applied sciences

MDPI

Article

WiGR: A Practical Wi-Fi-Based Gesture Recognition System with a Lightweight Few-Shot Network

Pengli Hu, Chengpei Tang *, Kang Yin and Xie Zhang

School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou 510006, China; hupli@mail2.sysu.edu.cn (P.H.); yink5@mail2.sysu.edu.cn (K.Y.); zhangx289@mail2.sysu.edu.cn (X.Z.)
* Correspondence: tchengp@mail.sysu.edu.cn; Tel.: +86-1382-607-1066

Abstract: Wi-Fi sensing technology based on deep learning has contributed many breakthroughs in gesture recognition tasks. However, most methods concentrate on single domain recognition with high computational complexity while rarely investigating cross-domain recognition with lightweight performance, which cannot meet the requirements of high recognition performance and low computational complexity in an actual gesture recognition system. Inspired by the few-shot learning methods, we propose WiGR, a Wi-Fi-based gesture recognition system. The key structure of WiGR is a lightweight few-shot learning network that introduces some lightweight blocks to achieve lower computational complexity. Moreover, the network can learn a transferable similarity evaluation ability from the training set and apply the learned knowledge to the new domain to address domain shift problems. In addition, we made a channel state information (CSI)-Domain Adaptation (CSIDA) data set that includes channel state information (CSI) traces with various domain factors (i.e., environment, users, and locations) and conducted extensive experiments on two data sets (CSIDA and SignFi). The evaluation results show that WiGR can reach 87.8–94.8% cross-domain accuracy, and the parameters and the calculations are reduced by more than 50%. Extensive experiments demonstrate that WiGR can achieve excellent recognition performance using only a few samples and is thus a lightweight and practical gesture recognition system compared with state-of-the-art methods.

Keywords: few-shot learning; gesture recognition; lightweight network; Wi-Fi sensing technology

Citation: Hu, P.; Tang, C.; Yin, K.; Zhang, X. WiGR: A Practical Wi-Fi-Based Gesture Recognition System with a Lightweight Few-Shot Network. *Appl. Sci.* **2021**, *11*, 3329. https://doi.org/10.3390/app11083329

Academic Editor: Juan-Carlos Cano

Received: 11 March 2021
Accepted: 6 April 2021
Published: 7 April 2021

1. Introduction

With the rapid development of the Internet of Things technology, various smart devices have changed people's lives. Human–computer interaction technologies, i.e., information interaction between humans and computers, have become essential. Since gestures have the advantages of easy learning, rich information, and simplicity, gesture recognition technology [1] has become a research hotspot in recent years. Gesture recognition technology can be widely used in virtual games, automatic driving assistance systems, sign language recognition, and intelligent robot control. Currently, the main problems of the existing gesture recognition methods based on wearable sensors [2,3] and cameras [4,5] are that they are not convenient enough, the required equipment is expensive, and there is a risk of privacy leakage, which limits the wide application of gesture recognition systems in reality. Gesture recognition technology is more practical than ever before under the booming development of Wi-Fi sensing technologies, progressively transitioning from theoretical research to practical landing application stages due to their advantages of a contactless manner, low cost, good privacy, and the fact that they do not require line-of-sight propagation (LoS) [1]. Specifically, the development of gesture recognition systems is moving from the single domain to the cross domain, from recognizing fixed types of gestures to new types of gestures. In addition, a gesture recognition system is increasingly deployed in the mobile environment, and its model has also transformed from heavyweight to lightweight to meet the requirements of mobile device deployment.

Wi-Fi sensing technologies recognize a gesture by analyzing a gesture's feature, extracted from the channel state information (CSI) of Wi-Fi signals, which are generated during the execution of the gesture. A convolution neural network [6–8], an important neural network model of deep learning, has excellent feature extraction capabilities. Therefore, Wi-Fi-based gesture recognition methods mainly adopt deep learning algorithms to recognize gestures [9–12]. However, these methods concentrate on single domain recognition. Once they face new types of gestures or gestures performed in a new domain, the recognition performance will dramatically degrade, and a large amount of testing data from the new domain is needed to adjust the model. This problem is called a "domain shift" and is a substantial challenge for improving the practicality of the gesture recognition system. In addition, deep-learning-based gesture recognition systems usually have a complex neural network model. Due to the limitations of storage space and computation consumption, the storage and calculation of neural network models on mobile devices are other substantial challenges. Therefore, designing a lightweight gesture recognition system with good recognition performance in the new domain using a small amount of data is an essential aspect of facilitating the application of gesture recognition technology.

Recently, there has been an increasing amount of literature adopting the transfer learning technique [13–15], generative adversarial networks [16], or a manually designed domain-independent feature body-coordinate velocity profile [17] to eliminate the domain shift problem. However, the excellent performance of these methods depends on high amounts of data, and the manual modeling method needs to analyze complex CSI data. Since the influence pattern of gestures on Wi-Fi signals is complicated, the model of velocity profiles is complicated as well.

In addition, inspired by the few-shot learning technique [18–22], Zou et al. [23] and Zhou et al. [24] combined a few-shot network and adversarial learning to remove domain-related information. Lan et al. [25] proposed a few-shot multi-task classifier to address the domain shift problem. The basic idea is to initialize the parameters of the classifier so that the classifier can quickly adapt to a new domain. Yang et al. [26] proposed a Siamese recurrent convolutional architecture to remove structured noise and used convolution neural network (CNN)-long short-term memory (LSTM) to extract temporal-spatial features. Although these methods can eliminate the domain shift problem with a small amount of data, they require more computation. Their complex models with many parameters are not suitable for deployment.

To address the challenges mentioned above, we proposed WiGR, a novel, practical Wi-Fi-based gesture recognition system. The key structure of WiGR is an improved few-shot learning network, which consists of a feature extraction subnetwork and a similarity discrimination subnetwork. The feature extraction subnetwork adopted a 2-D convolutional kernel [6] to simultaneously extract the spatial features and temporal dynamics of gestures. Similar to the relation network [22], the similarity discrimination subnetwork uses a learning-based neural network as the similarity measurement method to determine the type of gesture, and this is more accurate than using fixed functions as measurement methods [18–21]. The whole network can learn a transferable similarity evaluation ability from the training set and apply the learned knowledge to the new testing domain via an episode-based training strategy [20] to eliminate the problem of domain shift. In addition, there is evidence that lightweight networks [27–31] play a crucial role in mobile deployment. Therefore, we introduce depthwise separable convolution and an inverted residual layer of a linear bottleneck [30,31] in a few-shot learning network to reduce model computations and parameters. Simultaneously, to reduce the complexity of the model while the recognition performance does not decrease accordingly, we introduce a squeeze and excitation (SE) block [32] to improve the quality of the features generated from the network by explicitly modeling the interdependence between the network convolution feature channels. Later extensive experiments on two data sets (CSI-Domain Adaptation (CSIDA) and SignFi [10]) demonstrate that WiGR can achieve excellent recognition per-

formance in cross-domain evaluation, and our network design dramatically reduces the model computations.

Our contributions can be summarized as follows:

- We designed a novel Wi-Fi-based gesture recognition system called WiGR that is more practical than existing gesture recognition systems. The practicality is reflected in its ability to recognize new gestures or gestures performed in new domains using just a few new samples.
- A lightweight few-shot learning network, which consists of a feature extraction subnetwork and a similarity discrimination subnetwork, is proposed to address the hard domain shift problem. Lightweight and effective blocks are introduced in the network to achieve lower computational complexity and high performance.
- We made a CSIDA data set, which includes CSI traces with various domain factors, to simulate real scenes. The CSIDA data set was helpful for us to verify the accuracy of the proposed WiGR in cross-domain evaluation.
- Extensive experiments on the SignFi data set and the CSIDA data set show the superiority of the proposed WiGR over existing gesture recognition systems in terms of cross-domain accuracy and computational complexity.

2. Preliminary

2.1. Related Work

2.1.1. Wi-Fi-Based Gesture Recognition Methods

With the rise of Wi-Fi sensing technology, the CSI of Wi-Fi can convey rich information and achieve precise tracking. There are many different types of methods based on CSI to achieve gesture recognition. For example, WiGeR [33] employs a multilevel wavelet decomposition algorithm and the short-time energy algorithm dynamic time warping (DTW) to recognize gestures. WiCatch [34] utilizes the support vector machine (SVM) with the MUSIC signal processing algorithm to recognize gestures. Ma et al. [10] proposed SignFi, a deep learning method with a nine-layer CNN architecture, to recognize sign gestures. However, these methods have not dealt with the hard domain shift problem.

