Letter

Recurrent ConFormer for WiFi Activity Recognition

Miao Shang and Xiaopeng Hong

Dear Editor,

Human activity recognition (HAR) using WiFi signals has been a significant task due to its potential applications in for example, healthcare services and smart homes. This letter deals with the WiFi channel state information (CSI)-based HAR task. To capture the dynamics of human activities well from CSI without using a huge number of training samples, we propose a recurrent model of convolution blocks and transformer encoders. Firstly, the model utilizes the convolution blocks to capture local variation and the self-attention mechanism in transformer encoders to characterize long-range dependencies. Secondly and more importantly, the recurrent architecture models the context information well within CSI signals and allows the network to deepen without scale increase, making it particularly suited to learning from a small amount of CSI samples.

With the rapid development of deep learning models, an increasing number of approaches utilize deep learning to solve HAR tasks using WiFi signals. Various deep models for CSI-based human activity recognition have been proposed. One sort of studies [1]–[3] utilized CNN structures to extract features. Wang *et al*. [1] processed the temporal information of CSI signals by 1D-ResNet [4]. GS [2] encoded the input into a 2D RGB image and fed it into the 2D-CNN structure EfficientNet [5]. Zhang *et al*. [3] proposed a 3D-CNN to learn the spatiotemporal dynamic patterns. Considering the temporal characteristics of the CSI signal, there are studies performing time series classifications. Yousefi *et al*. [6] compared several traditional feature-based methods with the long short-term memory (LSTM), demonstrating the potential of employing recurrent deep models to process CSI signals. Afterwards, Meng *et al*. [7] proposed a modified attention-based bi-directional GRU network to learn features in two directions. In [8], CSI signals were transformed to the domain-independent feature and then fed into a CNN-GRU structure to capture spatial and temporal features. Yadav *et al*. [9] regarded the CSI-based HAR as a multi-variate time series problem and modified the InceptionTime network to extract features. OneFi in [10] transformed the CSI series to the Doppler spectrogram by short-time Fourier transform and applied transformer for classification. THAT in [11] adopted two-stream transformer to process raw CSI signals by the dimension of time and channel, respectively.

Though promising progress has been made, there are still limitations: 1) Over-parameterized deep models are easy to overfit on small datasets. Deep models usually require a large amount of training data while most datasets for CSI-based HAR only have a size from hundreds to thousands of samples [1], [6] and [10]. 2) Deep CNN models with fixed convolution kernel size can extract local features but have difficulty capturing long-range relations within signals [12].

To address these problems, in this letter, we propose a lightweight and efficient Recurrent model of CONvolution blocks and transFORMER encoders (Recurrent ConFormer) for HAR using CSI signals. The main contribution lies in two aspects: Firstly, we incorporate the recurrent mechanism into the transformer encoder and propose the recurrent transformer module, which allows for building the deep architecture with a fixed number of parameters. Secondly, we propose to cascade the recurrent convolution and transformer modules. This architecture captures the local variation by convolution blocks and models the long-range dependencies among local features by transformer encoders. To the best of our knowledge, this is the first time to combine the recurrent mechanism with the cascaded CNN and transformer for CSI-based recognition.

Problem statement: Given the raw CSI data of the received WiFi signals, we build a model with a feature extractor to capture the dynamics and a classifier head to recognize the types of human activities. In concrete, consider the system below:

$$\begin{cases} z = f_\omega(x) \\ \hat{y} = h(z) \end{cases} \tag{1}$$

where $x \in \mathbb{R}^{d\times T}$ is a time sequence of the amplitude of CSI singals as the input. As [1], we only use the information of CSI amplitude and ignore other information such as the CSI phrase to recognize the activities. d and T are the number of sub-carriers and timestamps, respectively. $\hat{y}$ is the output of the system to approach the ground-truth label y. The CSI samples and corresponding ground-truth labels consist of the training dataset $\mathcal{D}$. f_ω is the feature extractor with the parameters ω to extract the dynamic feature of inputs, and h is a classifier head to recognize the activities given the feature extracted.

