Joint Activity Recognition and Indoor Localization with WiFi Fingerprints

FEI WANG\textsuperscript{1,2}, JIANWEI FENG\textsuperscript{2}, YINLIANG ZHAO\textsuperscript{*1}, XIAOBIN ZHANG\textsuperscript{1}, SHIYUAN ZHANG\textsuperscript{1}, AND JINSONG HAN\textsuperscript{1}.
\textsuperscript{1}Department of Computer Science and Technology, Xi’an Jiaotong University, Xi’an, Shaanxi 710049 China
\textsuperscript{2}The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA

\*Corresponding author.

ABSTRACT Recent years have witnessed the rapid development in the research topic of WiFi sensing that automatically senses human with commercial WiFi devices. This work falls into two major categories, i.e., the activity recognition and the indoor localization. The former work utilizes WiFi devices to recognize human daily activities such as smoking, walking, and dancing. The latter one, indoor localization, can be used for indoor navigation, location-based services, and through-wall surveillance. The key rationale behind this type of work is that people behaviors can influence the WiFi signal propagation and introduce specific patterns into WiFi signals, called WiFi fingerprints, which further can be explored to identify human activities and locations. In this paper, we propose a novel deep learning framework for joint activity recognition and indoor localization task using WiFi channel state information fingerprints. More precisely, we develop a system running standard IEEE 802.11n WiFi protocol, and collect more than 1400 CSI fingerprints on 6 activities at 16 indoor locations. Then we propose a dual-task convolutional neural network with 1-dimensional convolutional layers for the joint task of activity recognition and indoor localization. Experimental results and ablation study show that our approach achieves good performances in this joint WiFi sensing task. Data and code have been made publicly available at https://github.com/geekfeiw/apl.

INDEX TERMS CSI Fingerprints, Activity Recognition, Indoor Localization, 1D Convolutional Neural Networks

I. INTRODUCTION

WiFi devices have been extensively explored as pervasive sensors for human sensing tasks such as activity recognition \cite{1}–\cite{4}, indoor localization \cite{5}–\cite{8}, and health-care applications \cite{9}–\cite{14}. This prosperity benefits from several special properties of WiFi, including the the ubiquitous deployment of commercial WiFi devices, the robustness to lighting condition and occlusion overcoming limitation of cameras, and the non-intrusiveness sensing requiring no user’s extra effort.

Though there is abundant work on the specific aforementioned WiFi human sensing task \cite{1}–\cite{4}, \cite{9}–\cite{15}, to the best of our knowledge, seldom work aims at completing the joint activity recognition and indoor localization task. Carrying out the joint task would breed numerous useful human-computer interaction applications. For example, in a smart home with Internet-of-Things (IoT) devices \cite{16}, \cite{17}, the devices could precisely respond differently to the same gesture command based on user’s location. More specifically, the user can use the gesture of ‘hand down’ to turn down the television in front of her, whereas she can also use the same gesture to lower the air conditioner’s temperature when standing close to the air conditioner.

The joint task can be summarized as the following two folds. (1) Recognizing activities conducted at different locations. (2) Localizing the user by the activities. However, there are two major challenges lying in the way. The first challenge is that WiFi fingerprint differs even when performing a same activity but at different locations, thus we need to look for a same representation for activities conducted at all locations. The second one is that WiFi fingerprints vary when performing activities, thus we have to explore features for indoor localization from the fingerprint variances.

To conclude the above challenges formally, WiFi fingerprint, $W$, contains two components at the same time, activity category ($A$) and user location ($L$). We denote the WiFi fingerprint as $W(A,L)$. Joint activity recognition and indoor localization task aims to learn a function $f$, which is capable to classify activity categories ($f : W(A,L) \to A$) and to localize the user ($f : W(A,L) \to L$), simultaneously. Thus we formalize the joint task as $f : W(A,L) \to (A,L)$.

To this end, in the paper we propose a novel 1-
dimensional Convolutional Neural Network (C1D) including
two branches, one for activity recognition and the other
for indoor localization. To date, conventional 2-dimensional
Convolutional Neural Networks (C2D), which have brilliant
ability to learn features from raw data, boost the development
of computer vision \[18\]–\[23\], robotics \[24\]–\[27\], machinery \[28\]–\[30\], etc. Unlike C2D that processes 2D spatial
data such as images, C1D is capable to process 1D temporal
data. For temporal WiFi fingerprints, we design a C1D based
on the ResNet \[18\] to carry out the joint task of activity
recognition and indoor localization.

