter size: 2×2 No of filters: 32, 32, 32, 32, 32, 32 Activation function after convolutional layers: ReLU No of max pooling layers: 3 Kernel size: 2×2 Strides: 1×1 Fully connected layers: 3200 nodes Hidden layer: 110 nodes Output: 110 nodes		
Size 4	No of convolutional layers: 4 Filter size: 2 × 2 No of filters: 32, 32, 32, 32 Activation function after convolutional layers: ReLU No of max pooling layers: 2 Kernel size: 2 × 2 Strides: 1 × 1 Fully connected layers: 3872 nodes Hidden layer: 110 nodes Output: 110 nodes		
Size 1	No of convolutional layers: 1 Filter size: 3×3 No of filters: 16 Activation function after convolutional layers: ReLU No of max pooling layers: 1 Kernel size: 2×2 Strides: 2×2 Fully connected layers: 576 nodes Output: 110 nodes		

Table 2. Configuration of the CNN-IPS network.

Tachniqua	Model Training	Hyperparameters		
Technique	Overall Path	Individual Path	Setting	
CNN-IPS (TM)	Size 8	Size 8	Epochs: 426	
CNN-IPS (SM)	Size 1	Size 1	Epochs: 100	
KD-CNN-IPS		Size 8	Epochs: 426	
	Size 8 -> Size 1	Size 8 -> Size 1	Epochs: 100 <i>T</i> : 2 α: 0.1	
	Size 8 -> Size 4 -> Size 1	Size 8	Epochs: 426	
TAKD-CNN-IPS (M1)		Size 8 -> Size 4	Epochs: 100 <i>T</i> : 2 <i>α</i> : 0.1	
		Size 4 -> Size 1	Epochs: 100 <i>T</i> : 2 α: 0.1	
		Size 8	Epochs: 426	
TAKD-CNN-IPS (M2)	Size 8 -> Size 6 -> Size 1	Size 8 -> Size 6	Epochs: 100 <i>T</i> : 2 α: 0.1	
		Size 6 -> Size 1	Epochs: 100 <i>T</i> : 2 <i>α</i> : 0.1	
		Size 8	Epochs: 426	
		Size 8 -> Size 6	Epochs: 100 <i>T</i> : 2 α: 0.5	
TAKD-CNN-IPS (M3)	Size 8 -> Size 6 -> Size 4 -> Size 1	Size 6 -> Size 4	Epochs: 100 <i>T</i> : 2 <i>α</i> : 0.3	
		Size 4 -> Size 1	Epochs: 100 <i>T</i> : 2 <i>α</i> : 0.1	

Table 3. Model training paths and setting of hyperparameters for various IPS techniques considered.

5.2. Results and Discussion

Before the performance of the proposed schemes can be discussed, it is crucial to understand the impacts of the α and T hyperparameters. Generally, knowledge distillation trains the student model by utilizing both the true label and studying how the bigger network represents and manipulates data. In order to allow the student network to gain more information from the larger network, the temperature scaling hyperparameter T is introduced to soften the peaky probability distribution of the trained larger network. It is done mostly because soft probabilities from trained networks disclose more information about the data, mainly on the relationship between the RSSI and location classes than the actual label. A higher value of T will generate a softer distribution. Hence, T larger than one is able to improve the performance of the student model. In this work, we have selected *T* = 2. As mentioned in Section 4, α is the weight of the student loss, while β is the weight of distillation loss used in the objective function to train the distilled model. Thus, by varying α , users can control the importance of the student loss and the amount of information being distilled from the larger model since β will automatically change as a result. More explicitly, a higher value of α will result in a higher contribution from student loss and a lower contribution from distillation loss, while the opposite will happen with a lower value of α . In order to illustrate how the α hyperparameter affects the positioning

performance of the proposed IPS, the performance gain index has been introduced. The performance gain index *P* is calculated using the following equation to demonstrate the amount of improvement attained by the knowledge-distilled schemes against the student model CNN-IPS (SM).

$$P = \frac{E_{SM} - E_{\rm KD}}{E_{SM}} \tag{10}$$

where E_{SM} and E_{KD} represent the average positioning errors of the CNN-IPS (SM) and knowledge-distilled schemes, respectively.

The graph of the performance gain index against α with T = 2 is displayed in Figure 3. From the figure, it is observed that all the positioning schemes considered exhibit the best performance at $\alpha = 0.1$. This result indicates that the positioning schemes attain better improvement when the contribution from distillation loss is larger. Besides that, it is also noteworthy that the performance gain index at $\alpha = 1$ is zero. This is because at $\alpha = 1$, the models are trained solely by the student loss. Since all the student models in the knowledge-distilled schemes have the same architecture as CNN-IPS (SM), their average positioning errors will be identical to that of the CNN-IPS (SM).



