Digital Communications and Networks 11 (2025) 337–348

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

A framework for locating multiple RFID tags using RF hologram tensors ☆,☆☆

Xiangyu Wang [a,c], Jian Zhang [b], Shiwen Mao [a,*], Senthilkumar CG Periaswamy [c], Justin Patton [c]

[a] Dept. of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA
[b] Department of Electrical and Computer Engineering, Kennesaw State University, Kennesaw, GA 30144, USA
[c] RFID Lab, Auburn University, Auburn, AL 36849, USA

ARTICLE INFO

Keywords:
Radio-frequency identification (RFID)
Ultra-high frequency (UHF) passive RFID tag
RF hologram tensor
Indoor localization
Deep learning (DL)
Swin Transformer
Self-supervised learning

ABSTRACT

In this paper, we present a Deep Neural Network (DNN) based framework that employs Radio Frequency (RF) hologram tensors to locate multiple Ultra-High Frequency (UHF) passive Radio-Frequency Identification (RFID) tags. The RF hologram tensor exhibits a strong relationship between observation and spatial location, helping to improve the robustness to dynamic environments and equipment. Since RFID data is often marred by noise, we implement two types of deep neural network architectures to clean up the RF hologram tensor. Leveraging the spatial relationship between tags, the deep networks effectively mitigate fake peaks in the hologram tensors resulting from multipath propagation and phase wrapping. In contrast to fingerprinting-based localization systems that use deep networks as classifiers, our deep networks in the proposed framework treat the localization task as a regression problem preserving the ambiguity between fingerprints. We also present an intuitive peak finding algorithm to obtain estimated locations using the sanitized hologram tensors. The proposed framework is implemented using commodity RFID devices, and its superior performance is validated through extensive experiments.

1. Introduction

Radio-frequency identification (RFID) is a radio-based identification technology, where a reader interrogates RFID tags and identifies each tag through its response. It has been widely used in a variety of applications, including supply chain management, inventory tracking, access control, toll collection, and animal management. Due to its widespread use and the low-cost tags, the RFID technology has recently been expanded to fields such as healthcare and environmental monitoring, thanks to the rapid development of the Internet of Things (IoT). By exploiting the measurements in RFID readings, an increasing variety of functions and applications are emerging based on RFID, e.g., localization [1], gesture recognition [2], vital sign monitoring [3,4], three-Dimensional (3D) human pose tracking [5,6], remote temperature sensing [7], and material recognition [8].

Indoor localization has consistently remained a popular research topic among both existing and emerging applications, owing to its pivotal role in solving position-related problems such as gesture recognition and human pose tracking. The RFID-based localization system is primarily based on two types of measurements: the Received Signal Strength Indicator (RSSI) and the phase angle. SpotOn [9] used RSSI along with a path loss model to perform trilateration. LANDMARC [10] leveraged RSSI readings from reference tags as fingerprints to estimate an unknown tag position via fingerprint matching. The RFID phase angle is extremely sensitive to environmental changes, particularly, to the variations in tag-antenna distance. Recent applications have achieved centimeter-level localization by predicting the Direction of Arrival (DoA) with the received RFID phase angle. SparseTag [1] used a spatial smoothing-based method with a novel sparse RFID tag array to predict angles. RF-Wear [11] achieved a mean error of $8° - 12°$ in tracking angles with a uniform linear array. Moreover, RF-Kinect [12] added a body geometry model to the RF hologram to determine limb orientation and human joint location.

On the other hand, Deep Neural Networks (DNN) have sparked a lot of interest and promise in domains like Computer Vision (CV) and

☆ Peer review under the responsibility of the Chongqing University of Posts and Telecommunications.
☆☆ This work is supported in part by the U.S. National Science Foundation (NSF) under Grants ECCS-2245608 and ECCS-2245607.
* Corresponding author.
E-mail addresses: xzw0042@auburn.edu (X. Wang), jzhang51@kennesaw.edu (J. Zhang), smao@ieee.org (S. Mao), szc0089@auburn.edu (S.C.G. Periaswamy), jbp0033@auburn.edu (J. Patton).

https://doi.org/10.1016/j.dcan.2023.12.004
Received 24 September 2022; Received in revised form 9 December 2023; Accepted 19 December 2023
Available online 28 December 2023
2352-8648/© 2023 Chongqing University of Posts and Telecommunications. Production and hosting by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Natural Language Processing (NLP). To take advantage of the superior classification performance of deep networks, researchers integrate deep networks into fingerprinting systems. Deep autoencoders, for example, have been used to extract WiFi CSI features as fingerprints [13–16]. With a deep residual sharing learning approach, ResLoc [17] further enhanced localization accuracy. CiFi [18] was the first work to leverage a Deep Convolutional Neural Network (DCNN) for indoor localization, where measured AoA images were utilized for training a 6-layer DCNN.

Although the introduction of deep networks improves the efficacy of such indoor localization systems, the inherent challenges of fingerprinting-based localization systems remain unaddressed. To begin with, a huge number of fingerprints are required for mapping the signal characteristics over space. The granularity of fingerprints determines the lowest positioning inaccuracy of the localization system [19]. Even when a classifier predicts the label correctly, errors can still occur due to the coverage gap between two fingerprints when the target is positioned in their middle. On the other hand, collecting a large amount of fingerprints would be time-consuming or even impractical in certain public places, such as shopping malls or airports [20]. Second, the fingerprints utilized in such systems are highly dependent on the equipment configuration. The AoA images utilized in CiFi, for example, are defined by the configuration and setup of the receivers. The measurement offset could not be eliminated clearly [21]. The network must be trained from scratch when a different setup or equipment is deployed. As a result, the transferability of CiFi could be poor. In this paper, we try to decouple data creation from the hardware setup and to find a deep network that can process data from various RFID device configurations. The tag position will be estimated with the deep network using data collected from any type of devices.

In this paper, we propose MulTLoc, a framework for Multiple RFID Tag localization utilizing RF Hologram tensors with DNNs. This approach aims to alleviate the fundamental difficulties of fingerprinting-based methods and to fully leverage the potential of deep neural networks. Radio Frequency (RF) hologram tensors are created using phase readings from antenna pairs of the reader. To generate ground truth tensors for supervised learning, a computer vision sensor (e.g., a Kinect V2) is used. Based on DCNN and Swin Transformer [22], two representative hologram filter networks are investigated with the suggested framework to clean the noisy input hologram tensors by exploiting the spatial relationship between tags. An intuitive peak detection technique will be used to infer the location of RFID tags.

The main contributions made in this paper are summarized as follows.

- To the best of our knowledge, this is the first study to utilize RF hologram tensors to train deep networks for locating multiple tags in a 3D space. The use of RF hologram tensor renders deep networks independent of environmental changes, thus considerably improving the robustness and transferability of the proposed system.
- We implemented two novel deep networks to clean up noisy RF hologram tensors. In the networks, the spatial information between multiple tags is leveraged to suppress the fake peaks that exist in original RF hologram tensors. We begin by introducing a DCNN-based network for cleaning RF hologram tensors. Then a Swin Transformer based network [22] is also proposed to filter RF hologram tensors. When the Swin Transformer is being trained, self-supervised learning is utilized to extract general features from hologram tensors. Position estimation is reduced to a simple peak detection problem that can be solved quickly with the sanitized hologram tensor.
- A prototype of the proposed MulTLoc framework is created using Commercial Off-The-Shelf (COTS) RFID devices. With extensive joint localization experiments, the performance of the proposed framework is evaluated. The experimental results show that the MulTLoc framework is capable of simultaneously localizing multiple tags in a 3D space with high accuracy.

The remainder of this paper is organized as follows. We present an overview of related works in Section 2. Section 3 introduces the preliminaries and motivation of our approach. We present the MulTLoc design in Section 4 and our experimental study in Section 5. Section 6 concludes this paper.

2. Related works

With the development of mobile communication technology over the last decade, academia and industry have paid close attention to location-based services. Signal processing has long been used to determine the position of a signal source by estimating the Time-of-Flight (ToF), Angle-of-Arrival (AoA), or a third signal parameter such as Doppler shift and Angle-of-Departure (AoD) [23–27]. The accuracy of parameter estimate, however, is governed by the number of antennas (for AoA) and the transmission bandwidth (for ToF), which are often fixed in a certain wireless communication system. As a result, the cost of improving parameter estimate would be prohibitively high.

On the other hand, the fingerprinting method, with its convenience and effectiveness, transforms the localization problem into a feature matching one. Researchers are working on two tracks to increase the accuracy of fingerprinting-based localization. First, more and more powerful classification algorithms are introduced. For example, K-Nearest Neighbors algorithm (KNN) and its modifications are commonly leveraged in indoor localization systems [28–31]. Machine learning algorithms, such as Random forest [32,33] and AdaBoost [34,35], are often used in promoting the performance of classification as well. Another important aspect influencing the localization accuracy of the fingerprinting-based localization system is the quality of fingerprints. Principal Component Analysis (PCA) is a common tool to extract features from the original fingerprints [36,37] for enhancing the fingerprint quality. Recently, with the development of deep learning, deep autoencoder has been implemented as feature extractors [15,38]. Motivated by the superior performance of DNNs for image classification tasks, feature extraction and classification can be unified in fingerprinting-based localization systems using DNNs. For example, features from AoA images were extracted and classified in CiFi [18] and ResLoc [39] in one effort. Despite the continuous development of new techniques aimed at enhancing performance, the inherent limitation of the fingerprinting method still needs to be addressed. The localization accuracy of this method is reliant on the resolution of fingerprints, and any changes in fingerprints would necessitate an update of the system. In this research, we attempt to present a novel framework to overcome fingerprint-related issues.

