A Real-time Object Detection for WiFi CSI-based Multiple Human Activity Recognition

Israel Elujide[1], Jian Li[1], Aref Shiran[1], Siwang Zhou[2], and Yonghe Liu[3]
[1,3]*Department of Computer Science & Engineering*, *The University of Texas at Arlington, USA*
[2]*College of Information Science and Technology*, *Hunan University, China*
Email: {israel.elujide, jian.li3, aref.shiran}@mavs.uta.edu, [2]swzhou@hnu.edu.cn, [3]yonghe@cse.uta.edu

2023 IEEE 20th Consumer Communications & Networking Conference (CCNC) | 978-1-6654-9734-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCNC51644.2023.10059647

***Abstract*—In the recent past, human activity recognition research has focused on using WiFi channel state information (CSI) as a viable alternative to legacy systems like video and sensor-based activity recognition having limitations such as privacy invasion, obtrusiveness, and the inconvenience of wearing sensory devices. While the performance of CSI-based activity recognition models is impressive, many of the models are built using offline processed data from regulated settings which hinders their application in real-time. However, real-life human activity recognition requires models to be responsive to identifying activities in real-time. To address the shortcoming of CSI-based activity recognition models, we propose a deep learning object detection framework and instance segmentation for multiple human activity recognition using WiFi signals. The real-time CSI data from the signal is captured on a sliding window and converted into time-frequency domain images of the activity stream using continuous wavelet transform (CWT). Since it is impossible to pre-segment activities within a stream in real-time, the power profile from the transformed images is exploited to provide insights for deep learning instance segmentation to identify each unique human activity. The evaluation is carried out using real-time CSI data with single and multiple human activities. The results show that real-time model classification accuracy is 93.80% on average and instance segmentation accuracy of 90.73%.**

***Index Terms*—WiFi, channel state information, activity recognition, deep learning, object detection framework**

I. Introduction

CSI-based human activity recognition has been a subject of intensive research because it is considered a feasible alternative to legacy recognition systems. The inexpensive nature of the WiFi devices for obtaining CSI, coupled with user privacy elimination and ready availability of wireless local area networks make CSI-based recognition attractive. Similarly, the change in society due to the COVID-19 pandemic has accentuated the focus on contact-free human activity recognition. Many public facilities like airports, restaurants, and shopping malls now render services with limited physical contact requiring devices to implement gesture-based authentication and learn unique user patterns by extracting subtle attributes from user mannerisms for authentication [1], [2]. The performance of CSI-based recognition models has been impressive, but many have cross-domain limitations and are non-real-time.

The cross-domain challenge is when a model with a good performance in one location cannot give satisfactory performance in another location. The limitation is due to the nature of raw CSI data because it contains both gesture and location-dependent information, and frameworks to address the challenge were proposed in [3], [4]. Similarly, the non-real-time problem is when a recognition model is trained with data from regulated settings comprising pre-segmented sequences of related activities [5]–[7]. The training data is processed offline and the model may produce satisfactory results for experimentation, but becomes inadequate for real-time recognition. It is noteworthy that working with real-time data is challenging. For example, the data stream contains continuously changing activity, the activities from the stream can be concurrent or interleaved, the duration and interval between activities might not be uniform, and information about activities from the stream cannot be determined in advance. Moreover, the deployment of human activity recognition models in real-life necessitates working with streaming data, and the models should be able to obtain contextual information from the stream and be responsive enough to recognize activity in real-time.

To address the non-real-time challenge, we propose a real-time deep learning object detection framework and instance segmentation for CSI-based multiple human activity recognition. This is the first WiFi CSI-based proposal to address the problem to the best of our knowledge. The closest work is [8] for a legacy system focusing on the classification of tagged sensor data and [9] for a software-defined radar system with specialized equipment. Unfortunately, the technique in [8] cannot be applied to CSI-based models because activities are not collected using sensors, thereby making it difficult to glean information about activities from sensor characteristics. Additionally, our model is a real-time object detection deep learning that classifies human activities and localizes each activity in the context of the stream.

