Location and Person Independent Activity Recognition with WiFi, Deep Neural Networks and Reinforcement Learning

YONGSEN MA, William & Mary, Williamsburg, VA, US
SHEHERYAR ARSHAD, University of Texas at Arlington, Arlington, TX, US
SWETHA MUNIRAJU, ERIC TORKILDSON, ENRICO RANTALA, KLAUS DOPPLER, Nokia Bell Labs, Sunnyvale, CA, US
GANG ZHOU, William & Mary, Williamsburg, VA, US

In recent years, Channel State Information (CSI) measured by WiFi is widely used for human activity recognition. In this paper, we propose a deep learning design for location and person independent activity recognition with WiFi. The proposed design consists of three Deep Neural Networks (DNNs): a 2D Convolutional Neural Network (CNN) as the recognition algorithm, a 1D CNN as the state machine, and a reinforcement learning agent for neural architecture search. The recognition algorithm learns location and person independent features from different perspectives of CSI data. The state machine learns temporal dependency information from history classification results. The reinforcement learning agent optimizes the neural architecture of the recognition algorithm using a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). The proposed design is evaluated in a lab environment with different WiFi device locations, antenna orientations, sitting/standing/walking locations/orientations, and multiple persons. The proposed design has 97% average accuracy when testing devices and persons are not seen during training. The proposed design is also evaluated by two public datasets with accuracy of 80% and 83%. The proposed design needs very little human efforts for ground truth labeling, feature engineering, signal processing, and tuning of learning parameters and hyperparameters.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Hardware** → **Wireless devices**; • **Computing methodologies** → *Neural networks*; *Reinforcement learning*.

Additional Key Words and Phrases: Activity Recognition, WiFi, Wireless Sensing, Channel State Information

ACM Reference Format:
Yongsen Ma, Sheheryar Arshad, Swetha Muniraju, Eric Torkildson, Enrico Rantala, Klaus Doppler, and Gang Zhou. 2020. Location and Person Independent Activity Recognition with WiFi, Deep Neural Networks and Reinforcement Learning. *ACM Trans. Internet Things* 1, 1, Article 1 (January 2020), 24 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In recent years, WiFi signals are widely used for non-intrusive sensing purposes. WiFi-based sensing applications are easy to deploy and have low costs by reusing the infrastructure that is designed for wireless communications. Multiple-Input Multiple-Output (MIMO) and Orthogonal Frequency-Division Multiplexing (OFDM) are two of the most important technologies to provide high performance for modern WiFi systems. MIMO-OFDM provides Channel State Information (CSI) which represents the power attenuation and phase shift from the transmitter to the receiver at certain carrier frequencies. In addition to improving the networking performance of WiFi networks, CSI can also be used for WiFi-based sensing applications since it captures how WiFi signals travel from the transmitter to the receiver through surrounding objects and humans. For example, when a person

Authors' addresses: Y. Ma and G. Zhou, Department of Computer Science, William & Mary, Williamsburg, VA, US; S. Arshad, Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, US; S. Muniraju, E. Torkildson, E. Rantala, and K. Doppler, Nokia Bell Labs, Sunnyvale, CA, US.

is moving or doing different activities around the WiFi transmitter or receiver, the reflected WiFi signals are changed accordingly. The CSI amplitude and phase are also impacted, and these CSI variations can be fed to pre-defined models or machine learning algorithms for human motion detection [2, 11, 13, 15, 18, 20, 22–25, 29, 37, 41, 50, 53, 54, 58] and activity recognition [3, 7–12, 16, 19, 32, 42–49, 52].

For WiFi-based activity recognition to be practical in real-world scenarios, the recognition algorithm should be location and person independent. In the training stage, the recognition algorithm is usually trained in a controlled environment. During testing in real-world deployments, the location and orientation of WiFi devices are usually unknown and testing persons are unseen during training. However, it is challenging for WiFi-based activity recognition to be robust in different scenarios, since WiFi signals are very sensitive to different factors. CSI is impacted by not only human activities but also the static and motion status of WiFi transmitters, receivers, and the surrounding environment. For example, the location and orientation of WiFi receivers and target persons have a great impact on how CSI amplitude and phase change. When a recognition algorithm is trained or modeled by CSI measurements from a certain WiFi receiver, it is challenging to make the algorithm still work for another WiFi receiver placed at a different location with different antenna orientations. Moreover, different persons may have different motion and activity patterns, so models trained on one person may not work for another person whose data are not seen during training or modeling.

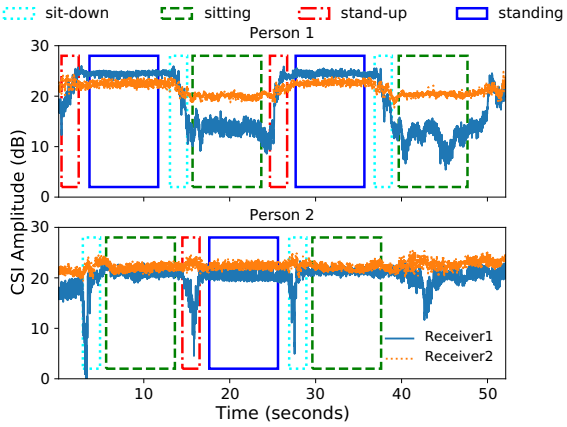


Fig. 1. CSI amplitude of TX0/RX0 of the 1st subcarrier from two WiFi devices for different activities performed by two persons. Different persons or receivers have different CSI patterns. It is challenging to distinguish different activities if the recognition algorithm is trained or modeled with CSI data of receiver 1 for person 1 and tested with CSI data of receiver 2 for person 2.

As shown in Fig. 1, CSI patterns of the same activity are very different for different persons or different receiver locations/orientations. Modeling-based and instance-based learning algorithms do not work if they are tested with unseen persons or unknown receiver locations/orientations. For example, when person 1 changes the status from standing to sitting, the CSI amplitude of receiver 1 decreases from ~25dB to ~10dB. But for receiver 2, the CSI amplitude changes from ~22dB to ~20dB. Moreover, there are no big differences for different activities for receiver 2 of person 2. Therefore, traditional recognition algorithms can hardly work if they are trained or modeled with CSI data of receiver 1 for person 1 and tested with CSI data of receiver 2 for person 2.

WiFi-based activity recognition can reuse Convolutional Neural Networks (CNNs) that have high performance for computer vision tasks. But reusing existing CNNs may result in low performance for WiFi-based activity recognition, since CSI has some unique characteristics that are different from images. CSI has much smaller spatial resolutions than images and contains noises and interferences from all directions. CSI amplitude and phase

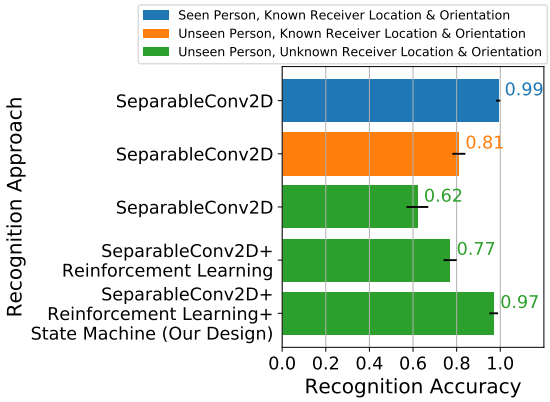


Fig. 2. Accuracy comparison of different deep learning solutions for WiFi-based activity recognition. The recognition accuracy of SeparableConv2D drops dramatically for unseen persons or unknown receiver locations & orientations. The accuracy is significantly improved by the proposed design with reinforcement learning and state machine.

are very sensitive to the surrounding environment and the location and orientation of WiFi receivers and target persons. Therefore, pre-trained CNNs have low accuracy for unseen persons or unknown receiver locations and orientations. As shown in Fig. 2, depthwise separable 2D convolutions, or SeparableConv2D, achieves 99% accuracy when the data of testing persons and receivers are seen during training. But the accuracy drops to 84% for unseen persons and 62% for unseen persons and unknown receiver locations and orientations. Therefore, it is necessary to find the suitable CNN types, neural architectures and learning parameters that are specially designed for CSI data.

In this paper, we propose a novel deep learning solution for robust WiFi-based activity recognition. The proposed design contains three neural networks: a 2D CNN as the recognition algorithm, a 1D CNN as the state machine, and a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) as the reinforcement learning agent for neural architecture search. In summary, the proposed design has the following three components.

Recognition Algorithm: 2D CNN. To learn location and person independent features from different perspectives of 4D CSI tensors in time, spatial, and frequency domains.

State Machine: 1D CNN. To learn temporal dependency information from previous classification results for improving the recognition performance of static and transition activities.

Neural Architecture Search: RNN with LSTM. To optimize the neural architecture of the recognition algorithm by reinforcement learning.

The combination of these three deep learning components provides location and person independent WiFi-based activity recognition with the following properties.

Robust: It is independent of the locations, placements, and orientations of WiFi devices and target persons. The pre-trained model also works when WiFi receivers are placed at unknown places with uncertain orientations and antenna placements and for new persons whose data are not seen during training.

Automatic: It requires very little human efforts for data collection, ground truth labeling, and training. It only needs simple CSI pre-processing and does not require manual efforts for ground truth labeling, feature engineering, signal processing, learning parameters tuning, or neural architecture search.

Reusable and adaptable: It can be trained on new data and pre-trained models without restarting the training process. It can evolve over time as there are more data measured in new scenarios with different settings.

The proposed design is evaluated by CSI measurements from real-world scenarios. There are 14555 instances of 5 activities, including sitting, standing, sit-down, stand-up, and walking, performed by 7 persons. There are 4 WiFi receivers placed at different locations with different antenna orientations. Each participant can sit or stand at two locations with random facing directions and walk randomly in a constrained area. As shown in Fig. 2, with reinforcement learning, the recognition accuracy is improved from 62% to 77% for unknown receiver locations and orientations and for unseen persons. The accuracy is further improved to 97% by adding both reinforcement learning and state machine. The proposed design is also evaluated by two public datasets, S.Yousefi-2017 [51] and FallDeFi [29], with accuracy of 80% and 83%. In summary, we make the following contributions.

- We propose a novel deep learning solution with a combination of separable 2D convolutional neural networks, state machine, and reinforcement learning for robust WiFi-based activity recognition.
- The propose design recognizes 5 activities with 97% average accuracy when the location and orientation of WiFi receivers and target persons are unknown and the data of target persons are not seen during training.
- The propose design needs very little human efforts for ground truth labeling, signal processing, feature engineering, and model tuning. It can be re-trained on new data to evolve over time and be adaptive to different scenarios.

The rest of the paper is organized as follows. Section 2 presents the proposed design. Section 3 shows the experiment setup and evaluation results. Section 4 presents some discussions on overhead of neural architecture search and robustness of new environments. Section 5 presents related works, and Section 6 concludes the paper.

