Deep Learning Driven Wireless Real-time Human Activity Recognition

Hanqing Guo, Nan Zhang, Shaoen Wu
Ball State University
{hguo, nzhang, swu}@bsu.edu

Qing Yang
North Texas University
qing.yang@unt.edu

***Abstract*—Human activity recognition based on wireless sensing is advantageous at various features such as privacy preservation, but also very challenging due to the instability of wireless signals. This paper proposes a deep learning driven wireless human activity recognition solution based on Multiple-Input-Multiple-Output (MIMO) radar sensing. User activities are first sensed by a low-power Frequency-Modulated Continuous Wave (FMCW) MIMO radar. Then a sequence of 3D images are generated out of the reflected signal strength. Next, deep neural networks (DNNs) are designed to analyze the correlation among the sequential 3D images to recognize various types of human activities. This work has developed: 1) a large dataset containing over 1,500 training videos of six different types of indoor activities, 2) a customized deep learning video data-loader to select proper training data in each training epoch, 3) a deep recurrent neural network (RNN) model to recognize human activities based on radar imaging results. This solution has been extensively evaluated in a research lab room. The results show that the solution is able to generate wireless imaging frame-by-frame, and it can achieve over 86.7% accuracy in recognizing six different types of human activities.**

I. Introduction

Human activity recognition (HAR) plays a key role in many smart systems, especially in smart health. For example, an effective system is able to alert family members when young and elderly people are in adversary conditions. Conventional HAR systems are either based video cameras or wearable devices, while camera-based solutions raise privacy issues and wearable devices are inconvenient in daily use. Hence, it is important to design an HAR system that is privacy friendly while requiring no users' attention. Therefore, an HAR system embedded and quietly working in the background environment is favored.

In this work, we propose a robust deep-learning based wireless HAR system with two main modules. The first module constructs 3D activity images from raw wireless signals by using an MIMO FMCW radar. The second module recognizes activities with a deep learning model working on the 3D activity images. The highlights of this work are as follows. **First**, the radar device is out of commodity at very low power. **Second**, the captured signal image is at very high resolution in that the radar operates from 3.3GHz to 10GHz. **Third**, a Convolution Neural Network (CNN) is designed to extract features from the 3D continuous signal images. **Forth**, a Recurrent Neural Network (RNN) is designed to exploit the temporal features in the sequence of 3D activity signal images for the HAR recognition.

The rest of the paper is as follows. Section II briefly reviews previous HAR solutions. Then Section III discusses our proposed deep learning driven MIMO radar based HAR system, including the system architecture, radar wireless imaging and the deep learning model designed for activity recognition. The performance evaluation is presented in Section IV. Section V concludes this work and hints the future work.

II. Related Work

The related works in HAR include camera based solutions, more recent RGB-D camera based solutions, wearable sensor based solutions and RF based solutions. All of those solutions will be investigated and discussed in this section.

A. Camera Based Solutions

Most traditional camera-based HAR solutions exploited trajectory features [1], [2] such as dense trajectories (DT) and improved dense trajectories (iDT). The recent years have seen many camera-based HAR solutions based on deep convolution networks and recurrent networks [3]–[8]. More recently, a two-stream (spatial stream and temporal stream) convolution network solution was proposed to improve action detection accuracy [3]. Another work adopted long-short term memory (LSTM) network and achieved an 88.6% accuracy on UCF-101 datasets [6]. In 2017, the structured segment network (SSN) was proposed to detect each action instance via a structured temporal pyramid, and was evaluated on untrimmed videos [4]. This novel design allows to not only recognize actions, but also localize the start and end frames of the action, which outperforms previous methods on THUMOS14 [9].

B. Sensor Based Solutions

Wearable sensor based solutions typically use accelerator and Gyroscope [10] to collect raw data, then analyze these data to recognize human motions and activities. Many of such solutions use smart phones as the data collection tool [11]–[13]. Inertial measurement unit (IMU) sensor that combines accelerometer and gyroscope sensor has also been widely used in wearable devices [14]. Those sensors can collect linear acceleration, rotation angle, the angular velocity of targets wearing these sensors. The sensors' raw data are then processed with various algorithms to recognize human

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

activities [15]–[17]. Recently some solutions have attempted to combine camera-based and wearable sensor-based data for improved performance in accuracy as well as efficiency [18].