The overall target of our system is to address issues with the CSI-based models produced using offline preprocessing and pre-segmentation to identify human activity in CSI data, thereby limiting their real-time application. Specifically, the system contains the following components: CSI collection, CSI-to-image transformation, and object detection network. In the CSI collection phase, data from user activities were collected using [10] during WiFi signal transmission from a transmitter to a receiver. The real-time signal containing the activity data is captured using a sliding window before the transformation. To leverage changes in both time and frequency domains of the signal simultaneously, we apply a

978-1-6654-9734-3/23/$31.00 ©2023 IEEE

wavelet transformation by converting the CSI data from the WiFi signal to images in the second phase [11], [12]. In the last step, an object detection deep learning network using Mask R-CNN is used to achieve human activity classification, localization, and instance segmentation within a continuous stream generated from images in the previous stage [13]. To summarize, this paper addresses the issues of CSI-based activity recognition models when dealing with real-time human activity data, and the main contributions are as follows:

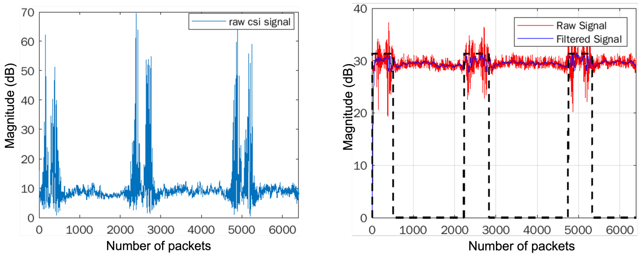
- We present a system that simultaneously tracks the changes in both frequency and time domains of a WiFi signal to fix the issue of multiple activities on a single stream without performing offline demarcation between the activities.
- We perform continuous wavelet transformation (CWT) on the CSI data and exploit the power profile from the transformed images to help resolve the difficulty of tracking frequency changes due to different activities.
- We implement an object detection deep learning network to accurately localize when each activity occurs within a stream, correctly classify the activity labels with their corresponding confidence level and provide a unique segmentation mask between different instances of the same activity.
- With thorough evaluation and comparison, we show that our deep learning object detection model achieves an accuracy of 93.80% on average when trained with multiple activities in a single stream.

The remainder of this paper is organized as follows. In Section II, we review related work. We present the system design and different modules in Section III. Implementation and evaluation are presented in Section IV. The results are discussed in Section V. Finally, the conclusion is presented in Section VI.

II. Related Work

The availability of CSI on low-cost commercial off-the-shelf (COTS) wireless network interface cards made it possible to use CSI for many applications like vehicular monitoring and alert system for dangerous driving, sleep monitoring, evaluation of a noisy and crowded wireless environments, and fall detection [14]–[16]. Many of these applications leveraged CSI data from a controlled setting where the setup is geared towards the intended empirical investigation. Recognition models from a controlled environment may be suitable for experimentation but face practical difficulty when the context changes to real life.

A critical task before training a CSI-based model is the extraction of human activity from CSI data, and the extracted information is processed as training features for the model. Determining the portion of CSI data where an activity lies is challenging and tracking the beginning and end in an online stream increases the complexity. The offline process of tracking activity in CSI data is shown in Fig. 1. Performing a similar task on streaming data is challenging. The work in [17] uses the variance of the CSI amplitude of subcarriers by using a preset threshold to determine the beginning and end of an activity, [18] locates the active part containing human action by the time interval of the mean Doppler shift. In [19], the authors adopt a peak-finding algorithm to determine the duration of activity within a stream by using a sliding window for extracting the start time and end time of each activity. Likewise, [20] detects the start and end of the action in the CSI data by calculating the variance using the second eigenvector and the corresponding principal component across the subcarriers.