DNN, as previously stated, has been frequently used in indoor localization systems because of its excellent feature extraction and classification capabilities. It has evolved over the last decade to meet the needs of various downstream jobs. Since the debut of AlexNet [40], DCNN has become a superstar in computer vision. ResNet [41] constructs a DCNN with hundreds of layers by utilizing shortcut connections. Hourglasss [42] and U-Net [43] use an encoder-decoder design to keep high-resolution representation of images. To date, CNN remains a key model for addressing computer vision challenges. In the area of NLP, Recurrent Neural Networks (RNN) are prominent for dealing with temporal sequence data [44]. Transformer [45] has recently emerged as a dominating successor by using an attention method to construct the global interdependence between input and output. The transformer has also been applied for computer vision tasks. Vision Transformer (ViT), Swin Transformer, and their modifications [46,47] keep improving the state-of-the-art performance in various CV tasks. In the proposed framework, we deploy two representative networks, DCNN and Swin Transformer, to implement the hologram filter network.

3. Preliminaries and motivation

3.1. RFID phase model

Sensitive and trustworthy measures from the original RFID readings are more useful for locating RFID tags in real-time. In contrast to RSSI, the phase value is commonly used in many RFID-based sensing applications [48,49,7]. As shown in (1), the phase reading $\theta_{i,m}$ on channel i from antenna m is a periodic function with a period of 2π.

$$\theta_{i,m} = \mathrm{mod}\left(\frac{4\pi \left|TA_m\right|}{\lambda_i} + \theta_{tag} + \theta_{eq}, 2\pi\right) \tag{1}$$

where $\left|TA_m\right|$ denotes the distance between tag T and antenna A_m, λ_i is the wavelength of channel i, and θ_{tag} and θ_{eq} are the phase offsets caused by the RFID tag and RFID hardware such as antenna and reader, respectively. Here θ_{eq} is a constant for a given RFID system; hence it can be calibrated and removed conveniently.

3.2. Hologram tensor

Tagoram [50] is the first work to introduce the concept of an RF hologram for RFID localization. The primary concept underlying an RF hologram is to compute the similarities between theoretical and measured phase values for each grid in the surveillance space. To eliminate the tag-related phase offset, i.e., θ_{tag} in (1), we use phase difference, instead, as the observation in our system. The measured phase difference, obtained with the phase values collected from a pair of antennas (m,n) on channel i, is denoted as

$$p_{i,m,n} = \mathrm{mod}(\theta_{i,m} - \theta_{i,n}, 2\pi) \tag{2}$$

When the coordinates of both antennas are known, the theoretical phase difference between antenna pair (m,n) on channel i can be determined. The theoretical phase difference for a tag located at grid position $G_{x,y,z}$ from antenna pair (m,n) is given by

$$q_{i,m,n}^{x,y,z} = \mathrm{mod}\left(\frac{4\pi \left|G_{x,y,z}A_m\right|}{\lambda_i} - \frac{4\pi \left|G_{x,y,z}A_n\right|}{\lambda_i}, 2\pi\right) \tag{3}$$

With the measured and theoretical phase differences, their similarity, $S_{x,y,z}$, can be estimated as follows:

$$S_{x,y,z} = \sum_{(M,N)} \sum_{I} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\delta_{i,m,n}^{x,y,z})^2}{2\sigma^2}\right)$$
$$\delta_{i,m,n}^{x,y,z} = \mathrm{mod}\left(p_{i,m,n} - q_{i,m,n}^{x,y,z}, 2\pi\right) \tag{4}$$

where (M,N) represents the set consisting of all available antenna pairs, and I denotes the set of all available channel indices. The hologram tensor, $\mathbf{S}$, is constructed as

$$\mathbf{S} = \begin{bmatrix} S_{1,1,z} & S_{1,2,z} & \cdots & S_{1,y,z} \\ S_{2,1,z} & S_{2,2,z} & \cdots & S_{2,y,z} \\ \vdots & \vdots & \ddots & \vdots \\ S_{x,1,z} & S_{x,2,z} & \cdots & S_{x,y,z} \end{bmatrix}, \ z = 1,2,\ldots,Z \tag{5}$$

where each element is scaled to have a value in $[0,1]$ in the proposed system.

3.3. Motivation

MulTLoc is, to the best of our knowledge, the first effort to train deep learning models for real-time 3D localization using hologram tensors. Although some indoor localization systems, e.g., [18,39,51], use RF signals to produce images or tensors for offline training, the generated data may lack a strong relationship between the observation and the spatial location. In these applications, images and tensors are employed as fingerprints, and deep networks are used as classifiers. The ambiguity between fingerprints may be lost throughout the dataset construction process, thus restricting the transferability of the localization model. In comparison to the images and tensors in the preceding studies, the hologram tensor is *interpretable*. The hologram tensors represent the possibility of a tag located at a grid position in the surveillance space. The similarity S is directly connected to the distances between the tag and antennas, and it exhibits a high degree of independence from the equipment utilized to generate the tensor.

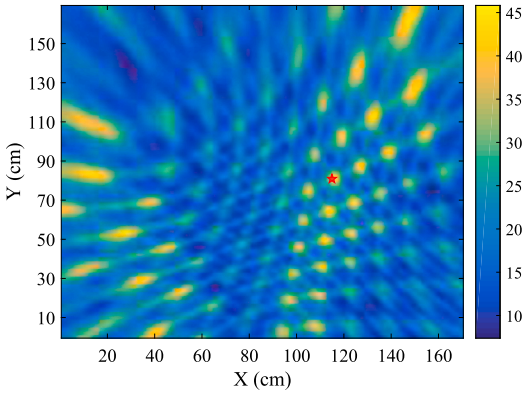


Fig. 1. Hologram of a 2D scenario. The red pentagram denotes the ground truth and a brighter color indicates higher similarity.

A hologram matrix formed in a two-dimensional area is shown in Fig. 1. It displays the two-dimensional projection of the hologram matrix. The exact location of the target tag is indicated by the red pentagram. As can be seen, there is a peak at the position of the ground truth. However, because of the multipath and phase wrapping effects, multiple fake peaks are also formed and distributed across the hologram. Some of the fake peaks have even greater similarity values. To avoid such issues, data prepossessing has become a key component of many RFID-based sensing systems. Some ways improve accuracy at the expense of real-time performance. Channel selection [1] and phase sanitation [52], for example, are used to keep systems away from phase readings tainted by the multipath effect. Such approaches, however, may be impractical for real-time localization systems. This is because multiple-round interrogations are required, and the tag (or target) will not remain stationary until the system performs a sufficient number of interrogations.

Moreover, some applications rely on specific hardware and deployment, such as the synthetic-aperture array [53] and multi-resolution filtering [54], to mitigate the detrimental effect caused by the phase wrapping ambiguity. Although these technologies provide sufficient precision and real-time performance, the requirement for customized hardware increases costs and restricts compatibility with COTS RFID systems. Furthermore, tag localization in 3D spaces is a more difficult challenge than in 2D spaces. In this paper, we present two unique neural networks that have been integrated into our proposed framework to address these challenges.

4. Overview of the MulTLoc system

In this paper, we present MulTLoc, an RFID-based localization framework for estimating the location of multiple tags *simultaneously* in 3D space utilizing noisy hologram tensors. Although MulTLoc, like numerous previous deep learning-based localization systems, is trained using ground truths provided by sensors like an RGB-D camera, the localization problem is treated as *regression* in this study. To estimate the coordinates of unknown sites, traditional fingerprinting methods leverage deep neural networks to treat location estimation as a *classification* problem. The size of the fingerprint database limits the accuracy of localization, and the granularity of the fingerprints determines the inherent inaccuracy of the system. The network would not give location

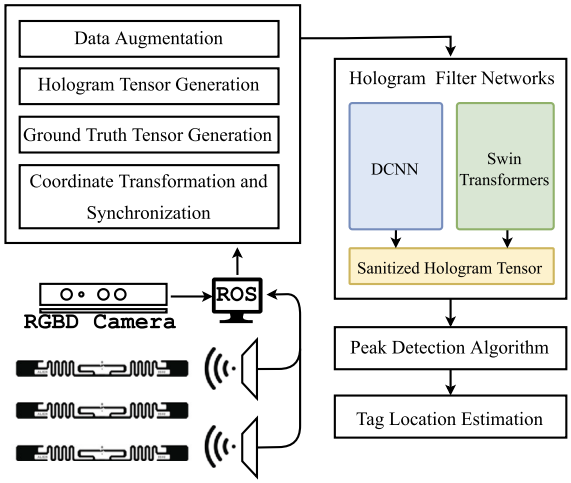


Fig. 2. The MulTLoc system architecture.

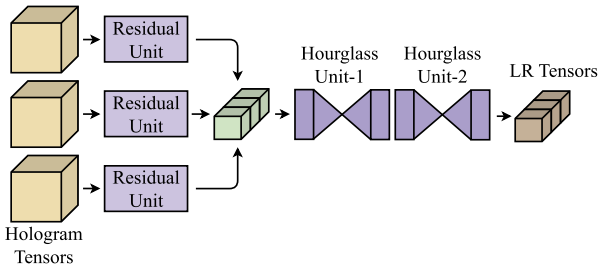


Fig. 3. Architecture of the DCNN based hologram filter network.

estimation instantaneously in the MulTLoc framework. Instead, noisy hologram tensors are regressed to single-peak hologram tensors, which are free from the fake peaks introduced by multipath and phase wrapping effects. Location estimation is then performed intuitively using the sanitized hologram tensor.

4.1. MulTLoc system architecture

Fig. 2 depicts the MulTLoc architecture. An RFID system collaborates with a vision-based sensor to generate hologram tensors and the accompanying ground truth tensors, respectively, for training the hologram filtering networks. Because the hologram tensors and ground truth coordinates provided by the vision-based sensor are typically in distinct coordinate systems, our proposed framework uses the Robot Operating System (ROS) to synchronize and unify the data acquired from diverse hardware. Generally, any deep neural network capable of sanitizing noisy hologram tensors would be compatible with MulTLoc. We utilize two typical neural networks in this paper to evaluate the performance of the proposed framework. Finally, the unknown tag position can be induced conveniently using a simple peak detection algorithm with the sanitized hologram tensors.