2 DESIGN

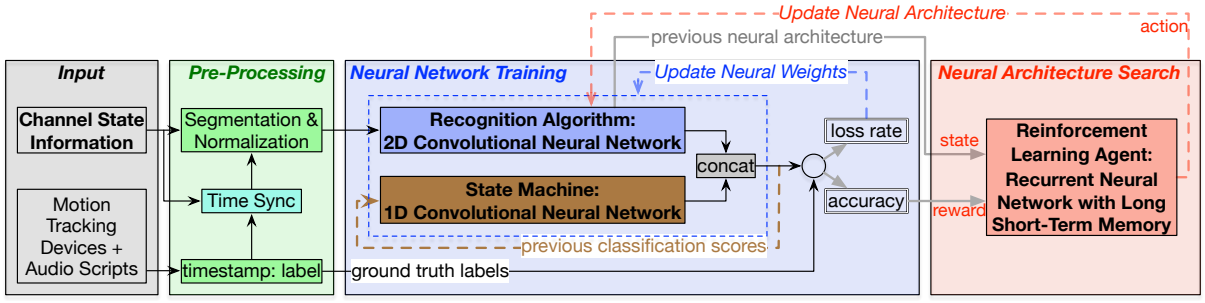


Fig. 3. The training process of the proposed design. The input is a time series of CSI matrices measured by WiFi packets. The recognition algorithm is a 2D CNN. The state machine is a 1D CNN. The final classification results are calculated by the concatenation of the recognition algorithm and the state machine. The neural architecture of the recognition algorithm is updated by the reinforcement learning agent by an RNN with LSTM. Motion tracking devices and audio signals are used for ground truth labeling during off-line training. During the inference stage, only CSIs are used as the input.

The overview of the proposed design, including pre-processing, recognition algorithm, state machine, and neural architecture search is shown in Fig. 3. During data collection, each participant follows the audio scripts to perform different activities. At the same time, CSI measurements are collected by WiFi receivers. A motion tracking system, i.e., HTC Vive, along with the audio scripts, are used to label the CSI data. Note that motion tracking devices and audio scripts are used only for off-line training. During the inference stage, only CSI measurements are used. The proposed design has the following components. First, a time series of CSI matrices

is synchronized and segmented by ground truth labels and then normalized by the training data. Second, the recognition algorithm uses a 2D or 3D CNN to learn features from different perspectives of 4D CSI tensors. Third, the state machine learns temporal dependency information from previous classification results by a 1D CNN. Finally, the neural architecture of the recognition algorithm is optimized by a reinforcement learning agent, i.e., an RNN with LSTM.

The combination of these components provides location and person independent WiFi-based activity recognition. The recognition algorithm is responsible for learning location and person independent features within one CSI segment in time, spatial, and frequency domains. The state machine tries to learn temporal dependencies across multiple CSI segments. The reinforcement learning agent optimizes the neural architecture of the recognition algorithm to maximize the accuracy. As a result, the proposed design is robust in new scenarios when the locations and orientations of WiFi devices and target persons are unknown and for new target persons whose data are unseen during training. It requires very little human efforts for ground truth labeling, signal processing, feature engineering, parameter tuning, and neural architecture search.

2.1 Pre-Processing: CSI Normalization

CSI represents how wireless signals travel from the transmitter to the receiver at certain carrier frequencies along multiple paths. For a MIMO-OFDM channel with N_t transmit antennas, N_r receive antennas, and N_c subcarriers, the CSI is a 3D matrix $H \in \mathbb{C}^{N_r \times N_t \times N_c}$. Each CSI entry is a complex number representing the Channel Frequency Response (CFR) of the multi-path channel:

$$h(f;t) = \sum_{i=1}^{N} a_i(t) e^{-j2\pi f \tau_i(t)}, \tag{1}$$

where $a_i(t)$ and $\tau_i(t)$ are the power attenuation and propagation delay, respectively, of the i-th path, f is the carrier frequency, and N is the number of multi-path components [38]. The CSI amplitude and phase represent the power attenuation and phase shift of the multi-path channel, which are impacted by the combined effects of Doppler frequency shift, multi-path channel propagation, and the static/mobility status of the transmitter, receiver, and nearby humans/objects. The multi-path profile is mainly impacted by the surrounding environment and the relative position of transmitter, receiver, and target persons. So the measured CSIs are different for when the person is sitting and standing. When there are moving persons or objects nearby, it also introduces time variations due to Doppler frequency shift and channel propagation variations. CSI phase is too sensitive to very small environmental changes, so we only use CSI amplitude as the input.

CSI measurements are collected from multiple WiFi receivers every 10 milliseconds and are synchronized with the timestamps of audio scripts. During off-line training, raw CSI measurements are segmented based on the corresponding ground truth labels from the audio scripts. Each CSI segment has time duration of 2 seconds containing 200 samples of CSI matrices. Shorter CSI segments are discarded. There are 3 transmit antennas, 3 receive antennas, and 30 subcarriers. Each training and testing sample is a time series of 3D CSI matrices, resulting in a 4D CSI tensor with size of $(200, 3, 3, 30)$. Each training and testing CSI segment is normalized by:

$$x^i_{train} = \frac{|csi^i_{train}| - mean(|csi_{train}|)}{std(|csi_{train}|)},\ i = 1, \cdots, N_{train}$$

$$x^j_{test} = \frac{|csi^j_{test}| - mean(|csi_{train}|)}{std(|csi_{train}|)},\ j = 1, \cdots, N_{test},$$

where $mean(|csi_{train}|)$ and $std(|csi_{train}|)$ are the mean and standard deviation of the CSI amplitude of training samples. Each dimension of the 4D CSI tensor is normalized separately. Note that $mean(|csi_{train}|)$ and $std(|csi_{train}|)$ do not include any testing CSI samples, so the information of testing CSI samples is not leaked to the normalized training data x^i_{train}.

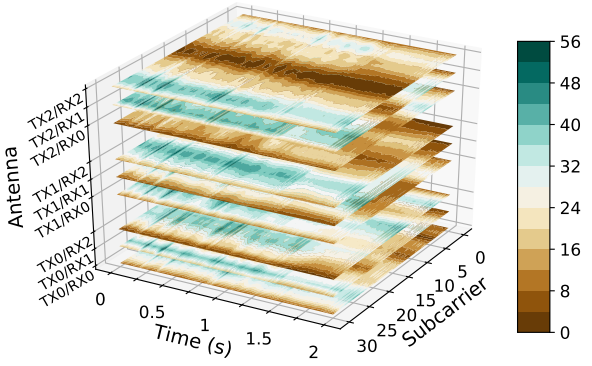


Fig. 4. A time series of CSI amplitude measurements provides information in time, spatial, and frequency domains. The CSI tensors are fed to different CNNs to automatically find location and person independent features.

We use as little pre-processing as possible to retain as much raw information as possible. CSI normalization is the only pre-processing before feeding the input to the recognition algorithm. We leverage the power of DNNs and reinforcement learning for extracting useful information from the normalized CSI input. When there are more pre-processing involved, there will be a higher probability of losing information embodied in the unprocessed data. Moreover, the CSI pre-processing has very low computation overhead, so it runs fast for both off-line training and real-time inference. Fig. 4 shows an example of the pre-processed CSI segment. It contains CSI amplitude variations in the time, spatial, and frequency domains. The pre-processed 4D CSI tensors are fed to different CNNs to learn useful features from different perspectives of the CSI data.

2.2 Recognition Algorithm: 2D/3D CNN

CSI matrices have some similar attributes as digital images. For a MIMO-OFDM channel with N_t transmit antennas, N_r receive antennas, and N_s subcarriers, the CSI matrix is similar to a digital image with spatial resolution of $N_t \times N_r$ and N_s color channels. Convolutional Neural Networks (CNNs) are proven to have very good performance and are used in almost all modern neural network architectures. Therefore, WiFi-based activity recognition can reuse the CNN models and architectures that have high performance for computer vision tasks.

However, CSI has some unique characteristics that are different from images and videos, so reusing existing CNNs may result in low performance for WiFi-based activity recognition. For example, the spatial resolution, which is 3×3 in our case, is much smaller than that of images. A digital image usually have 3 (RGB) or 1 (grayscale) color channels, while an uncompressed CSI matrix has 52 data subcarriers for a 20MHz WiFi channel. Besides, unlike images and videos that usually contain light signals from visible directions and distances, CSI may contain noises and interferences from all directions. So CSI amplitude and phase are very sensitive to the surrounding environment and the location and orientation of WiFi receivers and target persons. Therefore, we need to find which types of DNNs are suitable for CSI data.

A time series of CSI matrices characterizes MIMO channel variations in time, frequency, and spatial domains, as shown in Fig. 4. CSI can be processed, modeled, and trained in different domains for different WiFi sensing applications. Different CNNs can infer information from their specific aspects of the training data. Our task is to find the best type of CNNs and the corresponding neural architecture that provide robust WiFi-based activity recognition for unknown receiver locations/orientations and unseen persons. We consider the following three convolutions types: 2D convolution, 3D convolution, and depthwise separable 2D convolution.

2.2.1 2D Convolutions (Conv2D). Conv2D is calculated by

$$S(i,j) = (I * K)(i,j) = \sum_x \sum_y I(x,y)K(i-x, j-y), \tag{2}$$

where S is the convolution output, I is the input, and K is the kernel [14]. The convolutional layer learns the weights and biases while going through the input vertically and horizontally with the same kernel. For Conv2D, the same kernel is shared for different color channels.

2.2.2 3D Convolutions (Conv3D). Conv3D uses 3D kernels, instead of 2D kernels in Conv2D, as going through cubic regions of the input data. Conv3D is calculated by

$$S(i,j,d) = \sum_x \sum_y \sum_z I(x,y,z)K(i-x, j-y, d-z). \tag{3}$$

Conv3D learns features combined in all three domains: time, spatial, and frequency. Besides, CSI tensor reshaping is not needed for Conv3D, while it is necessary for Conv2D. One potential issue for using Conv3D for CSI data is that CSI matrices have too small spatial resolutions, i.e., 3×3 in our case. To address this issue, we reshape the CSI tensors from $\mathbb{R}^{N_s \times N_r \times N_t \times N_c}$ to $\mathbb{R}^{N_s \times N_c \times N_t \times N_r}$, with N_s as the number of CSI samples for each segment.

2.2.3 Depthwise Separable 2D Convolutions (SeparableConv2D). SeparableConv2D first uses different kernels (depthwise convolutions) for each color channel and then uses a 1×1 kernel (pointwise convolutions) along the input depth to get the combined features. SeparableConv2D is calculated by:

$$S(i,j,d) = \sum_d D(i,j,d)K_p(k-d) \text{ with } D(i,j,d) = \sum_x \sum_y I(x,y)K_d(i-x, j-y), \tag{4}$$

where $D(i,j,d)$ is the output of the first step, K_d is the kernel of the d-th color channel, K_p is a 1×1 pointwise convolution kernel, and $S(i,j,d)$ is the convolution output of the second step [4, 6]. For computer vision tasks, SeparableConv2D has 10 times less computation cost with a small reduction of accuracy compared with normal convolutions [4, 6]. For CSI data, SeparableConv2D has the best recognition accuracy, as shown in Section 3.