We formulate the feature extractor as a composition of two functions to fully characterize the dynamic information of CSI signals.

$$f_\omega(x) = f_g(f_l(x)) \tag{2}$$

where f_l aims to capture local variation and f_g further encodes long-range dependencies. Let z_l and z_g denote the outputs of f_l and f_g respectively. The system can be rewritten as

$$\begin{cases} z_l = f_l(x) \\ z_g = f_g(z_l) \\ \hat{y} = h(z_g). \end{cases} \tag{3}$$

Method: The proposed model consists of three parts: the recurrent CNN module f_l, the recurrent transformer module f_g and a classifier head h, as illustrated in Fig. 1. The recurrent CNN module uses convolution operations to extract local feature. The recurrent transformer module then takes advantage of the self-attention operation to get global correlations of these local features. Finally, the classifier head with a fully-connected layer is provided to make prediction.

Recurrent CNN: Fig. 2 illustrates the architecture of the recurrent CNN module, which consists of a downsample stage and a recurrent convolution stage. In the downsample stage, we downsample the raw CSI data x to $x^0 \in \mathbb{R}^{d'\times T'}$ by an 1-D convolution operation[1] with a kernel size of 7×1 and a max pooling operation as in [1]. Obviously, $T' = T/4$. d' is the number of convolution channel.

In the recurrent convolution stage, we incorporate the recurrent mechanism [13] with the residual structure. As shown in Fig. 2, the recurrent convolution stage is composed of the recurrent convolution block and a residual connection [4]. The convolution block $\psi(\cdot)$, as the basic unit in recurrence (abbreviated by recurrent unit) evolves over time steps through the recurrent and the feed-forward computation. In concrete, given the downsampled CSI data x^0 as the input, the recurrent convolution stage is formulated as follows:

$$\begin{cases} x^1 = \psi(x^0;\theta_0) \\ x^{k+1} = \psi(x^k;\theta) + \psi_0(x^0;\theta_0) \\ z_l = ReLU(x^{N_C} + x^0) \end{cases} \tag{4}$$

[1] Considering the time-sequence characteristics of CSI signals, 1-D convolutional operation (Conv-1D) is used to capture the feature along the temporal dimension.

Corresponding author: Xiaopeng Hong.

Citation: M. Shang and X. P. Hong, "Recurrent ConFormer for WiFi activity recognition," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 6, pp. 1491–1493, Jun. 2023.

M. Shang is with the College of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: miaos0522@gmail.com).

X. P. Hong is with Harbin Institute of Technology, Harbin 150001, China (e-mail: hongxiaopeng@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123291

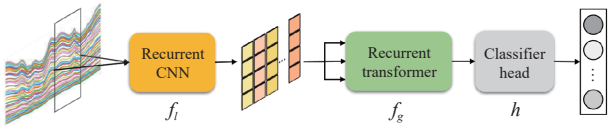

Fig. 1. The overall architecture of the proposed model. The feature extracted by recurrent CNN is divided into patches along the temporal dimension and then fed into the recurrent transformer. Finally, the prediction of activity can be obtained by the classifier head.

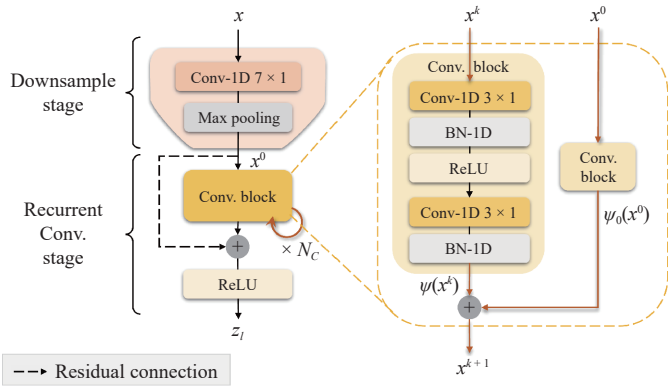

Fig. 2. Detailed structure of the recurrent CNN, where Conv. Block is short for the convolution block and Conv-1D 3×1 stands for Conv-1D with a kernel size of 3×1. The parameters of the convolution blocks are shared during iterations.