Based on the proposed framework, we deploy two representative deep network models for creating the hologram filter network. First, a DCNN-based hologram filter network is designed with the hourglass backbone to clean and compress noisy hologram tensors. To recover the original size of the hologram tensor, trilinear interpolation or equivalent approaches could be used before peak detection. Data augmentation is used to prevent overfitting in the training of the DCNN-based hologram filter network. Further, the DCNN network architecture in this paper is related to the channel of input tensors. The architecture shown in Fig. 3 is for locating three tags simultaneously, where three residual units are used to sanitize the hologram tensors from three tags. Additional residual units are required to cope with more channels, resulting in the expansion of network architecture. To resolve the issue, another hologram filter network is also proposed with the Swin Transformer for keeping the architecture stable for taking care of the tensors from more tags. The output of the network keeps the original size of the input tensors, which will be directly adopted for location prediction. Self-supervised learning is deployed in the training to extract latent features from noisy hologram tensors. Once the networks have been properly trained, the vision-based sensor will no longer be required when the system is applied for location estimation.

4.2. Training dataset generation

In order to successfully train the deep networks, it is essential to label the hologram tensor with the relevant ground truth tensor. However, the ground truth coordinates and hologram tensors are acquired by different sensors with distinct coordinate systems. The reported coordinates for most vision-based sensors are normally determined by the coordinate origin of the sensor space. For example, the center of the depth sensor is the origin of the coordinates for Kinect V2, whereas the surveillance space determines the coordinates of the antennae of MulTLoc. ROS is used in MulTLoc to integrate the hologram tensors from the RFID system and the coordinates from the vision-based sensor, to label hologram tensors with accurate ground truth tensors. We transfer all coordinates from the vision-based sensor into the frames of the hologram tensors depending on the sensor pose and position in the surveillance space. To ensure synchronization, timestamps are appended to both the hologram tensors and the ground truth coordinates. The coordinates with the most recent timestamp will be assigned an RF hologram tensor.

The ground truth tensor, $\mathbf{K}$, is constructed using a Gaussian kernel. Based on the synchronized ground truth coordinates by measuring the Euclidean distance $\left|G_{x,y,z}H\right|$ between the grid location $G_{x,y,z}$ and the ground truth location H, the ground truth tensor, $\mathbf{K}$, is formulated as

$$\mathbf{K}=\begin{bmatrix} K_{1,1,z} & K_{1,2,z} & \cdots & K_{1,y,z} \\ K_{2,1,z} & K_{2,2,z} & \cdots & K_{2,y,z} \\ \vdots & \vdots & \ddots & \vdots \\ K_{x,1,z} & K_{x,2,z} & \cdots & K_{x,y,z} \end{bmatrix},\ z=1,2,...,Z \tag{6}$$

Each element of $\mathbf{K}$ is given by

$$K_{x,y,z}=\frac{1}{\epsilon\sqrt{2\pi}}\exp\left(-\frac{\left|G_{x,y,z}H\right|^2}{2\epsilon^2}\right) \tag{7}$$

where ϵ controls the radius of the ground truth peak. In the MulTLoc framework, $\mathbf{K}$ supervises the training of hologram filter networks. To coordinate the compressed output of the DCNN-based hologram filter network, the ground truth tensor $\mathbf{K}$ is downsampled. Since the hologram tensor is interpretable spatially, our training dataset is augmented by the flipping and rotating operations in the training of the DCNN based hologram filter network.

4.3. Design of DCNN for filtering hologram tensors

As shown in Fig. 3, a DCNN-based hologram filter network is introduced to remove fake peaks from hologram tensors. Compared to the changing settings that compromise the efficacy of fingerprinting-based localization systems, the positional connection between tags is relatively constant, especially for passive tags attached to items. The hologram filter network is intended to learn the spatial connection between tags in order to differentiate the real peaks in RF hologram tensors. We downsample the hologram tensors from n tags and concatenate them into an n-channel tensor using residual units to reduce the amount of weights in the proposed network and accelerate training. The newly created n-channel tensor retains the detailed information in the

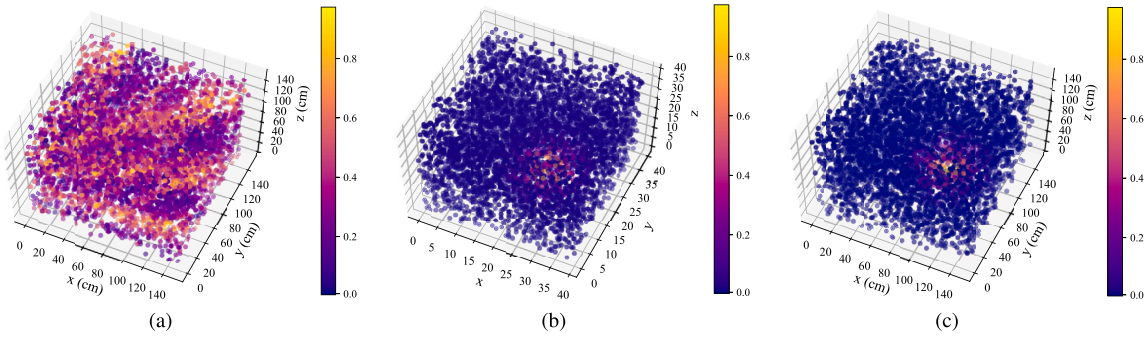


Fig. 4. (a) The original hologram tensor. (b) The filtered hologram tensor. (c) The full-size ground truth tensor.

original hologram tensors while also including a coherent understanding among the tags. In our following experiments, n is set to three, i.e., to locate three tags at the same time.

The residual unit of the hologram filter network consists of two residual blocks [55], each consisting of two three-dimensional convolutional layers. The hourglass blocks [42] are arranged end-to-end following the residual blocks as the backbone of the hologram filter network to extract features in the n-channel tensor at different sizes. The design of the hourglass unit is comparable to that of an encoder-decoder network, as shown in Fig. 3. The input tensor is first compressed and then upscaled in the unit. The bottom-up, top-down inference is repeated by stacking the hourglass units. By computing the loss between the ground truth tensors and the output tensors, the deep network is optimized with the Adam algorithm. We will discuss the selection of loss function in the following section. For accelerating training, intermediate supervision is applied at each hourglass unit in the DCNN-based hologram filter network. The hologram filter network produces a low resolution, n-channel tensor (i.e., the LR Tensor), which is divided into n low resolution hologram tensors for location estimation.

Fig. 4(a) and Fig. 4(b) display the input and output of the hologram filter network, respectively. Lower similarity values are shown as bluish pixels in the figures. The RF hologram tensor is produced using the phases gathered from our testbed, which covers a region of dimension 1.5 m×1.5 m×1.5 m (see Section 5.1 for details). Fake peaks spread out in the original hologram tensor, similar to the hologram matrix in the two-dimensional case. Although the space has four bands with greater similarity values, no clear peak can be recognized. The hologram filter network generates the sanitized hologram tensor, shown in Fig. 4(b), by mixing the holograms from the three tags. The majority of the fake peaks in the input tensor have now been muted. The single bright spot is in the center of the filtered hologram tensor, which is similar to that in the ground truth tensor shown in Fig. 4(c). The spatial connection among the tags has now helped to retrieve the accurate location information concealed among fake peaks.

4.4. Design of Swin Transformers for filtering hologram tensors

However, the cleaned tensor is compressed by the DCNN based network in Fig. 4(b). Because the input of our networks is a 4D tensor (i.e., it consists of 150×150×150×3 pixels in the following experiment), and the number of parameters escalates with the use of 3D convolution, the output size is reduced to save memory while utilizing the DCNN as the backbone. Data compression appears to be at the expense of location estimation accuracy. Furthermore, the number of input channels (i.e., the number of tags to locate) determines the architecture of the DCNN network, to deal with tensors from three tags, three residual units are included. Therefore, the compatibility of the framework remains limited when the DCNN backbone is used.

To address the issue, we employ a Swin Transformer-based network to sanitize the noisy hologram tensors. In our framework, this approach is functionally equivalent to the DCNN-based network but offers several advantages. The Swin Transformer backbone not only exhibits robustness to different sizes of the input tensor, but also produces output with the same size as the input tensor. Fig. 5 illustrates the architecture. The network is a 3D variation of the U-Net [43] with a Swin Transformer backend. The input tensor is first split into non-overlapping 3D tokens, which are subsequently fed into the Swin Transformer blocks. Our implementation sets the patch size to $2\times2\times2$. It consists of a raw feature dimension of $2\times2\times2\times3$ associated with the multi-tag hologram tensor with three channels. The raw features are projected into a 48-dimensional space using a linear embedding layer, which is consistent with the traditional Swin Transformer. Then, the processed tokens are applied with Swin Transformer blocks.

Fig. 6 plots the shifted window based self-attention in the Swin Transformer blocks, where W-MSA and SW-MSA represent the regular window based Multi-head Self-Attention (MSA) and shifted window based MSA, respectively. A LayerNorm (LN) layer is adopted before each MSA and MLP. The tokens are first partitioned into small cubes in the Swin Transformer block. For example, the token of $H\times W\times D$ would be divided into $\frac{H}{M}\times\frac{W}{M}\times\frac{D}{M}$ cubes with a window of $M\times M\times M$. In the following block, the window would shift $(\frac{M}{2},\frac{M}{2},\frac{M}{2})$ pixels, so that the connection between neighboring non-overlapping windows in the previous block would be introduced in the network. With this approach, the output of two consecutive Swin Transformer blocks is computed as

$$\begin{aligned}\hat{s}^{l} &= \text{W-MSA}(\text{LN}(s^{l-1})) + s^{l-1}\\ s^{l} &= \text{MLP}(\text{LN}(\hat{s}^{l})) + \hat{s}^{l}\\ \hat{s}^{l+1} &= \text{SW-MSA}(\text{LN}(s^{l})) + s^{l}\\ s^{l+1} &= \text{MLP}(\text{LN}(\hat{s}^{l+1})) + \hat{s}^{l+1}\end{aligned} \tag{8}$$

The self-attention is given by

$$\text{Attention}(Q,K,V) = \text{SoftMax}\left(\frac{QK^{T}}{\sqrt{d}} + B\right)V \tag{9}$$

where Q, K, V stand for queries, keys, and values, respectively; d is the scale-down factor; and B is the relative position bias. A patch merging layer always follows the Swin Transformer blocks for shrinking the size of the features by a factor of 2 in each stage. The output of each stage will not only be passed on to the subsequent stage, but also be fed into the DCNN-based decoders to regenerate the filtered hologram tensor.