C. RF Based Solutions

Radar sensors and RF devices have been recently exploited for smart home applications using either FMCW radar [19]–[22] or off-the-shelf devices [23]–[25]. A research group at MIT designed an FMCW MIMO radar sensing testbed to detect human motions [19] and it can even capture human figures through a wall [20], [21]. Off-the-shelf devices such as ultrasonic sensor or walabot [26] have been also used to recognize human activities [23]–[25]. Avrahami et al. proposed a human activity recognition scheme based on 2D heat maps generated by walabot, while Zhu et al. [24] used traditional signal processing algorithms with clustering machine learning scheme to recognize human actions. Both of them claim an accuracy over 80%.

III. Deep Learning Driven MIMO Radar Based HAR

In this work, we propose a deep learning driven MIMO radar based HAR solution, which consists of two modules: the first one capturing and constructing users' activity images and the second one recognizing the activities. The entire process is performed in five major steps: 1) establishing a labeled 3D wireless radar imaging dataset for training, 2) extracting high-level features of each 3D radar imaging frame with a Deep Neural Network (DNN), 3) learning temporal activity features of the labeled training data with a LSTM network, 4) fine-turning the learning procedure for each specific activity, and 5) recognizing a real-time activity based its 3D radar imaging sequence.

A. System Architecture

As shown in Figure 1, the radar sensed raw wireless signals of activity was first converted into 3D wireless image frames, and a DNN is used as the filter to get rid of non-continuous frames, which is shown in the purple block. Next, the activity wireless imaging clips are passed into a CNN-RNN network model, where the CNN is used to extract high-dimensional features of a frame, and RNN is trained to recognize the activity based on the given feature sequence from the CNN, which is illustrated in blue-green. The trained CNN-RNN is then used with a sliding window technique to recognize an activity based on the radar sensed raw signals.

B. MIMO Radar Imaging

To enable real-time activity imaging with an ambient radar, we design a framework, called *Human Image Capturing based on FMCW Radar* (HICFR), to scan the surroundings with FMCW chirps and an MIMO antenna array. HICFR builds a 3D coordinate system as shown in Figure 2(a). While FMCW chirps are used to compute the direct distance between an object and a receive antenna, the 2D antenna array can identify spatial directions. It emits parallel FMCW chirps from multiple antennas to scan the 3D volume of surroundings.

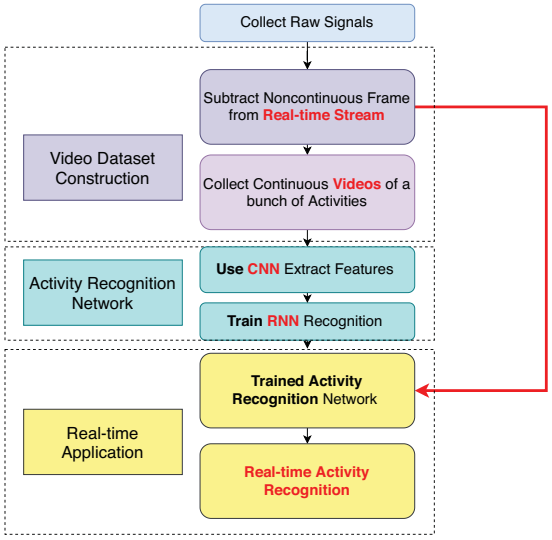
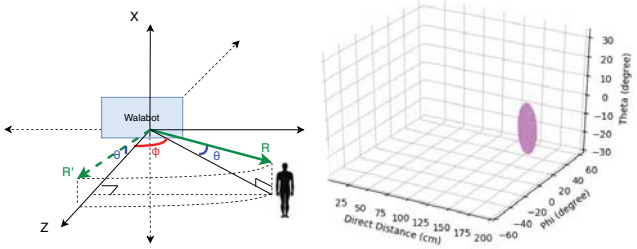


Fig. 1: System Architecture

HICFR converts a signal amplitude value to a color depth. Consequentially, a 3D image is constructed from the scan of the 3D volume of the surroundings with the MIMO FMCW radar. Due to the instability and unreliability of wireless signals in practice, it is likely that some of a sequence of scans will be missed or corrupted, which results in missed 3D wireless imaging frames during an activity. To address this problem, a Deep Neural Network (DNN) interpolator is designed to frames to make up those frames for a continuous sequence of activity wireless images.