Few-shot learning methods [18–22] have achieved great success in addressing the domain shift problem. Zou et al. [23] proposed a new few-shot domain adaptation scheme (F-CADA). F-CADA adopts adversarial learning to construct an embedding space, which needs a large number of unlabeled target data. It then enhances the performance of the target classifier by a few labeled target data via greedy label propagation. Zhou et al. [24] proposed three adversarial learning processions to remove the distribution discrepancy between source and target data, increasing the complexity of the system. Lan et al. [25] proposed a multi-task classifier to address the domain shift problem. The basic idea of addressing domain shift is to initialize the classifier with multi-task classifier parameters so that the classifier can quickly adapt to any new sensing domain while it is difficult for the cross-tasks classifier to converge. For the deep Siamese recurrent convolutional network [26], it is a typical method of using a few-shot learning network to recognize gestures. The Siamese network relies on CNN-LSTM architecture to extract spatial–temporal features, which also increases the complexity of the model. Taken together, these methods ignore the problem of model calculation complexity, which is not beneficial for model deployment. In this paper, our proposed system adopts a different feature extraction network, i.e., a 2-D convolutional neural network, which has a better performance in feature extraction compared with the CNN-LSTM architecture. In addition, we not only focus on the domain shift issue but also introduce a lightweight block to meet the performance requirements of mobile deployment.

2.1.2. Few-Shot Learning Network

The few-shot learning method [18–22], the key technology used in this paper, is committed to addressing the domain shift problem using just a few support samples. This is the key difference between the few-shot learning method and other domain adaptation

methods. Traditional few-shot learning methods use a certain measurement method to express the correlation of samples. For example, the Siamese network [18] is a two-way neural network that determines whether the samples belong to the same class depending on their distance. This network is fed a pair of samples in a sequence to calculate a contrastive loss function in each iteration process, which has less efficiency in updating the network's weights compared with the number of batch samples [19]. A matching network [20] utilizes the idea of metric learning based on deep neural features and augments the neural network with external memories to achieve few-shot learning. Snell et al. [21] proposed a prototype network that measures the similarity of features by a fixed equation (e.g., negative Euclidean distance and cosine similarity). In the above methods, the similarity measurements are fixed functions that are not flexible when applied in a complex embedding space. In 2017, Sung et al. [22] proposed a relational network that adopts a learning-based neural network as the similarity measurement method, and this measurement method helps determine the relationship between samples more accurately, compared with a fixed manual measurement method. Therefore, we introduce a relation network as the basic model for solving the domain shift problem in our system. Additionally, we introduce some lightweight blocks in the model to make the system more suitable for mobile devices.

2.1.3. Lightweight Network Designs

Lightweight neural networks have fewer parameters and consume fewer computer resources, so these networks are more suitable for deployment on mobile devices. SqueezeNet [27] reduces the network's parameters by replacing the 3×3 convolution kernel with a 1 x 1 convolution kernel and limiting the number of channels. ShuffleNet [28] adopts pointwise group convolution to reduce the model computational cost and uses channel shuffle to improve the information presentation ability of the network. InceptionV3 [7], Xception [29], MobileNetV1 [30], and MobileNetV2 [31] adopt depthwise separable convolution instead of traditional convolution to reduce parameters and computing consumption. In addition, MobileNetV2 uses an inverted residual layer of a linear bottleneck to achieve better performance with less computing consumption. Overall, these studies prove the effectiveness of the deep separable convolution model and the linear inverted residual lightweight structure. Therefore, we introduce these strategies into our network to make our system more lightweight.

2.2. Overview of CSI

As a signal descriptor of the Wi-Fi signal, CSI reflects the signal information of the communication link, such as signal scattering, multipath fading, and the power decay of distance. A wireless channel usually uses the channel impulse response (CIR) to describe the multipath propagation of the signal from the amplitude characteristics and the phase characteristics. The measurement of CSI is mainly used to obtain CIR values [35]. The CIR is mathematically expressed as

$$X(i) = ||X(i)||e^{j\angle X(i)} \quad (1)$$

where $||X(i)||$ represents the amplitude of CSI measurement at the ith subcarrier, and $\angle X(i)$ denotes the phase of CSI measurement at the ith subcarrier. Since the phase information is more sensitive to environmental changes, our interest is in obtaining the CSI phase information for each subcarrier.

Currently, some network interface cards (NICs) have been able to continuously monitor the state changes of signal frequency response in wireless signals [36], such as Intel 5300, Atheros 9390 [37,38], and Atheros AR9580 [39]. We can obtain CSI data directly from the NICs by modifying the open-source driver of the NICs.

3. Methods

3.1. Problem Definition

In actual testing scenarios for gesture recognition, the testing conditions are usually different from those for training procedures. It is not feasible to collect a large amount of data in new scenarios to adapt the system to the current scenario. Therefore, a practical gesture recognition system should achieve excellent performance using just a few samples of gestures when facing new types of gestures that have not been seen in a training procedure or when gestures are performed in a new domain. Formally, our system is trained by training set D, which consists of samples with corresponding labels of the old types of gestures. We then divide the samples with corresponding labels of the new types of gestures or gestures performed in the new domain into two subsets, i.e., support subset S and testing subset Q. Our goal is to train the system by training set D and then use the transferable knowledge learned from D and the feature knowledge learned from the support subset S to identify the label y_j of each sample x_j in the testing subset Q.

3.2. Overview of WiGR

In this section, we introduce the framework of the proposed WiGR system. As illustrated in Figure 1, WiGR mainly contains three parts: CSI data collection, data processing, and a lightweight few-shot network. First, the input of the system is CSI data containing gesture information. These CSI data collection methods are described in detail in Section 3.2.1. We will explain the data processing in Section 3.2.2. The key structure of the system is a lightweight few-shot network, which is explained in Section 3.2.3. We describe the episode-based training strategy [19] used to train the lightweight few-shot network in Section 3.2.4.

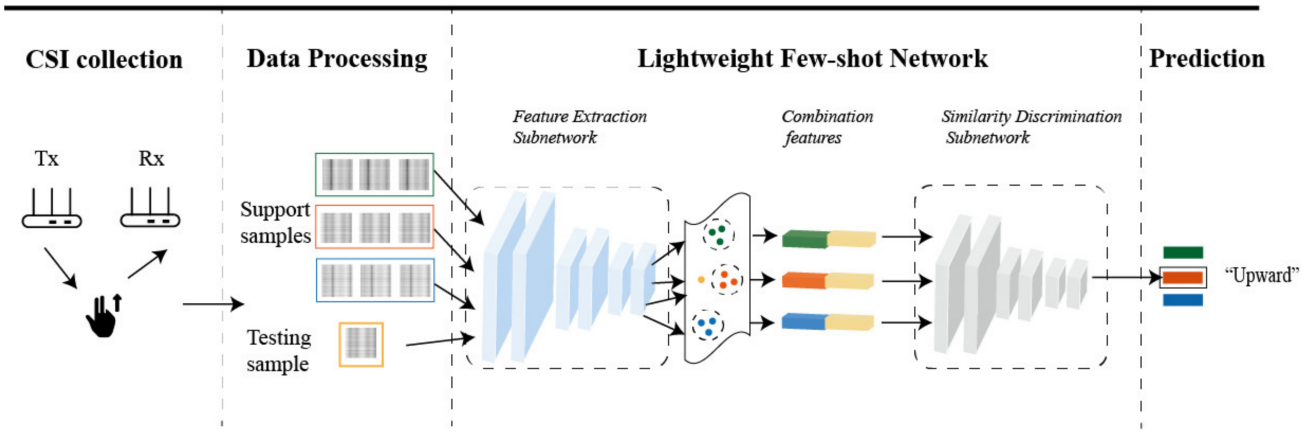


Figure 1. The framework of the Wi-Fi-based gesture recognition (WiGR) system contains four steps: (1) input the channel state information (CSI) data collected from the Wi-Fi environment; (2) process the CSI data; (3) feed the processed CSI data into the lightweight few-shot network; (4) train the network via an episode-based training strategy and output the predicted results.