DNNs are organized into multiple layers, and the convolutional layer is only one of the layers. Each convolutional layer is usually followed by other layers including batch normalization, Rectified Linear Unit (ReLU), max-pooling, and dropout layers. Before the output layer, a flatten layer and a full-connected layer with softmax are needed to calculate the loss rate of the classification algorithm. CNNs learn the training parameters of each layer, using an optimization algorithm, to minimize the loss rate. Since the convolution layer shares the same kernel for multiple input regions, it significantly reduces computation overhead for both training and inference.

2.3 State Machine: 1D CNN

There is temporal dependency information within a single CSI segment and across multiple CSI segments. Each CSI segment is a 4D tensor containing 200 samples of CSI matrices measured in 2 seconds. The recognition algorithm does not learn the temporal dependencies among neighboring CSI segments. For example, if the classification result of the current CSI segment is stand-up, the next CSI segment has a high probability to be standing. Therefore, we add a state machine to learn the temporal dependencies across CSI segments. Fig. 5 shows the state transition diagram of 5 human activities. Each state is in corresponding to a 4D CSI tensor with size of (200, 3, 3, 30). The state machine is used to learn the state transition probabilities and to predict the current state based on previous states. The final classification results are obtained by the concatenation of the output of the state machine and the classification scores of the recognition algorithm.

The state machine can be modeled by a Markov chain which represents a time series of possible activities. The probability of each activity depends on the state of the previous activity. Hidden Markov Model (HMM) is a widely used Markov model wherein the states are modeled by a Markov process with unobservable states, i.e., hidden states. HMM has a strong assumption that state transitions only depend on the current state which

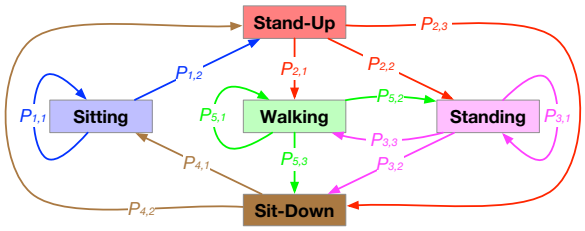


Fig. 5. State machine of 5 activities.

can be modeled by a linear transformation of the previous state. This assumption does not hold for CSI data wherein state transitions have non-linear relationships with the current and previous states. Besides, HMM needs parameter learning to find the best set of state transitions and emission probabilities. The learned parameters of HMM are highly dependent on the training data. When the state transitions of training and testing data have different distributions, the learned HMM will overfit to training data and give low accuracy for testing data. Moreover, the HMM needs to be trained separately in addition to the training of the recognition algorithm.

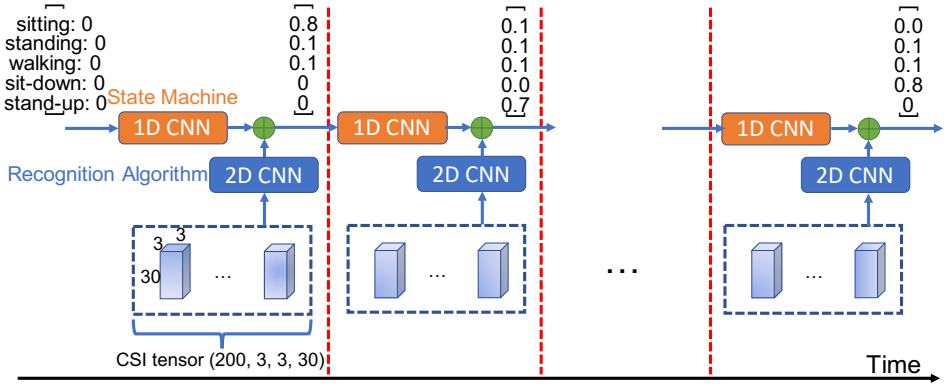
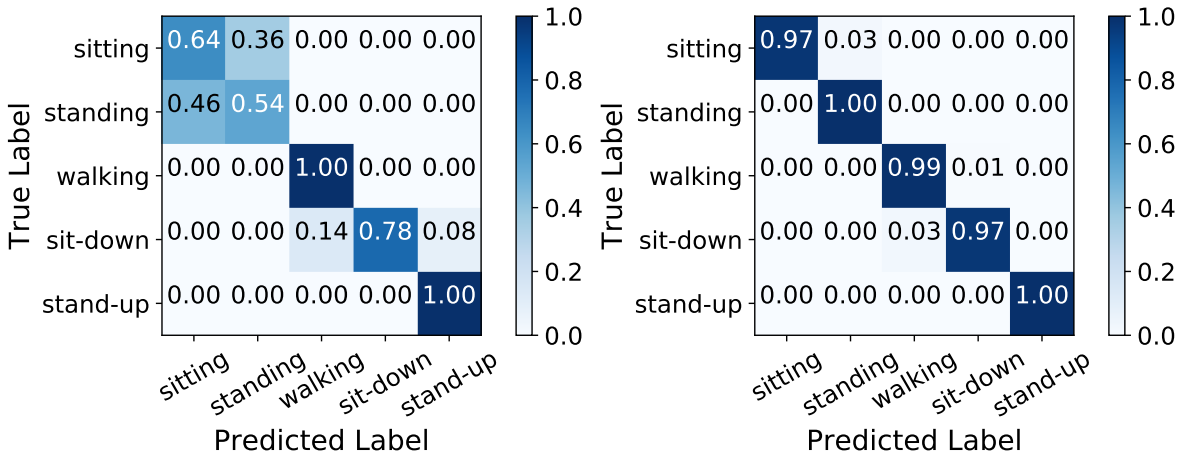


Fig. 6. Recognition algorithm and state machine.

Recently, DNNs, especially RNNs, are popular replacements of HMM for sequential problems. RNNs are proven to have good performance for complex sequential inputs that involve non-linearities and long-term temporal dependencies, which are hard to handle for HMM. DNNs do not have the Markov assumption. Instead, DNNs rely on the learning parameters, or neural weights, to extract complex features and state transitions. However, RNNs have have extremely high computation costs. Besides, we run experiments and find that RNNs have much lower accuracy than CNNs for CSI data. The major reason is the low spatial resolution of a single CSI matrix. It is hard for RNNs to capture short-term temporal dependencies within a CSI segment. To address this issue, we use a 1D CNN, i.e., Conv1D, as the state machine. 1D CNNs run much faster than RNNs and offer comparable or higher performance for CSI data. It has only one Conv1D layer with 5 kernels of size 1×1. The state machine is very lightweight with only 140 parameters, including 10 from the Conv1D layer and 130 from the softmax layer. It offers 20% accuracy improvements with very low computation and training costs, which is shown in Section 3. Moreover, the state machine is trained together with the recognition algorithm, so it does not require extra efforts to train the recognition algorithm and state machine separately. Fig. 6 shows how the state machine

and recognition algorithm work together for activity recognition using CSI tensors. The input of the recognition algorithm is a 4D CSI tensor with size of (200, 3, 3, 30). The input of the state machine is a vector representing the classification scores of different activities. For the first CSI tensor that has no previous classification scores, the input of the state machine is a vector of zeros. The final classification result is the concatenate of the recognition algorithm and the state machine, which is used as the input of the state machine for the next CSI tensor.



(a) SeparableConv2D with reinforcement learning, without state machine. Recognition accuracy = 78%.

(b) SeparableConv2D with reinforcement learning and state machine. Recognition accuracy = 99%.

Fig. 7. Confusion matrix of leave-one-person-out and leave-one-device-out testing results. The recognition performance for static activities, i.e., sitting and standing, is significantly improved by the state machine.

Fig. 7 shows the confusion matrices of SeparableConv2D with and without the state machine. Both confusion matrices are obtained from leave-one-person-out and leave-one-device-out validation for person 2. Without the state machine, the recognition algorithm has difficulties in distinguishing sitting and standing from each other. The reason is that CSI measurements have similar patterns for static activities. This issue is addressed by the state machine, which improves the overall recognition accuracy from 77% to 97%. As shown in Fig. 7b, the recognition performance of sitting and standing has significant improvements. This is accomplished by utilizing the high accuracy of motion transition activities and the temporal dependencies learned by the state machine. More details of the impact of the state machine are shown in Section 3.

2.4 Neural Architecture Search: Reinforcement Learning

Although the features, or learning weights, of CNNs can be automatically learned during training, it is non-trivial to find the best neural architecture, especially for CSI data. The neural architecture of a CNN refers to a set of hyperparameters such as the number of convolutional layers, number of convolutional kernels, size of convolutional kernels, size of max-pooling, and dropout rate. One way is to reuse the neural architectures that are proven to provide high performance for computer vision and natural language tasks. But these neural architectures do not necessarily give good performance for WiFi-based activity recognition, since CSI data are different from images, videos, and texts. Another approach is using neural architecture search which tries to optimize the CNN architecture for improving the classification performance.

We use a reinforcement learning agent, NASCell [59] from TensorFlow, for neural architecture search. It needs almost no human efforts for hyperparameters tuning. Reinforcement learning tries to maximize a numerical reward signal by learning how to interact with the environment in discrete time steps [35]. In the context of neural architecture search, the environment is the recognition algorithm which updates the state and reward to the agent. The action signal is the neural architecture of the recognition algorithm, and the reward signal is the classification accuracy. NASCell uses an RNN with LSTM to update the neural architecture of the recognition algorithm. For each training cycle, the training results and neural architecture of the recognition algorithm are fed to NASCell to find the next action output. NASCell updates the neural architecture to maximize the expected reward, which is done by the policy gradient of the empirical approximation of the expected reward

$$\nabla_{\theta_c} J(\theta_c) \approx \frac{1}{m} \sum_{k=1}^{m} \sum_{t=1}^{T} \nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c)(R_k - b), \tag{5}$$

where $\nabla_{\theta_c} J(\theta_c)$ is the policy gradient of the expected reward $J(\theta_c)$ with learning parameters θ_c, m is the number of neural architectures in one batch, T is the number of hyperparameters, a_t is the list of actions, R_k is the training accuracy of the k-th neural neural architecture, and b is the average training accuracy of previous neural architectures for preventing high variances [59]. The action output of NASCell is mapped to the neural architecture of the recognition algorithm to start the next training cycle.