where z_l is the output of the recurrent convolution stage obtained after N_C time steps. x^k is the output of the convolution block after the kth time step, for $k = 1, 2, \ldots, N_C - 1$, and works as the input to the next time step. The convolution block $\psi(\cdot;\theta)$ and $\psi_0(\cdot;\theta_0)$ represent the recurrent and the feed-forward computation functions, with the parameter settings θ and θ_0 respectively. $\psi(\cdot;\theta)$ is made up of two 1D convolution operators, two 1D batch normalizations, and an activation function $ReLU(\cdot)$. $\psi_0(\cdot;\theta_0)$ is obtained by using x^0 as the input to $\psi(\cdot)$ with the parameter θ_0, which is stored during the recurrent evolution. When $N_C = 1$, (4) degrades into the normal 1D residual convolution block in ResNet [4]. It is worth mentioning that compared with the original recurrent structure in [13], the presented one is with a residual connection and multiple convolution operations.

Recurrent transformer: This module deals with global correlations between the local features extracted in recurrent CNN. The overall architecture is shown on the left of Fig. 3. It has an embedding and a recurrent encoder stage, and takes the output of recurrent CNN z_l as the input. The embedding stage encodes z_l into embedding vectors. Specifically, we first split z_l into T' time patches and apply a linear projection to get the embedding e_i of each time patch

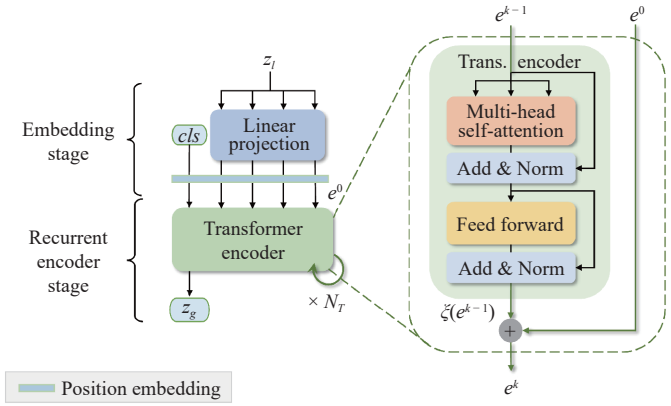

Fig. 3. Detailed architecture of the recurrent transformer. Parameters of transformer encoder are shared during iterations.

with the size of $\mathbb{R}^{d_h}$, where $\mathbb{R}^{d_h}$ reflects the hidden dimension of the transformer encoder, $i = 1, 2, \ldots, T'$. Moreover, a class token $cls \in \mathbb{R}^{d_h}$ is attached. After that, for better timestamp sensitivity, we add learnable position embedding $\boldsymbol{p} = (p_0, p_1, \ldots, p_{T'})$, where $p_i \in \mathbb{R}^{d_h}$, $i = 0, 1, \ldots, T'$, to maintain the absolute position information of each patch embedding. Thus, we have the embedding sequence $\boldsymbol{e}^0 = (cls + p_0, e_1 + p_1, e_2 + p_2, \ldots, e_{T'} + p_{T'})$ and input it into the recurrent encoder stage.

As shown in Fig. 3, the main part in the recurrent encoder stage is a transformer encoder [12], which also served as the recurrent unit. Inside the transformer encoder, the multi-head self-attention (MHSA) with h heads is used to characterize the long-range interactions in sequential embeddings and capture global high-level feature by computing the attention scores between any two embeddings. The feed-forward layer is an MLP with a depth of 2 and the feed-forward dimension of d_f. More details about Transformer Encoder can be found in [12]. The recurrent encoder is formulated as follows:

$$\begin{cases} \boldsymbol{e}^k = \xi(\boldsymbol{e}^{k-1};\delta) + \boldsymbol{e}^0 \\ \boldsymbol{e}^{N_T} = \xi(\boldsymbol{e}^{N_T-1};\delta) \\ z_g = e^{N_T}(1) \end{cases} \tag{5}$$

where the output z_g is the class embedding after N_T iterations, i.e., the first element of e^{N_T}, and $\xi(\cdot;\delta)$ is a transformer encoder with parameter setting δ. $\boldsymbol{e}^k$ is the output of the kth recurrent transformer encoder for $k = 1, 2, \ldots, N_T - 1$. Note that the module degrades into a normal transformer encoder when $N_T = 1$. The recurrent structure is similar to the one in the recurrent CNN module. The only difference is to replace the feed-forward computation by an identity connection.