The feature representation from the Swin Transformer backbone is firstly adjusted through a convolutional encoder before it is concatenated with the features from the decoder of the lower layer. The convolutional decoder processes the merged features, and the output returns to the higher layer. In our implementation, the filtered hologram tensor is obtained directly with the convolutional decoder from the top layer. To supervise the training of the network, a mixed loss function is formulated as

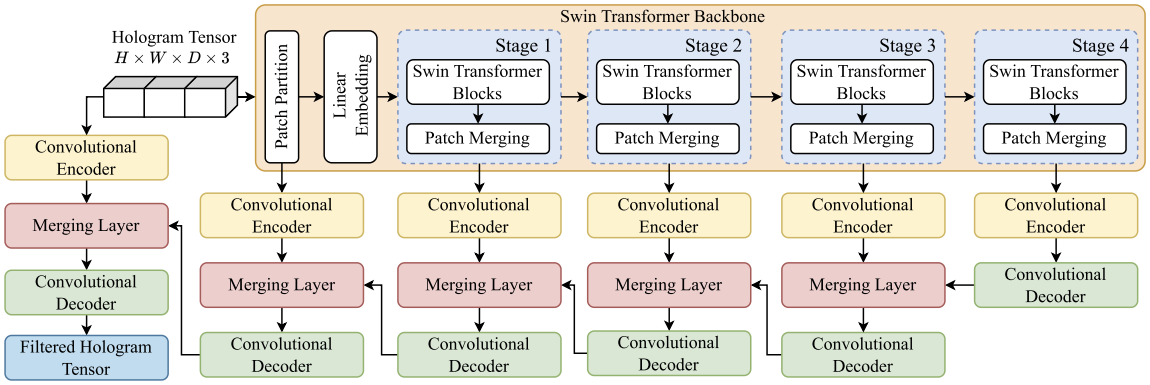


Fig. 5. Architecture of the Swin Transformer based hologram filter network.

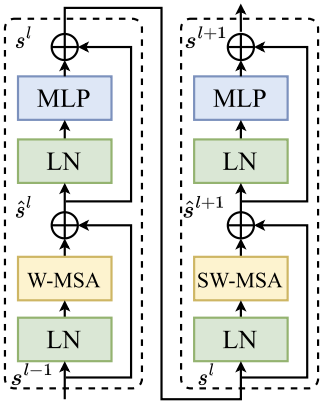


Fig. 6. Swin Transformer blocks.

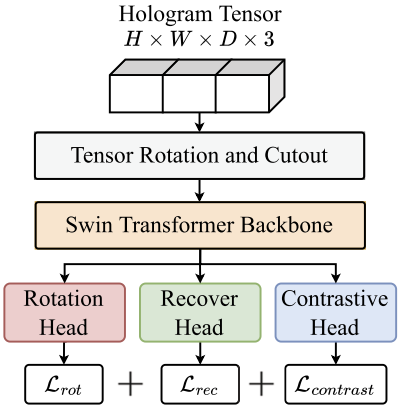


Fig. 7. Architecture of self-supervised pre-training.

$$\mathcal{L}_{mix} = \alpha \times \mathcal{L}_{\text{MS-SSIM}} + (1-\alpha) \times \mathcal{L}_{\ell_1} \tag{10}$$

where $\mathcal{L}_{\text{MS-SSIM}}$ is the multiscale structural similarity index, $\mathcal{L}_{\ell_1}$ represents the ℓ_1 loss, and α is a hyper-parameter [56].

4.5. Self-supervised pre-training of the Swin Transformer

Self-supervised pre-training has made a significant contribution to the development of cutting-edge models for a wide range of NLP tasks. Recent research has identified numerous self-supervised methods aimed at enhancing the capacity of deep neural networks in leaning feature representations for vision tasks as well [57,58]. In this paper, self-supervised pre-training is leveraged with the Swin Transformer-based hologram filter network to promote the performance of sanitizing noisy tensors in the proposed framework.

According to Fig. 7, we adopt three pretext losses to facilitate effective data representation learning in self-supervised pre-training, which are inspired by the prior work for medical image analysis [59]. The input hologram tensor **S** is firstly cropped and rotated to generate sub-volumes randomly. The Swin Transformer backbone is utilized to extract feature representations from sub-volumes. Simultaneously, three different projection heads are employed to fulfill the corresponding pretext tasks. The first task is to predict the angle rotation of the sub-volumes in six classes including 90°, −90° along with the x-axis, y-axis, and z-axis, respectively. The cross-entropy loss is utilized as follows:

$$\mathcal{L}_{rot} = -\sum_{m=1}^{6} r_m \times \log(\hat{r}_m) \tag{11}$$

where $\hat{r}_m$ is the SoftMax result from the rotation head, and r_m is the ground truth.

Tensor recovering task is also a part of self-supervised pre-training. We mask out a portion of pixels in the sub-volumes with a ratio s. The sub-pixel convolution in the recover head regenerates the masked pixels with the feature representation from the Swin Transformer backbone. The MSE loss $\mathcal{L}_{rec}$ is leveraged to measure the difference between the ground truth sub-volume S_{sub} and the recovered sub-volume $\hat{S}_{sub}$, which is given as

$$\mathcal{L}_{rec} = \frac{1}{P}\sum (S_{sub} - \hat{S}_{sub})^2 \tag{12}$$

where P is the number of pixels in the sub-volume.

Contrastive learning [57] is also a part of self-supervised training in the proposed framework. We leverage a simple instance discrimination task as the pretext task. Two correlated sub-volumes of the input hologram tensor, $\mathbf{s}$ and $\mathbf{s}_+$, are generated with the tensor rotation and cutout at first. With the Swin Transformer backbone and the contrastive head, the feature representations are extracted from $\mathbf{s}$ and $\mathbf{s}_+$ and denoted as q and k_0. For a minibatch of N tensors, only the feature representation from the same input tensor is treated as positive pair, while the feature representation $\{k_1, k_2, ..., k_{N-1}\}$ from the rest $N-1$ tensors are the negative examples. Apparently, the instance discrimination task is actually an N-way classification problem, which tries to classify q as k_0. Thus, the contrastive loss function, called InfoNCE [60], is defined as

$$\mathcal{L}_{contrast} = -\log\left(\frac{\exp(q \cdot k_0/\alpha)}{\sum_{m=0}^{N-1}\exp(q \cdot k_m/\alpha)}\right) \tag{13}$$

where the dot product is implemented to measure the similarity between the feature representations, and α is a temperature hyper-parameter.

Finally, the Swin Transformer backbone is self-supervised by minimizing the complex loss function, which is defined as follows:

$$\mathcal{L} = \mathcal{L}_{rot} + \mathcal{L}_{rec} + \mathcal{L}_{contrast} \tag{14}$$

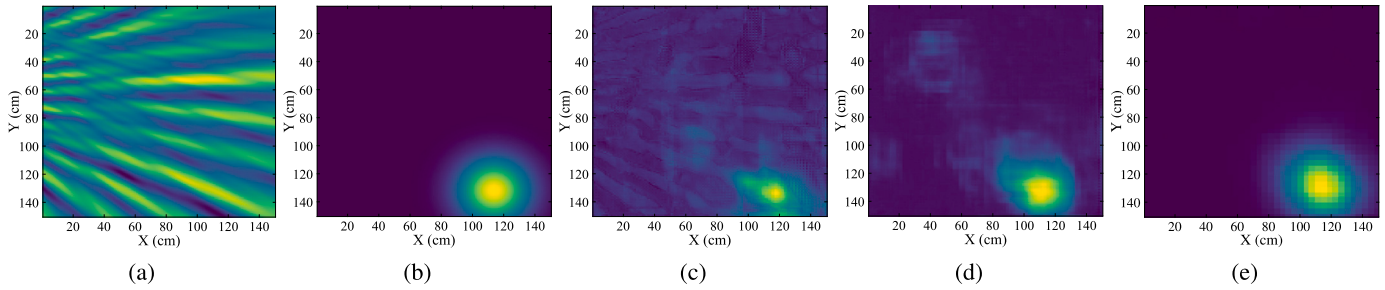


Fig. 8. (a) A slice of the input tensor. (b) A slice of the ground truth tensor. (c) A slice of the sanitized tensor using self-supervised pre-training. (d) A slice of the sanitized tensor using supervised learning. (e) A slice of the sanitized tensor obtained with the DCNN-based network.

Our Swin Transformer-based hologram filter network is fine-tuned using regular supervised learning after self-supervised pre-training. Fig. 8 depicts the progress brought by self-supervised pre-training. Fig. 8(a) and Fig. 8(b) show a noisy hologram tensor slice and the corresponding ground truth slice. In Fig. 8(c), a sanitized slice of the hologram tensor is generated using the pre-trained weight, while Fig. 8(d) shows the slice sanitized by the network without the self-supervised pre-training. By comparing Fig. 8(a) and Fig. 8(c), we notice that the stripe pattern in Fig. 8(a) is extracted and recovered in Fig. 8(c). The peak spot locates at one of the stripes in the slice, which is consistent with our observation in Fig. 1. The stripe pattern, however, vanishes in Fig. 8(d). Instead, the shadow area in Fig. 8(d) is consistent with the blur area in Fig. 8(a). It appears that the network learns how to sanitize the tensor via a "shortcut," which is not what we expect. Furthermore, the area of the peak spot in Fig. 8(c) is significantly more focused than that in Fig. 8(d). This result verifies that self-supervised pre-training is able to improve location estimation by extracting detailed and interpretative feature representations from noisy tensor inputs.