(a) CSI from multiple subcarriers

(b) Activity tracking from a single subcarrier

Fig. 1: Signal perturbation from human activity on CSI data and activity tracking in a filtered CSI signal on a single subcarrier.

It is important to note that these algorithms are applied offline to extract activity within CSI data, making them unsuitable for real-time activity detection. However, detecting multiple activities within a real-time stream requires a technique that works online and simultaneously tracks changes between the activities. To address the challenge, we employ a sliding window and time-frequency CSI to image transformation by monitoring the human activity streaming data in the time domain and tracking changes in the energy band in the frequency domain.

III. System Design

A. CSI and Real-time Human Activity

The wireless signals contain the CSI for human activity resulting from multipath and perturbation in the channel. We capture the CSI data containing the signal variation across the subcarriers indicating the perturbation. Training a model with CSI activity data involves extracting the portion having the active signal due to perturbation by human activity. For a given activity period, we collect CSI across all antenna streams. If n_t represents the number of transmission antennas and n_r represents the number of receiving antennas, then the received signal can be expressed by the following equation

$$y = Hx + n, \tag{1}$$

where x and y represent transmitted signal and received signal complex vector respectively, n is channel noise vector. The CSI, H, is the channel matrix between the transmitter and receiver.

The CSI values between the transmit-receive antenna pair contain the channel response for all the subcarriers. In our case, this represents $n_t \times n_r \times N_{sc}$ CSI values where N_{sc} denotes 30 number of subcarriers. Hence, the CSI for a single spatial link can be represented as

$$H = [h_1, h_2, \ldots, h_{N_{sc}}], \tag{2}$$

and each subcarrier CSI h_s, $s \in [1, \ldots, N_{sc}]$, in equation (2) is a complex value containing both amplitude and phase information.

B. CSI Imaging

The module performs a time-frequency analysis and converts the time-series CSI data to images in preparation for training by the object detection network. In preparation for the transformation, we captured the real-time activity data from the stream using a sliding window. If a real-time data stream, represented as an infinite sequence, is given by $S =< d_1, d_2, d_3, \ldots >$ and d_i is an n-dimension vector data item in the stream where $i \in \{1, 2, 3, ...\}$ and n is 30 subcarriers. The initial window, W, containing k data items from the stream, serves as a baseline. Then, the consequent window moves a time step with the new data item from the stream. The output of the window is used for CSI-to-image transformation. To address similarity and redundancy between frames as a result of the sliding windows, we applied the frame distance measure in [21].

We collect CSI of various human activities by leveraging tools on wireless devices [10] and feed the extracted CSI as input data for the next phase – CSI imaging. The input CSI is denoted as $X_i \in \{X_1, X_2, \ldots, X_N\}$, where N is the number of collected CSI samples. Each CSI sample is associated with human activity labels depending on the number of activities performed when the data was collected. The corresponding gesture label can be represented as $L_i^g \in \{0, \ldots, n-1\}$, where n is the number of unique human activities.

The time-frequency analysis is performed by examining the frequency content of a changing signal with time. FFT allows visibility of the dominant frequencies in a time-series signal but does not capture changes in time from the frequency domain. The inability to capture the changes in the time domain by FFT prevents its direct application to CSI data. The limitation is addressed by short-time Fourier transform (STFT); however, the resolution of the spectrogram generated by STFT is limited by its window function and was fixed by techniques in [22] using continuous wavelet transform (CWT). The CWT of a signal, s(t), is defined as

$$CWT(t, \omega) = \left(\frac{\omega}{\omega_o}\right)^{\frac{1}{2}} \int s(t')\Psi^*\left(\frac{\omega}{\omega_o}(t'-t)\right) dt', \tag{3}$$