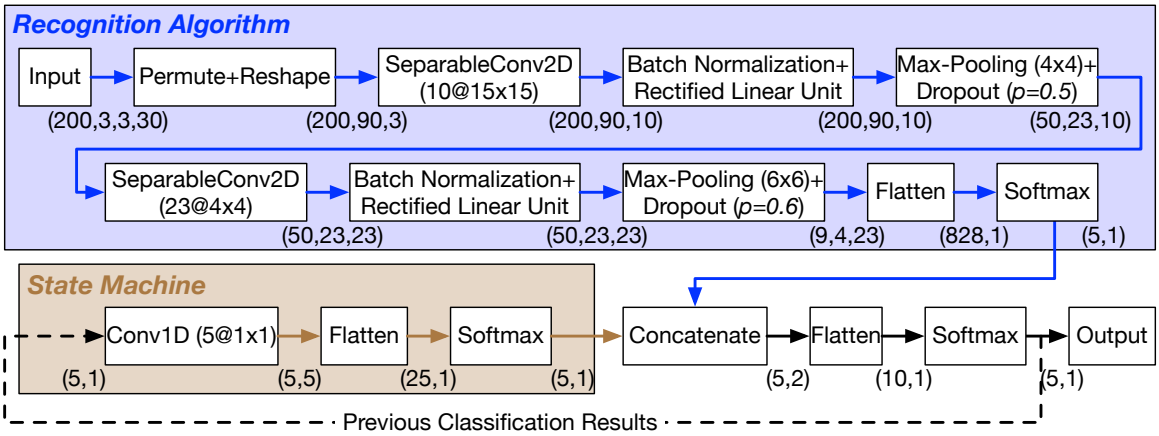


Fig. 8. Neural architecture of the best performing recognition algorithm. The input is pre-processed 4D CSI tensors, and the output is one of the five activities. The numbers inside the brackets indicate the output size of each layer.

Fig. 8 shows the best performing neural architecture of the recognition algorithm and state machine optimized by the reinforcement learning agent. Note that the neural architecture of the state machine is fixed and is not updated by the reinforcement learning agent. The recognition algorithm has two SeparableConv2D layers with each followed by batch normalization, ReLU, max-pooling, and dropout layers. The input of the recognition algorithm is 4D CSI tensors of size $(200, 3, 3, 30)$, and the output is the classification scores of 5 activities. The state machine contains a Conv1D layer with 5 kernels of size 1×1. The input of the state machine is the classification scores of the previous CSI segment, and the output is the classification scores of 5 activities. The final classification output is calculated by the concatenation of the outputs of the recognition algorithm and state machine.

Table 1. Overview of Performance Evaluation for Different Testing Scenarios

	Dataset	Size (GB)[1]	Testing Scenario	# (Rooms, Persons, RXs, TXs)	# Instances	Size of Each Instance	Testing Accuracy (CNN, CNN+RL, CNN+RL+SM)
§3.1	This paper	17.05	Same environment; unseen persons; unknown receiver location/orientation	(1, 7, 4, 1)	14555	(200, 3, 3, 30)	62%, 77%, 97%
§3.2	S.Yousefi-2017 [51]	2.78	Same environment; unseen persons; known receiver location/orientation	(1, 6, 1, 1)	2079	(2000, 3, 30)	45%, 63%, 80%
	FallDeFi [29]	1.06	Unseen environment; unseen persons; unknown receiver location/orientation	(6, 5, 5, 5)	397	(2000, 3, 30)	51%, 64%, 83%

[1] Size of numpy array of CSIs of 5 activities. Other activities of S.Yousefi-2017 [51] and FallDeFi [29] are not included.

3 EVALUATION

An overview of performance evaluation of different testing scenarios is shown in Table 1. The proposed design is evaluated by leave-one-person-out and leave-one-device-out tests with CSI measurements of 5 activities performed by 7 persons with different receiver locations and orientations in Section 3.1. We also evaluate the proposed design with other public datasets including FallDeFi [29] and S.Yousefi-2017 [51] in Section 3.2. Note that the proposed design is evaluated by unknown persons, unseen environments, and unknown receiver location/orientation, which is harder than the evaluations in [29] and [51] . Moreover, the output of FallDeFi [29] is binary classification (fall or not), which is less challenging than our evaluations of 5 mobile and static activities.

3.1 Evaluation Results of Unseen Persons and Unknown Receiver Locations/Orientations

This section presents evaluation results of the impact of convolutions, state machine, and reinforcement learning for unseen persons and unknown receiver locations/orientations.

3.1.1 Experiment Setup and Data Collection. The experiment setup is shown in Fig. 9. There are 4 WiFi receivers placed at different locations with different antenna orientations. CSI measurements of 3 receivers are used for off-line training and the other receiver is used for testing. There are 5 activities, sitting, standing, sit-down, stand-up, and walking, performed by 7 persons. The 7 participants have a wide variety of heights (from 64 to 75 inches) and weights (from 160 to 210 pounds), as shown in Table 2. In total, there are 14555 CSI segments, each with size of $(200, 3, 3, 30)$, measured from 4 WiFi receivers. WiFi receivers are placed at different places with different heights and antenna orientations. During data collection, each person follows the audio instructions to perform different activities. Each person can walk randomly in the walking area and sit/stand at two locations with different facing directions. All activities are performed normally as in the real-life. For example, the person can have minor motions, such as interacting with the smartphone, as sitting, standing, and walking. CSI measurements are collected at different dates.

Table 2. Height and Weight of Participants in the Experiments

Height (inches)	67	75	64	70	75	66	66
Weight (pounds)	190	200	160	180	185	195	210

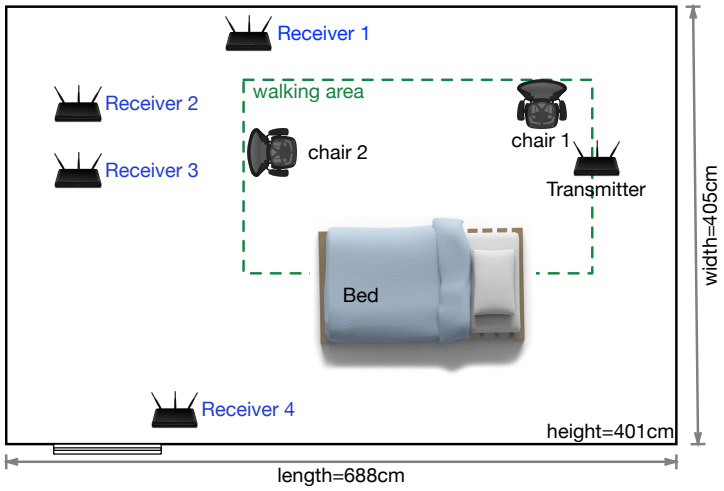


Fig. 9. Experiment setup. There are 4 WiFi receivers placed at different locations with different heights and antenna orientations. Each participant can walk randomly within the walking area, sit or stand at two chair locations with random facing directions. CSI measurements of receiver 1 are for testing and other receivers for training in Section 3.1.2 to 3.1.3. Other receivers are for leave-one-receiver-out testing in Section 3.1.6.

There are 4 WiFi receivers collecting CSI measurements as the participant is performing different activities following the audio instructions. Before each round of CSI measurements, the timestamps of audio instructions and the 802.11n CSI tool of multiple devices are synchronized by the Network Time Protocol (NTP) with millisecond-level accuracy. The motion tracking devices and audio instructions are used later for ground truth labeling and segmentation of CSI measurements. Each WiFi device is a HummingBoard Edge [34] with an Intel 5300 WiFi card installed. The 802.11n CSI tool [17] is used for sending WiFi packets and measuring CSIs every 10 milliseconds. There are three antennas for each WiFi device, and the antenna spacing of each WiFi device is 2.6cm.

Raw CSI measurements are fed to Python scripts for extracting CSI matrices and pre-processing including segmentation and normalization. Both CNNs and NASCell use the Adam optimizer with learning rate of 0.01 and 0.001, respectively. We first use different hyperparameters including learning rate and epochs and different optimizers using a small subset of the dataset. We started from the hyperparameters that are commonly used for computer vision models. We found that the Adam optimizer with learning rates of 0.01 and 0.001 gives the best results for the small dataset, and hence we use these parameters for the larger dataset. The off-line network training is performed on a GTX 1080 Ti GPU. The performance is evaluated with leave-one-person-out tests, i.e., the data of testing persons are not seen during training, and leave-one-device-out tests, i.e., the location and orientation of testing receivers are unknown. The following performance results are evaluated with both leave-one-person-out and leave-one-device-out tests, i.e., neither testing devices nor persons are seen during training, unless stated otherwise. Performance metrics include accuracy, recall, precision, and F1-measure scores [33].

3.1.2 Impact of Different Convolutions. Fig. 10 shows the performance results of Conv2D, Conv3D, and Separable-Conv2D with and without state machine. Both leave-one-person-out and leave-one-device-out validation are used, i.e., CSI samples of the testing persons and testing devices are not seen during training. SeparableConv2D provides the best recognition performance. First, the average score, including accuracy, precision, recall, and F1-measure, of SeparableConv2D is the highest for both with and without the state machine. This means SeparableConv2D gives the most accurate recognition results. Second, the stand deviation of SeparableConv2D is the smallest,

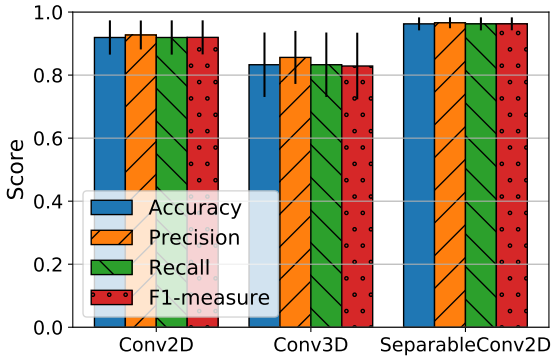


Fig. 10. Average testing results of different convolution types for leave-one-person-out and leave-one-device-out validation. SeparableConv2D has the highest average score and the smallest standard deviation of recognition performance.

which means it gives consistent recognition results for different persons. Conv2D shares the same kernel for different color channels, so it does not learn features in the depth axis. Conv3D uses a 3D kernel to go through the CSI data and learn features in time, spatial, and frequency domains. But the CSI tensor has very small spatial resolution, i.e., 3×3, so it does not help much learning spatial features. Conv3D learns features of different domains simultaneously using one shared kernel, while SeparableConv2D uses different kernels for different color channels and learns features in the depth axis separately in different kernels. So SeparableConv2D has the best recognition performance for CSI-based activity recognition.