Furthermore, Fig. 8(e) displays a slice of the sanitized tensor using the DCNN-based hologram filter network. Compared to the previous slices, Fig. 8(e) is almost identical to the ground truth slice in Fig. 8(b). Even though the slice is much cleaner than the slices from the Swin Transformer, the details from the original input tensors are lost through the DCNN-based network. It is difficult for us to discover how the network cleans up the tensor in the forward propagation. The phenomenon exhibits the difference between the DCNN backbone and the Swin Transformer backbone. The Swin Transformer has a stronger representation capacity, since the effective features are retained in Fig. 8(c). However, it usually suffers from data shortage because of the lack of the typical convolutional inductive bias. On the other hand, the DCNN backbone performs well even with a dataset of limited size. The detailed comparison between the two backbones will be presented in the following section.

4.6. Location estimation

The tag location can be readily inferred using a simple peak detection algorithm applied to the sanitized hologram tensors. Because the sanitized tensor from the DCNN-based hologram filter network is compressed, we employ trilinear interpolation to recover its size. The estimated location $\widehat{G}$ is computed as follows:

$$\widehat{G} = \left\{ G \mid f(\mathbf{S}_R, G) = \max(\mathbf{S}_R) \right\} \tag{15}$$

where $f(\cdot)$ extracts the similarity value at the grid location G from the sanitized hologram tensor $\mathbf{S}_R$.

5. Experimental study

5.1. Testbed configuration

To evaluate the performance of the proposed framework, we create a prototype using a Zebra FX9600 reader and eight Zebra AN720 antennas, as shown in Fig. 9. Besides, three UPM Raflatac Frog 3D tags are utilized as localization targets, which are attached to the body of a subject. In the experiment, we evaluate the proposed framework by concurrently localizing the three tags to ensure real-time performance, taking into account that a commercial RFID reader can interrogate tags at a rate of roughly 500 Hz [61]. A Kinect V2 device collaborates with a 3D human position estimation algorithm [62] to produce ground truth coordinates for supervised learning. For dataset creation and tag position estimation, the target tags are attached to the two shoulders and the neck. ROS Kinetic Kame is utilized to synchronize and unify the coordinates and tensors from Kinect V2 and the RFID reader. We adjust the requirement for hologram tensor creation to ensure real-time performance of the proposed framework. When five antenna pairs are available, the phases from seven channels will be used to construct the hologram tensors. The surveillance space of the testbed covers a space of dimension 1.5 m × 1.5 m × 1.5 m at a 0.5 m height above the ground. The grid size is set to 1 cm. Furthermore, the similarities at each grid position in the surveillance space are computed in parallel using CUDA GPU programming to speed up the construction of hologram and ground truth tensors.



Fig. 9. The MulTLoc testbed setup.

To train the deep networks, we collect phase readings and related coordinates from several volunteers, each having three tags attached to the two shoulders and the neck, and moving randomly in the surveillance space. Three hundred groups of data are included in the dataset. As illustrated in Algorithm 1, two tensors from the shoulder tags and one tensor from the neck tag (step 4) are first generated with the collected phase readings with eqs. (2), (3) and (5) and section 3.2. The corresponding ground truth tensors (step 6) are created using eqs. (6) and (7). Once the tensor-label pairs are ready, a general supervised training method is able to optimize the parameters for both deep networks (steps 13-18). The pseudocode for location prediction is presented in Algorithm 2. Similarly to offline training, the collected phase readings are used to build hologram tensors as deep network inputs (step 4). The filtered tensor is then directly produced using the well-trained networks (step 6). Trilinear interpolation is implemented to recover the size of the filtered tensors (steps 8). Finally, the location prediction of our MulTLoc system is the coordinates of the pixel with the

highest value in each filtered tensor (step 12). It should be noted that our proposed MulTLoc is compatible with the Swin Transformer backbone. As a result, self-supervised pre-training might be easily adapted to improve interpretative feature representations. The pseudocode for the self-supervised pre-training is provided by Algorithm 3. In each mini-batch, the hologram tensors, which are formed in the same manner as previous methods, are cropped and rotated to produce sub-volumes as the labels for self-supervised training (step 8). After the pre-training is completed, the optimized parameter w would be loaded for the general training of the Swin Transformer-based hologram filter network. For both schemes, the acquired data is divided randomly for training, validation, and testing. Specifically, 80% of the tensor groups are used to train the deep neural networks. The training dataset includes seven hundred and twenty RF hologram tensors in total, while the remaining sixty tensor groups are evenly separated for validation and testing. An Nvidia RTX3090 GPU and an RTX A6000 GPU are utilized to accelerate the computation of the two deep networks.

Algorithm 1: Weights Training of MulTLoc System.

Input: network architecture D, target area A, Max_epoch, phase groups $\phi = \{\theta_1, \theta_2, \theta_3\}$, ground truth locations $\mathcal{H} = \{H_1, H_2, H_3\}$,
Output: Trained parameters w;
//Generate hologram tensor groups $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3\}$ with collected phases ϕ in target area A;
$\mathcal{S} = HologramGenerator(\phi, A)$;
//Generate ground truth tensor groups $\mathcal{K} = \{\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3\}$ with tag locations $\mathcal{H}$;
$\mathcal{K} = truthGenerator(\mathcal{H}, A)$;
if *Swin Transformer is used as backbone* **then**
  Load the pre-trained w
end
else
  Randomly initialize w
end
while $epoch < Max_epoch$ **do**
  Shuffle the training data $\{\mathcal{S}, \mathcal{K}\}$;
  for *mini-batch* $\{s', k'\}$ *in* $\{\mathcal{S}, \mathcal{K}\}$ **do**
    compute the loss, $L(D, w, s', k')$;
  end
  update w with the computed loss
end

Algorithm 2: Location Prediction of MulTLoc System.

Input: phase groups $\phi = \{\theta_1, \theta_2, \theta_3\}$, target area A, network parameter w, network architecture D
Output: Location estimation $\hat{\mathcal{G}} = \{\hat{G}_1, \hat{G}_2, \hat{G}_3\}$;
//Generate hologram tensor groups $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3\}$ with collected phases ϕ in target area A;
$\mathcal{S} = HologramGenerator(\phi, A)$;
//Filter hologram tensors with the deep network D and the well-trained parameters w ;
$\mathcal{S}' = D(w, \mathcal{S})$;
// Employ trilinear interpolation to recover the size of $\mathcal{S}'$;
$\mathcal{S}' = interpolate(\mathcal{S}')$;
// Choose the coordinates of the pixel with the highest value in the filtered tensor as the output ;
$\hat{\mathcal{G}} = \left\{\mathcal{G} \mid f(\mathcal{S}', \mathcal{G}) = \max(\mathcal{S}')\right\}$

5.2. Implementation details

For the DCNN based hologram filter network, the input tensors are first compressed by residual units and passed into the hourglass backbone as shown in Fig. 3. The residual unit consists of two ResUnit_block.

Algorithm 3: Self-supervised Pretraining of MulTLoc System.

Input: network architecture D, Max_epoch, phase groups $\phi = \{\theta_1, \theta_2, \theta_3\}$, target area A
Output: Trained parameters w;
//Generate hologram tensor groups $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3\}$ with collected phases ϕ in target area A;
$\mathcal{S} = HologramGenerator(\phi, A)$;
while $epoch < Max_epoch$ **do**
  Shuffle the training data $\mathcal{S}$;
  for *mini-batch* s' *in* $\mathcal{S}$; **do**
    Generate sub-volumes groups s'_{cut} and rotation label r with the mini-batch s';
    Compute the loss $L(D, w, s', s'_{cut}, r)$;
  end
  update w with the computed loss
end

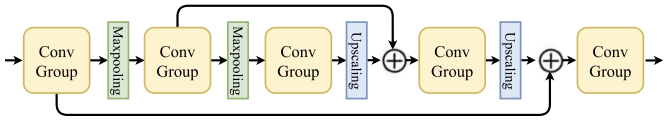


Fig. 10. Illustration of the structure of the Hourglass Unit.

Each block is made up of two $[3 \times 3 \times 3]$ convolution layers. The hourglass units are used after residual units to retrieve location information across all scales. In the unit, a $[3 \times 3 \times 3]$ convolution layer is inserted between two $[1 \times 1 \times 1]$ convolution layers to build a convolution block. Four blocks are binded as a convolution group in the hourglass unit. As shown in Fig. 10, the output from the group is then downsampled using a max-pooling of 2x2 with a stride of 2. The residual connection is inserted between convolution blocks. Also, the inputs of groups are also added back to the tensor when they are upsampled to the original size. It is worth noting that the depth of the hourglass backbone may be raised recursively. Fig. 10 exhibits an hourglass unit with a depth of 2, however, our implemented hourglass unit has a depth of 3.

The detailed information about the Swin Transformer backbone is presented in Table 1. As is shown in Fig. 5, four stages are included in the Swin Transformer backbone. The feature dimension and the number of attention heads will increase by a factor of two in each stage. With the patch merging, the output size from stages is shrunk by a factor of two. Two blocks are included in each stage by default. We will discuss the effect of feature dimension and the number of blocks on the performance of location estimation later in this section.

5.3. Experiment results and discussions

5.3.1. Overall performance

Fig. 11 presents the Cumulative Distribution Function (CDF) of localization errors, showcasing the overall localization precision achieved by different network configurations. The best localization performance is achieved by the DCNN backbone cooperating with the MS-SSIM loss function, which has a mean error of 0.0558 m. When the L2 loss is leveraged in the DCNN training, the mean localization error increases to 0.0688 m. For the Swin Transformer based network, a mean error of 0.0961 m is achieved when self-supervised learning is leveraged in training, whereas the mean error is 0.1041 m without self-supervised training. Even though a precision improvement in location estimation is brought by self-supervised training, DCNN-based hologram filter networks, in general, outperforms the Swin Transformer based networks. This result is not unexpected. In [46], the authors showed that a large Vision Transformer underperforms models with ResNet backbone when a small dataset is utilized in training. Due to the fact that our dataset only has three hundred groups of input tensors, it is acceptable to achieve a comparable location precision with a Swin

Table 1
Details of the Swin Transformer Backbone.