In Figure 2(a), θ is the elevation angle to detect the height of a target, and ϕ is width angle to capture the width of the a target. R is the FMCW signal travel distance from a transmission antenna to a target head, and R' is the hypotenuse of a triangle whose angle is θ when R rotates a degree of ϕ. The scan range is the sector where the triangle passes. In our case, θ is from $-30°$ to $30°$ and ϕ is from $-60°$ to $60°$. The direct travel distance R can be calculated with FMCW in formula (1) as below:

$$R = \frac{c|\triangle t|}{2} = \frac{c|\triangle f|}{2(df/dt)} \tag{1}$$



(a) 3D Coordinate (b) Signal Image

Fig. 2: 3D MIMO Radar Imaging

Figure 2 shows a generated wireless image, where it can be clearly observed that an object with purple color has direct distance R as $150cm$ to the radar, with a width angle ϕ being $40°$, elevation angle θ being from $-30°$ to $-5°$.

C. Deep Learning Driven Activity Recognition

The recognition module of this work is called **D***eep Learning Driven Human* **A***ctivity* **R***ecognition based on MIMO Radar* **I***maging* (DARI), which recognizes human activity in six categories: *fall*, *stand up still*, *walk*, *sit-to-stand* and *stand-to-sit*. It takes a sequence of continuous 3D MIMO radar images as videos for its input. For every 3D frame of an input wireless imaging video, CNN is designed to extract the spatial features. Then the extracted features of each frame are used as input to an RNN, particularly LSTM, to further extract the temporal correlation among the frames of an activity. The output of the last RNN cell gives the confidence score of each type of activity. The difference or loss between the predicted confidence with the ground-truth label is then backpropagated to update the LSTM parameters in training.

1) Video Dataset Construction: Video dataset is constructed in two steps: 1) collecting continuous videos for activities; 2) segmenting the frames and labeling them.

In the very beginning, our system generates real-time stacked 3D MIMO radar images that are then passed through the DNN interpolator as described in Section III-B above. To gather enough training data, the data collection runs for a long time with a variety of activities performed. Then, the videos are manually segmented and cropped to make sure every activity is associated with a fixed number of frames, which are labeled with the corresponding activity category. Note that an activity may be labeled with more than one categories, e.g. the "fall" activity has two category labels: fall and stand-up. At the end, six categories have been labeled: sit-to-stand, stand-to-sit, still, walk, fall, and stand-up. Each category has many video clips of 3D radar frames which represent the corresponding activity. The reason for cropping and segmentation into a fixed number of frames is to meet the input requirement of the deep learning activity recognition model as discussed as follows.

2) Deep Learning Activity Recognition: The deep learning activity recognition comprises two deep learning networks: 1) a Convolution Neural Network (CNN) and 2) an Recurrent Neural Network (RNN). While the CNN extracts the spatial features of a 3D wireless image, the RNN analyzes the temporal features among those 3D wireless frames. The RNN infers the activity category.

In our work, the CNN is a feature extractor. It contains input, convolution, pooling, and fully connected layers, but not the output layer as in a classifier. Given a 3D radar image frame, our CNN extracts and builds the feature map, which is the input to the RNN recognition module.

We denote fp the final feature map, and $fp[t]$ the feature map of the 3D radar image at moment t. We also introduce K as the number of frames. Equation 2 shows the detail of the RNN. When given input $(fp[t-K], fp[t-K+1], ..., fp[t])$, RNN produces $y[t]$ as the human activity at moment t. Each frame of an activity has almost equal contribution to the output. The RNN module exploits the sequential (temporal) correlation among the given frames of an activity, which means it predicts upon the order of input.

Refer to Equation 2 where the output of the k_{th} unit is denoted as h_k. There are three kinds of weight matrices, W_{xh}, W_{hh} and W_{hy} respectively. While W_{xh} is associated with the input fp, W_{hh} relates to output of the recurrent unit h_k, and W_{hy} is the key factor of the output sample $y[t]$. In the middle of two recurrent units, an activation function σ_h enables the non-linearation between the previous output and the present input. The other activation function, denoted as σ_y, usually a $tanh$ function, regulates the output falling between $+1$ to -1. To describe the forward propagation of the RNN model, we denote $fp[t-K]$ and h_0 as the initial input. In the beginning, $fp[t-K]$ is fed to the first recurrent unit, and then multiplies its weight matrix W_{xh}. Meanwhile, the weight matrix W_{hh} combines with h_0 and forms the initial previous output. The sum of $W_{hh}h_0$, $W_{xh}fp[t-K+1]$ and bias b_h passes through the activation function σ_h and produces h_1. This process repeats at every recurrent unit and finally ends at the K_{th} layer, where $y[t]$ is computed with h_K, W_{hy}, b_y and the activation function σ_y. The output $y[t]$ is a vector that contains the probability of every activity at moment t.