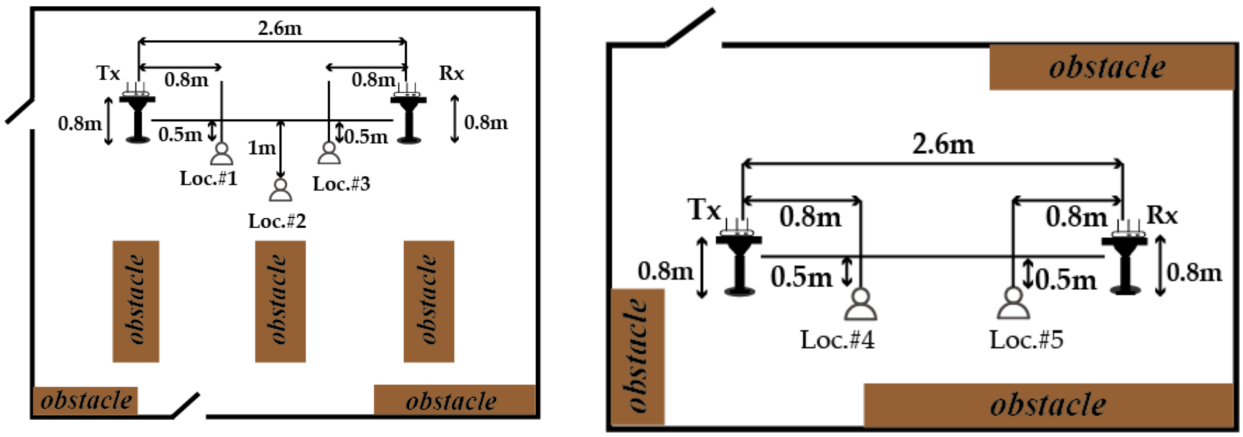
3.2.1. CSI Data Collection

In this section, we introduce the collection method of the CSI data. The CSI data used in this paper came from two Wi-Fi data sets, i.e., our own CSIDA data set and the public SignFi data set [10], which were created via different data collection methods.

The collection of the CSIDA data set. We used two Atheros AR9580 Wi-Fi chipsets supporting the IEEE 802.11n standard as a transmitter (Tx) and a receiver (Rx) [39], respectively. Each Wi-Fi chipset was equipped with three antennas with an interval of 0.1 m (m). It should be noted that, considering the performance of the computer, only one transmitting antenna and three receiving antennas were used in our experiment. Therefore, there were 3 (1 × 3) Tx–Rx pairs in total. The bandwidth was 40 MHz, and the Wi-Fi frequency was

5 GHz. Since orthogonal frequency division multiplexing (OFDM) was used in protocol 802.11n [40], many subcarriers could be obtained. Therefore, one CSI datum included 114 subcarriers for each Tx–Rx pair. In addition, each CSI datum was collected in 1.8 s (s) with a sampling rate of 1000 data frames/s. Denote the number of antennas at the Tx as N_{Tx}, the number of antennas at the Rx as N_{Rx}, the number of subcarriers as Nc, and the sampling data frames as T. The CSI data can be represented as a complex matrix of T × Nc × (N_{Tx} × N_{Rx}) (i.e., 1800 × 114 × 3), which indicates the size of the input data of our proposed network.

We collected CSI data in two different indoor environments (Room 1 and Room 2). The layout of the indoor environment is shown in Figure 2. From Figure 2, we can see that the distance between Tx and Rx is 2.6 m [41]. In Room 1, we marked three locations on which the users stood and performed predesigned gestures. In Room 2, we marked two locations. The distance between the user and the transmitter/receiver refers to [17,41]. The user stood on the premarked locations and saw the instructions on the screen of a computer. The computer was used to automatically label the CSI data generated during the execution of the gestures.



(**a**) Room 1 (**b**) Room 2

Figure 2. The layout of two different indoor environments: (**a**) Room 1; (**b**) Room 2.

Five users performed predesigned gestures. As shown in Figure 3, the predesigned gestures were of six types: upward, downward, leftward, rightward, circle, and zigzag, which are commonly used in the field of human–computer interaction. When collecting the data, the user stood on the premarked location and faced the computer screen. Before data collection, the screen showed the type of gesture and reminded the user to raise their hand to prepare for the action. After 3 s of preparation time, the user started performing the gesture, and the duration time of each gesture was 1.8 s. At the same time, the computer started collecting CSI data frames with a sampling rate of 1000 data frames/s, and each CSI datum had four labels: the identity of the users, the room number, the location number, and the gesture category. Afterward, the screen showed instructions to stop for 2 s, and the user took a short break. Thus, the CSI data collection of one gesture was completed. We kept repeating the above process until the data collection was completed.

Table 1 shows a summary of the CSIDA data collection. The five users with different figures stood on five different locations (three locations in Room 1 and two locations in Room 2) to perform six predesigned gestures. Each gesture was repeated 20 times. Therefore, there were 1800 (5 × 3 × 6 × 20) samples of gestures in Room 1 and 1200 (5 × 2 × 6 × 20) samples of gestures in Room 2.

The collection of the SignFi data set. The SignFi data set was collected using an 802.11n CSI tool based on Intel 5300 NIC [10]. The CSI collection system contained a Tx and an Rx, equipped with one and three antennas, respectively. In addition, there were

30 subcarriers, and the sampling time was 200 data frames in that system. Therefore, the size of one CSI datum inputted into the lightweight few-shot learning network was 200 × 30 × 3. The SignFi data set contains 276 sign gestures collected by five users, and each gesture was repeated 10 times. There are 14,280 gesture samples in total, which consist of 11,520 gesture samples obtained in a lab and 2760 gesture samples obtained in a home. A detailed description is given in Table 2 [10], where "Number of Samples" denotes the total number of samples (number of gestures × number of repetitions).

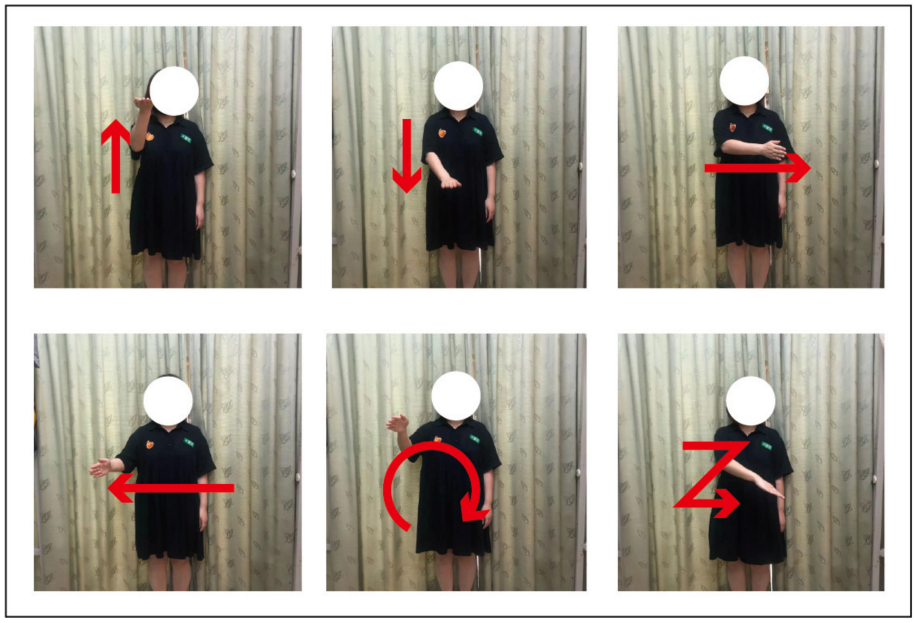


Figure 3. Six common types of gestures: upward, downward, leftward, rightward, circle, and zigzag.

Table 1. CSI-Domain Adaptation (CSIDA) data collection summary.