	Stage-1	Stage-2	Stage-3	Stage-4
Layer Size	$\begin{bmatrix}\text{win. sz. } 7\times7\times7\\ \text{dim } 48, \text{head } 3\end{bmatrix}\times2$	$\begin{bmatrix}\text{win. sz. } 7\times7\times7\\ \text{dim } 48\times2, \text{head } 6\end{bmatrix}\times2$	$\begin{bmatrix}\text{win. sz. } 7\times7\times7\\ \text{dim } 48\times4, \text{head } 12\end{bmatrix}\times2$	$\begin{bmatrix}\text{win. sz. } 7\times7\times7\\ \text{dim } 48\times8, \text{head } 24\end{bmatrix}\times2$
Output Size	(24×24×24)	(12×12×12)	(6×6×6)	(3×3×3)

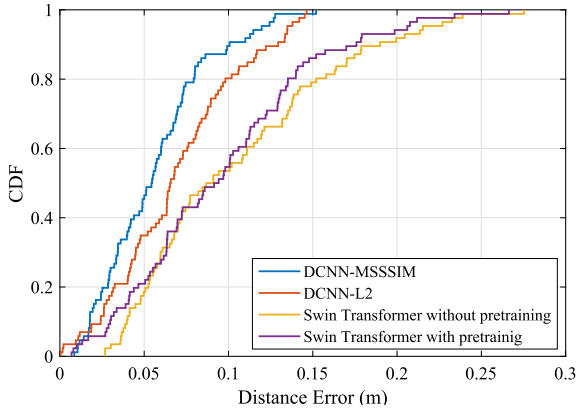


Fig. 11. CDFs of location estimation errors obtained with different hologram filter networks.

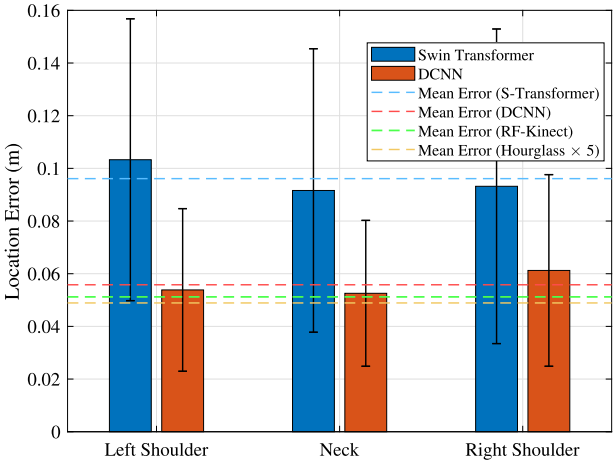


Fig. 12. Location estimation for tags.

Transformer-based network. Furthermore, the interpretable filtered result is the main reason for us to investigate the Swin Transformer-based network. As observed in Fig. 1, where peaks consistently appear on a highlighted stripe, Fig. 8(c) recovers the real peak based on the stripe pattern in Fig. 8(a), which meets our expectation in location estimation. Additionally, large Vision Transformer models outperform the ResNet-based model as the dataset grows in computer vision tasks. Thus Swin Transformer-based hologram filter network has a great potential when there is sufficient data.

Fig. 12 depicts the mean locating errors for tags as well as the overall average errors of various network configurations. Apparently, the DCNN-based hologram filter network beats the network with a Swin Transformer backbone in terms of accuracy. The location error obtained from the tag attached to the left shoulder, with the Swin Transformer backbone, is approximately 0.103 m, which doubles the error of 0.053 m achieved with the DCNN backbone. The lowest error is obtained for the tag attached to the neck with both networks, which is 0.0525 m for the network with the DCNN backbone, and 0.0916 m for

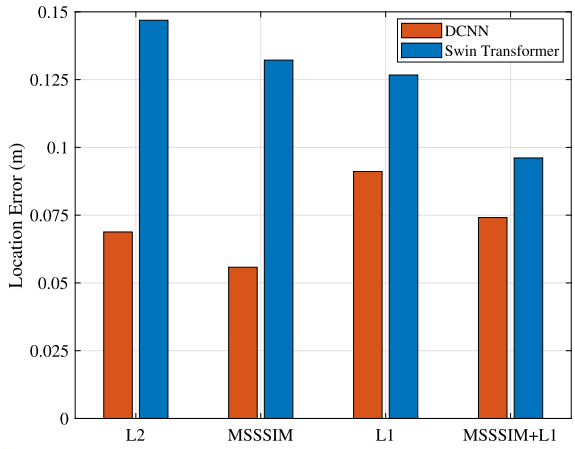


Fig. 13. Location estimation effected by different loss function.

the Swin Transformer based network. In Fig. 12, the mean errors for various systems are also indicated with dashed lines. The blue and red lines show the overall errors for networks using the Swin Transformer and DCNN backbones, respectively. Because RF-Kinect [12] also performs an experiment in a 1.5 m × 1.5 m scanning region, its average location error of 0.0512 m is displayed as a green line for comparison with our proposed approaches. As illustrated in Fig. 12, even though RF-Kinect exhibits a small improvement in localization accuracy over the DCNN-based network, the extended version of the DCNN-based hologram filter network, denoted as Hourglass×5, achieves an error of 0.0489 m, outperforming all other methods. The network extension will be discussed next.

5.3.2. *Effect of loss function on location error*

Two representative deep neural networks are used as backbones in this paper to sanitize the noisy hologram tensors. First, the framework employs a 3D variant of the U-Net with a Swin Transformer backbone. This type of network is proposed in [63] for semantic segmentation. However, the purpose of semantic segmentation is to label each pixel in the image with the right label, which does not match our task of tensor sanitizing. Labels are not utilized in our task. The pixels are filtered with the network to produce a smooth and continuous sanitized tensor. Tensor sanitizing, from this perspective, is similar to image restoration and denoising. The fake peaks related to phase wrapping and multipath effect can be treated as blur and noise in the hologram tensor. Therefore, two loss function in image restoration, L1 loss and MS-SSIM loss [56], are introduced to the Swin Transformer-based network aimed at enhancing the performance of tensor sanitizing.

According to Fig. 13, the mean location error is 0.1469 m when the L2 loss is utilized in the training. Location estimate continues to improve with the use of MS-SSIM loss and L1 loss. The MS-SSIM loss reduces the mean error to 0.1322 m, while the L1 loss contributes to a mean error of 0.1267 m. Moreover, we use a joint loss function that includes both MS-SSIM loss and L1 loss to improve the performance of the Swin Transformer-based hologram filter network. When self-supervised learning is used, the error is optimized to 0.0961 m.

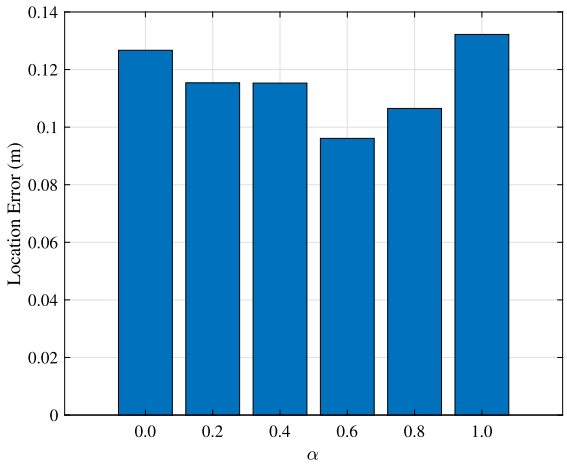


Fig. 14. Location estimation affected by α using the Swin Transformer backbone.

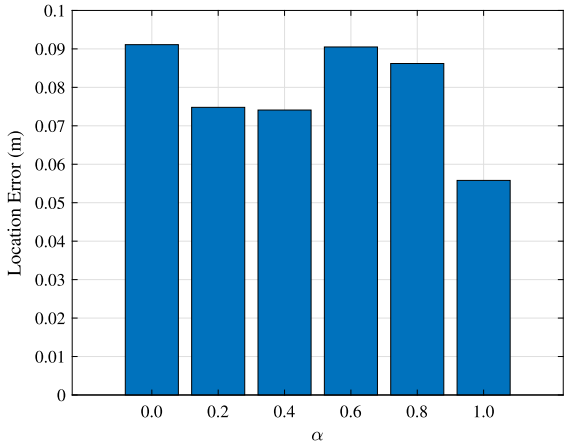


Fig. 15. Location estimation affected by α using the DCNN backbone.

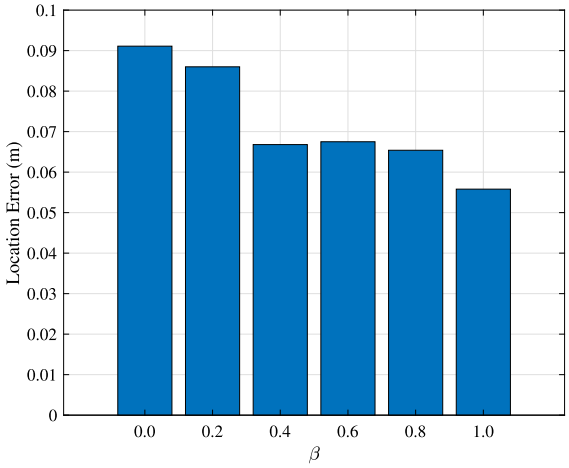


Fig. 16. Location estimation obtained by the joint function consisting of the L2 loss and MS-SSIM loss.

Since the joint loss function achieves an outstanding performance in improving the localization accuracy for Swin Transformer-based hologram filter network, we examine the effect of the ratio between L1 loss and MS-SSIM loss on location error in Fig. 14. It is evident that the location error remains high when L1 loss, or the MS-SSIM loss, is deployed in the training individually (i.e., when $\alpha = 0.0$, or when $\alpha = 1.0$). With the increment of α, the MS-SSIM loss is introduced into the supervised training. The location error drops to 0.1154 m when α is 0.2. Even though a slight stagnation happens as α is 0.4, the lowest location error is achieved when α rises to 0.6. After that, as the L1 loss disappears, the localization accuracy continues to deteriorate.