where $\Psi(\cdot)$, and $\frac{\omega}{\omega_o}$ are the *mother* wavelet and scale parameter respectively. For a *mother* wavelet with zero time origin oscillating at ω_o, CWT is essentially a decomposition of signal $s(t')$ to many shifted and scaled wavelets $\Psi[\omega/\omega_o(t'-t)]$. When the transformation into wavelets is done at a fixed t or ω, the equation (3) becomes

$$CWT(t, \omega) = \frac{\left(\frac{\omega_o}{\omega}\right)^{1/2}}{2\pi} \int S(\omega')\Psi^*\left(\frac{\omega_o}{\omega}\omega'\right) e^{j\omega' t} d\omega', \tag{4}$$

and the $\Psi(\omega')$ in equation (4) is the Fourier transform of $\Psi(t')$.

C. Object Detection Network

The training network is a deep learning object detection based on a region-proposal detection framework. The network comprises feature extraction, Region Proposal Network (RPN), RoIAlign, Fully Convolution Network (FCN), softmax, and boundary box regressor. Each of the components is explained in detail below.

a) Feature Extraction: This component uses a deep Convolutional Neural Network (CNN) backbone to obtain low-level feature representations like edges and corners at the early layers and high-level features at later layers. Our implementation is based on ResNet-50. CNN is suitable for this purpose because of the image training input. The feature pyramid network (FPN)improves the feature extraction from the backbone by allowing the extracted features in each layer to be shared between the preceding and succeeding layers. Thereby generating a pyramid of information flow for the backbone network from the top down.

b) Region Proposal Network: It takes the inputs and extracts a low-dimensional vector using a sliding window over the convolution feature map from the feature extraction stage. The main goal of this phase is to scan the image for areas containing human activity of interest called anchors. RPN avoids duplication of feature computation by using the output from the backbone and FPN networks. The outputs of this stage are rectangular proposals for each activity in the image with its corresponding score.

c) RoIAlign: This stage works on region-of-interest (RoI), which is the part of the image containing the human activity to generate a fixed-size feature map. The ROIAlign layer improves detection accuracy by eliminating misalignment necessary for segmentation.

d) Bounding Box Regressor: The aim of the layer is to learn scale-invariant and log-scale transformation between a proposed box and a ground-truth box. The algorithm works by considering a set of N training pairs $(p^i, g^i)_{i=1,\ldots,N}$ where p^i and g^i are the proposal and ground truth respectively. For the bounding box shown in Fig. 3, coordinates of the proposal's bounding box and ground truth bounding box are given as $\mathbf{p} = (p_x, p_y, p_w, p_h)$, $\mathbf{g} = (g_x, g_y, g_w, g_h)$ respectively and the transformation by the regressor is

$$\begin{aligned} \hat{g}_x &= p_w d_x(\mathbf{p}) + p_x & \hat{g}_y &= p_h d_y(\mathbf{p}) + p_y \\ \hat{g}_w &= p_w \exp(d_w(\mathbf{p})) & \hat{g}_h &= p_h \exp(d_h(\mathbf{p})) \end{aligned} \tag{5}$$

The function $d_i(\mathbf{p})$ where $i \in \{x, y, w, h\}$ is modeled as linear transformation given by $d_i(\mathbf{p}) = \mathbf{w}_i^T \phi(\mathbf{p})$ where $\mathbf{w}_i$ is a vector of model parameters. If the target is

$$\begin{aligned} t_x &= (g_x - p_x)/p_w & t_y &= (g_y - p_y)/p_h \\ t_w &= log(g_w/p_w) & t_h &= log(g_h/p_h), \end{aligned} \tag{6}$$

then the regression equation for minimizing the sum of squares loss can be expressed as

$$\mathcal{L}_{reg} = \underset{\hat{\mathbf{w}}_\mathbf{i}}{\arg\min} \sum_i (t_i - d_i(\mathbf{p}))^2 + \lambda \|\hat{\mathbf{w}}\|^2. \tag{7}$$

