mmFiT: Contactless Fitness Tracker Using mmWave Radar and Edge Computing Enabled Deep Learning

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

06-09-2021 / 10-09-2021

CITATION

Tiwari, Girish; Bajaj, Parveen; Gupta, Shalabh (2021): mmFiT: Contactless Fitness Tracker Using mmWave Radar and Edge Computing Enabled Deep Learning. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.16574588.v1>

DOI

[10.36227/techrxiv.16574588.v1](https://doi.org/10.36227/techrxiv.16574588.v1)

mmFiT: Contactless Fitness Tracker Using mmWave Radar and Edge Computing Enabled Deep Learning

Girish Tiwari^{id}, *Graduate Student Member, IEEE*, Parveen Bajaj and Shalabh Gupta^{id}

Abstract—Internet of things (IoT) is transforming the way we imagine healthcare with ubiquitous connectivity, faster response and deeper personalized insights using large amounts of data. Fitness trackers provide useful insights to maintain balance of a healthy lifestyle. Nowadays, fitness trackers are available as wearable devices which creates a sense of unease in exercise and may cause skin irritation. In this paper, we present *mmFiT*, an edge computing enabled, contactless, real-time fitness tracker using a single mmwave radar point cloud data. It has the inherent advantage of user privacy preservation while tracking indoor fitness activities. Experimental results show that the system can classify various exercises with real-time accuracy of 95.53% and is also capable of counting repetitions of exercises. This implementation is computationally inexpensive, and therefore, the system can be deployed in an IoT connected edge device for real-time operations. This system will be an ideal fit in a smart home or smart gymnasium setting.

Index Terms—mmwave radar, point-cloud, deep learning, CNN, IWR1642, non contact, real-time, privacy preserving, contactless, human activity recognition, fitness tracker, exercise, edge computing.

I. INTRODUCTION

CONNECTED healthcare is opening up new avenues for people's health management systems. IoT provides unprecedented flexibility in achieving the goal of connected healthcare. It provides the required framework for the connectivity and data exchange capability to the devices. An IoT device in a crude sense is a source of data. The data processing usually happens at a central place. As the number of IoT devices increase, this approach will require huge computation power, big data centers, increased infrastructure cost, and high network bandwidth apart from increased delay. This is where a new computing paradigm, edge computing, comes to rescue. Edge computing is the computing which takes place near the device or data source. Edge devices have enough processing power and memory to collect, process, and make real-time decisions based on the processed data with little to no help from other parts of the network [1]. Artificial intelligence (AI) is growing at a rapid pace. Concurrently, advances in hardware technology have facilitated low cost, compact, and powerful processing devices bringing artificial intelligence in the realm of edge computing. The convergence of AI and edge computing with IoT is called Edge Intelligence [2] or Artificial Intelligence of Things (AIoT) [3]. Deng et. al. differentiates

edge intelligence into 'AI for edge' and 'AI on edge' [2]. The AI on edge aims to provide insights and decisions from massive data available at edge while delivering advantage of performance, low-cost, privacy, reliability, and efficiency. This work is based on the AI on edge paradigm.

Commercialization of single-chip mmwave radar sensors in the last decade has created opportunities in various fields such as healthcare [4], high resolution radar imaging [5], human activity recognition [6], and many more. Camera sensors can also be used for accurate human activity recognition [7]. However, user privacy remains a grave concern apart from issues such as poor lighting and other environmental factors. One of the biggest advantages of using mmwave radar sensors in a smart home setting is user privacy preservation while providing feature rich data [8]-[9]. Radar micro-doppler spectrograms have been extensively used for human activity classification [10]-[13]. Most state of the art approaches require raw radar data for spectrogram generation. These methods are not suited for real-time operations on resource limited edge devices, as spectrogram generation is computationally expensive and raw data collection requires high data transfer speed [14]. Moreover, using only micro-doppler spectrograms may not be an effective approach for classification of symmetric activities since the spectrograms will be similar. On the other hand, processed radar point cloud has lower data size and can be acquired in real-time. However, sparsity of point cloud creates additional challenges for accurate classification.