Classifier head: The classifier head is a fully connected linear layer with parameters weight $W \in \mathbb{R}^{d_h \times c}$ and bias $b \in \mathbb{R}^c$, where c is the number of categories. The prediction turns out in the form of score with the softmax function σ as follows:

$$\hat{y} = \sigma(W^T z_g + b). \tag{6}$$

Loss function: The cross-entropy loss is used as loss function to optimize our recurrent ConFormer network

$$L = \sum_{(x,y)\in\mathcal{D}} -y \log \hat{y}. \tag{7}$$

Datasets: We conduct extensive experiments on two WiFi human activity recognition datasets to verify our proposed method.

ARIL: ARIL [1] contains 1398 samples of 6 activities, i.e., hand up, hand down, hand left, hand right, hand circle and hand cross, each performed 15 times at 16 different locations. One out of every five trials are evenly selected to build the test set (278) and the others are used for training (1116). All the instances are collected by universal software radio peripherals with one WiFi antenna to broadcast and receive WiFi signals. The raw CSI data includes CSI phase and amplitude. The number of sub-carriers and the streams for each CSI data is 52 and 192.

UT-HAR: UT-HAR [6] contains 557 samples of 7 activities (lie down, fall, pick up, run, sit down, stand up, walk) with raw CSI phase and amplitude. The samples are collected by transmitter and receiver with 3 antennas. The receiver is equipped with Intel 5300 NIC, with the sampling rate of 1 kHz. We randomly split the dataset into non-overlapping training set (80%) and test set (20%). The number of sub-carriers and the streams for each sample are 30 and 2000. Therefore, the shape of one CSI amplitude sequence sample is $(30 \times 3) \times 2000$.

Results and discussion: In this section, we provide the implementation details and the comparative results.

When implementing the network, we set the hidden dimension d_h and feed-forward dimension d_f in transformer encoder to 128 and 256, and the output dimension d' for Conv-1D in recurrent CNN to 128. The number of heads h in MHSA is set to 8 and 4 for ARIL and UT-HAR, respectively. The recurrent depth of recurrent CNN N_C and recurrent transformer N_T are both set to 4.

The network is implemented by Pytorch 1.12.0 with Python 3.9 and trained on an NVIDIA GeForce RTX 3060 GPU. The model is

trained for 100 and 50 epochs in total on ARIL and UT-HAR, respectively. In the training process, we set the batch size to 64 and the learning rate to 0.0001, which decreases by 0.1 after 75 epochs for ARIL. As for UT-HAR, we set the batch size to 16 and the learning rate decreases by 0.1 after 40 epochs. We repeat the experiments with different random seeds for ten times for reliable evaluation and report the average top-1 accuracy (avg Acc.) and the standard deviation (Std.).

Results on ARIL: We compare our network with two state-of-art works on ARIL, i.e., ResNet1D [1] and Gimme' signals (GS) [2]. Wang *et al*. [1] used the 1D ResNet along the temporal dimension. Memmesheimer *et al*. [2] encoded the CSI series to an image and fed it into the 2D CNN EfficientNet [5]. They used a re-implementation and pre-trained weights of EfficientNet.

We use the source code of the method and repeat it for ten times for reliable and fair comparisons. To exclude the interference of other issues, we only list the accuracy of the methods without data augmentation. As reported in Table 1, the accuracy of our proposed model is 95.83% which is 6% higher than ResNet1D and 2.7% higher than GS. It demonstrates our recurrent ConFormer framework achieve a superior performance on HAR tasks. Moreover, our model has far fewer parameters than the other two, indicating the role of the recurrence mechanism in reducing the complexity of the model.

Table 1. Comparative Experimental Results on ARIL

Method	Framework	Params	Avg Acc.	Std.
ResNet1D [1]	ResNet-1D	2.69 MB	89.78%	1.73
GS [2]	EfficientNet-2D	9.11 MB	93.07%	1.47
Ours	Recurrent ConFormer	0.40 MB	95.83%	0.62

Results on UT-HAR: We also compare our network with two state-of-art works on UT-HAR, including the methods based on LSTM [6] and two-stream augmented transformer (THAT) in [11]. The comparative results are presented in Table 2.

Table 2. Comparative Experimental Results on UT-HAR.