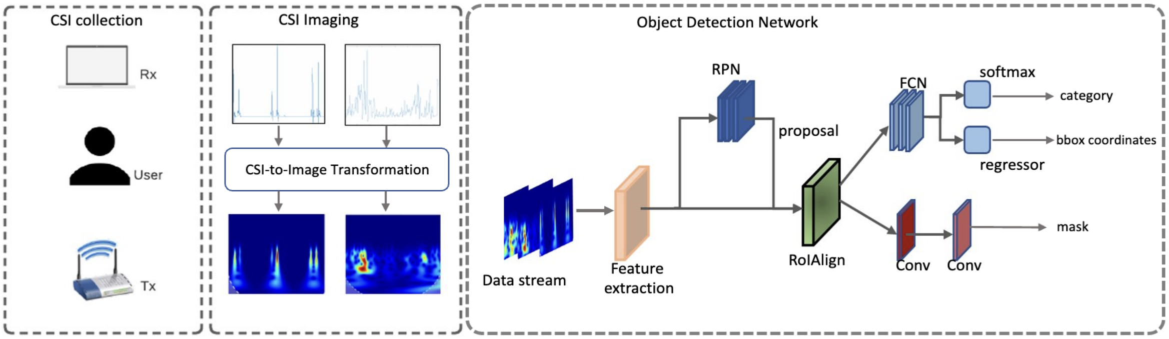


Fig. 2: The system architecture of real-time object detection for CSI-based human activity recognition.

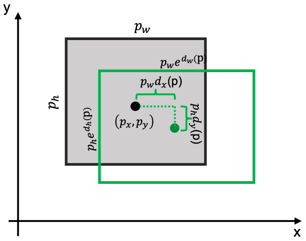


Fig. 3: Object detection predicted and ground truth bounding boxes.

D. Loss Function of Object Detection Network

During training, the loss function calculates the sum of each of the loss components - softmax loss, regression loss, and mask loss. By optimizing the loss function, the model gains better performance. The loss of the object detection network is given as

$$L = L_{cls} + L_{bbox} + L_{mask}, \tag{8}$$

$$\begin{aligned} L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, g_i) \; + \\ \lambda \frac{1}{N_{reg}} \sum_i g_i L_{reg}(t_i, t_i^*) \; + \\ \frac{-1}{m^2} \sum_{1 \le i,j \le m} \left[z_{i,j} \log \hat{z}_{i,j}^k + (1 - z_{i,j}) \log(1 - \hat{z}_{i,j}^k) \right], \end{aligned} \tag{9}$$

where $z_{i,j}$ and $\hat{z}_{i,j}^k$ are mask for cell (i,j) for the ground truth region of size $m \times m$ and predicted mask region of class k respectively. The L_{cls} is the classification loss and a cross-entropy represented in equation (9), L_{bbox} is the bounding box regression loss, and L_{mask} is the binary cross-entropy loss of the mask branch. If there are k number of classes and the mask branch generates a mask of $m \times m$ per RoI of each class, then the total output of the mask branch becomes $k * m^2$.

E. Performance Evaluation Metrics

The training network is an object detection model based on a Mask R-CNN objection detection framework. The model performance is measured using metrics called Intersection over Union and mean Average Precision. The performance evaluation metric and objective function of the training process are explained in the following subsections.

a) Intersection over Union (IoU): This is the ratio of the area of overlap of the prediction boundary box and ground truth boundary box to the area obtained by the union of both ground truth and prediction boundary boxes. An IoU score of more than 50% is considered a satisfactory prediction.

b) mean Average Precision (mAP): This is the average IoU for all classes divided by the classes' average. In other words, the mAP of a class is the total of the IoU of all data points divided by the total class labels for the class.

IV. Implementation and Evaluation

A. Experimental Setup

We capture human activity data from the transmitter and receiver to model real-time object detection for CSI data and perform a time-frequency transformation to generate a fixed-sized set of images. The images are similar to frames in a video stream, and the frames are treated as the real-time output of streaming CSI data. The output represents the entry point into the training network.