Environment	Locations	Users	Weight/Height	Number of Samples
Room 1	# 1 # 2 # 3	User 1 User 2 User 3	50 kg/160 cm 48 kg/163 cm 65 kg/170 cm	1800
Room 2	# 4 # 5	User 4 User 5	56 kg/168 cm 81 kg/185 cm	1200

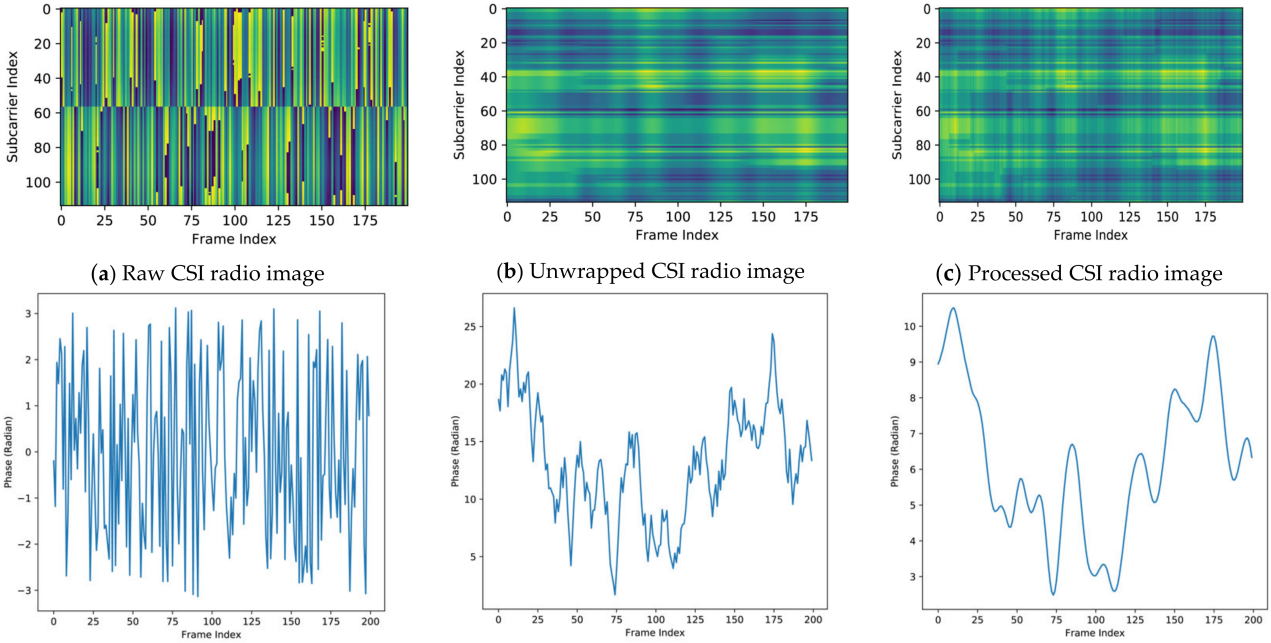
Table 2. The SignFi data collection summary.

Environment	Users	Weight/Height	Number of Samples
Lab	User 1	90 kg/170 cm	1500 (150 × 10)
	User 2	61 kg/174 cm	1500 (150 × 10)
	User 3	55 kg/168 cm	1500 (150 × 10)
	User 4	65 kg/180 cm	1500 (150 × 10)
	User 5	68 kg/171 cm	5520 (276 × 20)
Home	User 5	68 kg/171 cm	2760 (276 × 10)

3.2.2. CSI Data Processing

Before feeding the raw CSI data into the proposed WiGR model, we needed to remove noises to improve gesture recognition accuracy. As we know, pulse and burst noise are usually at a higher frequency than the reflected signal caused by human movement, whereas static reflectors usually have a lower interference frequency [42,43]. Therefore, it is necessary to filter this interference noise. In our experiments, we adopted a finite impulse

response (FIR) filter [44] designed by the least-squares method, with the cutoff frequencies set to 2 and 80 Hz. Figure 4 shows the CSI phase waveform of the "upward" gesture in 200 data frames and its corresponding CSI radio image. As shown in Figure 4a–c, the CSI radio images have both spatial and temporal characteristics that are useful for recognition. The x-axis represents the duration of one CSI datum collection, which shows the temporal characteristics of the CSI data. The y-axis represents 114 subcarriers, which shows the spatial change of the CSI data. For the sake of clarity, we randomly selected one subcarrier of the CSI data to show its phase waveform in Figure 4d–f. In addition, the sampling clock and carrier frequency of the Tx and Rx were not synchronized in the real-world Wi-Fi systems, and this led to sampling time offset and sampling frequency offset, which introduce random phase shift. Therefore, the raw CSI phases were wrapped in the range of $[-\pi, \pi]$, as shown in Figure 4d, which wrongly shows the changing trend of CSI phases. We unwrapped the CSI phases to recover the lost information by removing random phase shifts [10], as shown in Figure 4e. Unwrapped CSI phases were then filtered with an FIR filter to remove noise interference, as shown in Figure 4f.



(**a**) Raw CSI radio image (**b**) Unwrapped CSI radio image (**c**) Processed CSI radio image

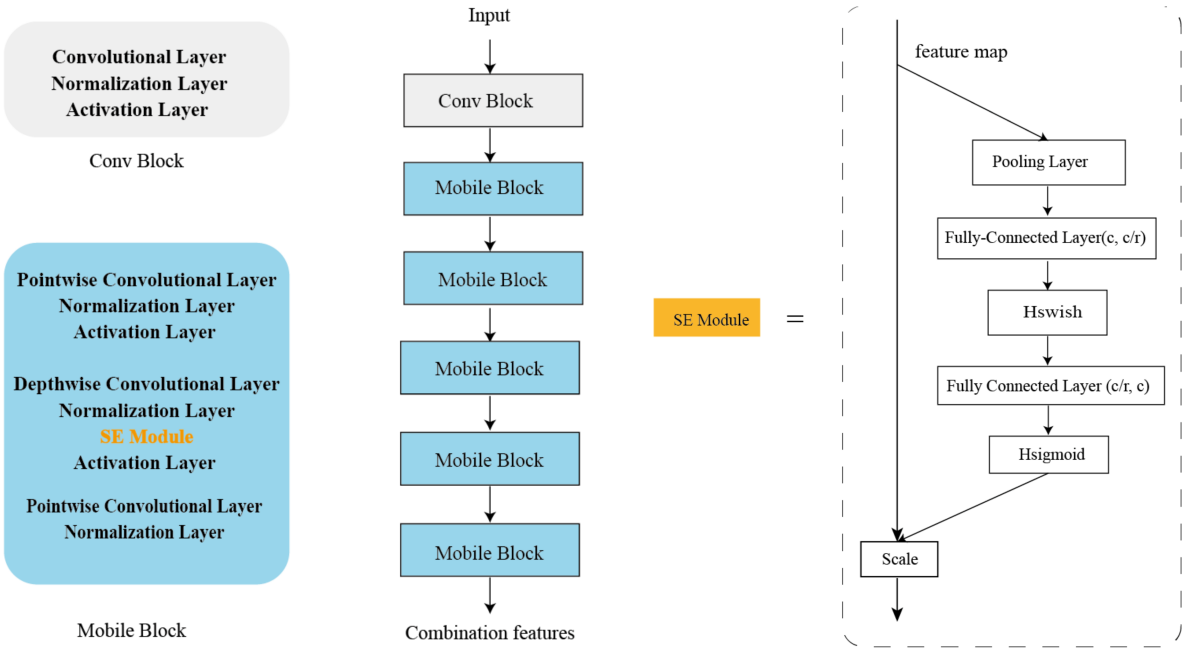
(**d**) Raw CSI phase (**e**) Unwrapped CSI phase (**f**) Processed CSI phase

Figure 4. Comparison of the CSI phase before and after processing. The upper row shows CSI radio images, and the lower row shows the waveform of the CSI phase.

3.2.3. Lightweight Few-Shot Network

The key structure of WiGR is a lightweight few-shot network that consists of a feature extraction subnetwork and a similarity discrimination subnetwork. The function of the feature extraction subnetwork is to extract advanced features of support samples and testing samples, and the features of testing samples and support samples are then combined in-depth. The function of the similarity discrimination subnetwork is to determine the relationship of combination features and output the similarity score of these gestures. The samples with the highest score are considered to be of the same type. Additionally, we introduce depthwise separable convolution and an inverted residual layer of a linear bottleneck [30,31] in the network to reduce model computations and parameters.

Feature extraction subnetwork. As shown in Figure 5, we adopted the one "conv block" and five "mobile blocks" to construct the feature extraction subnetwork.



(**a**) Feature extraction subnetwork (**b**) SE module

Figure 5. An illustration of the structure of the feature extraction subnetwork (**a**) and the SE (Squeeze and Excitation) module (**b**). The feature extraction subnetwork consists of a conv block and a mobile block.