Figure 3. Performance gain index of knowledge distillation scheme for T = 2 with different α .

The performance of the CNN-IPS (TM) and CNN-IPS (SM) in terms of accuracy, loss, average positioning error, minimum, maximum and percentile positioning error is tabulated in Table 4. The average positioning errors for *D* test samples are measured using the Euclidean distance between the predicted location (\hat{x}_d, \hat{y}_d) and the ground truth $(\tilde{x}_d, \tilde{y}_d)$, which can be calculated using the following formula:

$$E = \frac{1}{D} \sum_{d=1}^{D} \sqrt{(\tilde{x}_d - \hat{x}_d)^2 + (\tilde{y}_d - \hat{y}_d)^2}$$
(11)

The results are displayed as a benchmark for the proposed scheme. From the table, CNN-IPS (TM) outperforms CNN-IPS (SM) as expected in training and testing accuracy. More specifically, the training accuracy and testing accuracy of CNN-IPS (TM) exceeds CNN-IPS (SM) by 0.37% and 25%, respectively. Additionally, a similar trend is also observed

for the loss, where there is an improvement of 58.36% and 29.02% when we compare CNN-IPS (TM) to CNN-IPS (SM) in the training and testing phases, respectively. It is also noteworthy that the average positioning error of CNN-IPS (TM) is only 35.9% of that of CNN-IPS (SM). More specifically, there is a performance gap of 1.4288 m between the average positioning error of CNN-IPS (TM) and CNN-IPS (SM).

Phase	Classification Performance	CNN-IPS (TM)	CNN-IPS (SM)
Training	Accuracy	0.9912	0.9875
-	Loss	0.0341	0.0819
Testing	Accuracy	0.7368	0.4868
-	Loss	3.2395	4.5638
	Average positioning error (m)	0.8003	2.2291
	Min positioning error (m)	0	0
	Max positioning error (m)	10.6820	15.1630
	25th percentile (m)	0	0
	50th percentile (m)	0	0.9900
	75th percentile (m)	1.0100	3.0804
	95th percentile (m)	3.9280	8.3371

Table 4. Classification performance comparison between CNN-IPS (TM) and CNN-IPS (SM).

Comparing the performance of CNN-IPS (TM) and CNN-IPS (SM) gives a rough idea of how the size of a network affects the localization performance of CNN-IPS. Next, we will investigate how these knowledge distillation schemes improve the estimation of the CNN-IPS. The training accuracies of all techniques are quite close to each other. In fact, all the methods can achieve a training accuracy of approximately 98%. However, the testing accuracy of the models ranges from 48.68% to 73.68%. Therefore, the testing accuracy clearly indicates which techniques give the greatest and the worst positioning performance. The training accuracy, as well as the average positioning error, of each technique are presented in Figure 4. The training accuracy of all the techniques are quite close to each other, and they manage to achieve at least 98%. Additionally, the average positioning error for each of the techniques is less than 0.05 m. When we compare the positioning performance of the CNN-IPS, KD-CNN-IPS and all of the TAKD-CNN-IPS schemes, the difference in terms of average positioning error is around 0.025 m. For an indoor positioning system, a difference of 0.025 m can be regarded as insignificant. Figure 5 presents the improvement in terms of positioning accuracy of the proposed technique as compared to the baseline CNN-IPS (SM) and the KD-CNN-IPS technique. The figure shows that TAKD-CNN-IPS (M3) is up to 10.53% improved in terms of testing accuracy compared with CNN-IPS (SM). Besides that, the testing accuracies of TAKD-CNN-IPS (M3) and TAKD-CNN-IPS (M2) are superior to those of the KD-CNN-IPS, while TAKD-CNN-IPS (M1) has the same accuracy as KD-CNN-IPS. Hence, this provides evidence that when an intermediate network is introduced, the student network can capture more knowledge because it is trained using the soft distribution produced by a network with a closer capacity.