The experiment's environment consists of an access point serving as a transmitter and a laptop as a receiver. The network interface card on the laptop is Intel NIC5300 for collecting the CSI, and the transmitter is a dual-band TP-Link AC1750 operating in the 2.4 GHz frequency band. The receiver runs on Ubuntu Linux 12.04 LTS with a modified kernel to support the NIC5300 [10]. Finally, the architecture for model training is implemented using Pytorch, and the training is performed on a Google Colab notebook dual-core Intel(R) CPU @ 2.20GHz and using the TPU backend.

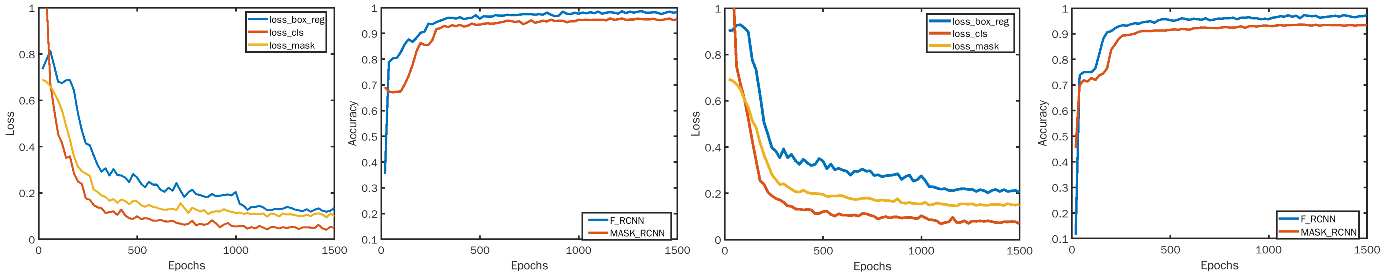
B. Data Collection

For performance analysis, we ask multiple subjects to perform three actions. These actions are the human activities considered in our evaluation. The human activities are hand movement, running, and walking. The sampling rate of the experiment is 80 packets/second. The data is split into a 70% training set, a 15% validation set, and a 15% test set. For instance, the walk activity dataset consists of 312, 81, and 62 activity instances for training, validation, and testing.

V. Results

A. Accuracy of a single human activity

We first analyzed the performance of the proposed model on the CSI dataset for one human activity. The sample was split into 70% for training, 15% for testing, and 15% for



(a) Training loss of run activity (b) Accuracy of run activity (c) Training Loss of walk activity (d) Accuracy of walk activity

Fig. 4: Evaluation of system using a single human activity.

validation dataset. The model is trained using the training data and evaluated using the validation set. After the model training is completed, the trained model is evaluated using the test dataset. To reduce the model's training time and improve the performance, we leveraged transferred learning by importing pre-trained weights of a base architecture and then fine-tuning them during training with our dataset. The base architecture in our case is ResNet50. Each model training lasted 1500 epochs, and evaluation was performed every 500 steps on validation data. The performance during training and testing for average precision for single and multiple activities is shown in Table I and II, respectively. Likewise, the mAP for single and multiple activities are presented in Table III and IV respectively.

TABLE I: Average precision result of validation data.

Model Accuracy				
Activity	AP (%)	AP_{50} (%)	AP_{75} (%)	Recall (%)
walk	60.34	100	60.30	66.4
run	73.65	99.55	87.45	77.7
walk-wave-run	58.05	96.94	62.99	65.4