Specifically, conv block is a normal CNN structure that consists of a convolutional layer, a normalization layer, and an activation layer. Each convolutional layer has multiple three-dimensional (3-D) convolutional kernels, and each 3-D convolutional kernel consists of multiple two-dimensional (2-D) convolutional kernels. Each 2-D convolutional kernel performs a convolution operation on the CSI data and can simultaneously extract the spatial features and temporal dynamics of the CSI data. The CSI datum of each gesture obtained from one antenna is a 2-D radio image (see Figure 4a), which can be denoted as

$$X \in R^{N_c \times T} \tag{2}$$

where R is a real number, N_c is the number of subcarriers, and T is the number of data frames. As an analogy to the image recognition problem, one CSI datum is analogized to a video frame, where N_c looks like the pixel in one frame and T looks like the number of frames. If there are N CSI data ($N = N_{Tx} \times N_{Rx}$), the output Q_i of the ith 3-D convolutional kernel can be denoted as

$$Q_i = \sum_{N=1}^{N} X_N * W_i^N + b_i \tag{3}$$

where X_N is the Nth 2-D CSI datum, W_i^N is the Nth 2-D convolutional kernel of the ith 3-D convolutional kernel, and b_i is the ith bias parameter. Benefitting from the excellent feature extraction capabilities of CNN [6–8], the feature extraction process of the feature extraction subnetwork is effective. In addition, we use depthwise separable convolution instead of ordinary convolution operations in the convolutional layer to reduce the network's parameters and computing consumption. The normalization layer can accelerate network training by reducing the internal covariate shift. The activation layer adopts two different activation functions, i.e., H-swish [45] (HS) and ReLU [46] (RE). H-swish is an improved version of rectified linear unit (ReLU) and can work on more features, but the calculation consumption of H-swish will also increase compared with the ReLU. Therefore, we used H-swish and ReLU alternately to balance the complexity and accuracy of the network.

The mobile block is based on a linear bottleneck with an inverted residual structure [30], which is beneficial for deployment on mobile devices. Firstly, the block makes

a low-dimensional compressed feature high-dimensional using a pointwise convolution layer consisting of M convolution kernels with a kernel size of 1 × 1. M is the number of convolution kernels, whose size is determined by the parameter fac. fac is used to change the number of feature dimensions proportionally, which is beneficial for reducing the model calculation complexity. It then uses a depthwise convolution layer consisting of M convolution kernels with a kernel size of 3 × 3 or 5 × 5 to further extract features and uses an SE module to enhance the robustness of the feature map. The SE module is optional in the mobile block. Finally, the features are projected back to a low-dimensional representation using another pointwise convolution kernel.

The SE module, as per [32], is a lightweight attention model based on the squeeze and excitation structure. The SE module is used to enhance the robustness of the feature map by generating relation weights for each channel of the feature map. Firstly, the SE block uses a global average pooling layer to squeeze the feature map obtained from the upper layer into a 1 × 1 × c feature channel vector, where c is the number of channels. To show the correlation between feature channels, it then uses two fully connected layers to reduce c to c/r, where r is the reduction factor, and then return c/r to c to obtain a feature attention weight of the feature map. This operation reduces the consumption of the calculation. The Scale operation multiplies the feature attention weight with the feature map and outputs a robust feature. Table 3 shows the specifications for the feature extraction subnetwork, where #Out denotes the number of channels of output features map, SE denotes whether there is an SE module in block, NL denotes the types of activation function, and S stands for stride.

Table 3. Specifications for the feature extraction subnetwork.

Input	Operator	Kernel Size	M	#Out	SE	NL	S
1800 × 114 × 3	Conv Block	3 × 3	-	16	False	HS	(2, 1)
900 × 114 × 16	Mobile Block	3 × 3	16 × *fac*	16 × *fac*	True	RE	2
450 × 57 × (16 × *fac*)	Mobile Block	3 × 3	72 × *fac*	24 × *fac*	False	RE	2
225 × 29 × (24 × *fac*)	Mobile Block	5 × 5	96 × *fac*	40 × *fac*	True	HS	2
113 × 15 × (40 × *fac*)	Mobile Block	5 × 5	240 × *fac*	80 × *fac*	True	HS	2
57 × 8 × (80 × *fac*)	Mobile Block	5 × 5	240 × *fac*	80 × *fac*	True	HS	2

Similarity discrimination subnetwork. Similar to the feature extraction subnetwork, we also adopted a CNN structure to construct the similarity discrimination subnetwork. As shown in Figure 6, we utilized a conv block to further analyze the representational information of the combination features obtained from the feature extraction subnetwork. We used an average pooling layer made up of a 3 × 3 kernel to reduce the number of parameters. The following layer is a convolutional layer used to further extract features. A flatten layer was used to condense multidimensional features into one dimension, and it is usually used in the transition from a convolutional layer to a fully connected layer. The fully connected layer mapped the learned distributed feature representation to the sample labeling space with a sigmoid as an activation function and output the similarity score of gestures in the range of 0 to 1. A large score means that the combined features belong to the same type of gestures. Thus, the similarity discrimination subnetwork could determine the relationship of samples accurately. Table 4 shows the specifications for the similarity discrimination subnetwork.

3.2.4. Episode-Based Training Strategy

We adopted an episode-based training strategy [20] to train the lightweight few-shot network. In each episode, we extracted K (e.g., 5) types of gestures uniformly at random from the training set D without replacement, and took G (e.g., up to 5) samples from each gesture to simulate support set S. We then took the remaining samples of each gesture to simulate testing set Q. Subsequently, we used a feature extraction subnetwork $f_\Phi(\cdot)$ to calculate the feature e of support sample x_i and the feature o of testing sample x_j. These features were combined in-depth with the operator concate (e, o). Finally, we fed these combined features into the similarity discrimination subnetwork $g_\Phi(\cdot)$, which produced

a similarity score $h_{i,j}$, representing the similarity between x_i and x_j. The $h_{i,j}$ is defined as follows:

$$h_{i,j} = g_\varphi(concate(f_\varphi(x_i), f_\varphi(y_i))) \quad (4)$$

Additionally, the lightweight few-shot network adopts the mean square error loss function J, defined as follows:

$$J = \sum_{j=1}^{Q} \sum_{i=1}^{K} (h_{i,j} - P(y_i == y_j))^2 \quad (5)$$

where y_i and y_j represent labels of sample $x_i$and sample x_j, respectively. $P\ (y_i == y_j)$ indicates whether y_i and y_j are equal. If they are equal, $P\ (y_i == y_j) = 1$; otherwise, $P\ (y_i == y_j) = 0$. Moreover, we adopted the gradual warmup learning scheduler [6] to minimize the loss function J.

Based on the episode-based training strategy, in each training episode, we can randomly produce a training support set and a training query set to simulate the support set and testing set encountered in the test scenario. We repeated the above process until the model could learn a robust transfer knowledge from the labeled training set D. We then applied the learned transfer knowledge to the new testing domain to address the domain shift problem.

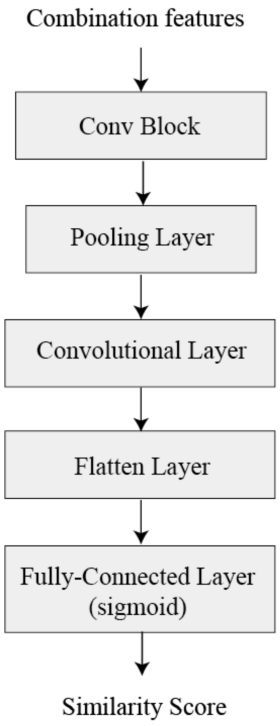


Figure 6. Structure of the similarity discrimination subnetwork.

Table 4. Specifications for the similarity discrimination subnetwork.

Input	Operator	Kernel Size	#Out	SE	NL	S
$57 \times 8 \times (160 \times$ fac)	Conv Block	1×1	$576 \times fac$	False	HS	1
$57 \times 8 \times (576 \times fac)$	Pooling Layer	3×3	-	False	-	1
$57 \times 8 \times (576 \times fac)$	Conv Layer	1×1	$128 \times fac$	False	HS	1

4. Results

We conducted extensive experiments on two data sets (SignFi [10] and CSIDA) to verify WiGR's effectiveness. We implemented the proposed system on a PyTorch 1.8.0 framework on an Intel(R) Xeon(R) CPU E5-2630 v4 @2.20GHz with an Nvidia Titan X Pascal GPU and 32.0 GB of RAM.