However, the DCNN-based hologram filter network does not benefit from the joint loss function consisting of the L1 loss and MS-SSIM loss. The hourglass network, a convolutional network architecture for human pose estimation, serves as the backbone of the DCNN-based hologram filter network. The main idea of the hourglass network is to capture the spatial interactions associated with the key points using a repeated bottom-up, top-down convolutional structure. It converges to our tensor filter task, in which we attempt to extract the true peaks existing in a multiple-channel hologram tensor by using the spatial relationship between peaks in different channels. However, the loss functions for image restoration cannot match the hourglass network perfectly. The L1 loss produces the largest error of 0.0911 m in Fig. 13. Despite the fact that the MS-SSIM loss leads to the best localization accuracy with a mean error of 0.0558 m, the joint loss function cannot replicate its effect on the Swin Transformer backbone. The location error related to the L1 loss is even higher than the error achieved by the L2 loss, which are 0.0741 m and 0.0688 m, respectively. In Fig. 15, we also examine the effect of ratio α between the L1 loss and MS-SSIM loss on location error. Although the location error drops as the MS-SSIM loss is involved in the training, the overall localization accuracy is not elevated by the joint loss function. When α is 0.6, the localization accuracy even gets worse.

To acquire the best localization accuracy with the DCNN-based hologram filter network, a joint loss function comprising both the L2 loss and MS-SSIM loss is investigated. Let β be the ratio between the L2 loss and MS-SSIM loss, which follows the similar way as the α in (10). The effect of β on location error is presented in Fig. 16. With the growth of β, the MS-SSIM loss is utilized to advance the localization accuracy. As β reaches 0.4, the MS-SSIM loss significantly improves the situation, but after that the growth rate slows down. Eventually, the optimized location error is obtained when the loss function is composed by the MS-SSIM loss only.

Table 2
Location Estimation Affected by the Setup of the Swin Transformer based Hologram Filter Network (*: the default setting).

Structure	# of parameters (M)	Location Error (m)
(2,2,2,2)*	62.2	0.0961
(2,2,4,2)	63.1	0.0985
(2,2,6,2)	64.1	0.1072
(2,2,8,2)	65.0	0.1064

5.3.3. *Effect of model size on location error*

The backbone architecture has a considerable impact on the performance of the hologram filter networks. To evaluate the localization accuracy resulted by the structural changes, we conduct experiments with different number of trainable parameters using two hologram filter networks. In Table 2, the number of layers in a Swin Transformer-based network is firstly varied. According to [22], only the layers in Stage-3 are varied, where $(2, 2, 2, 2)$ indicates a default layer configuration where two layers are included in each stage. As we can see, the number of parameters grows with the increasing number of layers. However, the location error does not improve significantly. Despite an increase in the number of parameters from 62.2M to 65.0M, the distance error remains at about 0.1 m.

Another key parameter influencing the scale of the Swin Transformer backbone is dimension size, given as dim in Table 1. It determines the output dimension of the linear layer in a Transformer block.

Table 3
Location Estimation Affected by the Feature Size (*: the default setting).

Dimension Size	# of parameters (M)	Location Error (m)
12	4.07	0.1175
24	15.7	0.1135
48*	62.2	0.0961

Table 4
Location Estimation Affected by the Setup of the DCNN based Hologram Filter Network (*: the default setting).

Structure	# of parameters (M)	Location Error (m)
Hourglass ×2	33.4	0.0565
Hourglass ×3*	49.9	0.0558
Hourglass ×4	66.3	0.0523
Hourglass ×5	82.7	0.0489

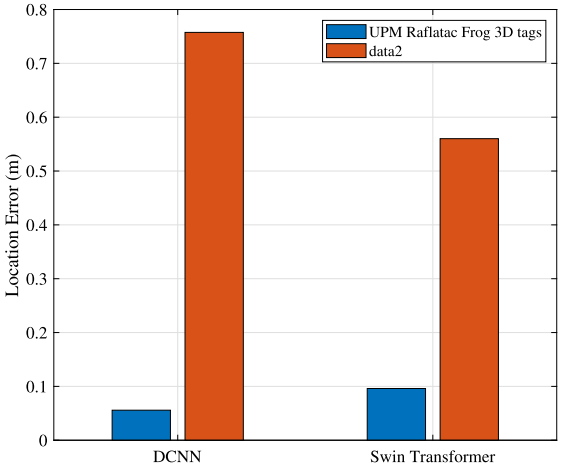


Fig. 17. Location estimation obtained for different tags.

Because the dimension of the current stage is determined by the dimension of the preceding stage, any change in the first stage would drastically alter the scale of the entire network. Table 3 employs three different dimension sizes, i.e., 12, 24, and 48, to investigate their impact on the number of parameters and localization accuracy. When the dimension size is 12, the network consists of just 4.07M trainable parameters; therefore, the capability of the network is constrained by the confined size. The location error degrades to 0.1175 m. As we double the dimension size to 24, the number of trainable parameters grows to 15.7M. Correspondingly, the extent of the network size contributes to the enhanced localization accuracy. The error drops to 0.1135 m. We further increase the dimension size to 48 to examine the improvement in localization accuracy brought by the enlarged network. In this scenario, the hologram network is composed of 62.2M parameters and the location error becomes 0.0961 m.

We also study the influence of network size on the localization precision of the DCNN-based hologram filter network. Due to the architecture of the hourglass backbone, it is convenient to stack several hourglass units for network expansion. Four variations of the hourglass backbone are deployed and the results are presented in Table 4. The number of parameters grows in direct proportion to the number of hourglass units leveraged in the backbone. With the increment of the number of parameters, the location error declines gradually. The lowest error of 0.0489 m is obtained when five hourglass units, with 82.7M parameters, are used in the network.

5.3.4. Robustness of hologram filter networks

To investigate the robustness of the hologram filter networks, SML GBe4U7 tags are deployed in the framework to collect hologram tensors. The tags are attached to the human body at the same position as in the previous experiments. The newly collected tensors are not used to train, or fine tune, the hologram filter networks. Instead, the experimental results are derived directly from the newly collected tensors using networks that were previously trained. Fig. 17 delineates the performance degradation resulted from different tags. Obviously, a significant performance degradation occurs with both networks. However, the Swin Transformer-based network is more robust to the change of tags. Even though DCNN-based network achieves a location error of less than 6 cm, its accuracy suffers when facing tags that have never been used in training. In the new tag test, the Swin Transformer-based network outperforms the DCNN network by approximately 20 cm. It potentially shows that the interpretable pattern extracted by the Swin Transformer-based network is beneficial to tensor sanitization in different tags, whereas the DCNN-based network might rely on certain "shortcuts" for tensor cleaning, which could adversely affect its transferability.

6. Conclusions and future work

In this paper, we presented MulTLoc, a framework that utilizes deep neural networks for filtering RF hologram tensor in order to locate multiple RFID tags in 3D spaces. To our knowledge, this paper represents the first attempt to train deep neural networks using hologram tensors for the purpose of 3D localization based on RFID tags. Two representative deep learning models were incorporated in the MulTLoc framework. First, we built a DCNN-based hologram filter network. The network successfully recovered cleaned hologram tensors. Centimeter-level multiple tag localization was achieved successfully with sanitized hologram tensor. In addition, a Swin Transformer-based network was also used to sanitize the hologram tensors aimed at enhancing the compatibility of the proposed framework. The network architecture was not related to the number of target tags. By adopting self-supervised training, the network was effectively trained with a small dataset. We evaluated the proposed framework using a task of multi-joint location estimation. The results demonstrated the outstanding performance of the proposed framework. For future investigations, it becomes increasingly difficult to improve the accuracy of localization systems using a single signal observation. Consequently, sensor fusion could be a practical method of taking advantage of various radio frequency signals, such as Wi-Fi, RFID, and mmWave. Furthermore, deep learning based signal processing, especially after the rise of chatGPT, could be an attractive method to eliminate noise brought by the cluttered environment and enhance localization accuracy.

CRediT authorship contribution statement

Xiangyu Wang: Data curation, Investigation, Methodology, Validation, Writing – original draft. **Jian Zhang:** Conceptualization. **Shiwen Mao:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Senthilkumar CG Periaswamy:** Project administration. **Justin Patton:** Project administration.

References

[1] C. Yang, X. Wang, S. Mao, SparseTag: high-precision backscatter indoor localization with sparse RFID tag arrays, in: Proc. IEEE SECON'19, Boston, MA, 2019, pp. 1–9.
[2] S. Pradhan, et al., RIO: a pervasive RFID-based touch gesture interface, in: Proc. ACM MobiCom'17, New York, NY, 2017, pp. 261–274.
[3] C. Yang, X. Wang, S. Mao, Respiration monitoring with RFID in driving environments, IEEE J. Sel. Areas Commun. 39 (2) (2021) 500–512.
[4] C. Yang, X. Wang, S. Mao, Unsupervised detection of apnea using commodity RFID tags with a recurrent variational autoencoder, IEEE Access 7 (1) (2019) 67526–67538.

[5] C. Yang, X. Wang, S. Mao, RFID-Pose: vision-aided 3D human pose estimation with RFID, IEEE Trans. Reliab. 70 (3) (2021) 1218–1231.

[6] C. Yang, X. Wang, S. Mao, Subject-adaptive skeleton tracking with RFID, in: Proc. the 16th IEEE International Conference on Mobility, Sensing and Networking (MSN 2020), Tokyo, Japan, 2020, pp. 599–606.

[7] X. Wang, J. Zhang, Z. Yu, S. Mao, S. Periaswamy, J. Patton, On remote temperature sensing using commercial UHF RFID tags, IEEE Int. Things J. 6 (6) (2019) 10715–10727.

[8] J. Wang, J. Xiong, X. Chen, H. Jiang, R.K. Balan, D. Fang, Simultaneous material identification and target imaging with commodity RFID devices, IEEE Trans. Mob. Comput. 20 (2) (2021) 739–753.

[9] J. Hightower, R. Want, G. Borriello, SpotON: an indoor 3D location sensing technology based on RF signal strength, UW CSE Technical Report, 2000.