a) run activity: There are 115 activity instances in the training dataset. Similarly, the validation and test datasets comprise 16 and 12 unique human activity instances for evaluation, respectively. The training loss for both the faster-RCNN branch and mask branch is shown in Fig. 4a. Likewise, the performance evaluation of the model in terms of accuracy is shown in Fig. 4b. In Table I, the average precision for IoU with 50 percent overlap after complete training is 99.55%, while the average precision for 0.75 area overlap and range with 0.5 - 0.95 overlap is 87.45% and 73.65% respectively. Table II shows the evaluation with the test dataset when the detection threshold for RoI is set at 85 percent gives 100%, 72.95%, and 66.55% average precision for 50 percent, 75 percent, and 50 - 95 percent IoU respectively. As shown in Table III, the average precision per category to successfully detect run activity during the training is 67.07% and 80.22% for no activity. For the test dataset, the performance is 63.97% average precision for run activity detection and 76.931% for no activity.

TABLE II: Average precision result of test data.

Model Accuracy				
Activity	AP (%)	AP_{50} (%)	AP_{75} (%)	Recall (%)
walk	63.00	99.96	81.84	69.0
run	66.55	100	72..95	76.2
walk-wave-run	64.67	96.94	83.00	65.0

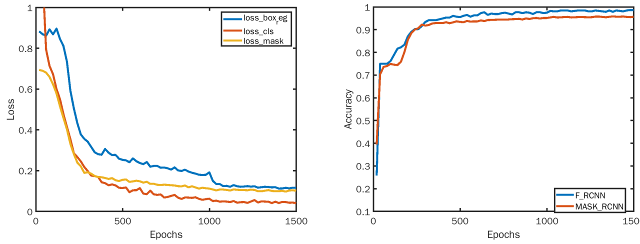
TABLE III: mAP single activity

Activity	Validation (%)	Testing (%)
run	67.073	63.969
walk	48.313	55.372

TABLE IV: mAP multiple activities

Activity	Validation (%)	Testing (%)
run	47.366	53.267
walk	61.337	62.772
wave	59.901	73.366

b) walk activity: The dataset comprises 312 activity instances for training, 81 instances for validation, and 62 instances for testing. At the end of the training, the average precision for IoU with 50, 75, and 50 - 95 percent area overlap is 100%, 60.30%, and 60.34%, respectively. The result for average precision is present in Table I. The average precision for segment per category is 48.31% for walking and 72.38% for periods when no human activity was detected. The test dataset results are in Table II with 99.96%, 81.48%, and 63.00% for IoU with 50 percent, 75 percent, and 50 - 95 percent area overlap. For the segment category evaluation, walk average activity precision is 55.372% and 70.63% for no activity instances in the test dataset.



(a) Training loss of walk-wave-run activities (b) Accuracy of walk-wave-run activities

Fig. 5: Evaluation results of multiple activities

B. Accuracy of multiple activities

We investigated the performance of our model when different activities are interleaved in a stream. We combine three different activities for the evaluation, namely running, hand wave, and walking. The training dataset comprises 108 instances of multiple human activities and periods with no human activity. Likewise, the validation and test datasets both

TABLE V: Comparison of real-time and non-real-time models.

Model Accuracy				
Activity	M-RCNN	Seg. Mask M-RCNN	d-CNN	iCNN
walk	0.929	0.895	100	100
run	0.948	0.913	99.0	100
walk-wave-run	0.937	0.914	95.8	99.4
Average	0.938	0.907	99.8	98.3

have 22 instances of different activities. The output of the training loss and accuracy is shown in Figure 5. Similarly, the performance in terms of average precision and mAP is shown in Table IV. After training, the overall performance is 96.94% for average precision for IoU for validation proposals with 0.5 area overlap and 62.99% for proposals with 0.75 area overlap. For a proposal in the range of 0.5 - 0.95 IoU, the average precision is 58.05%. The average precision for each category is 59.90%, 61.34%, and 47.34% for hand waves, walking, and running, respectively. The average precision when there is no activity is 63.60%. The trained model is evaluated using the test dataset, and the result is 93.81%, 83.00%, and 64.67% for an average of 0.5 IoU, 0.75 IoU, and 0.5 - 0.95 IoU, respectively. The metrics for each category in the test data are 73.37% for hand wave, 53.27% for running, 62.77% for walking, and 69.25% when none of the activities is detected.