As shown in Figures 6 and 7, CNN-IPS (SM) is the worst performer in terms of average positioning error. It can be seen that all of the models based on the TAKD-CNN-IPS technique outperform both the CNN-IPS (SM) and KD-CNN-IPS counterparts. As expected, TAKD-CNN-IPS (M3) exhibits the best performance amongst the 3 TAKD-CNN-IPS models considered in terms of the average positioning error. More explicitly, the average positioning error of TAKD-CNN-IPS (M3) is 1.4125 m, which only differs by 0.6122 m from CNN-IPS (TM), while the average positioning error of CNN-IPS (SM) is 1.4288 m, which is inferior to that of the CNN-IPS (TM). Additionally, it is found that the proposed TAKD-CNN-IPS scheme with multiple teacher assistants attains a higher testing accuracy than those with a single teacher assistant due to the fact that TAKD-CNN-IPS with multiple teacher assistants has a better capability to pass helpful knowledge from CNN-IPS (TM) to the student model. Interestingly, although the testing accuracies of TAKD-CNN-IPS (M1)

and KD-CNN-IPS are identical, as indicated in Figure 5, the TAKD-CNN-IPS (M1) is found to exhibit a performance gain of 9.82% over KD-CNN-IPS in terms of average positioning error. This is because the Euclidean distances between the actual and predicted locations of the test samples are misclassified by the TAKD-CNN-IPS (M1) and are much closer than those of the test samples, which KD-CNN-IPS incorrectly predicts.



Figure 4. Training accuracy and average positioning error of CNN-IPS (TM), CNN-IPS (SM), KD-CNN-IPS, TAKD-CNN-IPS (M1), TAKD-CNN-IPS (M2) and TAKD-CNN-IPS (M3).



Figure 5. The difference in testing accuracy for all techniques as compared to CNN-IPS (SM) and KD-CNN-IPS.



Figure 6. Average positioning error for all techniques.



Figure 7. Percentage of improvements in terms of average positioning error for all techniques as compared to CNN-IPS (SM) and KD-CNN-IPS.

The cumulative distribution function (CDF), as shown in Figure 8, provides insight into the distribution of the positioning errors for each technique. From the CDF, it can be seen that all the models based on TAKD-CNN-IPS schemes outperform the KD-CNN-IPS and CNN-IPS (SM) schemes. More explicitly, the probability of positioning errors within 2 m achieved by CNN-IPS (SM) and CNN-IPS (TM) are 0.6184 and 0.8092, respectively. On the other hand, the KD-CNN-IPS scheme and all models based on the proposed TAKD-CNN-IPS technique are capable of achieving better localization performance as compared

to CNN-IPS (SM). For example, the probability of positioning errors within 2 m ranges for KD-CNN-IPS and TAKD-CNN-IPS (M3) are 0.7039 and 0.7434, respectively. Additionally, all of the TAKD-CNN-IPS schemes manage to surpass 90% for the probability of positioning errors within 4 m, while CNN-IPS (SM) and KD-CNN-IPS only attained a probability of positioning errors within 4 m of 82.23% and 88.15%, respectively.



Figure 8. The cumulative distribution function of positioning errors for CNN-IPS (TM), CNNIPS (SM), KD-CNN-IPS, TAKD-CNN-IPS (M1), TAKD-CNN-IPS (M2) and TAKD-CNN-IPS (M3).

Figure 9 illustrates the average execution time of various IPS techniques considered during the testing phase. As anticipated, all the models developed based on the proposed TAKD-CNN-IPS technique can execute much faster than the CNN-IPS (TM). Quantitatively, the testing time required by the proposed TAKD-CNN-IPS schemes is only around 14.91% of that of the CNN-IPS (TM). This performance advantage is because the testing time of CNN is governed by the architecture and complexity of the model. Because the architecture of the student model adopted in TAKD-CNN-IPS is significantly more simple than that of the CNN-IPS (TM), it results in substantially lower computational complexity and shorter execution time compared to CNN-IPS (TM). It is also noteworthy that the average testing time incurred by the TAKD-CNN-IPS models is similar to those of the KD-CNN-IPS and CNN-IPS (SM), which is 0.17 s. This observation can be explained by the following. During the testing phase, the KD-CNN-IPS and CNN-IPS (SM) and the student model for location prediction. Since the architectures of CNN-IPS (SM) and KD-CNN-IPS (M2), TAKD-CNN-IPS (M3) and KD-CNN-IPS are identical, all these techniques result in the same testing time.



Figure 9. Testing time of CNN-IPS (TM), CNN-IPS (SM), KD-CNN-IPS, TAKD-CNN-IPS (M1), TAKD-CNN-IPS (M2) and TAKD-CNN-IPS (M3).