$$\begin{aligned} h_1 &= \sigma_h(W_{hh}h_0 + W_{xh}fp[t-K+1] + b_h) \\ h_k &= \sigma_h(W_{hh}h_{k-1} + W_{xh}fp[t-K+k] + b_h) \\ y[t] &= \sigma_y(W_{hy}h_k + b_y) \end{aligned} \tag{2}$$

D. Sliding Window Enhancement

Because of the wireless uncertainties, the radar image sequence likely contains outliers and exceptions. To reduce the impact of such abnormalities, we design a real-time sliding window scheme to smooth out the abnormalities for improved recognition performance. This enhancement is based on our trained RNN model. As shown in Figure 3, suppose our trained RNN takes four frames as input at one time, which means K in equation 2 is equal to 4. As discussed earlier, the output of the HRCIF system is a sequence of continuous 3D radar frames generated in real-time. To predict the current activity, we need to retrieve previous frames and put them together as the input of RNN. For example, to predict the human activity at moment $t = 4$, we need the previous three frames, which represent by the blue dotted block in Figure 3, which is also considered in the RNN module. The output of this block $y[4]$ is the vector contains the probabilities of all activities. With time passing, the block dotted block slides one frame to the next to generate the next activity probability vector.

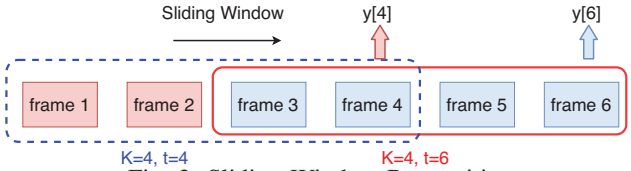


Fig. 3: Sliding Window Recognition

IV. Performance Evaluation

A. Evaluation Platform and Settings

We reuse the radar platform in our previous work [27] that introduced the whole process of generating radar images stream, which is an off-the-shelf radar sensor called

Walabot [26]. It is compact and low-cost with a board size of $72mm * 140mm$. The average power is lower than -41dbm/MHz. The frequency range of FMCW chirp emitted by Walabot is 3.3GHz-10GHz, which is capable to detect a distance of 10 meters.

Referring to Section III-C, there are three main steps in the DARI solution: 1) constructing dataset, 2) training the CNN-RNN framework, and 3) using the CNN-RNN model for recognition. The performance evaluation has been conducted over these steps.

1) Video Dataset Construction: For each activity type, we collect more than 20 radar videos, each of which is longer than ten minutes. Since the HICFR system processes one frame every second, the recorded video has a frame rate as fps, which means we have overall $20 * 10 * 60 = 12000$ frames in total. Since those frames are recorded in real-time, one video contains different activities. For example, when recording the activity "fall", the user also walks before falling down. However, the RNN model demands that a sequence of frames only correspond to exactly one activity. Hence, we manually cut and crop our original videos into the fixed number of frames with only one ground truth label. In our case, we tested the video size (number of frames) of 3, 5 and 8. For each video size, we collect 93 samples for fall, 79 samples for stand-up, 93 samples for sit-to-stand, 82 samples for stand-to-sit, 96 samples for walk and 81 samples for still. The total training data set contains $3 * sum(samples) = 1572$ activity samples, for the six different categories with three video sizes.

2) Activity Recognition Network: Section III-C2 introduces a CNN-RNN framework to achieve activity recognition. Figure 4 shows the main flow graph of the DARI system. It can be clearly observed that the target gradually falls from the bottom row in the figure, which is used as ground truth. The second row from the bottom representing the output of HICFR is our training samples for the "fall" activity, with a sample size 5, which means the K mentioned in Section III-C2 is equal to five. From the first radar image to the last one, the frames record the position changes of the target during the "fall" activity. The first frame shows that the target stands with the angle of $60°$ and the direct distance $150cm$. The purple volume going down along with the wide angle because the target moves to the centering direction of the radar from frame #2 to frame #3. With falling onto the ground, the last two frames show that the detected target position in elevation angle is very low, and that the visualization indicates the target falling to the ground. The RNN model enables the system to understand what activity is shown in the five frames. Before this RNN model, a CNN model, a revised ResNet18, is used to extract the spatial features of those frames: the last fully connected layer is revised to generate 64 features and the softmax layer is removed. Then we take the 64 features of every frame as the feature map (fp) to feed into the RNN model.