To evaluate our proposed approach, we implement the
standard IEEE 802.11n protocol in two universal software
radio peripheral (USRP) sets, Ettus N210\[1\] where one broad-
casts WiFi signals and the other parses Channel State Infor-
mation (CSI) fingerprints of WiFi for joint task. We define
6 hand gestures for potential human-computer interaction
applications, namely, hand up, hand down, hand left, hand
right, hand circle and hand cross. One volunteer repeats these activities 15 times at each location (16 locations in all)
and contributes a dataset with 1394 samples (after excluding
the invalid data). We evaluate our proposed C1D on this
dataset and present the results with several metrics such as in
confusion matrix, f1 scores, convolutional feature maps, etc.
Experimental results show our proposed C1D achieves a very
promising performance in the joint task. We summarized our
contributions as follows.

1. We propose a novel 1-dimensional Convolutional Neu-
ral Network for the joint activity recognition and indoor
localization task with the CSI fingerprints as inputs.

2. We implement IEEE 802.11n protocol in two USRP sets
and build a dataset specifically for the joint task. We evaluate
the performance of proposed deep network on this dataset
and fully discuss the results.

II. RELATED WORK

A. CSI FINGERPRINTS

CSI fingerprints of WiFi have been widely utilized for activity
recognition \[1\]–\[4\], \[9\]–\[11\] and indoor localization \[31\]–\[34\]. As for activity recognition, in \[5\], \[9\]–\[11\], CSI finger-
prints are used to detect user falling especially for the elderly-care system. In \[2\], CSI fingerprints are used to infer user keystroke. Further in \[35\], researchers find that CSI fingerprints can reveal people’s typing when they use
smart phones in public WiFi. In \[1\], \[36\] CSI fingerprints are designed for hand sign recognition for human-computer
interactions. As for indoor localization, \[31\]–\[34\] collect CSI fingerprints corresponding to people locations, and train clas-
sifiers to localize people with collected CSI fingerprints. To
our best survey, there is no work on joint activity recognition
and indoor localization, which is very useful in controlling
different smart devices at different locations with a set of
pre-defined activities. We achieve this task by a dual-branch
Convolutional Neural Network.

B. CSI FINGERPRINTS CLASSIFICATION

There exist three popular approaches in CSI fingerprints
classification. (1) Hand-crafted features + Support Vector
Machine (SVM) \[37\]–\[39\], \[41\] apply statistical values of CSI
time-series such as the mean, maximum, minimum, entropy,
etc., as features to train SVM with kernel methods for CSI
fingerprints classification. This approach requires expertise
in designing features, which is even much harder on joint
activity recognition and indoor localization. (2) Dynamic
Time Wrapping (DTW)+ k Nearest Neighbors (kNN): \[1\],
\[3\], \[36\] first build a dataset with CSI fingerprints. When
classifying a test CSI sample, this approach requires com-
puting all distances between the test sample and all samples
in the dataset, which is time-consuming compared to pre-
training a classifier first. (3) Deep learning: \[31\], \[34\] utilizes
deep Boltzmann Machine (DBM) to do indoor localization.
However DBM relies heavily on careful design and tricks
to converge. \[32\], \[33\] apply 3-5 convolutional layers on
activity recognition. In general, the shortage in the depth
limits the performance. \[36\] utilizes ResNet \[18\] and In-
ception \[20\] to categorize CSI fingerprints, whereas it only
handles single moment CSI, i.e., rather than handles temporal
CSI fingerprints. In this paper, we propose a ResNet-based
Convolutional Neural Network to do CSI fingerprints classi-
fication.