The SignFi data set and our CSIDA data set are both Wi-Fi data sets and include CSI data with various domain factors. The SignFi data set includes two domain factors, i.e., different environments and users. The CSIDA data set includes three domain factors, i.e., different environments, users, and locations.

4.1. Recognition Performance Evaluation

Recognizing new types of gestures. The ability to recognize new types of gestures is important for enhancing the scalability of a gesture recognition system. The few-shot learning method, the key technology used in this paper, can realize the generalization of the model through just a few support samples. This is the key difference between the few-shot learning method and other domain adaptation methods. To verify the ability of the proposed system to identify new types of gestures through just a few samples, we compared it with other few-shot learning methods on the SignFi data set and the CSIDA data set. Table 5 demonstrates that the WiGR model can achieve 98.6%, 97.2%, and 95.8% accuracy when it recognizes 10, 20, and 30 new types of gestures, respectively, under the condition that 100 new types of gestures are used for training and each new gesture has three support samples. Compared with other methods, the improvement in accuracy is more than 10 percentage points.

Table 5. Accuracy of recognizing new types of gestures with three samples using the SignFi data set.

Models	Number of Training Types	Number of New Testing Types		
		10	20	30
Siamese Network [18]	100	84.1%	85.2%	79.5%
Matching Network [20]		78.4%	73.2%	69.9%
Prototype Network [21]		76.6%	71.5%	67.8%
Relation Network [22]		89.2%	85.5%	81.4%
WiGR (fac = 1/6) (ours)		98.2%	96.8%	93.0%
WiGR (fac = 1/4) (ours)		98.6%	97.2%	95.8%

Table 6 demonstrates that our WiGR model has better recognition performance than the other few-shot learning models. When WiGR was trained with three old types of gestures, it achieved 91.4% and 84.9% recognition accuracies for two and three new types of gestures, and each new type of gesture had three support samples. Because our CSIDA data set does not have enough training types for training, the recognition accuracy dropped slightly compared with the SignFi data set. In general, the accuracy of the proposed WiGR model is remarkably higher than the other few-shot learning models in all evaluations.

Cross-domain evaluation. To verify that the proposed WiGR system does play a role in cross-domain recognition, we conducted extensive cross-domain experiments by splitting the data set according to the layout of the environment, the user who performs the gestures, and the user's location. We compared our model with other traditional gesture recognition systems, such as WiGeR [33], which utilizes a classifier with a DTW algorithm, and WiCatch [34], which employs SVM with a MUSIC signal processing algorithm. In addition, since the proposed WiGR adopts components of a CNN to construct a network, then, to verify the superiority of the CNN-based WiGR, the selected comparison systems were based on machine learning algorithms (i.e., WiGeR and WiCatch) or based on only a sample structure of a CNN without the capability of cross-domain recognition (i.e., SignFi [10]). Moreover, Siamese-LSTM [26], using a Siamese network that consists of a CNN and LSTM to address domain shift problems, is a typical few-shot domain adaptive method and was used as a baseline method. These competitive methods were useful in verifying the effectiveness of our WiGR model in cross-domain evaluation.

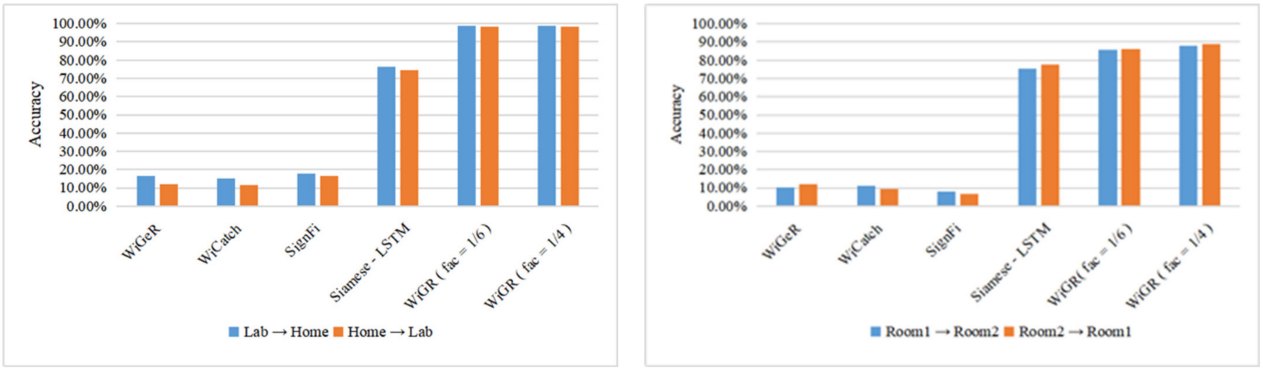
- Cross-environment evaluation. For the environmental shift, we used CSI data from two different environments. All the data from one environment were used for training, while data from the other environment were used for testing. Figure 7 shows the accuracy for recognizing gestures that are collected in a new environment with three support samples for each gesture, where A $\rightarrow$ B denotes that A is the training set and B is the testing set. We can see that traditional machine learning methods, such as WiGeR and WiCatch, and an ordinary convolutional network, such as SignFi, have almost no shift ability when testing samples from a totally new environment, while our proposed WiGR model could achieve an average accuracy of 98% and 88% using the SignFi and CSIDA data sets, respectively, and therefore remarkably outperforms the other methods.
- Cross-user evaluation. For the user shift, we evaluated all methods in the same environment to control variables, and then conducted leave-one-user-out cross-validation using CSI traces from different users. In other words, we adopted CSI traces collected by some users as the training set and utilized the CSI traces of the other users as the testing set. Figure 8 shows the results of recognizing new user's gestures, and each gesture has three support samples. From Figure 8, we can see that the cross-user recognition accuracies of WiGeR, WiCatch, and SignFi are no more than 80%, but still better than the cross-environment performance. The reason is that the training data set has abundant user domain information for extracting common features. Our WiGR model achieves state-of-the-art performance with a recognition average recognition accuracy of 92% and 91% using the SignFi and CSIDA data sets, respectively. Compared with the domain-adaptive Siamese-LSTM, our method improves its performance by about 10%, which demonstrates that WiGR alleviates the problem of domain shift effectively by learning transferable knowledge from the training set and using the features extracted from the support samples to recognize gestures.
- Cross-location evaluation. For the location shift, we evaluated all the methods in the same environment to control variables, and then performed leave-one-location-out cross validation using CSI traces. As shown in Figure 9, our proposed WiGR model still shows excellent performance with an average recognition accuracy 90.8%, and, therefore, outperforms other methods. In addition, when the testing CSI data are collected at Loc. 1 and Loc. 3, the recognition performance is slightly reduced compared with the data collected at Loc. 2. This is because that the user performed gestures at Loc. 1 or Loc. 3 is very close to Rx or Tx. In this case, the user's body will block more signals, resulting in weaker signal propagation, which in turn affects gesture recognition performance.

Table 6. Accuracy of recognizing new types of gestures with three samples using the CSIDA data set.