A healthy lifestyle keeps one physically as well as mentally fit to enjoy all aspects of life. Daily exercising is one of the important foundations of a healthy lifestyle. Keeping track of the daily exercise routine has become a necessity in this connected and fast world. Fitness trackers nowadays are widely available in the market as various types of wearables [15]-[16]. However, performing intense workout while wearing these gadgets may give a sense of unease and cause skin irritation. Tiwari et. al. proved the concept of using a single mmwave radar sensor for realizing a real-time contactless fitness tracker using deep convolutional neural networks [17]. They achieved accuracy of 99.85% using grayscale feature plots. However, this accuracy was achieved while putting the user in a predefined position. The accuracy of the system was not satisfactory when user moved from one position to another and the CNN with range-doppler image as input could not differentiate in symmetrical activities.

In this paper, we present *mmFiT*, a user privacy preserving, contactless, highly accurate, real-time fitness tracker system for indoor fitness activities, capable of running on edge devices for applications in IoT connected healthcare. We propose

Manuscript received. This work was supported by the Indian Institute of Technology Bombay, Mumbai, India.

Girish Tiwari and Shalabh Gupta are with the Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai, India (e-mail: girishtiwari@ieee.org)

Parveen Bajaj is an independent researcher from Hyderabad, India.

Fig. 1. Block diagram of the system. The mmwave radar performs onboard radar signal processing and gives point cloud data. This point cloud data is used to generate feature plots for classification and repetition counting of exercises.

simple yet effective following approaches to improve the classification accuracy of CNNs: (i) Use of color feature plots with intensity vectors for improving classification of range-doppler image CNN for symmetrical activities. (ii) Data transformation technique using translation of range-angle values and angle normalization, which eliminates the requirement of additional training data and re-training the CNN for position variation of the user. We propose methods to extract repetition counting of exercises from the sparse point cloud data. Based on this data, we can estimate the calorie burnout of users in a workout session. We also propose an algorithm for real-time implementation of the fitness tracker for activity classification and exercise repetition counting. This system can be deployed in a smart home or smart gymnasium where the user can perform exercises freely without the need of attaching any wearable sensor for workout tracking. Using the IoT connection, the user can be informed about details of his/her workout session in real-time.

This paper is structured as follows: Section II presents the system design of the fitness tracker followed by brief description of the dataset and radar point cloud processing. Section III presents analysis of various deep learning classifiers for fitness activity classification. In section IV, we analyse the effect of position variation of users on classifiers and propose a data transformation technique to improve accuracy. Section V illustrates the process of extracting repetition count of exercises from the radar point-cloud data. Section VI presents the analysis and results of real-time performance of the system followed by memory and computational requirement analysis of CNN for deployment on edge devices.

II. SYSTEM DESIGN

To realise an edge computing enabled, privacy preserving, real-time, contactless fitness tracker, we use the feature-rich point cloud data output of a mmwave radar sensor. Fig. 1 shows the block diagram of the entire system chain. The user performs exercises in front of the radar. The radar collects the reflected returns and performs onboard radar signal processing i.e. FFT, CFAR, and angle of arrival estimation to generate point cloud data output in real-time. This radar data is collected on a computer using serial interface and processed. This processed data is then used to generate feature plots for the classifier. We use a dataset of seven fitness activities to train and test the classifiers. Once the classification of the data is performed, the data is sent for repetition counting. After the workout, the system generates a session report which shows the duration and repetitions of each exercise. This data can be used to estimate the calorie burnout of the user during the workout. Calorie burn in an activity can be calculated as [18]:

Fig. 2. Setup for data collection using IWR1642 mmwave radar.

TABLE I
DETAILS OF THE COLLECTED DATASET. EACH BINARY SAMPLE IS OF 10 SECONDS DURATION.