C. Comparison with non-real-time model

The performance of our model is compared with a non-real-time deep learning classification model [17] that was also trained with images after offline activity extractions and preprocessing. The comparison with our model is shown in Table V. The results show a slightly lower accuracy for the object detection real-time model than the non-real-time (i-CNN) model, with a 0.076 and 0.055 decrease in accuracy for walking and running, respectively. The tradeoff in accuracy between non-real-time and real-time detection is a 0.061 reduction for hand waves.

VI. Conclusion

Many existing works have considered the performance of human activity recognition with CSI in traditional lab settings and produced models that cannot be directly used to recognize activities in real-time settings. To address this challenge, we proposed a CSI-based object detection model for recognizing human activity in real-time. Our proposed network achieved desirable performance with real-time data streams containing single and multiple activities. We consider our model as a possible solution to systems requiring real-time contact-free and sensorless human activity recognition. In future work, we would like to investigate the influence of the sliding window overlap on training performance and how changing the backbone of the object detection network affects the real-time performance of the human activity recognition model.

References

[1] T. Nguyen and N. Memon, "Tap-based user authentication for smartwatches," *Computers & Security*, vol. 78, pp. 174–186, 2018.

[2] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, 2019.

[3] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wi-fi," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1342–1355, 2018.

[4] I. Elujide, C. Feng, A. Shiran, J. Li, and Y. Liu, "Location independent gesture recognition using channel state information," in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*. IEEE, 2022, pp. 841–846.

[5] D. Zhang, H. Wang, Y. Wang, and J. Ma, "Anti-fall: A non-intrusive and real-time fall detector leveraging csi from commodity wifi devices," in *International Conference on Smart Homes and Health Telematics*. Springer, 2015, pp. 181–193.

[6] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: A real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2016.

[7] R. M. Keenan and L.-N. Tran, "Fall detection using wi-fi signals and threshold-based activity segmentation," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–6.

[8] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2014.

[9] B. Tan, K. Woodbridge, and K. Chetty, "Awireless passive radar system for real-time through-wall movement detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2596–2603, 2016.

[10] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.

[11] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE transactions on information theory*, vol. 38, no. 2, pp. 569–586, 1992.

[12] J. Sadowsky, "Investigation of signal characteristics using the continuous wavelet transform," *johns hopkins apl technical digest*, vol. 17, no. 3, pp. 258–269, 1996.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[14] S. Arshad, C. Feng, I. Elujide, S. Zhou, and Y. Liu, "Safedrive-fi: A multimodal and device free dangerous driving recognition system using wifi," in *IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[15] I. Elujide and Y. Liu, "An entropy-based wlan channel allocation using channel state information," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)(50308)*. IEEE, 2020, pp. 74–79.

[16] T. Nakamura, M. Bouazizi, K. Yamamoto, and T. Ohtsuki, "Wi-fi-csi-based fall detection by spectrogram analysis with cnn," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[17] S. Arshad, C. Feng, R. Yu, and Y. Liu, "Leveraging transfer learning in multiple human activity recognition using wifi signal," in *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2019, pp. 1–10.

[18] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3209–3221, 2017.

[19] X. Liu, H. Chen, X. Jiang, J. Qian, G. Aceto, and A. Pescape, "Wi-cr: Human action counting and recognition with wi-fi signals," in *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. IEEE, 2019, pp. 1–8.

[20] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1118–1131, 2017.

[21] G. Pan, Y. Zheng, R. Zhang, Z. Han, D. Sun, and X. Qu, "A bottom-up summarization algorithm for videos in the wild," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, pp. 1–11, 2019.

[22] I. Daubechies, *The wavelet transform, time-frequency localization and signal analysis*. Princeton University Press, 2009.