C. 1D CONVOLUTIONAL NEURAL NETWORK

Conventional Convolutional Neural Network (C2D) \[18\]–\[20\] are designed for 2D inputs such as images. C2D applies
2D convolutional kernels to sweep along the width and height
of an image to capture its semantic and structural informa-
tion for image classification \[18\]–\[20\], object detection \[38\],
instance segmentation \[21\], etc. In \[39\], \[40\], researchers
apply 3D CNN (C3D) on video data, which sweeps along the
width, height, and time of the video to capture information
both in spatial and in temporal. In this paper, we apply 1D
convolutional kernels to sweep along the time axis of the CSI
fingerprint series to capture the temporal information of CSI
fingerprints, which works well in the joint task of activity
recognition and indoor localization.

III. DATA COLLECTION

A. HARDWARE

We implement the standard IEEE 802.11n protocol in two
universal software radio peripherals (USRPs) to collect CSI
fingerprints. As shown in FIGURE \[1\] the first two figures
are the top view and front view of the USRP (Ettus N201),
respectively. The USRP is composed mainly of a mother
board, a daughter board and a WiFi antenna, which is used
to broadcast or receive WiFi signals under the control of
GNU Radio\[4\]. The details are listed below. Meanwhile, the
assembling diagram is shown in FIGURE \[2\].
FIGURE 1: Main hardware: Ettus USRP N210 and Ettus Clock.

FIGURE 2: System framework. The system contains two sets of personal computers and USRPs, which work as the WiFi transmitter and receiver, respectively. An Ettus clock synchronizes the two sets.

B. ACTIVITY AND LOCATION

We design 6 activities, namely, hand up, hand down, hand left, hand right, hand circle, and hand cross, for human-computer interaction applications, as shown in FIGURE 3. This cluster of activities covers the majority of daily commands for smart Internet-of-Things home, where using cameras are not practical due to security and privacy concerns. Here we illustrate how our proposed activities work by the case of television. “Hand up” and “hand down” can be used to turn up and down the voice volume, respectively; “Hand left” and “hand right” indicate switching channels; “Hand circle” and “hand cross” are for CONFIRM command and CANCEL command, respectively.

Besides recognizing activity in smart home, localizing the user when s/he is doing an activity is also crucial for the joint task. By combining user’s activity and location together, we are able to infer user’s intention more precisely and make it possible for users to control a range of smart devices with the same activity. For example, the user may want to communicate with the television on the sofa, whereas s/he probably needs to control the air conditioner (AC) when standing in front of an AC. To make a proof-of-concept experiment, we collect CSI fingerprints when a volunteer does 6 activities at 16 locations shown in FIGURE 3.

C. CSI FINGERPRINT ANALYSIS

The volunteer repeats each of the 6 activities 15 times at 16 locations and contributes a dataset with 1440 samples. Next we visualize some samples varying in activities and locations to present the challenges of joint activity recognition and indoor localization task.

FIGURE 4 shows CSI fingerprints when the volunteer plays 6 activities at #10 location, in which the x-axis is the sampling index (time) and the y-axis is the amplitude of the CSI fingerprints. There are 52 time series in each CSI fingerprints, differed with 52 colors in FIGURE 4. 52 is the number of orthogonal frequency division multiplexing (OFDM) [41] sub-carriers that carry data in parallel in WiFi protocol. FIGURE 4 demonstrates that CSI fingerprints vary when the user conducts 6 activities at a same location.

FIGURE 5 illustrates 3 CSI fingerprint samples when the volunteer carries out “hand circle” at #3 location. FIGURE 5 shows that though the volunteer plays the same activity at the same location, CSI fingerprints are still very different in time-serial profile (left and middle), and in the start point of the activity (left and right). Besides, performing the same activity at different locations also largely varies CSI fingerprints as illustrated in FIGURE 6, making it challenging to find out shared features for one activity at all locations.

IV. METHODOLOGY

A. 1D CONVOLUTIONAL NEURAL NETWORK

As illustrated in FIGURE 4, FIGURE 5, and FIGURE 6, CSI fingerprints are time series with 52 sub-carriers. We donate it as $C \in R^{52 \times t}$, where $t$ is for sampled time, $R$ means Real number. Convolutional Neural Network approaches boost current pattern recognition applications due to the ability of learning powerful features directly from raw data. In this paper, we apply 1-dimensional Convolutional Neural Network (C1D) on CSI fingerprints for joint activity recognition and indoor localization task.
FIGURE 3: We define 6 gesture commands mainly for human-computer interaction applications in smart home, i.e., hand circle, hand up, hand cross, hand left, hand down, and hand right.