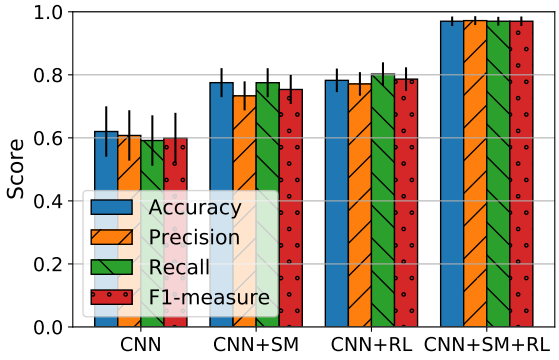
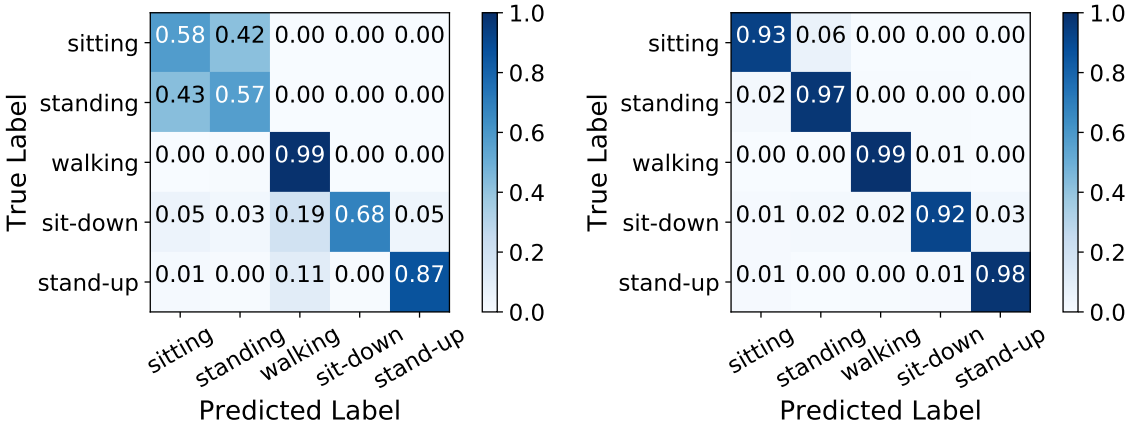


Fig. 11. Average testing results of SeparableConv2D for leave-one-person-out and leave-one-device-out testing. CNN+SM+RL has the highest average score and the smallest standard deviation of recognition performance.

3.1.3 Impact of State Machine and Reinforcement Learning. Fig. 11 shows performance results of SeparableConv2D with or without state machine and reinforcement learning for leave-one-person-out and leave-one-device-out testing. By using state machine and reinforcement learning together, the average accuracy of SeparableConv2D is improved from 62% to 97%. State machine alone, i.e., the neural architecture is not optimized by reinforcement learning, provides about 15% improvement compared to SeparableConv2D without state machine. If only

reinforcement learning is used without state machine, the performance improvement is also around 15%. Reinforcement learning and state machine improves each other when they are used at the same time, which provides 35% higher accuracy than the baseline SeparableConv2D model architecture without state machine.

3.1.4 Impact of State Machine. The average recognition accuracy of SeparableConv2D is 77% without state machine and 97% with state machine. The major contribution of state machine is on improving the recognition performance for static activities, i.e., sitting and standing, by taking advantage of the temporal dependencies of multiple CSI segments. Motion activities have different CSI patterns, so they have relatively high recognition accuracy even without state machine. The state machine learns temporal dependencies of neighboring CSI segments and utilizes the high accuracy of motion activities to improve the accuracy of static activities. The performance of transition activities that are misclassified as walking is also improved.



(a) SeparableConv2D with reinforcement learning, without state machine. Average recognition accuracy = 77%.

(b) SeparableConv2D with reinforcement learning and state machine. Average recognition accuracy = 97%.

Fig. 12. Confusion matrix of leave-one-person-out and leave-one-device-out testing results for all persons. The recognition performance for static activities, i.e., sitting and standing, is significantly improved by the state machine.

Fig. 12 shows the confusion matrices of the best performing SeparableConv2D architecture without the state machine. First, walking has close to 100% accuracy for all testing persons. Second, sitting and standing have very low recognition accuracy for all testing persons. Sitting and standing are very easy to be misclassified with each other because they have similar CSI patterns. The only difference between them is the heights of siting and standing. Some persons have no big differences between siting and standing heights. Sitting and standing are static activities, so almost none of them are misclassified as motion activities. Finally, transition activities, i.e., sit-down and stand-up, have higher accuracy than static activities but lower accuracy than walking. Both sit-down and stand-up are motion activities and they have different impacts on CSI patterns, so they are easier to recognize compared with static activities. The issue for transition activities is that they are sometimes misclassified as walking. Some testing persons have some different static/motion patterns compared with other persons. For example, one person could be playing with a smartphone during data collection. This introduces minor movements which could confuses the recognition algorithm to misclassify transition activities as walking. Therefore, the major issue is how to improve the recognition performance of static and transition activities.

The recognition accuracy of static and transition activities is significantly improved by the state machine, as shown in Fig. 12b. This is achieved by using the state machine to learn time dependencies and context information

from neighboring CSI segments. The recognition algorithm utilizes the relatively high accuracy of transition activities and the temporal dependencies to improve the recognition performance of static and transition activities. Person 5 and 7 have slightly lower recognition accuracy than other persons. For person 5 and 7, 14% to 20% of sitting are misclassified as standing, and 11% to 17% of sit-down are misclassified as standing. The major reason is that the state machine sometimes gives wrong classification results of the temporal dependency information. The state machine of the proposed design is a small 1D CNN with fixed neural architecture. It is possible to improve the recognition performance by using a deeper 1D CNN as the state machine and with its neural architecture optimized by the reinforcement learning agent. We leave the optimization of the state machine as future work.

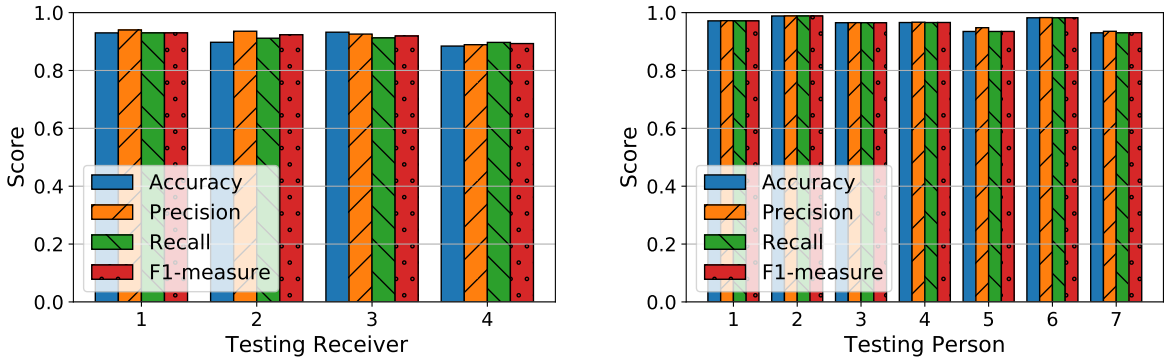
3.1.5 Impact of Reinforcement Learning. To check the impact of reinforcement learning, we use SeparableConv2D as the recognition algorithm with or without reinforcement learning for neural architecture search. Fig. 2 shows the average recognition accuracy of SeparableConv2D with or without reinforcement learning. When the location and orientation of the WiFi receiver is known, the accuracy is 99% if the data of testing persons are seen during training. But the accuracy drops to 84% for leave-one-person-out validation wherein the data of testing persons are not seen during training. When the location and orientation of the WiFi receiver is unknown during training, the recognition accuracy of SeparableConv2D drops to 62%. When reinforcement learning is used for neural architecture search, the recognition accuracy of SeparableConv2D is improved to 77% for unknown receiver locations and orientations. When Conv1D is added as the state machine, the accuracy of SeparableConv2D with reinforcement learning is improved to 97%.

Table 3. Number of parameters and inference time consumption per instance for different convolution types.

Convolution Type	State Machine	Reinforcement Learning	Number of Parameters		Inference Time per Instance	Average Accuracy (std)
			Trainable	Non-Trainable		
Conv2D	None	Yes	426324	82	18.5 milliseconds	69.14% (6.8%)
Conv3D			127167	65	20.6 milliseconds	71.29% (4.0%)
SeparableConv2D			7992	72	18.7 milliseconds	76.86% (2.3%)
Conv2D	Conv1D	Yes	7500	50	11.3 milliseconds	92.57% (5.5%)
Conv3D			39911	60	14.3 milliseconds	83.29% (9.9%)
SeparableConv2D			13743	46	12.2 milliseconds	96.60% (2.0%)

Table 3 shows the neural architecture summary of the trained model and the corresponding performance results with and without state machine. SeparableConv2D has a higher recognition accuracy and comparable inference time consumption as Conv2D and Conv3D for both with and without state machine. The inference time consumption per instance is calculated by running the trained CNN on each testing CSI instance one by one. Non-trainable parameters are from batch normalization layers. These parameters are updated with the mean and variance of the batch normalization input, but are not trained with backpropagation.

3.1.6 Impact of Receiver Location/Orientation and Target Person: Location and Person Independence Test. Fig. 13a shows performance results of different receiver locations/orientations for leave-one-device-out testing of person 7. The average accuracy of different receivers is 93%, which demonstrates that our design is robust for different unknown receivers. Recognition scores of the best performing SeparableConv2D architecture with and without the state machine are shown in Fig. 13. With state machine and reinforcement learning, the recognition accuracy of each testing person is 97%, 99%, 97%, 97%, 95%, 99%, and 93%, as shown in Fig. 13b. The overall recognition accuracy is improved by 20% when the state machine and reinforcement agent are added.



(a) Testing results of different receivers with unknown person. (b) Testing results of different persons with unknown receiver.

Fig. 13. Testing results of different persons for leave-one-person-out and leave-one-device-out testing.

Table 4. Impact of CSI Sampling Rate

Packet Interval (milliseconds)	Performance Score			
	Accuracy	Precision	Recall	F1-measure
10	97%	97%	97%	97%
100	91%	93%	92%	92%

3.1.7 Impact of CSI Sampling Rate. Table 4 shows the performance results of different CSI sampling rate for leave-one-person-out (person 1) and leave-one-device-out (receiver 1) testing. The recognition accuracy is 97% for packet interval of 10 milliseconds and 91% for packet interval of 100 milliseconds. This means that our design can still provide accurate performance without actively sending CSI measurement packets. Our design can utilize existing WiFi packets, such as beacon packets that usually have 100 milliseconds packet interval, for passive and accurate activity recognition.

3.2 Evaluation Results of New Datasets and New Environments

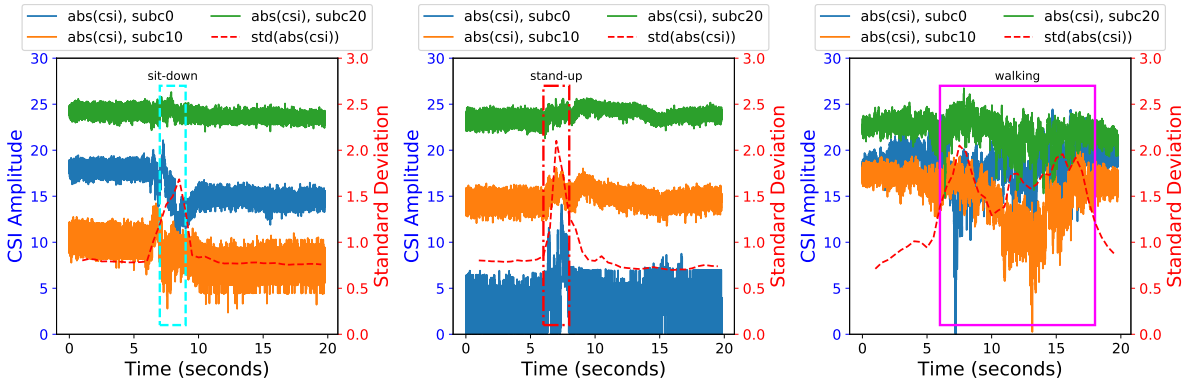
This section presents evaluation results of two public datasets from S.Yousefi-2017 [51] and FallDeFi [29].