Exercise	No. of Samples	Class Label Value
Dumbbell	442	1
Shoulder press	442	2
Squat	442	3
Lateral raise	440	4
Boxing	443	5
Right triceps	442	6
Left triceps	443	7
Total	3094	

$$\text{Calorie burn/minute} = \frac{\text{MET} \times 3.5 \times \text{Body weight}(Kg)}{200}$$

Here, MET is metabolic equivalent of task, which is the ratio of metabolic rate of the body in an activity compared to when the body is at rest.

A. Dataset Description

A dataset of total seven fitness activities namely dumbbell, shoulder press, squat, lateral raise, boxing, right arm tricep, and left arm tricep were collected. Block diagram for data collection is shown in Fig. 2. An mmwave FMCW radar system IWR1642 by Texas Instruments was used in this experiment. The user performed the exercises approximately 4 meters away from the radar. Static clutter removal algorithm was enabled in the radar for detection of only moving targets. The radar sends detection of point-cloud in real-time using a serial connection to the host computer. This point cloud data can be saved as a binary file to the computer. Two volunteers participated for data collection in 11 sessions. In each session, 20 samples for each of the seven exercises were collected corresponding to each volunteer. The duration of each sample was 10 seconds. The details of the dataset are shown in Table I. Movement of body parts in various exercises are shown in Fig. 3.

Fig. 3. Movement of body parts in various exercises tagged as class labels.

TABLE II
IMAGE DATASET FOR CONVOLUTION NEURAL NETWORKS.

Exercise	No. of Image Samples	Class Label Value
Dumbbell	2652	1
Shoulder press	2652	2
Squat	2652	3
Lateral raise	2640	4
Boxing	2658	5
Right triceps	2652	6
Left triceps	2658	7
Total	18564	

B. Radar Point-Cloud Processing

The mmwave radar system performs 1D FFT, 2D FFT, CFAR, and angle processing using the on-chip DSP. Thereafter, it sends out range, velocity, and angular information for the detected points. This point cloud information is transmitted over the serial interface in the form of packets. This serial interface has data rate limitation ($< 1Mbps$). Each packet comprises multiple types of TLV (Tag-Length-Value) structures. There are 10 types of TLVs for the selected radar board and the required TLVs may be selected while configuring the system parameters. However, choosing all TLVs results in increased data for each frame which may restrict the real-time radar performance due to data rate limitation. We make use of only two TLVs which are: (a) list of detected points, and (b) statistics for timing information. The structure of a saved binary file for a sample is shown in Fig. 4. After processing these TLVs, we can extract range, velocity, angle of arrival, and timing information for each detection point. Although radar frame rate is 10Hz, it sends these TLVs only when radar detects an object. It means that the number of detections in each sample may vary. Additionally, since radar is capable of detecting multiple targets, we may also get multiple detection points at the same instant. Fig. 5 shows point cloud of doppler, range, and angle variations over time for one sample of shoulder press exercise.

III. DEEP LEARNING CLASSIFIERS

Authors in [17] showed that convolutional neural networks cannot differentiate symmetrical movements just by using the grayscale Range-Doppler image (RDI) as feature input. This

Fig. 4. Structure of a collected binary file sample.

Fig. 5. Point cloud variations of doppler, range, and angle values for one sample of shoulder press exercise.

is because the RDI for symmetrical activities will be similar. Joint input of Range-Doppler image (RDI), Range-Angle image (RAI), and Angle-Doppler image (ADI) significantly enhances the CNN performance. However, the accuracies drop when there is position variation of user w.r.t. the radar as RDI, RAI, and ADI are dependent on the position of the user.

To improve the accuracy of RDI CNN, we propose to use color images of Range-Doppler variations and use angle values as intensity vector with the help of [19] in matlab. Range-Doppler variations with angle as intensity for one sample

Fig. 6. Range-Doppler variations of one sample of squat exercise with angle as intensity vector.

TABLE III
COMPARISON OF CNN ACCURACIES WITH [17].