Method	Framework	Params	Avg Acc.	Std.
Yousefi *et al*. [6]	LSTM	0.24 MB	90.90%	2.27
THAT [11]	Transformer	49.3 MB	95.71%	1.32
Ours	Recurrent ConFormer	0.50 MB	96.16%	0.74

As reported in Table 2, our method outperforms LSTM by more than 5% and THAT by about 0.4%. Meanwhile, including 1.3 MB in the channel stream and 48 MB in the temporal stream, the amount of parameters evolved in THAT is about 100 times than ours. The comparison with THAT indicates that the recurrent ConFormer is a lighter model with comparable performance, which can be well-trained when meets smaller datasets. The results demonstrate that our model has a good balance of accuracy and complexity.

Discussion: The results demonstrate the proposed recurrent mechanism, which is built on the recurrent units, and the recurrent and feed-forward connections. The recurrent units are formed by the convolution block and the transformer encoder in the recurrent CNN and the recurrent transformer modules, respectively. Fig. 4 shows how unfolding the recurrent units leads to recurrence. As we can see, it degrades into a single layer when $t = 1$. More importantly, the depth of the network built by recurrent units is increased when t goes larger. As a result, deep enough networks can be efficiently built without adding any parameters.

Conclusion: This letter has investigated the problem of human activity recognition using WiFi signals. To learn the model efficiently from a limited number of training samples, we propose the recurrent ConFormer for CSI-based HAR. The proposed recurrent model not only combines the advantages of CNN and transformer but also builds a deep enough structure with a fixed number of parameters. Results of comparative experiments on ARIL and UT-HAR indicate the superiority of the proposed method in both accuracy and efficiency. Future work will focus on developing more lightweight and training-efficient solutions.

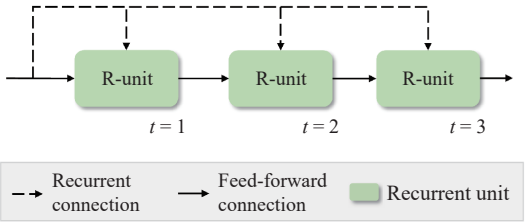

Fig. 4. Illustration of unfolding a recurrent unit for $t = 3$ time steps.

Acknowledgments: This work was supported by the National Key Research and Development Project of China (2019YFB1312000) and the Fundamental Research Funds for the Central Universities (AUGA5710011522).

References

[1] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han, "Joint activity recognition and indoor localization with WiFi fingerprints," *IEEE Access*, vol. 7, pp. 80058–80068, 2019.

[2] R. Memmesheimer, N. Theisen, and D. Paulus, "Gimme signals: Discriminative signal encoding for multimodal activity recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.,* 2020, pp. 10394–10401.

[3] R. Zhang, C. Jiang, S. Wu, Q. Zhou, X. Jing, and J. Mu, "Wi-Fi sensing for joint gesture recognition and human identification from few samples in human-computer interaction," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2193–2205, 2022.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2016, 770–778. DOI: 10.1109/CVPR.2016.90.

[5] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 6105–6114.

[6] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 98–104, 2017.

[7] W. Meng, X. Chen, W. Cui, and J. Guo, "Wihgr: A robust wifi-based human gesture recognition system via sparse recovery and modified attention-based BGRU," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10272–10282, 2021.

[8] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3.0: Zero-effort cross-domain gesture recognition with Wi-Fi," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 44, no. 11, pp. 8671–8688, 2021.

[9] S. K. Yadav, S. Sai, A. Gundewar, H. Rathore, K. Tiwari, H. M. Pandey, and M. Mathur, "Csitime: Privacy-preserving human activity recognition using WiFi channel state information," *Neural Networks*, vol. 146, pp. 11–21, 2022.

[10] R. Xiao, J. Liu, J. Han, and K. Ren, "OneFi: One-shot recognition for unseen gesture via cots WiFi," in *Proc. Conf. Embed. Networked Sens*., 2021, pp. 206–219.

[11] B. Li, W. Cui, W. Wang, L. Zhang, Z. Chen, and M. Wu, "Two-stream convolution augmented transformer for human activity recognition," in *Proc. AAAI Conf. Artificial Intelligence*, 2021, vol. 35, no. 1, pp. 286–293.

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

[13] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2015, pp. 3367–3375.