6. Conclusions

Despite the great success of CNN-IPS in achieving impressive positioning performance, it is impractical to implement such cumbersome deep models on resource-constrained devices due to the prohibitively high computational complexity and massive storage requirements associated with CNN-IPS. One possible solution is to apply KD-CNN-IPS to distill essential information acquired by the complex pre-trained CNN-IPS (teacher model) to a lightweight CNN-IPS (student model). However, if the teacher model is far more complicated than the student model, the performance of the KD-CNN-IPS will degrade, as the logits produced by the teacher network will be left soft and the lightweight student model has insufficient capacity to imitate the behavior of the teacher network. To circumvent these issues, this paper proposes a novel TAKD-CNN-IPS, whereby teacher-assistant models are employed as intermediate networks and muti-step knowledge distillation is performed. Extensive simulations have been conducted to evaluate the performance of the proposed technique and the effects of the hyperparameters have been analyzed. The results demonstrate that the proposed TAKD-CNN-IPS techniques are capable of achieving larger performance improvements in terms of average positioning error over the baseline CNN-IPS (SM) as compared to that of their KD-CNN-IPS counterparts. Quantitatively, in terms of an average positioning error, the improvement of the proposed TAKD-CNN-IPS models over the baseline CNN-IPS (SM) ranges from 33.81% to 38.79%, while KD-CNN-IPS only attains a gain of 29.73%. In addition, the testing time incurred by the proposed models is also substantially shorter than that of the CNN-IPS (TM) since the testing time of TAKD-CNN-IPS is only 14.91% of that associated with CNN-IPS (TM). The combined features of excellent localization performance, simple architecture and shorter execution time make the TAKD-CNN-IPS an appealing option for deployment on edge devices, such as smartphones or embedded sensor nodes, to support various real-time indoor applications, which include, but are not limited to, positioning, tracking and navigation.

Author Contributions: Conceptualization, Y.H.N.; methodology, A.B.M.; software, A.B.M.; writing original draft preparation, A.B.M.; writing—review and editing, Y.H.N. and C.K.T.; supervision, Y.H.N.; project administration, Y.H.N. and C.K.T.; funding acquisition, Y.H.N. and C.K.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Higher Education Malaysia, grant number FRGS/1/2019/ICT02/MMU/03/13.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- 1. Gupta, R.; Rao, U.P. An Exploration to Location Based Service and Its Privacy Preserving Techniques: A Survey. *Wirel. Pers. Commun.* 2017, 96, 1973–2007. [CrossRef]
- Statista. Number of Location-Based Service Users in the United States from 2013 to 2018 (in Millions). 2015. Available online: http://www.statista.com/statistics/436071/location-based-service-users-usa/ (accessed on 30 May 2022).
- Mahida, P.; Shahrestani, S.; Cheung, H. Deep Learning-Based Positioning of Visually Impaired People in Indoor Environments. Sensors 2020, 20, 6238. [CrossRef] [PubMed]
- Hung, C.-H.; Fanjiang, Y.-Y.; Lee, Y.-S.; Wu, Y.-C. Design and Implementation of an Indoor Warning System with Physiological Signal Monitoring for People Isolated at Home. *Sensors* 2022, 22, 590. [CrossRef] [PubMed]
- Indoor Positioning and Navigation Market—Forecast (2020–2025). Available online: https://www.researchandmarkets.com/ reports/4531980/indoor-positioning-and-navigation-market#:~{}:text=The%20global%20Indoor%20Positioning%20and,at%20 a%20CAGR%20of%2027.9%25 (accessed on 30 May 2022).
- Mainetti, L.; Patrono, L.; Sergi, I. A Survey on Indoor Positioning Systems. In Proceedings of the 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19 September 2014.
- Obeidat, H.; Shuaieb, W.; Obeidat, O.; Abd-Alhameed, R. A Review of Indoor Localization Techniques and Wireless Technologies. Wirel. Pers. Commun. 2021, 119, 289–327. [CrossRef]
- 8. Gentile, C.; Alsindi, N.; Raulefs, R.; Teolis, C. Geolocation Techniques: Principles and Applications; Springer: New York, NY, USA, 2013.
- 9. Liu, Y.; Yang, Z.; Wang, X.; Jian, L. Location, Localization, and Localizability. J. Comput. Sci. Technol. 2010, 25, 274–297. [CrossRef]
- 10. Yu, X.-M.; Wang, H.; Wu, J. A method of fingerprint indoor localization based on received signal strength difference using compressive sensing. *EURASIP J. Wirel. Commun. Netw.* **2020**, 2020, 72. [CrossRef]
- 11. Zhu, X.; Qu, W.; Qiu, T.; Zhao, L.; Atiquzzaman, M.; Wu, D.O. Indoor Intelligent Fingerprint-Based Localization: Principes, Approaches and Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2634–2657. [CrossRef]
- 12. Basiri, A.; Lohan, E.S.; Moore, T.; Winstanley, A.; Peltola, P.; Hill, C.; Amirian, P.; e Silva, P.F. Indoor location based services challenges, requirements and usability of current solutions. *Comput. Sci. Rev.* **2017**, *24*, 1–12. [CrossRef]
- Al-Ayyoub, M.; Nuseir, A.; Alsmearat, K.; Jararweh, Y.; Gupta, B. Deep learning for Arabic NLP: A survey. J. Comput. Sci. 2018, 26, 522–531. [CrossRef]
- Nagajyothi, D.; Siddaiah, P. Speech Recognition Using Convolutional Neural Networks. Int. J. Eng. Technol. 2018, 7, 133–137. [CrossRef]
- 15. Jiao, L.; Zhao, J. A Survey on the New Generation of Deep Learning in Image Processing. *IEEE Access* 2019, 7, 172231–172263. [CrossRef]
- Jang, J.-W.; Hong, S.-H. Indoor Localization with Wi-Fi Fingerprinting Using Convolutional Neural Network. In Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, 3–6 July 2018.
- Mirzadeh, S.I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; Ghasemzadeh, H. Improved Knowledge Distillation via Teacher Assistant. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
- Bahl, P.; Padmanabhan, V.N. RADAR: An In-Building RF-Based user Location and Tracking System. In Proceedings of the Conference on Computer Communications. nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel Aviv, Israel, 20–26 March 2000.
- Torres-Sospedra, J.; Montoliu, R.; Martinez-Uso, A.; Avariento, J.P.; Arnau, T.J.; Benedito-Bordonau, M.; Huerta, J. UJIIndoorLoc: A New Multi-Building and Multi-Floor Database for WLAN Fingerprint-Based Indoor Localization Problems. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014.
- Torres-Sospedra, J.; Montoliu, R.; Trilles, S.; Belmonte, O.; Huerta, J. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning system. *Expert Syst. Appl.* 2015, 42, 9263–9278. [CrossRef]
- 21. Yim, J. Introducing a decision tree-based indoor positioning technique. Expert Syst. Appl. 2008, 34, 1296–1302. [CrossRef]