	RDI	RAI	ADI
[17] (Grayscale image)	92.77	99.35	98.54
Present work (Color image with intensity as third variable)	99.8	99.9	99.7

Fig. 7. Framework of the CNN [17].

of squat exercise is shown in Fig. 6. Further, to increase the number of image samples in the dataset, each sample of 10 seconds was used to generate 6 samples using a sliding time window of 5 seconds. The number of image samples generated from the dataset are shown in TABLE II. We use the same framework for CNN as used in [17], apart from increasing the pool size of max pooling layer to (3,3) with stride (3,3). This helps in downsizing the feature map for fully connected layer and eases memory requirement for training the CNN. For ease of reference, the framework of the CNN is shown in Fig. 7. The CNN was trained with this color RDI plot using angle as intensity. The samples were randomly distributed in training, validation, and test dataset in the ratio of 80:10:10. The validation frequency was set for 50 iterations. The CNN was trained on the MATLAB R2021a version with Nvidia GTX 1060 graphics having 3GB GDDR5 memory. Input image size was $205 \times 269 \times 3$. The total time for training was approximately 10 minutes. The CNN achieves an accuracy of 99.8%. The confusion matrix of this CNN for the test dataset is shown in Fig. 8.

Similar to the RDI plot, we train the CNN using color plots of Range-Angle image (RAI) with doppler as intensity and Angle-Doppler image (ADI) with range as intensity. A

1	266						
2		266					
3			266				
4				254			
5					272		
6						264	2
7						1	271
	1	2	3	4	5	6	7

Fig. 8. Confusion matrix of test data for CNN having feature input of Range-Doppler image with angle as intensity.

Fig. 9. Data collection at various positions w.r.t. radar.

comparison of the CNN accuracy with [17] is shown for RDI, RAI, and ADI feature inputs in TABLE III. There is significant improvement in accuracy for RDI CNN, because of the additional angle information provided as intensity vector. With angle information, the CNN is able to differentiate in symmetrical right tricep and left tricep exercises.

IV. POSITION VARIATION OF THE USER

The classifiers were trained on data collected from User 1 and User 2 at position 0 (refer Fig. 9). RDI, RAI, and ADI feature plots are dependent on the position of the user w.r.t. radar. When the user positions change, it affects the feature plots which in turn affects the CNN accuracy. The drop in accuracies in [17] were primarily due to two factors: first is that at position 1, 2, 5 & 6 the angle variations affect the RAI and ADI plots, and second is that at position 3 and 4, the spread of point cloud is more and less respectively compared to position 0 due to change in both range and angle variations.

Apart from using the color image as feature plots, we filtered angle values centered around the median angle of the user position. It was observed that all the exercises had angular variations not more than 50 deg. For example, position 1 was located at a median angle of 116 deg. Therefore, we filtered angle values from 91 deg to 141 deg. Apart from normalization of angle variations in feature plots, this also helped in filtering unwanted clutter due to other objects. We tested the CNNs trained for position 0 for various other positions of the user w.r.t. radar. A total of 210 image samples (30 for each exercise) were generated for each position 1 to 6. Table IV shows that there is significant enhancement in accuracy when compared to [17]. However, this technique does not help much in improving accuracies at position 3 and 4. Both the range and angle values change at position 3 and 4 compared to position 0. Therefore, we propose a data transformation technique to improve accuracies for position 3 and 4.

TABLE IV
PREDICTION ACCURACIES OF CNNs AT VARIOUS POSITIONS USING COLORED FEATURE PLOTS WITH ANGLE NORMALIZATION.

Position	RDI	RAI	ADI
1	85.96	78.95	96.49
2	82.05	87.61	97.86
3	17.84	25.82	61.50
4	16.99	21.84	83.01
5	100	86.67	93.33
6	92.59	98.15	97.22

Fig. 10. Data transformation from position 4 to position 0.

A. Translation of Range-Angle Values

When the user moves from position 0 to position 3 or 4, the movement of body parts are similar as the user performs the exact same exercises. However, the feature plot input image changes. This change is due to the fact that when a user moves farther from radar, the angular variations w.r.t. radar will be lesser for similar movement of body parts because of change in the range w.r.t. radar. Therefore, we propose translation of range and angle values for position 3 and 4 as if the user is performing exercises at position 0.