Training strategy: The training strategy for the DARI system is unique since it contains two types of neural network. We use a pre-trained resNet18 to extract features, and this period does not update any parameters of resNet18. In this way, it is guaranteed that the CNN extracts exactly the same features from the same input frame during the whole training process, which means whatever is the difference between prediction and label, it does not change the original weights, kernels and bias parameters of the CNN model. In other words, the training process is not end-to-end. We keep the CNN parameters and only train the RNN model. The backpropagation function also only updates the parameters of the RNN model.

Video loader: In the training stage, the training dataset needs to shuffle data to reduce the variance and making sure that models remain general and not overfit. The shuffling process demands that dataset is loaded in a random order. In the beginning of each iteration, we create a list of dictionary named l_1 to store all training videos, with the key being the video name and the value is the label. Then we randomly create a list l_2 with the same length of l_1, but the value in each cell represents the order of video index. Then we can generate l_3 that is shuffled from l_1 and l_2. For example, $l_1 = [\{"walk\_1" : "walk"\}, \{"walk\_2" : "walk"\}, \{"fall\_1" : "fall"\}, \{"still\_1" : "still"\}]$ $l_2 = [3, 0, 1, 2]$ $l_3 = [\{"walk\_2" : "walk"\}, \{"fall\_1" : "fall"\}, \{"still\_1" : "still"\}, \{"walk\_1" : "walk"\}]$. The video is loaded in the order shown in l_3.

Hypo-parameters	Settings
CNN body	ResNet18
CNN pooling	Maxpooling
CNN last-layer	Fully Connected (512, 64)
RNN hidden size	100
RNN layers	1
RNN optimizer	SGD optimizer
RNN learning rate	0.01
RNN momentum	0.9
RNN lr scheduler	step size = 30, gamma=0.4

TABLE I: CNN and RNN Configurations

Training hypo-parameters: The configurations of the CNN and RNN models are shown in Table I. The hypo-parameters include: learning rate α, momentum β and initial h_0. We adopt the SGD optimizer to update parameters with $learning\ rate = 0.01$ and $momentum = 0.9$, and a $lr\ schedular$ is used to adjust the learning rate with $stepsize = 30$ and $gamma = 0.4$. The loss function is as in Equation 3, where x is the output of RNN, the dimension of x is (1, number of activity types), and $label$ is the label of the current activity. Thus $x[j]$ means the confidence with which our system recognizes the given video as j_{th} activity, and $x[label]$ is the confidence with which the recognition is correct. We trained the system with 1572 videos for 100 iterations on the platform with two GTX 1080 Ti GPUs with cuda acceleration.

$$loss(x, label) = -log(\frac{e^{x[label]}}{\sum_j e^{x[j]}}) \quad (3)$$

B. Result Analysis

1) Accuracy: We trained three RNNs with the input video size of $3, 5, 8$ respectively. Table II shows the training accuracy of these three RNN models. As can be observed, the RNN with the video size 3 has the worst prediction accuracy: less than 60%. The reason is that video size 3 is too short for the