- 22. Calderoni, L.; Ferrara, M.; Franco, A.; Maio, D. Indoor localization in a hospital environment using Random Forest classifiers. *Expert Syst. Appl.* **2015**, *42*, 125–134. [CrossRef]
- 23. Wang, P.; Fan, E.; Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognit. Lett.* **2020**, *141*, 61–67. [CrossRef]
- 24. Zhang, W.; Liu, K.; Zhang, W.; Zhang, Y.; Gu, J. Deep Neural Networks for wireless localization in indoor and outdoor environments. *Neurocomputing* **2016**, 194, 279–287. [CrossRef]
- 25. Nowicki, M.; Wietrzykowski, J. Low-Effort Place Recognition with WiFi Fingerprints Using Deep Learning. In *Automation* 2017; Szewczyk, R., Zieliński, C., Kaliczyńska, M., Eds.; Springer: Cham, Switzerland, 2017; Volume 550, pp. 575–584.
- 26. Kim, K.S.; Lee, S.; Huang, K. A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting. *Big Data Anal.* **2018**, *3*, 4. [CrossRef]
- 27. Mitall, A.; Tiku, S.; Pascricha, S. Adapting Convolutional Neural Networks for Indoor Localization with Smart Mobile Devices. In Proceedings of the 2018 on Great Lakes Symposium on VSLI, Chicago, IL, USA, 23–25 May 2018.
- Song, X.; Fan, X.; He, X.; Xiang, C.; Ye, Q.; Huang, X.; Fang, G.; Chen, L.L.; Qin, J.; Wang, Z. CNNLoc: Deep-Learning Based Indoor Localization with Wi-Fi Fingerprinting. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, Liecester, UK, 19–23 August 2019.
- 29. Mazlan, A.B.; Ng, Y.H.; Tan, C.K. A Fast Indoor Positioning Using a Knowledge-Distilled Convolutional Neural Network (KD-CNN). *IEEE Access* 2022, 10, 65326–65338. [CrossRef]
- Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. In Proceedings of the NIPS Deep Learning and Representation Learning Workshop, Montréal, QC, Canada, 12 December 2014.
- 31. Alkhulaifi, A.; Alsahli, F.; Ahmad, I. Knowledge distillation in deep learning and its application. *PeerJ Comput. Sci.* **2021**, *7*, 474. [CrossRef] [PubMed]
- Montoliu, R.; Sansano, E.; Torres-Sospedra, J.; Belmonte, O. IndoorLoc platform: A Public Repository for Comparing and Evaluating Indoor Positioning Systems. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017.