Models	Number of Training Types	Number of New Testing Types	
		2	3
Siamese Network [18]	3	65.0%	63.4%
Matching Network [20]		60.2%	55.4%
Prototype Network [21]		63.6%	58.4%
Relation Network [22]		77.4%	74.6%
WiGR (fac = 1/6) (ours)		89.9%	83.6%
WiGR (fac = 1/4) (ours)		91.4%	85.9%

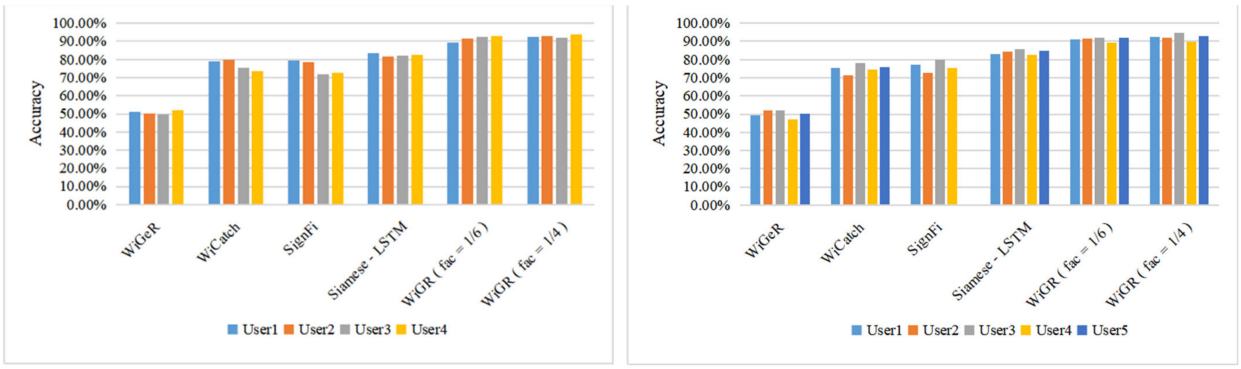
Different users have different physical body conditions, gesture speeds, and hand movements for the same gestures, and there are two different layout environments. Moreover, different locations can result in different signal propagation paths. These three factors may result in different CSI signal patterns, even for the same gesture. However, due to the excellent feature extraction capabilities of the CNN, CNN-based gesture recognition systems (i.e., WiGR and Siamese-LSTM) have superior cross-domain recognition performance compared to other gesture recognition systems based on traditional machine learning methods (i.e., WiGeR and WiCatch). Although SignFi also adopts the components of a

CNN, the structure of SignFi is too simple to play a role in cross-domain recognition. Additionally, the WiGR model can learn more robust transferable knowledge through supervised training, thereby eliminating the influence of individual, environmental, and location factors on gestures, which allows WiGR to achieve gesture recognition under a new domain with only a few samples.



(**a**) The SignFi data set. (**b**) The CSIDA data set.

Figure 7. Accuracy of cross-environment evaluation.



(**a**) The SignFi data set. (**b**) The CSIDA data set.

Figure 8. Accuracy of cross-user evaluation.

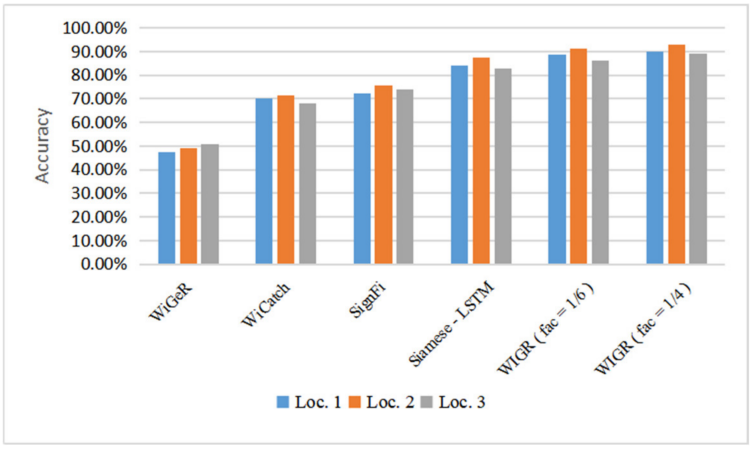


Figure 9. Accuracy of cross-location evaluation using the CSIDA data set.

4.2. Model Complexity Analysis

The complexity of a gesture recognition model, affecting storage space and computational cost, plays a vital role in mobile deployment. We utilized indicators *Params* and *MACs* to reflect the complexity of the model. Params refers to the model's parameters—the

smaller the value, the smaller the storage space required by the model. *MACs* refer to the calculations required by the model, and a smaller value corresponds to fewer computing resources be consumed. *M* is an abbreviation for million. The key network of the WiGR is an improved few-shot learning model, in which lightweight blocks are introduced. Therefore, to verify the effectiveness of these lightweight blocks, we compared them with normal few-shot learning models [18,20,21]. Table 7 shows that the WiGR outperforms other popular few-shot learning methods [18,20–22] about the model's complexity by a clear margin, and we can see that Params and MACs have the smallest value when scaling factor fac = 1/6. Thus, the value of fac also plays an important role in the model's computational complexity. Experimental results show that WiGR is a state-of-the-art lightweight gesture recognition model by far when *fac* = 1/6.

Table 7. Models' complexity performance.

Models	Complexity	
	Params (M)	MACs (M)
Siamese Network [18]	33.081921	98.998656
Matching Network [20]	0.087254	489.910816
Prototype Network [21]	0.014248	2.693704
Relational network [22]	0.155681	80.842768
WiGR (fac = 1/4) (ours)	0.095417	38.393904
WiGR (fac = 1/6) (ours)	0.005617	1.349448

4.3. The Influence of The Number of Antennas

Since only some high-end mobile devices are tailored for multiple input, multiple output (MIMO) communication with several antennas, it is necessary to study the influence of the number of antennas on the recognition performance of the WiGR model.

With different numbers of receiving antennas, we conducted cross-domain recognition evaluation and single-domain recognition evaluation. Specifically, the CSI data collected in Room 2 are selected as the test data in cross-environment evaluation, the CSI data performed by User 5 are selected as test data in cross-user evaluation, and the CSI data collected in Location 3 are selected as test data in cross-location evaluation. In single-domain recognition evaluation, we selected some CSI data of six gestures performed by User 1 at Location 1 of Room 1 as training data, and the remaining CSI data of each gesture as testing data. Similarly, there are three support samples provided for each gesture. From Table 8, we can see that the larger the number of receiving antennas, the better the recognition performance. This is because multiple receiving antennas can transmit richer CSI data, which helps the WiGR model recognize gestures more accurately. In addition, when only one transmitting antenna and one receiving antenna are used, the cross-domain recognition accuracy of the WiGR model can only reach 70.2–73.2%, and the single-domain recognition accuracy of the WiGR model can reach 91.3%. To a certain extent, it can still show a cross-domain recognition ability and a good single-domain recognition ability, although the effect is not as good as using MIMO.

Table 8. The influence of the number of antennas on the performance of cross-domain recognition.

Number of Transmitting Antennas	Number of Receiving Antennas	Cross-Domain Evaluation			Single-Domain Evaluation
		Room 2	User 5	Location 3	
1	1	70.2%	79.8%	73.2%	91.3%
1	2	78.4%	85.4%	80.7%	95.2%
1	3	87.8%	92.8%	89.2%	98.4%

5. Discussion

There are several limitations to our proposed WiGR, and they can become fruitful directions of further investigation. Firstly, we only discuss the impact of finite domains (i.e., environment, users, and locations). In fact, CSI signals will also be affected by the

orientation of the face [17] and other signal sources. These factors need to be considered in future work.

Secondly, in many human–computer interaction scenes, such as virtual games, automatic driving assistance systems, sign language recognition, and intelligent robot control, the distance between the user and the transmitter/receiver, or the distance between the transmitter and the receiver, is not fixed. Therefore, we simply set these distances according to [17,41]. In future work, we will focus on a specific application scenario (e.g., controlling a mobile phone with gestures) and discuss the setting of distance based on the application scenario.

Finally, in our experiment, the gestures were performed in the LoS. Wi-Fi signals do not require LoS propagation. Therefore, we are interested in expanding WiGR to the LoS scenario. For example, we can separate the transmitter and receiver with a wall, and then study the impact on the Wi-Fi signal in this case.

6. Conclusions

In this paper, we propose WiGR, a novel and practical Wi-Fi-based gesture recognition system. This system uses a lightweight few-shot network that is trained by an episode-based training strategy to eliminate the influence of domain shift. Lightweight and effective blocks are introduced into the network to achieve lower computational complexity and high performance. In addition, we made a CSIDA data set that includes CSI traces with various domain factors to verify the accuracy of the proposed WiGR in cross-domain evaluation. Extensive experiments on the SignFi [10] and CSIDA data sets show that the proposed WiGR is excellent in cross-domain recognition and computational complexity evaluation. It is a practical and lightweight gesture recognition system compared with existing gesture recognition systems.

Author Contributions: Conceptualization, P.H. and K.Y.; methodology, P.H. and K.Y.; software, K.Y.; validation, P.H. and X.Z; formal analysis, P.H. and X.Z.; investigation, P.H. and X.Z.; resources, K.Y.; writing—original draft preparation, P.H.; writing—review and editing, P.H. and C.T.; visualization, C.T.; supervision, C.T.; project administration, C.T.; funding acquisition, C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guangdong Provincial Applied Science and Technology Research and Development Program, grant number 2016B010125001, and the Natural Science Foundation of Guangdong Province, grant number 2018A030313797.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available at https://yongsen.github.io/SignFi/ (accessed on 15 March 2020).