Referring Fig. 10, let position 0 is situated at range r_0 from the radar and position 4 is at range r_4 . The difference in range of position 4 and position 0 is $\Delta r = r_4 - r_0$. At position 4, we assume that in a certain exercise, the movement of body parts extend within a circle with radius d . Let a point (x_4, y_4) exists on the circle having range cr_4 and angle θ_4 , where $x_4 = cr_4 \times \cos(90 - \theta_4)$ and $y_4 = cr_4 \times \sin(90 - \theta_4)$. As discussed before, the movement of body parts in exercises remains the same irrespective of the position of the user. When we translate the point from position 4 to position 0, the point becomes $(x_4, y_4 - \Delta r)$. Now we can calculate cr_0 and θ as,

$$cr_0 = \sqrt{(x_4)^2 + (y_4 - \Delta r)^2} \quad (1)$$

$$\theta = 90 - \tan^{-1} \left(\frac{y_4 - \Delta r}{x_4} \right) \quad (2)$$

TABLE V
PREDICTION ACCURACIES OF CNNs FOR POSITION 3 & 4 AFTER TRANSLATION OF RANGE-ANGLE VALUES.

Position	RDI	RAI	ADI
3	91.08	84.98	92.96
4	56.80	46.60	96.12

TABLE VI
FEATURES SELECTED FOR EXTRACTING REPETITION COUNT OF VARIOUS EXERCISES.

Exercise	Feature for Repetition Count
Dumbbell	Doppler
Shoulder Press	Doppler
Squat	Range
Lateral Raise	Angle
Boxing	Range
Right Tricep	Doppler
Left Tricep	Doppler

Using this technique, we translate the data of position 3 & 4 and generate the RDI, RAI, and ADI feature plots. TABLE V shows the accuracy of the CNNs after using the translation technique for position 3 & 4 data. This technique significantly enhances the accuracy for all three types of feature plot inputs.

The data collection for training the CNN is a highly demanding task. The discussed two techniques of angle normalization and translation of range-angle values eliminate the requirement of additional training data and re-training the CNN for various other positions. The ADI CNN trained for position 0 performs remarkably well for other positions as well.

V. COUNTING REPETITIONS OF EXERCISES

Apart from accurate classification of exercises, a fitness tracker shall have the capability to count the number of repetitions of exercise. We can see in Fig. 5, that there are distinct repeating variations in doppler, range, and angle data over time which represent the repetition of the exercise. Based on the body part movements in various exercises, we select one dominant feature (i.e. doppler, range or angle variations) for each exercise. The repetition count can be calculated by extracting the smoothed envelope of the data and counting the number of peaks. To extract the boundary of the scattered point cloud, we create an alpha shape and extract boundary facets. We used a shrink factor of 0.6. Using the boundary facets, we can find the boundary points and extract the minimum and maximum envelope. This envelope of the point cloud data is then smoothed using Local Weighted Scatterplot Smoothing (LOWESS) technique. LOWESS takes the weights of the scattered points into account in a defined window size based on the least square regression [20]. This helps in reducing the effect of outliers in smoothed data. We take the window size of 5 to 8 points depending on the exercise. Fig. 11(a) shows the extracted max and min envelope from the time-doppler point cloud data for shoulder press exercise. The solid

TABLE VII
COMPARISON OF PREDICTION ACCURACIES OF CNNs WITH VARIABLE WINDOW SIZE AT VARIOUS POSITIONS USING COLORED FEATURE PLOTS WITH ANGLE NORMALIZATION AND TRANSLATION FOR POSITION 3 AND 4.