FIGURE 4: CSI fingerprint samples of 6 activities on #10 position.

FIGURE 5: Play ‘circle’ at the #3 position. CSI samples vary in profiles. CSI may be partly captured because of late action start.

FIGURE 6 (left) shows a 2-dimensional convolutional operation (C2D) for spatial data such as images, and FIGURE 7 (right) illustrates C1D for temporal inputs such as WiFi fingerprints. For a C2D, the input size is $7 \times 7$, a 2D convolutional kernel sized $3 \times 3$ sweeps along the width and height of the input with the stride of 2, and it leads to a result of $3 \times 3$. With this sweeping operation, spatial information of the input, such as the object location in an image, can be captured. Differing from C2D, C1D only sweeps along the time axis and temporal information in $C$ can be captured in this way, which is highly correlated with the user activities. Besides, consulting the widely used method that C2D takes
3 (3 for RGB color channels) as the channel for image data, in this paper, we take the 52 of $C$ as the channel of C1D operation (52 for 52 OFDM channels) for CSI fingerprints.

### B. PREPROCESSING

As shown in FIGURE 5 CSI fingerprints vary according to different activity start time and finish time. Thus we manually annotate the activity duration to calibrating the useful signal after data collection. We take the time series of 29th sub-carrier as the visualization example to show a duration annotation in FIGURE 8 (left). This annotating process enables us to directly use the segmented CSI fingerprints for the joint task. Meanwhile, the size of inputs needs to stay the same to train a C1D, so we upsample the segmented CSI fingerprints to make them the same size using the linear interpolation (in our experiment, The size is 192 that equals the original size). One interpolated sample is shown in FIGURE 8 (right).

### C. NETWORK FRAMEWORK

In computer vision community, ResNets [18] have been proved to be effective and advanced in many task, such as image classification [18], object detection [33], instance segmentation [31], etc. However the standard ResNets are implemented to process 2D inputs such as images, thus we re-implement a ResNet specifically for our temporal CSI fingerprints, termed as ResNet1D.

The main component of ResNet1D is the basic residual block as shown in FIGURE 9. We denote the input as $x$, and the output as $y$. With a shortcut link, $x$ becomes a part of $y$. Besides two convolutional layers convert $x$ to $f(x)$. In all the output of the basic residual block is

$$y = f(x) + x.$$ 

In FIGURE 9 (right), we illustrate a basic residual block in our implementation in details. In the $f(x)$ branch, $x$ is scanned by two C1Ds with size of $3 \times 1$ (‘C1D3×1’). Moreover, 1D batch normalization [42] (‘BN1D’) follows at each ‘C1D3×1’, and a Rectified Linear Unit activation function follows the first ‘C1D3×1’. In the shortcut channel, $x$ is processed by a C1D with the size of $1 \times 1$ (‘C1D1×1’) and a 1D batch normalization. Because $f(x)$ is from two C1Ds, the size of $f(x)$ and $x$ may be different, making it an error to do element-wise addition between $f(x)$ and $x$. The ‘C1D1×1’ in the shortcut branch is designed to do size matching between $f(x)$ and $x$.

Based on the basic residual block above, we build the ResNet1D as shown in FIGURE 10. The network takes CSI time series as input and predict user activity and location in parallel as output. The network contains 11 C1D layers, where 9 are shared, and each sub-task has one C1D independently, whose parameters are listed in FIGURE 10. Taking the first C1D, ‘C1D7 × 1, 128’, as an example, the C1D is with the kernel size of $7 \times 1$ and the output channel number of 128. Besides C1Ds, there is 1 max pooling operation following the first C1D and 2 average pooling operations following the last 2 C1Ds, respectively. Moreover, we use a fully-connected layer to predict each of the 6 activities with a separate score. The output of activity recognition is the activity with the highest score. Meanwhile we use a fully-connected layer to predict one location out of 16 locations where the activity is carried out.