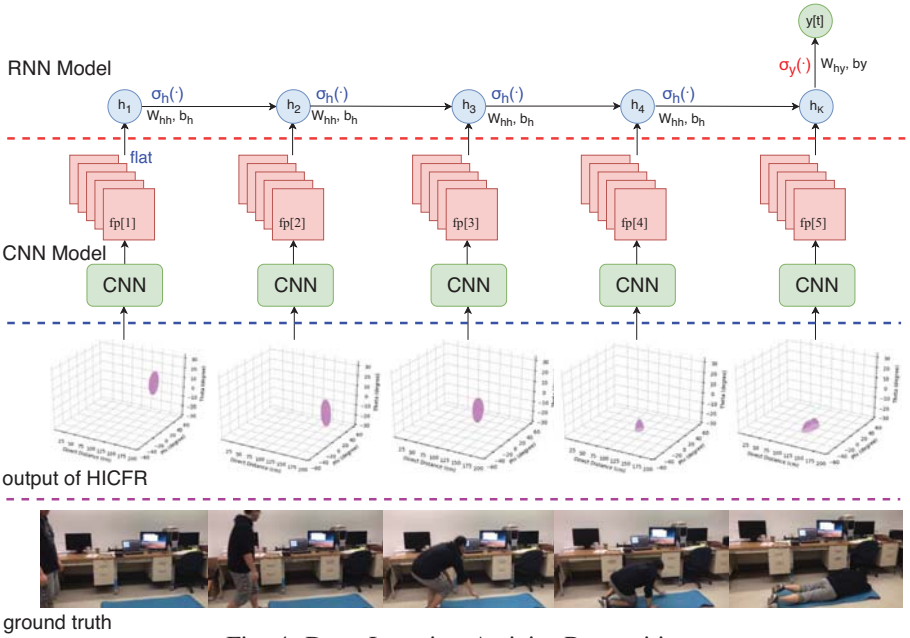


Fig. 4: Deep Learning Activity Recognition

video to contain all information about an activity. From human perspective, we need at least 3 continuous frames which indicate human stay on the ground to determine that is a fall activity. Overall, the RNN model with video size 5 performs best with 86.7% accuracy. The RNN with video size 8it has the best performance for fall and stand-up detection. However, the larger the video size more likely contains information of more than one activity, which will hurt the recognition of the short activities, e.g, sit-to-stand, and stand-to-sit.

Activity	frame length=3	frame length=5	frame length=8
fall	33.3%	89.2%	91.4%
stand-up	46.8%	91.1%	97.4%
sit-to-stand	40.8%	78.4%	38.8%
stand-to-sit	58.5%	80.4%	42.7%
walk	45.8%	82.3%	45.8%
still	96.2%	100%	100%
overall	52.3%	**86.7%**	68.5%

TABLE II: Recognition Accuracy

2) Confusion Matrix and Convergence: We also compare the convergence of the prediction and the ground truth, then generate the confusion matrix as shown in Figure 5(a)(b)(c). The confusion matrix shows the number of correct recognitions and wrong recognitions. For example, the first row and first column of Figure 5(a) is 31, which means that there are 31 samples of "fall" recognized correctly as "fall". The first row and second column of Figure 5(a) is 9, meaning that there are 9 samples of "fall" incorrectly recognized as stand-up. It turns out the RNN with $frame_length = 3$ is likely to misunderstand "fall" and stand-to-sit, stand-up and sit-to-stand, walk and stand-to-sit. The RNN with $frame_length = 8$ performs well with first two categories, but not for some quick movement and activities e.g., sit-to-stand, stand-to-sit and walk.

The RNN with $frame_length = 3$ has problem in converge for one hundred epochs. The RNN with $frame_length = 8$ finally converges at very low loss after wide fluctuations. The RNN with $frame_length = 5$ converges very quickly and stays stable during the rest of epochs.

V. Conclusion

This paper proposes a deep learning driven wireless human activity recognition solution based on Multiple-Input-Multiple-Output (MIMO) radar sensing. User activities are first sensed by a low-power Frequency-Modulated Continuous Wave (FMCW) MIMO radar array. Then 3-dimension images are generated out of the radar signals. Next, deep neural networks are designed to analyze the correlation among the sequential images and consequently recognize various types of human activities. This solution has been extensively evaluated in a research lab. The result shows the high performance of the solution in recognizing six different types of human activities.

VI. Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. #1923712 and #1726017. The radar equipment is partly sponsored by Vayyar at an education discounted price.

References

[1] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *CVPR 2011-IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176. IEEE, 2011.

[2] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[3] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[4] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017.