Position	CNN (1 Sec Window)			CNN (2 Sec Window)			CNN (5 Sec Window)		
	RDI	RAI	ADI	RDI	RAI	ADI	RDI	RAI	ADI
0	97.6	96.2	94.2	98.9	98.9	98.8	99.8	99.9	99.7
1	88.06	85.67	91.64	85.67	79.75	94.08	85.96	78.95	96.49
2	74.64	69.23	78.63	81.16	83.28	96.05	82.05	87.61	97.86
3*	86.67	74.44	73.06	91.67	84.57	90.12	91.08	84.98	92.96
4*	57.78	35	54.72	54.32	44.75	74.38	56.80	46.60	96.12
5	97.26	86.59	82.62	99.34	88.78	94.72	100	86.67	93.33
6	88.92	85.03	88.62	89.81	89.49	98.41	92.59	98.15	97.22

*After applying data transformation

Fig. 11. Extraction of repetition count of exercises from the doppler, range, and angle scattered point cloud. The dashed black rectangle shows the repetition of exercise (a) Smoothed max and min doppler envelope variation over time for shoulder press exercise, (b) Smoothed min range envelope variation over time for squat exercise, (c) Smoothed max and min angle envelope variation over time for lateral raise exercise.

lines show the LOWESS smoothed data. The four dashed black rectangles show the repetition of the exercise. Fig. 11(b) and Fig. 11(c) show the repetition counting for squat and lateral raise exercises using range and angle data respectively. Similarly, we can extract the repetition count of other exercises by processing the selected dominant features shown in TABLE VI.

VI. REAL-TIME PERFORMANCE OF SYSTEM

The radar gives point cloud data output in real-time using the serial connection. The connected computer receives the data collected for 1 second duration and sends it to the processor function. To simulate the real-time output of the radar, four random samples (of nearly 10 seconds each) for each of the seven exercises for position 0 were selected and processed from a test dataset. The processed data had time, range, doppler, angle, and actual activity information. Every second, the data for the last 1 second duration from this processed radar data was filtered and sent to the processor function. In Section III, we took a window size of 5 seconds for feature plots generation considering that at least one iteration of the exercise would have been performed. TABLE VII shows the comparison of accuracy of CNNs with

different window sizes. As we increase the window size, the accuracy of the CNNs increase. This makes sense, since as we increase the window size, more variations of the exercise will be captured in the generated feature plot. However, a large window size negatively affects the real-time accuracy. The larger the window size, the more time it will take for CNN to detect transitions from one exercise to another. Keeping this in mind, we selected the ADI CNN with a window size of 2 seconds for real-time testing of the system as it had good accuracy for all positions. Fig. 13 shows the control flow graph of the algorithm used for real-time execution. The data was sent for classification only when there was at least 2 seconds of continuous data. If the confidence of the predicted activity was less than 80 percent, the activity was classified as none of the seven (class label 0). Once the exercise was classified by the CNN, the data was sent for repetition counting only when there was a change in current predicted exercise to the last predicted exercise and at least 5 seconds of continuous data for last predicted exercise was available. Fig. 14 shows the real-time processing output of the system. The real-time angle-doppler plot is shown on the left and command window on the right shows the predicted activity, its confidence, and actual activity. There were 235 correct predictions by CNN out

Fig. 12. (a) Comparison of real-time prediction and actual activity for window size of 2 seconds, (b) Comparison of real-time prediction and actual activity for window size of 5 seconds, (c) Session report showing duration and repetitions of exercises.

Fig. 13. Flow chart of the real-time implementation algorithm.

of a total 246. Therefore, real-time accuracy of the system was 95.53%. Fig. 12(a) and (b) show the comparison of predicted activity to the actual activity with a window size of 2 and 5 seconds respectively. When the window size is 5 seconds, the CNN performs poorly in detecting the transition from one exercise to another. At the end, the system generates a session

report as shown in Fig. 12(c). The report shows the duration of each exercise in seconds followed by repetition count.