Since this network contains 4 residual blocks (RBs) and in each block there exists one basic residual block, we call it ResNet1D-[1,1,1,1]. ResNet1D-[1,1,1,1] is an expandable framework by customizing the number of basic residual block in RBs. In the evaluation section, we will compare the performances of ResNet1D-[1,1,1,1], ResNet1D-[2,2,2,2], and ResNet1D-[3,3,6,3] (the later two RB settings are defaults in ResNet [18]).

### D. LOSS FUNCTION

The loss, $L$, that optimizes ResNet1D is the sum of two sub-tasks, activity recognition and indoor localization. We term it as follows.

$$L = L_{activity} + \lambda L_{location}$$

where $L_{activity}$ and $L_{location}$ are losses of activity recognition and indoor localization, respectively. $\lambda$ is to balance
FIGURE 8: Signal preprocessing. we manually annotate the action duration (leftmost), split CSI samples during action (middle), and upsample the splitted CSI series to be size of 192 (rightmost). (take the 29th subcarrier series when the volunteer plays a ‘circle’ action at #2 position for example.)

FIGURE 9: The basic residual block (left) and a detailed implementation. ‘C1D3×1’ and ‘C1D1×1’ stand for the 1D convolutional operation with the size of 3 × 1 and 1 × 1, respectively. ‘BN1D’ is for 1D Batch Normalization. ReLU is for the Rectified Linear Unit activation function.

these two losses. Before computing \( L_{\text{activity}} \) and \( L_{\text{location}} \), we first normalize the prediction scores with SoftMax function,

\[
s'_i = \frac{e^{s_i}}{\sum_{j=1}^{K} e^{s_j}}, i \in [1, 2, ..., K],
\]

where \( K \) is the categories of activities (\( K = 6 \) for \( L_{\text{activity}} \) and \( K = 16 \) for \( L_{\text{location}} \)), \( s_i \) and \( s'_i \) are the predicted score and normalized score for the i-th activity, respectively. Using (3), all prediction scores are normalized to 0-1 range.

Then we apply the Cross Entropy Loss function on the normalized score to compute \( L_{\text{activity}} \) as follows.

\[
L_{\text{activity}} = -\log(s'_t)
\]

where \( s'_t \) means the normalized prediction score that belongs to (resulted from) t-th activity. With the same approach, indoor localization loss, \( L_{\text{location}} \), can be computed. In our experiment, we assume activity recognition and indoor localization are of the same importance, thus we set \( \lambda \) in (2) as 1 to optimize ResNet1D-[1,1,1,1].

E. IMPLEMENTATION
We implement ResNet1D with Pytorch 1.0.0 in a desktop that is with the Window 7 OS and one Nvidia Titan Xp GPU. The network is trained for 200 epochs by Adam optimizer [43] with default settings (\( \beta_1 = 0.9, \beta_2 = 0.999 \)). The mini-batch size is 128 and the initial learning rate is 0.005. The learning rate decays by 0.5 every 10 epochs. Before each epoch, all training data are shuffled.

V. EVALUATION
A. DATASET
As III-B described, our dataset involves 6 hand activities, i.e., hand up, hand down, hand left, hand right, hand circle, and stand still. The in-door environment is a typical home model with walls, doors, and windows. The volunteer is equipped with a mobile phone to collect CSI samples.

Input: CSI Series

C1D7x1, 128
MaxPool1D3x1
RB1
C1D3x1, 128
C1D3x1, 512
C1D3x1, 128
C1D3x1, 256
C1D3x1, 256
C1D3x1, 512
C1D3x1, 512

AvgPool1D4x1
AvgPool1D4x1
FC, 6
FC, 16

Output: activity Output: location

FIGURE 10: Deep framework. ‘RB’ is the abbreviation of residual block. For all RBs having one residual shortcut, we term this framework as ResNet1D-[1,1,1,1].

FIGURE 11: Deep framework. ‘RB’ is the abbreviation of residual block. For all RBs having one residual shortcut, we term this framework as ResNet1D-[1,1,1,1].
Loss leading to a final dataset with 1394 samples. We select one activity is repeated for 15 times. Thus we collect totally 16 × 6 × 15 = 1440 samples. In hand cross, conducted at 16 locations. At each location, each activity is repeated for 15 times. Thus we collect totally 16 × 6 × 15 = 1440 samples. In [IV-B] we manually discard samples with extremely late start point to ensure data quality, leading to a final dataset with 1394 samples. We select one out every five samples to build the test set (278), and leave the remaining 1116 samples for the training set. We then build up our model with the training set and validate the model on the test set.