Table 5. Summary of Datasets

Dataset	Size[1]	# Rooms	# RX	# TX	# Persons	# Instances	Input Shape of Each Instance
FallDeFi [29]	1.06 GB	1 corridor, 1 kitchen, 1 lab, 1 bathroom, 2 bedrooms	5	5	5	397	(2000, 30, 3, 1)
S.Yousefi-2017 [51]	2.78 GB	1 office	1	1	6	2079	(2000, 30, 3, 1)
This paper	17.05 GB	1 lab	4	1	7	14555	(200, 30, 3, 3)

[1] Size of numpy array of CSIs of 5 activities. Other activities of S.Yousefi-2017 [51] and FallDeFi [29] are not included.

3.2.1 Dataset Overview. A summary of the two datasets is shown in Table 1. For S.Yousefi-2017 [51], CSI measurements are collected in 1 room from 6 persons with 1 transmitter and 1 receiver. For FallDeFi [29], there are 6 rooms, 5 persons, 5 transmitters and 5 receivers. There are 2079 and 397 instances for S.Yousefi-2017 [51] and FallDeFi [29], respectively. The size of each instance for both datasets is (2000, 3, 30) representing 2000 CSI matrices with 3 receive antennas and 30 subcarriers measured in 2 seconds. Both datasets have CSI measurements of other activities, like fall, bend and pickup, but only 5 activities, i.e., sitting, standing, sit-down, stand-up and walking, are included in the evaluation. More details of these two datasets can be found in [51] and [29].

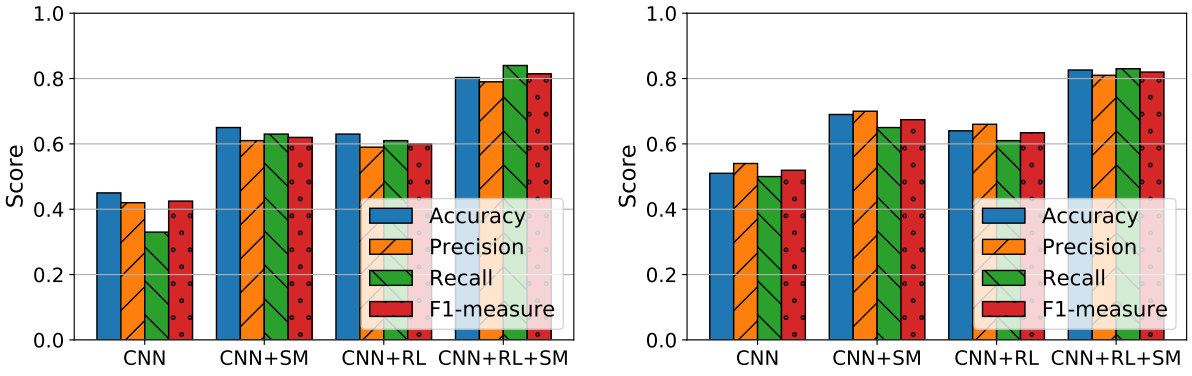


(a) Standing → sit-down → sitting (b) Sitting → stand-up → standing (c) Standing → walking → standing

Fig. 14. CSI segmentation of different activities using standard deviation of CSI amplitude. CSI data is from S.Yousefi-2017 [51].

These two datasets only have ground truth labels for sitting, standing and walking but do not have ground truth labels for transition activities including stand-up and sit-down. To address this issue, we use the standard deviation of CSI amplitude to calculate the start and end time of transition activities. This is similar to the method used in FallDeFi [29] for CSI segmentation. Fig. 14 shows some examples of CSI segmentation of different activities. The standard deviation is calculated by the amplitude of CSI measurements within a 2 seconds time window with a sliding window of 0.5 second. The start and end of transition activities are calculated by comparing the standard deviation with a pre-defined threshold which is $0.8 \times max(std(csi))$ in our case. Based on the ground truth label of the CSI sequence, the ground truth labels of the calculated transition window and the CSI segments before and after the transition window can be determined. For example, because the ground truth label of the CSI sequence in Fig. 14a is known as sitting, after the transition window is detected within the time window from 5 seconds to 7 seconds, the labels can be determined as standing before 5 seconds, sit-down from 5 seconds to 7 seconds, and sitting after 7 seconds. The labeled CSI segments are fed to different neural networks for evaluation.

3.2.2 Evaluation Results. The evaluation results of new datasets and new environment are shown in Fig. 15. The recognition accuracy of the proposed design, i.e., SeparableConv2D with state machine and reinforcement learning, is 80% and 83% for S.Yousefi-2017 [51] and FallDeFi [29], respectively. For SeparableConv2D with reinforcement learning but without Conv1D as the state machine, the accuracy drops to 63% and 64% for S.Yousefi-2017 [51] and FallDeFi [29], respectively. For SeparableConv2D without state machine or reinforcement learning, the accuracy further drops to 45% and 51%. Although the accuracy of the proposed design for these two datasets is 14% to 17% lower than when it is evaluated by our dataset, the reinforcement learning agent and state machine provide similar accuracy improvements for all the three datasets. Evaluation results demonstrate that our design is robust for different rooms, unknown transmitter/receiver deployments, and unseen persons.



(a) Evaluation results for the dataset from S.Yousefi-2017 [51]

(b) Evaluation results for the dataset from FallDeFi [29]

Fig. 15. Evaluation results of new datasets and dew environments: (a) test of unseen persons, (b) test of unseen environments, unseen persons, and unknown locations and orientations of the WiFi transmitter and receiver.

4 DISCUSSIONS

4.1 Overhead of Neural Architecture Search

The neural architecture search agent, NASCell, has very high overhead. For computer vision tasks, NASCell takes hundreds of GPU hours to find a good neural architecture. For our dataset with 14555 CSI instances each with size of (200, 3, 3, 30), NASCell takes about three weeks for a 1080 Ti GPU to find the neural architecture for SeparableConv2D without state machine. For the SeparableConv2D with Conv1D as the state machine, the searching time is about 10 days. Recently, there are some new approaches for more efficient neural architecture search, such as Efficient Neural Architecture Search (ENAS) [30], Differentiable Architecture Search (DARTS) [21], EfficientNet [36], and RegNet [31], for computer vision tasks. Compared with NASCell, these two approaches has about 1000 times less overhead with comparable accuracy. These efficient neural architecture search approaches can also be used to find the suitable neural architecture for CSI data. Moreover, since CSI is different from images, new neural architectures and new neural architecture search methods are needed to design and search for the best neural network models that are specifically designed for CSI data.

4.2 Robustness in New Environments

The accuracy of the proposed design is 97% for our dataset, and it drops to 80% and 83% for S.Yousefi-2017 [51] and FallDeFi [29], respectively. The major reasons of lower accuracy for the two public datasets are the dataset size and the quality of CSI segmentation and labeling. There are 14555 instances for our dataset, while there are only 2079 and 397 instances, respectively, for S.Yousefi-2017 [51] and FallDeFi [29]. Increasing the dataset size should also bring performance improvements. In our dataset, motion tracking devices and audio instructions are used to calculate fine-grained CSI segmentation and accurate ground truth labeling. But the two public datasets do not have accurate labeling information for transition activities, which could also impact the recognition performance. Besides, the length of CSI sequences of the two datasets is less than 10 seconds, so these two datasets contain less temporal dependency information for the state machine to learn. Collecting more CSI data with fine-grained labeling information and enough temporal dependency information could improve the performance of the proposed design. Another challenge for WiFi-based activity recognition is to get accurate and robust performance in real-world scenarios such as multiple persons and other head/hand/body motions. When there

are multiple persons, the multi-path channel is much more complex than when there is only one person. For example, it is hard to locate and recognize the activity of each person when there is one person siting static and another person walking. In this case, it needs other sensing capabilities such as human counting, device-free localization, and human identification. New models are needed to utilize these sensing functions and to combine multi-domain knowledge for practical activity recognition in real-world scenarios. Our goal is to first make single-person activity recognition independent of receiver locations and target persons and then try to extend it for multi-person scenarios in the future.

5 RELATED WORK

Recently, CSI is widely used for human motion detection and activity recognition. The survey [26] gives a review of signal processing techniques, algorithms, performance results, challenges, and future trends for different WiFi sensing applications.

Table 6. Related Works of Fall Detection and Motion Detection with CSI

Reference	Signal Processing	Algorithm	Performance
WiFall [18]	Weighted Moving Average (WMA), Local Outlier Filter (LOF)	kNN, One-Class SVM	Fall Detection Precision: 87%
FallDeFi [29]	Wavelet Filter, DWT, STFT, PCA, Interpolation, Thresholding	One-Class SVM	Fall Detection Accuracy: 93%/80% (same/different environments)
RT-Fall [41]	STFT, Band-Pass Filter (BPF), Interpolation, Thresholding	One-Class SVM	True Positive Rate: 91%, True Negative Rate: 92%
Anti-Fall [53]	Interpolation, Low-Pass Filter (LPF), Threshold-Based Sliding Window	One-Class SVM	Precision: 89%, False Alarm Rate: 13%
WiSpeed [54]	Median Filter, ℓ_1 Trend Filter, Thresholding	Statistical Modeling, Peak Detection	Fall Detection Rate: 95%
WiKey [1, 2]	LPF, PCA, DWT	kNN+DTW	Keystroke Detection: 97.5%
MAIS [11]	LPF, Outlier Filter, Thresholding	kNN	Anomaly Detection: 98.04%
FRID [13]	N/A	CSI Phase Coefficients	Motion Detection Precision: 90%
MoSense [15]	LPF, Euclidean Distance, Thresholding	Binary Classification	Motion Detection Accuracy: 97.38%/93.33% (LoS/NLoS, 5 activities)
AR-Alarm [20]	Interpolation, BPF, Duration-Based Filter	Binary Classification	True Positive Rate: 98.1%/97.7%
Liu-2017 [22]	Signal Isolation by Skewness	One-Class SVM	Motion Detection Rate: 90.89%
Wi-Sleep [23, 24]	Hampel Filter, Wavelet Filter, DWT, Interpolation	Pattern Matching	Posture Change Detection: 83.3%
SEID [25]	Signal Compression by CSI Amplitude Variance	HMM	Motion Detection Precision: 98%
WiStep [50]	Long Delay Removal, DWT, BPF, PCA	Peak Detection, Threshold-Based Detection	True Positive Rate: 96.41%, False Positive Rate:1.38%
NotiFi [58]	PCA	Dynamic Hierarchical Dirichlet Process, Bayesian Nonparametric Model	Abnormal Activity Detection Accuracy: 89.2%/ 85.6%/75.3% (LoS/NLoS/through-wall)
Khan-2017 [28]	Cross-Ambiguity Function	Doppler Frequency Shift	Fall Detection Accuracy: 98%

5.1 Motion Detection with CSI

In recent years, CSI is widely used for fall detection [18, 29, 37, 41, 53, 54] and motion detection [2, 11, 13, 15, 20, 22–25, 50, 58]. Table 6 shows a summary of the signal processing techniques, algorithms, and performance results of CSI-based fall and motion detection. Motion detection is a relatively simple task and sometimes has no clear borderline between signal processing and the detection algorithm. After some signal processing techniques such as low-pass filters and thresholding, the detection result can be directly derived without detection or classification algorithms. Modeling-based algorithms, e.g., threshold-based detection and peak detection, and very simple learning-based algorithms, e.g., one-class Support Vector Machine (SVM), are widely used for WiFi-based motion detection. Theoretical and statistical models are usually very sensitive to noises and outliers, so noise reduction is usually needed, such as the Hampel filter, wavelet filter, and local outlier filter. Aryokee [37] also uses CNNs and state machine, but its objective is fall detection with radar signals while ours is activity recognition with WiFi signals. Radar is designed for sensing and has finer granularity than WiFi which is designed for communication but not for sensing. So activity recognition with WiFi is much harder than fall detection with radar. Our goal is to not only detect motions but also recognize different motion and static activities with high and robust performance. Besides, Aryokee uses HMM as the state machine which needs to be trained separately and has low performance for CSI data that involve non-linearities and long-term temporal dependencies. Our design has the recognition algorithm and state machine trained together and the neural architecture automatically optimized by reinforcement learning.