The deployment feasibility of a neural network on any device will be ultimately governed by the available memory and computational power of the device. The memory requirement of the network during test time is dominated by the memory required to store intermediate layer outputs and model parameters [21]. Considering single precision floating point format, the total memory requirement of the designed CNN during test time is approximately 7.96 MB. Computation complexity of a CNN can be approximated by calculating the number of floating point operations (FLOPs). The FLOPs count in a forward pass of CNN are dominated by the convolution layer and the fully connected layer [21]. FLOPs for a fully connected layer are equal to product of number of inputs and number of outputs, and for a convolution layer it can be calculated as product of height of input, width of input, previous layer depth, current layer depth, kernel width, and kernel height. The FLOPs in forward pass of the designed CNN are 61.95 million. The memory and FLOPs details of the CNN are shown in TABLE VIII. This network can be deployed on edge computing devices such as Raspberry Pi 4 or Nvidia Jetson Nano. Raspberry Pi 4B has Broadcom Videocore VI graphics which is capable of 32G (FP32) floating point operations per second (FLOPS) [22]. The Nvidia Jetson Nano has 128 CUDA cores capable of 236G (FP32) FLOPS [23]. Both the devices have available memory in excess of 2 GB and are capable of running deep learning models. Considering the computational power and available memory, both the edge devices will be able to run the trained CNN in real-time.

VII. CONCLUSION

In this paper, we presented mmFit, an edge computing enabled, user privacy preserving, real-time, and contactless fitness tracker system for indoor fitness activities using a single mmwave radar sensor. The contactless configuration helps to avoid the hassle of attaching tracking sensors to body and skin irritation. The system is capable of highly accurate classification and exercise repetition counting in real-time. It is computationally inexpensive to be deployed on IoT connected edge devices. The system can be deployed in a smart home

Fig. 14. Snapshot of real-time processing output of the system. The left window shows the real-time angle-doppler variations with range as intensity. The right window shows the predicted activity, its confidence, and actual activity

TABLE VIII
MEMORY AND FLOPS CALCULATION FOR CNN DURING TEST TIME.

Layer Information	CNN Model
Input Size	205×269×3
Convolution Layer	
No. of Convolution Layers	3
Kernel Size	5
Strides	1
No. of Filters	8,16,32
Fully Connected Layer	
No. of FC Layers	1
No. of Inputs	22080
No. of Outputs	7
Model Parameters	171,447
Parameters of Intermediate Layer Outputs	1,913,976
Model Test Time Memory (Single Precision)	7.96 MB
FLOPs (Forward Pass)	61.95 Million
Accuracy (Angle-Doppler Image)	98.8%

or smart gymnasium setting with inherent advantage of user privacy preservation.