B. LEARNING CURVES

We display learning curves of loss and accuracy for the activity recognition and indoor localization in FIGURE [11]. The loss curve of activity recognition (1st subfigure), the training loss (blue line) decreases gradually, and reaches a relatively low state around the 50th epoch. Whereas the test loss curve (red line) wildly swings within the first 45 epochs, and gradually reaches to a steady state around the 75th epoch. A phenomenon needs to be addressed is that though the training loss curve keeps relatively steady after the 50th epoch, the test loss still decreases when more training epochs are involved. We ascribe it to the process of shuffling training dataset before each training epoch, described in [IV-E]. The shuffle process makes the network, i.e., ResNet1D-[1,1,1,1], be optimized with different mini-batch samples in each epoch. After training loss curve reaching a steady condition, the shuffling process continuously generates (keeps generating) more mini-batch combinations and these combinations are continuously updating the network.

The accuracy curves of activity recognition are plotted in the 2nd sub-figure of FIGURE [11], where the training curve (blue line) reaches a steady condition around the 70th epoch, and the test curve reaches a steady condition around the 100th epoch. Similarly, we ascribe it to shuffling process as explained above. Besides, the learning curves of indoor localization are plotted in the 3rd and 4th sub-figures of FIGURE [11]. Comparing the learning curves between the activity recognition and indoor localization, we find that the task of indoor localization converges faster and achieves a better performance.

C. QUANTITATIVE RESULTS

We demonstrate the quantitative results including confusion matrix, prediction accuracy, precision, recall, and F1 scores in the following section.

The confusion matrix of ResNet1D-[1,1,1,1] on activity recognition and indoor localization are shown in FIGURE [12] and FIGURE [13] respectively. As shown in the two figures, we achieve an accuracy of 88.13% for activity recognition and
95.68% for indoor localization. The gap between two accuracies accords with learning curves. For activity recognition, the majority of mis-predictions happen at recognizing the gesture of hand cross. Precisely, ResNet1D-[1,1,1,1] wrongly predicts 8% of hand cross to hand left, and wrongly predicts 6% of hand cross to hand circle. Meanwhile for indoor localization, a major error is wrongly predicting 15% of #16 location as #4 location. However, ResNet1D-[1,1,1,1] generally works well on both activity recognition and indoor localization.

We further compute the precision, recall, and F1 score from the confusion matrix, and list the results in Table 1 and Table 2. There exists a big gap between precision and recall for the activity of hand circle. A precision of 0.97 means ResNet1D-[1,1,1,1] effectively figures out (recognize) the hand circle activity, while a recall of 0.77 indicates that ResNet1D-[1,1,1,1] tends to categorize other activities into hand circle, decreasing the F1 score of hand circle to 0.82. Besides in Table 2, the lowest F1 score is on #4 location prediction, 0.81, due to the low recall. In general, ResNet1D-[1,1,1,1] achieves very promising performances.

### D. DATA VISUALIZATION

We visualize the test set by t-SNE [44] to explore the behaviors of ResNet1D-[1,1,1,1] on the joint task. Taking the activity recognition task as an example (FIGURE 14), we reduce the input into 2 dimensions data by a t-SNE tool and display the 2-d data in the figure. For an input test sample, the reducing procedure is as follows. As [IV-A] said, one original CSI fingerprint $C \in R^{52 \times t}$, where $t$ is 192 after the cutting and linear interpolating preprocess, described in [IV-B]. However t-SNE requires the input to be a long 1D vector, thus we reshape $C$ to be a vector, making a $C' \in R^{1 \times 9984}$ ($192 \times 52 = 9984$). In addition, we repeat the reshaping over all test samples and finally visualize the reshaped samples on the 1st sub-figure in FIGURE 14 marked with the green box. We can see that the inputs are highly disordered in term of activity recognition.