5.2 Activity Recognition with CSI

CSI is commonly used for recognizing human activities, including daily activities [3, 7, 9, 11, 12, 16, 19, 42–48], shopping [52], driving [8, 32], exercising [49], and head & mouth activities [10]. Table 7 shows a summary of the signal processing techniques, algorithms, and recognition accuracy of CSI-based activity recognition. Almost all the recognition applications use learning-based algorithms as the classifier. SVM and k Nearest Neighbor (kNN) are two popular classifiers for CSI-based activity recognition. Dynamic Time Wrapping (DTW) is usually used for kNN as the distance metric. Learning-based algorithms are usually not very sensitive to noises and outliers. Many learning-based algorithms use none or very simple noise reduction methods such as averaging and median filters. Noise reduction is usually used for modeling-based algorithms which are typically sensitive to noises. The major issue for modeling-based and instance-based learning algorithms is that they are not location or person independent when the data of testing devices or persons are not seen during training or modeling. Another issue for instance-based learning algorithms is that they need to calculate the distance from the testing instance to all the training instances. This introduces high overhead when there are multiple classes and each class instance has many CSI data points. SVM, kNN, and DTW have high inference costs for calculating the distance of different samples, so they usually employ feature extraction, subcarrier selection, or dimension reduction to reduce the input size. EI [19] uses a CNN as the recognition algorithm, but its recognition accuracy is less than 75%. SignFi [27] also uses a CNN for gesture recognition with WiFi, but it is tested with known WiFi receiver locations and orientations and has low accuracy for leave-one-person-out tests. WiMU [39] recognizes 2 to 6 simultaneously performed gestures with accuracy of 95.0% to 90.9%. CrossSense [56] uses CNN and expert models, including Naive Bayes, Random Forest, SVM with linear/RBF kernels, kNN, and Adaboost, to get over 80%/90% accuracy for gait identification/gesture recognition for 100 users and 40 gestures. WiRadar3.0 [57] uses Doppler frequency shift to extract body-coordinate velocity profile and proposes CNN-RNN models to recognize 6 gestures with accuracy of 92.7% (in-domain) and 82.6% to 92.4% (cross-domain) using WiFi data of 16 users measured from 3 environments. Since CSI data are different from images and videos, it may result in low recognition performance by just reusing CNNs that are designed for computer vision tasks. It is necessary to find the suitable CNNs, including the CNN types, neural architectures, and neural weights, that are specifically designed for CSI

Table 7. Related Works of Activity Recognition with CSI

Reference	Signal Processing	Algorithm	Accuracy
Wi-Chase [3]	LPF	kNN, SVM	97% (3 activities)
WIBECAM [7]	N/A	Autoregressive Model	73% to 100% (4 activities)
BodyScan [9]	LPF, PCA, Thresholding	SVM	72.3% (5 activities)
MAIS [11]	LPF, Outlier Filter, Thresholding	kNN	93.12% (3 activities)
DFLAR [12]	N/A	Sparse Auto-Encoder	90% (8 activities)
HuAc [16]	Outlier Filter, WMA; LPF, Thresholding, k Means	SVM	93% (16 activities)
EI [19]	Hampel Filter; Thresholding	CNN	<75% (10 users, 6 activities)
Wang-2018 [42]	Median Filter, Linear Fitting, LPF	SOM, Softmax Regression	>85% (8 activities)
CARM [43, 44]	DWT, Thresholding, PCA	HMM	>96% (8 activities)
Wang-2015 [45]	Gaussian Filter, LOF, k Means	DTW, SVM	80% (13 activities)
E-eyes [46]	LPF, Thresholding, Clustering	Multi-Dimensional DTW, Pattern Matching	90%/95% (single device/ multiple devices, 13 activities)
Wei-2015 [47]	Exponential Smoothing	Sparse Representation	<90% (8 activities)
ARM [48]	Wavelet Filter; DWT	DTW, HMM	>75% (6 activities)
Zeng-2015 [52]	BPF	Decision Tree, Simple Logistic Regression	89.6%/94.75 (entrance/in store, 4 activities)
WiDriver [8]	Signal Compression by Neural Network	Fresnel Zone Model, Finite Automata	96.8% (11 postures), 90.76% (7 activities)
HeadScan [10]	LPF, PCA	Sparse Representation, ℓ_1 Minimization	86.3% (5 activities)
WiBot [32]	LPF, PCA	kNN	94.5%/90.5% (3/5 activities)
SEARE [49]	LPF, Median Filter, PCA, Thresholding	DTW	97.8%/91.2% (LoS/NLoS, 4 activities)
WiMU [39]	STFT, PCA, Thresholding	Threshold-Based Detection, Pattern Matching	95.0%, 94.6%, 93.6%, 92.6%, 90.9% (2, 3, 4, 5, 6 concurrent gestures)
CrossSense [56]	PCA, Feature Selection	CNN, Expert Models (Naive Bayes, Random Forest, SVM, kNN, Adaboost)	>80%/90% (gait identification/ gesture recognition, 100 users, 40 gestures)
WiRadar3.0 [57]	Doppler Frequency Shift, Body-coordinate Velocity Profile	CNN, RNN	82.6% to 92.4% (16 users, 6 gestures, 3 environments)
Zhang-2019 [55]	Least-Square Smoothing Filter	Fresnel Zone Model, CNN	95%/92.1% (3/9 activities)
Chen-2016 [5]	micro-Doppler	Sparse Representation Classifier	90.2%/85.2% (Channel1/ Channel2, 6 motions)

data. To address this issue, we use different convolutions as the recognition algorithm for learning location and person independent features from different perspectives of CSI data. Moreover, we use reinforcement learning for optimizing the neural architecture of the recognition algorithm and a lightweight 1D CNN as the state machine for learning temporal dependencies. Wang et al. [40] and Zhang et al. [55] evaluate the impact of user location and body orientation on human respiration detection with commodity wifi devices. Our evaluation include not only unknown user location/orientation but also unknown location/orientation of WiFi receivers.

6 CONCLUSION

In this paper, we propose a novel deep learning solution for robust activity recognition with WiFi. The proposed solution uses a 2D CNN as the recognition algorithm, a 1D CNN as the state machine, and a reinforcement learning agent to find the best neural architecture for the recognition algorithm. We evaluate the proposed design with real-world traces of 5 activities performed by 7 persons. The proposed design provides 97% average recognition accuracy for unknown receiver locations/orientations and for unseen persons. The reinforcement learning agent provides 15% accuracy improvement compared with when the neural architecture is manually searched. The state machine, along with the reinforcement learning agent, provides another 20% accuracy improvement by learning temporal dependencies from history classification results. The proposed design is also evaluated by two public datasets and achieves 80% and 83% accuracy respectively. The proposed design requires very little human efforts for ground truth labeling, signal processing, feature engineering, parameter tuning, and neural architecture search.

REFERENCES

[1] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke Recognition Using WiFi Signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom '15)*. ACM, 90–102. https://doi.org/10.1145/2789168.2790109

[2] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2017. Recognizing Keystrokes Using WiFi Devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (May 2017), 1175–1190. https://doi.org/10.1109/JSAC.2017.2680998

[3] Sheheryar Arshad, Chunhai Feng, Yonghe Liu, Yupeng Hu, Ruiyun Yu, Siwang Zhou, and Heng Li. 2017. Wi-Chase: A WiFi based Human Activity Recognition System for Sensorless Environments. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 1–6. https://doi.org/10.1109/WoWMoM.2017.7974315

[4] Eli Bendersky. 2018. Depthwise separable convolutions for machine learning. Retrieved December 12, 2018 from https://eli.thegreenplace.net/2018/depthwise-separable-convolutions-for-machine-learning/

[5] Qingchao Chen, Bo Tan, Kevin Chetty, and Karl Woodbridge. 2016. Activity recognition based on micro-Doppler signature with in-home Wi-Fi. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*. 1–6.

[6] François Chollet. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *CVPR 2017*. arXiv:1610.02357

[7] Mauro De Sanctis, Ernestina Cianca, Simone Di Domenico, Daniele Provenziani, Giuseppe Bianchi, and Marina Ruggieri. 2015. WIBECAM: Device Free Human Activity Recognition Through WiFi Beacon-Enabled Camera. In *Proceedings of the 2Nd Workshop on Workshop on Physical Analytics (WPA '15)*. ACM, 7–12. https://doi.org/10.1145/2753497.2753499

[8] Shihong Duan, Tianqing Yu, and Jie He. 2018. WiDriver: Driver Activity Recognition System Based on WiFi CSI. *International Journal of Wireless Information Networks* (Feb 2018). https://doi.org/10.1007/s10776-018-0389-0

[9] Biyi Fang, Nicholas D. Lane, Mi Zhang, Aidan Boran, and Fahim Kawsar. 2016. BodyScan: Enabling Radio-based Sensing on Wearable Devices for Contactless Activity and Vital Sign Monitoring. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '16)*. ACM, 97–110. https://doi.org/10.1145/2906388.2906411

[10] Biyi Fang, Nicholas D. Lane, Mi Zhang, and Fahim Kawsar. 2016. HeadScan: A Wearable System for Radio-based Sensing of Head and Mouth-Related Activities. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 1–12. https://doi.org/10.1109/IPSN.2016.7460677

[11] Chunhai Feng, Sheheryar Arshad, and Yonghe Liu. 2017. MAIS: Multiple Activity Identification System Using Channel State Information of WiFi Signals. In *Wireless Algorithms, Systems, and Applications*. Springer International Publishing, 419–432.