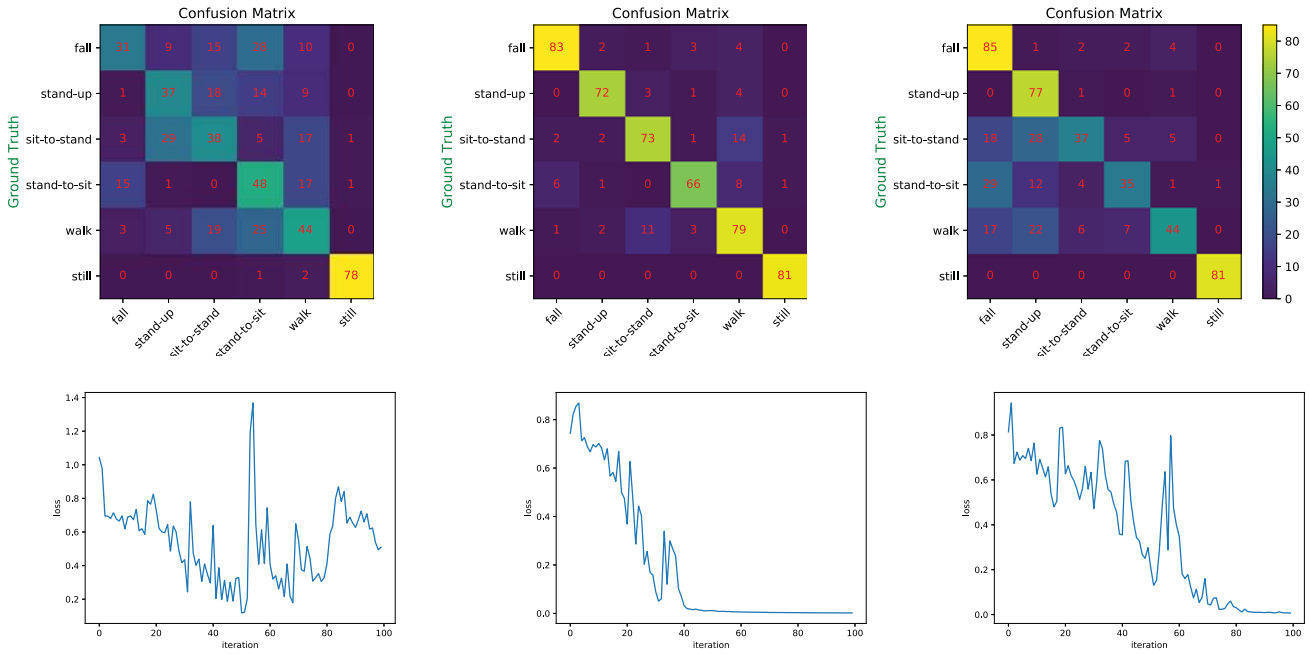


Fig. 5: Accuracy vs Loss Convergence with frame length 3, 5, 8 from left to right

[5] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

[6] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[7] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[9] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[10] Subhas Chandra Mukhopadhyay. Wearable sensors for human activity monitoring: A review. *IEEE sensors journal*, 15(3):1321–1330, 2015.

[11] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[12] Mohammed Mehedi Hassan, Md Zia Uddin, Amr Mohamed, and Ahmad Almogren. A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81:307–313, 2018.

[13] Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018.

[14] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M Khairi, and Vijayabaskar Kasi. Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, 1(2):256–262, 2013.

[15] Mi Zhang and Alexander A Sawchuk. A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *Proceedings of the 6th International Conference on Body Area Networks*, pages 92–98. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[16] Mi Zhang and Alexander A Sawchuk. Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 631–640. ACM, 2012.

[17] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016.

[18] Jun Ye, Guojun Qi, Naifan Zhuang, Hao Hu, and Kien A Hua. Learning compact features for human activity recognition via probabilistic first-take-all. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[19] Fadel Adib and Dina Katabi. *See through walls with WiFi!*, volume 43. ACM, 2013.

[20] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C Miller. 3d tracking via body radio reflections. In *NSDI*, volume 14, pages 317–329, 2014.

[21] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. Capturing the human figure through a wall. *ACM Transactions on Graphics (TOG)*, 34(6):219, 2015.

[22] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[23] Shangyue Zhu, Hanqing Guo, Junhong Xu, and Shaoen Wu. Distance based user localization and tracking with mechanical ultrasonic beamforming. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 827–831. IEEE, 2018.

[24] Shangyue Zhu, Junhong Xu, Hanqing Guo, Qiwei Liu, Shaoen Wu, and Honggang Wang. Indoor human activity recognition based on ambient radar with signal processing and machine learning. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[25] Daniel Avrahami, Mitesh Patel, Yusuke Yamaura, and Sven Kratz. Below the surface: Unobtrusive activity recognition for work surfaces using rf-radar sensing. In *23rd International Conference on Intelligent User Interfaces*, pages 439–451. ACM, 2018.

[26] Walabot. https://walabot.com/.

[27] Hangqing Guo, Nan Zhang, Wenjun Shi, ALI-AlQarni Saeed, Shaoen Wu, and Honggang Wang. Real-time indoor 3d human imaging based on mimo radar sensing. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1408–1413. IEEE, 2019.