Besides the raw inputs visualization, we also visualize feature maps produced by ResNet1D-[1,1,1,1] in multiple layers (FIGURE 10), i.e., feature maps after max pooling layer, RB1, RB2, RB3, RB4, feature maps before FC (the 7th sub-figure), and feature maps after FC (outputs, the 8th-subfigure). We reshape all feature maps to 1D long vectors with the same approach used for visualizing the raw inputs. FIGURE 14 shows that ResNet1D-[1,1,1,1] gradually increases the discriminative power of feature maps for the activity recognition task step by step, making classification more accurate in the deeper layers of the network. Finally in the outputs (the last sub-figure of FIGURE 14), features are learned to be effective for activity recognition.

With the similar approach, we visualize the raw inputs and feature maps after multiple layers of ResNet1D-[1,1,1,1] for indoor localization in FIGURE 15. It demonstrates that the network can effectively learn features for indoor location. In FIGURE 14 ResNet1D-[1,1,1,1] generates discriminative features after RB4, whereas in FIGURE 15 it generates discriminative features still before FC. Moreover ResNet1D-[1,1,1,1] enables to generate better features for the indoor localization than the activity recognition because the class clusters are more tighter compared the last sub-figure of FIGURE 14 and the last sub-figure of FIGURE 15.

More importantly in the activity recognition, we find the features are largely enhanced through its own branch because the feature before FC (7th) is much better than the features after the shared RB4 (6th). Under this consideration, we just add one more ‘C1D3 × 1, 512’ between the ‘C1D3 × 1, 512’ and the ‘AvgPool1D4×1’, named ResNet1D-[1,1,1,1]+. We train ResNet1D-[1,1,1,1]+, and find it with better performance on activity recognition, listed in Table 3.

### E. EXPANSIBLE STUDY AND BASELINES

ResNet1D is expandable by simply customizing the number of residual block in each RBs, shown in FIGURE 10. Following the default settings in ResNet [18], we evaluate the accuracy of ResNet1D-[2,2,2,2] and ResNet1D-[3,4,6,3]. As listed in Table 4, all ResNet1Ds work well in joint activity recognition and indoor localization. Meanwhile it deserves

---

**TABLE 1:** Precision, recall, and F1 score of the activity recognition task.

| Location | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 | #13 | #14 | #15 | #16 |
|----------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|
| Precision | 0.90 | 0.91 | 0.95 | 0.92 | 0.92 | 0.97 | 0.84 |
| Recall    | 0.98 | 0.89 | 0.83 | 0.92 | 0.77 | 0.89 |
| F1 score  | 0.94 | 0.90 | 0.84 | 0.92 | 0.82 | 0.86 |

**TABLE 2:** Precision, recall, and F1 score of the indoor localization task.

| Model                     | Activity Recognition | Indoor Localization |
|---------------------------|----------------------|---------------------|
| ResNet1D-[1,1,1,1]        | 88.13%               | 95.68%              |
| ResNet1D-[1,1,1,1]+       | 89.57%               | 95.68%              |

**TABLE 3:** Inspired by t-SNE visualization, we propose ResNet1D-[1,1,1,1]+, which outperforms ResNet1D-[1,1,1,1] on activity recognition.

---

[https://lvdmaaten.github.io/tsne]
mentioned that deeper ResNet1Ds tend to work better on indoor location, whereas work worse on activity recognition.

Besides in Table 4, we show the comparison between ResNet1Ds and two baseline methods, Dynamic Time Wrapping (DTW)+kNN [45], [46], and Support Vector Machine (SVM) [47] with radial basis function kernel (RBF). All our proposed ResNet1Ds outperform the baseline DTW+kNN, and SVM-RBF. We also record the time cost of these methods, which are the sum of time cost in training and testing. SVM-RBF costs least but performs worst. DTW+kNN is a very strong baseline in time-serial classification, however it is time-consuming.

| Method         | AR   | IL   | Time Cost |
|----------------|------|------|-----------|
| DTW+kNN        | 83.45% | 95.32% | 3356s     |
| SVM-RBF [47]   | 40.64% | 70.32% | 69s       |
| ResNet1D-[1,1,1] | 88.13% | 95.68% | 161s      |
| ResNet1D-[2,2,2,2] | 87.77% | 96.40% | 240s      |
| ResNet1D-[3,4,6,3] | 85.17% | 97.12% | 412s      |

TABLE 4: Expansible study and comparison with DTW+KNN. Deeper networks work better on indoor localization task, whereas work worse on activity task. All ResNet1Ds outperform the baseline methods, i.e., DTW+kNN and SVM [47]. ‘AR’ and ‘IL’ are abbreviations of activity recognition and indoor localization, respectively.