[12] Qinhua Gao, Jie Wang, Xiaorui Ma, Xueyan Feng, and Hongyu Wang. 2017. CSI-based Device-free Wireless Localization and Activity Recognition Using Radio Image Features. *IEEE Transactions on Vehicular Technology* 66, 11 (Nov 2017), 10346–10356. https://doi.org/10.1109/TVT.2017.2737553

[13] Liangyi Gong, Wu Yang, Dapeng Man, Guozhong Dong, Miao Yu, and Jiguang Lv. 2015. WiFi-based Real-Time Calibration-Free Passive Human Motion Detection. *Sensors* 15, 12 (2015), 32213–32229.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[15] Yu Gu, Jinhai Zhan, Yusheng Ji, Jie Li, Fuji Ren, and Shangbing Gao. 2017. MoSense: An RF-based Motion Detection System via Off-the-Shelf WiFi Devices. *IEEE Internet of Things Journal* 4, 6 (Dec 2017), 2326–2341. https://doi.org/10.1109/JIOT.2017.2754578

[16] Linlin Guo, Lei Wang, Jialin Liu, Wei Zhou, and Bingxian Lu. 2018. HuAc: Human Activity Recognition Using Crowdsourced WiFi Signals and Skeleton Data. *Wireless Communications and Mobile Computing* (2018). https://doi.org/10.1155/2018/6163475

[17] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *SIGCOMM Comput. Commun. Rev.* 41, 1 (Jan. 2011), 53–53. https://doi.org/10.1145/1925861.1925870

[18] Chunmei Han, Kaishun Wu, Yuxi Wang, and Lionel M. Ni. 2014. WiFall: Device-free Fall Detection by Wireless Networks. In *2014 IEEE Conference on Computer Communications (INFOCOM)*. 271–279. https://doi.org/10.1109/INFOCOM.2014.6847948

[19] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. ACM, 289–304. https://doi.org/10.1145/3241539.3241548

[20] Shengjie Li, Xiang Li, Kai Niu, Hao Wang, Yue Zhang, and Daqing Zhang. 2017. AR-Alarm: An Adaptive and Robust Intrusion Detection System Leveraging CSI from Commodity Wi-Fi. In *Enhanced Quality of Life and Smart Living*. Springer International Publishing, 211–223.

[21] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. DARTS: Differentiable Architecture Search. In *ICLR 2019*. arXiv:1806.09055

[22] Jialin Liu, Lei Wang, Linlin Guo, Jian Fang, Bingxian Lu, and Wei Zhou. 2017. A Research on CSI-based Human Motion Detection in Complex Scenarios. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*. 1–6. https://doi.org/10.1109/HealthCom.2017.8210800

[23] Xuefeng Liu, Jiannong Cao, Shaojie Tang, and Jiaqi Wen. 2014. Wi-Sleep: Contactless Sleep Monitoring via WiFi Signals. In *2014 IEEE Real-Time Systems Symposium*. 346–355. https://doi.org/10.1109/RTSS.2014.30

[24] Xuefeng Liu, Jiannong Cao, Shaojie Tang, Jiaqi Wen, and Peng Guo. 2016. Contactless Respiration Monitoring Via Off-the-Shelf WiFi Devices. *IEEE Transactions on Mobile Computing* 15, 10 (Oct 2016), 2466–2479. https://doi.org/10.1109/TMC.2015.2504935

[25] Jiguang Lv, Dapeng Man, Wu Yang, Xiaojiang Du, and Miao Yu. 2018. Robust WLAN-based Indoor Intrusion Detection Using PHY Layer Information. *IEEE Access* 6, 99 (2018), 30117–30127. https://doi.org/10.1109/ACCESS.2017.2785444

[26] Yongsen Ma, Gang Zhou, and Shuangquan Wang. 2019. WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.* 52, 3, Article 46 (June 2019), 36 pages. https://doi.org/10.1145/3310194

[27] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. 2018. SignFi: Sign Language Recognition Using WiFi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 23 (March 2018), 21 pages. https://doi.org/10.1145/3191755

[28] Usman Mahmood Khan, Zain Kabir, and Syed Ali Hassan. 2017. Wireless health monitoring using passive WiFi sensing. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 1771–1776.

[29] Sameera Palipana, David Rojas, Piyush Agrawal, and Dirk Pesch. 2018. FallDeFi: Ubiquitous Fall Detection Using Commodity Wi-Fi Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 155 (Jan. 2018), 25 pages. https://doi.org/10.1145/3161183

[30] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. 2018. Efficient Neural Architecture Search via Parameter Sharing. In *ICML 2018*. arXiv:1802.03268

[31] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Designing Network Design Spaces. In *CVPR 2020*. arXiv:2003.13678

[32] Muneeba Raja, Viviane Ghaderi, and Stephan Sigg. 2018. WiBot! In-Vehicle Behaviour and Gesture Recognition Using Wireless Network Edge. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 376–387. https://doi.org/10.1109/ICDCS.2018.00045

[33] scikit learn. 2019. Classification metrics. Retrieved April 25, 2019 from https://scikit-learn.org/stable/modules/model_evaluation.html

[34] SolidRun. 2019. HummingBoard. Retrieved May 13, 2019 from https://www.solid-run.com/nxp-family/hummingboard/

[35] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction* (2ed ed.). MIT press.

[36] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *ICML 2019*. arXiv:1905.11946

[37] Yonglong Tian, Guang-He Lee, Hao He, Chen-Yu Hsu, and Dina Katabi. 2018. RF-Based Fall Monitoring Using Convolutional Neural Networks. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 137 (Sept. 2018), 24 pages. https://doi.org/10.1145/3264947

[38] David Tse and Pramod Viswanath. 2005. *Fundamentals of Wireless Communication*. Cambridge University Press.

[39] Raghav H. Venkatnarayan, Griffin Page, and Muhammad Shahzad. 2018. Multi-User Gesture Recognition Using WiFi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services* (Munich, Germany) *(MobiSys '18)*. 401–413. https://doi.org/10.1145/3210240.3210335

[40] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. 2016. Human Respiration Detection with Commodity WiFi Devices: Do User Location and Body Orientation Matter?. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. 25–36. https://doi.org/10.1145/2971648.2971744

[41] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. 2017. RT-Fall: A Real-Time and Contactless Fall Detection System with Commodity WiFi Devices. *IEEE Transactions on Mobile Computing* 16, 2 (Feb. 2017), 511–526. https://doi.org/10.1109/TMC.2016.2557795

[42] Jie Wang, Liming Zhang, Qinghua Gao, Miao Pan, and Hongyu Wang. 2018. Device-free Wireless Sensing in Complex Scenarios Using Spatial Structural Information. *IEEE Transactions on Wireless Communications* 17, 4 (April 2018), 2432–2442. https://doi.org/10.1109/TWC.2018.2796086

[43] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom*

'15). ACM, 65–76. https://doi.org/10.1145/2789168.2790093
[44] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2017. Device-free Human Activity Recognition Using Commercial WiFi Devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (May 2017), 1118–1131. https://doi.org/10.1109/JSAC.2017.2679658
[45] Yi Wang, Xinli Jiang, Rongyu Cao, and Xiyang Wang. 2015. Robust Indoor Human Activity Recognition Using Wireless Signals. *Sensors* 15, 7 (2015), 17195–17208. https://doi.org/10.3390/s150717195
[46] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-eyes: Device-free Location-oriented Activity Identification Using Fine-grained WiFi Signatures. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom '14)*. ACM, 617–628. https://doi.org/10.1145/2639108.2639143
[47] Bo Wei, Wen Hu, Mingrui Yang, and Chun Tung Chou. 2015. Radio-based Device-free Activity Recognition with Radio Frequency Interference. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks (IPSN '15)*. 154–165. https://doi.org/10.1145/2737095.2737117
[48] Wei Xi, Dong Huang, Kun Zhao, Yubo Yan, Yuanhang Cai, Rong Ma, and Deng Chen. 2015. Device-free Human Activity Recognition Using CSI. In *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition (CSAR '15)*. ACM, 31–36. https://doi.org/10.1145/2820716.2820727
[49] Fu Xiao, Jing Chen, Xiao Hui Xie, Linqing Gui, Juan Li Sun, and Wang Ruchuan. 2018. SEARE: A System for Exercise Activity Recognition and Quality Evaluation Based on Green Sensing. *IEEE Transactions on Emerging Topics in Computing* (2018). https://doi.org/10.1109/TETC.2018.2790080
[50] Yang Xu, Wei Yang, Jianxin Wang, Xing Zhou, Hong Li, and Liusheng Huang. 2018. WiStep: Device-free Step Counting with WiFi Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 172 (Jan. 2018), 23 pages. https://doi.org/10.1145/3161415
[51] Siamak Yousefi, Hirokazu Narui, Sankalp Dayal, Stefano Ermon, and Shahrokh Valaee. 2017. A Survey on Behavior Recognition Using WiFi Channel State Information. *IEEE Communications Magazine* 55, 10 (Oct 2017), 98–104. https://doi.org/10.1109/MCOM.2017.1700082
[52] Yunze Zeng, Parth H. Pathak, and Prasant Mohapatra. 2015. Analyzing Shopper's Behavior Through WiFi Signals. In *Proceedings of the 2Nd Workshop on Workshop on Physical Analytics (WPA '15)*. ACM, 13–18. https://doi.org/10.1145/2753497.2753508
[53] Daqing Zhang, Hao Wang, Yasha Wang, and Junyi Ma. 2015. Anti-fall: A Non-intrusive and Real-Time Fall Detector Leveraging CSI from Commodity WiFi Devices. In *Inclusive Smart Cities and e-Health*. Springer International Publishing, 181–193.
[54] Feng Zhang, Chen Chen, Beibei Wang, and K. J. Ray Liu. 2018. WiSpeed: A Statistical Electromagnetic Approach for Device-Free Indoor Speed Estimation. *IEEE Internet of Things Journal* (2018). https://doi.org/10.1109/JIOT.2018.2826227
[55] Fusang Zhang, Kai Niu, Jie Xiong, Beihong Jin, Tao Gu, Yuhang Jiang, and Daqing Zhang. 2019. Towards a Diffraction-Based Sensing Approach on Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1, Article 33 (March 2019), 25 pages. https://doi.org/10.1145/3314420
[56] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. 2018. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. ACM, 305–320. https://doi.org/10.1145/3241539.3241570
[57] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, Republic of Korea) *(MobiSys '19)*. ACM, 313–325. https://doi.org/10.1145/3307334.3326081
[58] Dali Zhu, Na Pang, Gang Li, and Shaowu Liu. 2017. NotiFi: A Ubiquitous WiFi-based Abnormal Activity Detection System. In *2017 International Joint Conference on Neural Networks (IJCNN)*. 1766–1773. https://doi.org/10.1109/IJCNN.2017.7966064
[59] Barret Zoph and Quoc V. Le. 2017. Neural Architecture Search with Reinforcement Learning. In *ICLR 2017*. arXiv:1611.01578

Received January 2020; revised June 2020; accepted September 2020