REFERENCES

- [1] Redhat, "EDGE COMPUTING: Why IoT and edge computing need to work together", <https://www.redhat.com/en/topics/edge-computing/iot-edge-computing-need-to-work-together>
- [2] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7457-7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.
- [3] J. Zhang and D. Tao, "Empowering Things With Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things," in IEEE Internet of Things Journal, vol. 8, no. 10, pp. 7789-7817, 15 May15, 2021, doi: 10.1109/JIOT.2020.3039359.
- [4] B. Wang, L. Guo, H. Zhang and Y. -X. Guo, "A Millimetre-Wave Radar-Based Fall Detection Method Using Line Kernel Convolutional Neural Network," in IEEE Sensors Journal, vol. 20, no. 22, pp. 13364-13370, 15 Nov.15, 2020, doi: 10.1109/JSEN.2020.3006918.
- [5] M. E. Yanik, D. Wang and M. Torlak, "Development and Demonstration of MIMO-SAR mmWave Imaging Testbeds," in IEEE Access, vol. 8, pp. 126019-126038, 2020, doi: 10.1109/ACCESS.2020.3007877.
- [6] A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, "Rad-HAR: Human activity recognition from point clouds generated through a millimeter-wave radar," in Proc. 3rd ACM Workshop Millim.-Wave Netw. Sens. Syst., New York, NY, USA, 2019, pp. 51-56, doi:10.1145/3349624.3356768
- [7] M. Ehatisham-Ul-Haq et al., "Robust Human Activity Recognition Using Multimodal Feature-Level Fusion," in IEEE Access, vol. 7, pp. 60736-60751, 2019, doi: 10.1109/ACCESS.2019.2913393.
- [8] Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. 2021. Vid2Doppler: Synthesizing Doppler Radar Data from Videos for Training Privacy-Preserving Activity Recognition. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, Article 292, 1-10. DOI:<https://doi.org/10.1145/3411764.3445138>
- [9] E. L. Chuma, L. L. B. Roger, G. G. de Oliveira, Y. Iano and D. Pajuelo, "Internet of Things (IoT) Privacy-Protected, Fall-Detection System for the Elderly Using the Radar Sensors and Deep Learning," 2020 IEEE International Smart Cities Conference (ISC2), 2020, pp. 1-4, doi: 10.1109/ISC251055.2020.9239074.
- [10] B. Çağlıyan and S. Z. Gürbüz, "Micro-Doppler-Based Human Activity Classification Using the Mote-Scale BumbleBee Radar," in IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 10, pp. 2135-2139, Oct. 2015, doi: 10.1109/LGRS.2015.2452946.
- [11] Y. Kim and T. Moon, "Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks," in IEEE Geoscience and Remote Sensing Letters, vol. 13, no. 1, pp. 8-12, Jan. 2016, doi: 10.1109/LGRS.2015.2491329.
- [12] J. Zhu, H. Chen and W. Ye, "A Hybrid CNN-LSTM Network for the Classification of Human Activities Based on Micro-Doppler Radar," in IEEE Access, vol. 8, pp. 24713-24720, 2020, doi: 10.1109/ACCESS.2020.2971064.
- [13] Y. He, Y. Yang, Y. Lang, D. Huang, X. Jing and C. Hou, "Deep Learning based Human Activity Classification in Radar Micro-Doppler Image," 2018 15th European Radar Conference (EuRAD), 2018, pp. 230-233, doi: 10.23919/EuRAD.2018.8546615.
- [14] J. Pegoraro and M. Rossi, "Real-Time People Tracking and Identification From Sparse mm-Wave Radar Point-Clouds," in IEEE Access, vol. 9, pp. 78504-78520, 2021, doi: 10.1109/ACCESS.2021.3083980.
- [15] J. Savacool, "Finally, a Fitness Tracker Fit for Weightlifting, HIIT, and CrossFit," <https://www.mensjournal.com/health-fitness/nexus-the-best-fitness-tracker-for-weightlifting-hiit-and-crossfit/>
- [16] Wahoo, "Bluetooth and ANT+ Heart Rate Sensor with Calorie Tracking and Memory," <https://eu.wahoorfitness.com/devices/heart-rate-monitors/tickr-x/buy>.
- [17] G. Tiwari and S. Gupta, "An mmWave Radar Based Real-Time Contactless Fitness Tracker Using Deep CNNs," in IEEE Sensors Journal, vol. 21, no. 15, pp. 17262-17270, 1 Aug.1, 2021, doi: 10.1109/JSEN.2021.3077511.
- [18] J. Roland, "What Exactly Are METs, and What Should You Know About Them?", Oct 2019, <https://www.healthline.com/health/what-are-mets#calorie-connection>
- [19] Morel, Pierre. "Grammar: Grammar of Graphics Plotting in Matlab." The Journal of Open Source Software, vol. 3, no. 23, The Open Journal, Mar. 2018, p. 568, doi:10.21105/joss.00568.
- [20] William S. Cleveland (1979), "Robust Locally Weighted Regression and Smoothing Scatterplots", Journal of the American Statistical Association, 74:368, 829-836, DOI: 10.1080/01621459.1979.10481038
- [21] K. McGuinness, "Memory Usage and Computational Considerations", Deep Learning for Computer Vision, Summer seminar UPC TelecomBCN (July 4-8, 2016), <http://imatge-upc.github.io/telecombcn-2016-dlcv/>
- [22] Broadcom VideoCore VI Technical Details, https://www.cpu-monkey.com/en/igpu-broadcom_videocore_vi-221
- [23] Nvidia Jetson Nano Technical Specifications, <https://developer.nvidia.com/embedded/jetson-nano>