[10] L.M. Ni, Y. Liu, Y.C. Lau, A.P. Patil, LANDMARC: indoor location sensing using active RFID, in: Proc. IEEE PerCom'03, Dallas, TX, 2003, pp. 407–415.

[11] H. Jin, Z. Yang, S. Kumar, J.I. Hong, Towards wearable everyday body-frame tracking using passive RFIDs, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1 (4) (2018) 145.

[12] C. Wang, J. Liu, Y. Chen, L. Xie, H. Liu, S. Lu, RF-kinect: a wearable RFID-based approach towards 3D body movement tracking, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2 (1) (2018) 41.

[13] X. Wang, L. Gao, S. Mao, S. Pandey, DeepFi: deep learning for indoor fingerprinting using channel state information, in: Proc. IEEE WCNC'15, New Orleans, LA, 2015, pp. 1666–1671.

[14] X. Wang, L. Gao, S. Mao, S. Pandey, CSI-based fingerprinting for indoor localization: a deep learning approach, IEEE Trans. Veh. Technol. 66 (1) (2017) 763–776.

[15] X. Wang, L. Gao, S. Mao, PhaseFi: phase fingerprinting for indoor localization with a deep learning approach, in: Proc. GLOBECOM'15, San Diego, CA, 2015, pp. 1–6.

[16] X. Wang, L. Gao, S. Mao, CSI phase fingerprinting for indoor localization with a deep learning approach, IEEE Int. Things J. 3 (6) (2016) 1113–1123.

[17] X. Wang, X. Wang, S. Mao, Indoor fingerprinting with bimodal CSI tensors: a deep residual sharing learning approach, IEEE Int. Things J. 8 (6) (2021) 4498–4513.

[18] W. Wang, X. Wang, S. Mao, Deep convolutional neural networks for indoor localization with CSI images, IEEE Trans. Netw. Sci. Eng. 7 (1) (2020) 316–327.

[19] J. Talvitie, E.S. Lohan, M. Renfors, The effect of coverage gaps and measurement inaccuracies in fingerprinting based indoor localization, in: Proc. IEEE ICL-GNSS'14, 2014, pp. 1–6.

[20] S. He, S.-H.G. Chan, Wi-fi fingerprint-based indoor positioning: recent advances and comparisons, IEEE Commun. Surv. Tutor. 18 (1) (2016) 466–490.

[21] P.V. Nikitin, R. Martinez, S. Ramamurthy, H. Leland, G. Spiess, K. Rao, Phase based spatial identification of uhf rfid tags, in: 2010 IEEE International Conference on RFID (IEEE RFID 2010), IEEE, 2010, pp. 102–109.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: hierarchical vision transformer using shifted windows, in: Proc. IEEE CVPR'21, Virtual Conference, 2021, pp. 10012–10022.

[23] J. Gjengset, J. Xiong, G. McPhillips, K. Jamieson, Phaser: enabling phased array signal processing on commodity WiFi access points, in: Proc. ACM Mobicom'14, Maui, HI, 2014, pp. 153–164.

[24] J. Xiong, K. Jamieson, Arraytrack: a fine-grained indoor location system, in: Proc. ACM NSDI'13, IL, Lombard, 2013, pp. 71–84.

[25] Y. Xie, J. Xiong, M. Li, K. Jamieson, md-track: leveraging multi-dimensionality for passive indoor wi-fi tracking, in: Proc. ACM Mobocom'19, Los Cabos, Mexico, 2019, pp. 1–16.

[26] M. Kotaru, K. Joshi, D. Bharadia, S. Katti, SpotFi: decimeter level localization using WiFi, in: Proc. ACM SIGCOMM'15, London, UK, 2015, pp. 269–282.

[27] X. Li, S. Li, D. Zhang, J. Xiong, Y. Wang, H. Mei, Dynamic-music: accurate device-free indoor localization, in: Proc. ACM UbiComp 2016, Heidelberg, Germany, 2016, pp. 196–207.

[28] P. Bahl, V.N. Padmanabhan, Radar: an in-building RF-based user location and tracking system, in: Proc. IEEE INFOCOM'00, Tel Aviv, Israel, 2000, pp. 775–784.

[29] J. Oh, J. Kim, Adaptive k-nearest neighbour algorithm for wifi fingerprint positioning, Ict Express 4 (2) (2018) 91–94.

[30] D. Li, B. Zhang, C. Li, A feature-scaling-based *k*-nearest neighbor algorithm for indoor positioning systems, IEEE Int. Things J. 3 (4) (2016) 590–597.

[31] Y. Xie, Y. Wang, A. Nallanathan, L. Wang, An improved k-nearest-neighbor indoor localization method based on spearman distance, IEEE Signal Process. Lett. 23 (3) (2016) 351–355.

[32] L. Calderoni, M. Ferrara, A. Franco, D. Maio, Indoor localization in a hospital environment using random forest classifiers, Elsevier Expert Syst. Appl. 42 (1) (2015) 125–134.

[33] X. Guo, N. Ansari, L. Li, H. Li, Indoor localization by fusing a group of fingerprints based on random forests, IEEE Int. Things J. 5 (6) (2018) 4686–4698.

[34] Y. Zhang, D. Li, Y. Wang, An indoor passive positioning method using csi fingerprint based on adaboost, IEEE Sens. J. 19 (14) (2019) 5792–5800.

[35] Z. Liu, D. Liu, J. Xiong, X. Yuan, A parallel adaboost method for device-free indoor localization, IEEE Sens. J. 22 (3) (2022) 2409–2418.

[36] J.-R. Jiang, H. Subakti, H.-S. Liang, Fingerprint feature extraction for indoor localization, MDPI Sens. 21 (16) (2021) 5434.

[37] J. Luo, L. Fu, A smartphone indoor localization algorithm based on wlan location fingerprinting with feature extraction and clustering, MDPI Sens. 17 (6) (2017) 1339.

[38] X. Wang, L. Gao, S. Mao, S. Pandey, BiLoc: bi-modal deep learning for indoor localization with commodity 5 GHz WiFi, IEEE Access 5 (2017) 4209–4220.

[39] X. Wang, X. Wang, S. Mao, ResLoc: deep residual sharing learning for indoor localization with CSI, in: Proc. IEEE PIMRC'17, Montreal, Canada, 2017, pp. 1–6.

[40] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems, vol. 25, Curran Associates, Inc., 2012, pp. 1–9.

[41] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: ACM ECCV'16, Amsterdam, Nertherlands, 2016, pp. 630–645.

[42] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: ACM ECCV'16, Amsterdam, Netherlands, 2016, pp. 483–499.

[43] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[44] X. Li, X. Wu, Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition, in: Proc. IEEE ICASSP'15, South Brisbane, Australia, 2015, pp. 4520–4524.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 6000–6010.

[46] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, arXiv preprint, arXiv:2010.11929.

[47] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F.E. Tay, J. Feng, S. Yan, Tokens-to-token vit: training vision transformers from scratch on imagenet, in: Proc. IEEE CVPR'21, Virtual Conference, 2021, pp. 558–567.

[48] X. Wang, C. Yang, S. Mao, TensorBeat: tensor decomposition for monitoring multiperson breathing beats with commodity WiFi, ACM Trans. Intell. Syst. Technol. 9 (1) (2017) 1–27.

[49] J. Zhang, Z. Yu, X. Wang, Y. Lyu, S. Mao, S. Periaswamy, J. Patton, X. Wang, RFHUI: an RFID based human-unmanned aerial vehicle interaction system in an indoor environment, Elsevier Digit. Commun. Netw. J. 6 (1) (2020) 14–22.

[50] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, Y. Liu, Tagoram: real-time tracking of mobile RFID tags to high precision using COTS devices, in: Proc. ACM Mobicom'14, Maui, HI, 2014, pp. 237–248.

[51] H. Chen, Y. Zhang, W. Li, X. Tao, P. Zhang, ConFi: convolutional neural networks based indoor Wi-Fi localization using channel state information, IEEE Access 5 (1) (2017) 18066–180747.

[52] G. Wang, C. Qian, K. Cui, X. Shi, H. Ding, W. Xi, J. Zhao, J. Han, A universal method to combat multipaths for RFID sensing, in: Proc. IEEE INFOCOM'20, Toronto, Canada, 2020, pp. 277–286.

[53] L. Shangguan, K. Jamieson, The design and implementation of a mobile RFID tag sorting robot, in: Proc. ACM MobiSys'16, Singapore, 2016, pp. 31–42.

[54] J. Wang, D. Vasisht, D. Katabi, Rf-idraw: virtual touch screen in the air using rf signals, in: ACM SIGCOMM'14, 2014, pp. 235–246.

[55] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE CVPR'16, Las Vegas, NV, 2016, pp. 770–778.

[56] H. Zhao, O. Gallo, I. Frosio, J. Kautz, Loss functions for image restoration with neural networks, IEEE Trans. Comput. Imaging 3 (1) (2017) 47–57.

[57] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proc. IEEE CVPR'20, 2020, pp. 9729–9738.

[58] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proc. IEEE CVPR'22, New Orleans, LA, 2022, pp. 16000–16009.

[59] Y. Tang, D. Yang, W. Li, H.R. Roth, B. Landman, D. Xu, V. Nath, A. Hatamizadeh, Self-supervised pre-training of swin transformers for 3d medical image analysis, in: Proc. IEEE CVPR'22, New Orleans, LA, 2022, pp. 20730–20740.

[60] A.v.d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748.

[61] J. Zhang, X. Wang, Z. Yu, Y. Lyu, S. Mao, S.C. Periaswamy, J. Patton, X. Wang, Robust rfid based 6-dof localization for unmanned aerial vehicles, IEEE Access 7 (2019) 77348–77361.

[62] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, T. Brox, 3D human pose estimation in RGBD images for robotic task learning, in: Proc. IEEE ICRA'18, Brisbane, Australia, 2018, pp. 1986–1992.

[63] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. Roth, D. Xu, Swin unetr: swin transformers for semantic segmentation of brain tumors in mri images, arXiv preprint arXiv:2201.01266.