Acknowledgments: We are grateful that the College of William and Mary has provided the SignFi data set.

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

References

1. Ahmed, H.F.T.; Ahmad, H.; Aravind, C.V. Device free human gesture recognition using wi-fi csi: A survey. *Eng. Appl. Artif. Intel.* **2020**, *87*, 103281. [CrossRef]
2. Fang, B.; Lv, Q.; Shan, J.; Sun, F.; Zhao, Y. Dynamic Gesture Recognition Using Inertial Sensors-based Data Gloves. In Proceedings of the IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), Toyonaka, Japan, 3–5 July 2019; pp. 390–395.
3. Zhang, Y.; Chen, Y.; Yu, H.; Yang, X.; Lu, W.; Liu, H. Wearing-independent hand gesture recognition method based on emg armband. *Pers. Ubiquitous Comput.* **2018**, *22*, 511–524. [CrossRef]
4. Kumar, P.; Saini, R.; Roy, P.P.; Dogra, D.P. A position and rotation invariant framework for sign language recognition (slr) using kinect. *Multimed. Tools Appl.* **2017**, *77*, 8823–8846. [CrossRef]

5. Marina, L.G.; Yingxu, W.; Faisal, A.; Padma, P.P. Kinect sensor gesture and activity recognition: New applications for consumer cognitive systems. *IEEE Consum. Electron.* **2018**, *7*, 88–94.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 6 June–1 July 2016; pp. 770–778.
7. Szegedy, C.; Vanhoucke, V.; Loffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 6 June–1 July 2016; pp. 2818–2826.
8. Huang, G.; Liu, Z.; Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
9. Zou, H.; Zhou, Y.; Yang, J.; Jiang, H.; Xie, L.; Spanos, C.J. WiFi-enabled Device-free Gesture Recognition for Smart Home Automation. In Proceedings of the 2018 IEEE 14th International Conference on Control and Automation (ICCA), Anchorage, AK, USA, 12–18 June 2018; pp. 476–481.
10. Ma, Y.; Zhou, G.; Wang, S.; Zhao, H.; Jung, W. SignFi: Sign language recognition using WiFi. In *Proceedings of the ACM on Interactive, Mobile, Wearable Ubiquitous Technologies*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–21.
11. Akhtar, Z.U.A.; Wang, H. WiFi-Based Gesture Recognition for Vehicular Infotainment System—An Integrated Approach. *Appl. Sci.* **2019**, *9*, 5268. [CrossRef]
12. Aly, S.; Aly, W. DeepArSLR: A Novel Signer-Independent Deep Learning Framework for Isolated Arabic Sign Language Gestures Recognition. *IEEE Access* **2020**, *8*, 83199–83212. [CrossRef]
13. Zhang, J.; Tang, Z.; Li, M.; Fang, D.; Nurmi, P.; Wang, Z. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18), New Delhi, India, 29 October–2 November 2018; pp. 305–320.
14. Wang, J.; Chen, X.; Fang, D.; Wu, C.Q.; Yang, Z.; Xing, T. Transferring compressive-sensing-based device-free localization across target diversity. *IEEE Trans. Ind. Electron.* **2015**, *62*, 2397–2409. [CrossRef]
15. Chang, L.; Chen, X.; Wang, Y.; Fang, D.; Wang, J.; Xing, T.; Tang, Z. FitLoc: Fine-grained and low-cost device-free localization for multiple targets over various areas. In Proceedings of the 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
16. Xiao, C.; Han, D.; Ma, Y.; Qin, Z. CsiGAN: Robust Channel State Information-based Activity Recognition with GANs. *IEEE Internet Things J.* **2019**, *6*, 10191–10204. [CrossRef]
17. Zheng, Y.; Zhang, Y.; Qian, K.; Zhang, G.; Liu, Y.; Wu, C.; Yang, Z. Widar3.0: Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, Seoul, Korea, 17–21 June 2019; pp. 313–325.
18. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; Volume 2.
19. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), San Diego, CA, USA, 20–25 June 2005; pp. 539–546.
20. Vinyals, Q.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS '16); Curran Associates Inc.: Red Hook, NY, USA, 2016; pp. 3637–3645.
21. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical networks for few-shot learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Curran Associates Inc., Red Hook, NY, USA, 4–9 December 2017; pp. 4080–4090.
22. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1199–1208.
23. Zou, H.; Zhou, Y.; Yang, J.; Liu, H.; Das, H.P.; Spanos, C.J. Consensus adversarial domain adaptation. In Proceedings of the AAAI conference on artificial intelligence, Honolulu, Hawaii, USA, 27 January–1 February 2019; pp. 5997–6004.
24. Zhou, Z.; Zhang, Y.; Yu, X.; Yang, P.; Li, X.; Zhao, J.; Zhou, H. XHAR: Deep Domain Adaptation for Human Activity Recognition with Smart Devices. In Proceedings of the 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Como, Italy, 22–25 June 2020; pp. 1–9.
25. Lan, G.; Heit, B.; Scargill, T.; Gorlatova, M. GazeGraph: Graph-based few-shot cognitive context sensing from human visual behavior. In Proceedings of the 18th Conference on Embedded Networked Sensor Systems, Virtual Event, Japan, 16–19 November 2020; pp. 422–435.
26. Yang, J.; Zou, H.; Zhou, Y.; Xie, L. Learning Gestures from WiFi: A Siamese Recurrent Convolutional Architecture. *IEEE Internet Things J.* **2019**, *6*, 10763–10772. [CrossRef]
27. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.

28. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856.
29. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Anfreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
32. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2820–2828.
33. Al-Qaness, M.A.A.; Li, F. Wiger: WiFi-based gesture recognition system. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 92. [CrossRef]
34. Tian, Z.; Wang, J.; Yang, X.; Zhou, M. WiCatch: A Wi-Fi based hand gesture recognition system. *IEEE Access* **2018**, *6*, 16911–16923. [CrossRef]
35. Sen, S.; Lee, J.; Kim, K.; Congdon, P. Avoiding multipath to revive inbuilding WiFi localization. In Proceedings of the International Conference on Mobile Systems, Applications, and Services, Taipei, Taiwan, 25–28 June 2013; pp. 249–262.
36. Xiao, Y. IEEE 802.11n: Enhancements for higher throughput in wireless LANs. *IEEE Wirel. Commun.* **2005**, *12*, 82–91. [CrossRef]
37. Halperin, D.; Hu, W.; Sheth, A.; Wetherall, D. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 53. [CrossRef]
38. Welch, L.R. Hidden Markov Models and the Baum-Welch Algorithm. *IEEE Inf. Theory Soc. Newsl.* **2003**, *53*, 194–211.
39. Atheros CSI Tool. Available online: https://wands.sg/research/wifi/AtherosCSI/ (accessed on 20 May 2020).
40. Xia, P.; Zhou, S.; Giannakis, G.B. Adaptive MIMO-OFDM based on partial channel state information. *IEEE Trans. Signal Process.* **2004**, *52*, 202–213. [CrossRef]
41. Wang, F.; Feng, J.; Zhao, Y.; Zhang, X.; Zhang, S.; Han, J. Joint Activity Recognition and Indoor Localization with WiFi Fingerprints. *IEEE Access* **2019**, *7*, 80058–80068. [CrossRef]
42. Qian, K.; Wu, C.; Yang, Z.; Jamieson, K. Widar: Decimeter-Level Passive Tracking via Velocity Monitoring with Commodity WiFi. In Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Chennai, India, 10–14 July 2017; pp. 1–10.
43. Qian, K.; Wu, C.; Zhang, G.; Yang, Z.; Liu, Y. Widar2.0: Passive Human Tracking with a Single Wi-Fi Link. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, 10–15 June 2018; pp. 350–361.
44. Restuccia, F.; D'Oro, S.; Al-Shawabka, A.; Belgiovine, M.; Angioloni, L.; Ioannidis, S.; Chowdhury, K.; Melodia, T. DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms. In Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19), Catania, Italy, 2–5 July 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 51–60.
45. Howard, A.; Sandler, M.; Chu, G.; Chen, L.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 1314–1324.
46. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.