VI. CONCLUSION

In this paper, we propose a 1D Convolutional Neural Network with two branches for the joint task of activity recognition and indoor localization with WiFi fingerprints.
evaluate the proposed network, we implement IEEE 802.11 In protocol in a software-defined-radio hardware, Etuss N210, collect a dataset mainly for human-interaction applications, and fully discuss the results in various aspects. Experimental results show that our proposed network can achieve joint activity recognition and indoor localization well.

REFERENCES

[1] H. Li, W. Yang, J. Wang, Y. Xu, and L. Huang, “Wifinger: Talk to your smart devices with finger-grained gesture,” in Ubicomp. ACM, 2016, pp. 250–261.
[2] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, “Keystroke recognition using wifi signals,” in MobsCom. ACM, 2015.
[3] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, “Fallfedefi: Ubiquitous fall detection using commodity wi-fi devices,” IMWUT, vol. 1, no. 4, p. 155, 2018.
[4] H. Abdelnasser, M. Youssef, and K. A. Harras, “Wiget: A ubiquitous wifi-based gesture recognition system,” in INFOCOM. IEEE, 2015, pp. 1472–1481.
[5] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “Spotfi: Decimeter level localization using wifi,” in ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, 2015, pp. 269–282.
[6] X. Li, S. Li, D. Zhang, J. Xiong, Y. Wang, and H. Mei, “Dynamic-music: Accurate device-free indoor localization,” in ACM Ubicomp, 2016, pp. 196–207.
[7] M. Kotaru and S. Katti, “Position tracking for virtual reality using commodity wifi,” in IEEE CVPR, 2017.
[8] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity wifi,” in ACM MobiCom, 2015.
[9] Y. Wang, K. Wu, and L. M. Ni, “Wifall: Device-free fall detection by wireless networks,” IEEE Transactions on Mobile Computing, vol. 16, no. 2, pp. 381–394, 2017.
[10] B. Fang, N. D. Lane, M. Zhang, A. Boran, and F. Kawsar, “Bodyscan: Enabling radio-based sensing on wearable devices for contactless activity and vital sign monitoring,” in MobiSys. ACM, 2016, pp. 97–110.
[11] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, “Rt-fall: A real-time and contactless fall detection system with commodity wifi devices,” Transactions on Mobile Computing, vol. 16, no. 2, pp. 511–526, 2017.
[12] X. Wang, C. Yang, and S. Mao, “Tensorbeat: Tensor decomposition for monitoring multi-person breathing beats with commodity wifi,” arXiv preprint arXiv:1702.02046, 2017.
[13] H. Wang, D. Zhang, J. Ma, Y. Wang, D. Wu, T. Gu, and B. Xie, “Human respiration detection with commodity wifi devices: Do user location and body orientation matter?” in Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp). ACM, 2016, pp. 25–36.
[14] Y. Zhao, J. Wang, L. Zhang, A. Bhat, and F. Kawsar, “Bodyscan: Enabling radio-based sensing on wearable devices for contactless activity and vital sign monitoring,” in MobiSys. ACM, 2016, pp. 97–110.
[15] K. Qian, C. Wu, Z. Zhou, Y. Zheng, Z. Yang, and Y. Liu, “Infering motion direction using commodity wi-fi for interactive exergames,” in ACM CHI, 2017.
[16] L. Azzori, A. Iera, and G. Morabito, “The internet of things: A survey,” Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.
[17] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” Future generation computer systems, vol. 29, no. 7, pp. 1645–1660, 2013.
[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in IEEE CVPR, 2016, pp. 770–778.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in NIPS, 2012, pp. 1097–1105.
[20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in ICCV. IEEE, 2017.
[22] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world.” arXiv preprint arXiv:1607.02533, 2016.
[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in IEEE ICCV, 2017.