2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD) | 979-8-3503-4918-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CSCWD61410.2024.10580232

Privacy-preserving Human Activity Recognition via Video-based Range-Doppler Synthesis

Zhiyuan Cui[1], Luoyu Mei[1,2], Siyuan Pei[1], Borui Li[1], Xiaolei Zhou[3*]
[1]School of Computer Science and Engineering, Southeast University
[2]Department of Computer Science, City University of Hong Kong
[3]The Sixty-Third Research Institute, National University of Defense Technology
Email: { zycui, lymei-, siyuanpei, libr }@seu.edu.cn, zhouxiaolei@nudt.edu.cn

***Abstract*—As an important branch of IoT applications, Human activity recognition (HAR) is widely used in daily life, particularly through vision-based methods. However, vision-based HAR has serious privacy issues. How to better and low-cost protect the privacy of users who have already installed the relevant devices is a problem that needs to be solved. To address this challenge, we can solve it by transforming video to privacy-preserving mmWave data. Existing studies have primarily focused on synthesizing micro-Doppler data from video, but there is a lack of methods for synthesizing range-Doppler data. Thus, we present a comprehensive method for synthesizing range-Doppler data from videos and subsequently utilize this synthetic data for HAR. Experimentally, we deploy our range-Doppler synthesis method and classification model on a custom dataset. Experimental results indicate that the model trained with synthetic data achieves accuracy on the custom dataset by 95.7%, which is comparable to the accuracy of vision-based HAR works, and demonstrate that the scheme proposed in this paper achieves privacy-preserving HAR.**

***Index Terms*—Human Activity Recognition, Wireless Sensing, Deep Learning, Privacy-preserving**

I. Introduction

Human Activity Recognition (HAR), a branch of IoT applications, provides intelligent and convenient services for people and has been widely used in human-computer interaction [1, 2], healthcare, smart driving [3] and monitoring.

HAR can be categorized into three modalities: sensor-based, vision-based, and multimodal, based on the sensing approach [4]. Vision-based HAR has made significant advancements, but it often raises concerns about privacy leakage. Incidents such as the exposure of home surveillance images on websites have caused widespread alarm among users. As a sensor-based approach, mmWave-based HAR has the advantage of protecting visual privacy and good signal richness, but it requires the specialized acquisition of usage scene data. For users already employing vision-based devices, adopting mmWave-based HAR for privacy preservation would require equipment replacement and the extensive re-collection of data, which is costly. To address this issue, we propose a solution to achieve privacy-preserving HAR with vision-based devices by synthesizing mmWave data from video and then performing HAR based on the synthesized data.

Prior works in synthesizing mmWave data from video include Vid2doppler [12] and SynMotion [13]. Vid2doppler focuses on synthesizing micro-Doppler from video and achieves micro-Doppler-based HAR. On the other hand, SynMotion synthesizes mmWave data at the signal level, which is further processed into micro-Doppler for HAR. However, neither of these two works synthesize and utilize the advantageous range-Doppler. In contrast to micro-Doppler, range-Doppler data offers the advantage of an additional dimension, making it more informative and distinctive. And synthesizing range-Doppler requires less storage space and less computational cost compared to synthesizing mmWave signal for HAR.

However, synthesizing range-Doppler data from RGB video poses a challenging task due to the heterogeneity between the two data types, as they have different dimensions and represent different information. To address this challenge, we propose a novel approach in this paper for synthesizing range-Doppler data from unstructured human activity video data. Our approach involves performing cross-domain translation to bridge the gap between RGB video and range-Doppler data, enabling the extraction of range and velocity information from video and synthesizing range-Doppler sensing features. Furthermore, we leverage the synthesized range-Doppler data to train a classification model, allowing us to perform accurate HAR tasks. In summary, our work makes the following contributions:

- We propose a scheme for synthesizing range-Doppler data from RGB video. The method utilizes computer vision and camera imaging principles, etc., to realize the extraction of range and velocity information from video data and synthesize range-Doppler sensing features.
- We propose an action recognition scheme based on the synthetic range-Doppler data. We transform the synthesized range-Doppler into a spatial-temporal map and use a classification model for the action recognition tasks.
- We evaluate the performance of the scheme based on a custom dataset. Our model achieves an impressive accuracy of 95.7% in near-realistic scenarios. Through comparisons with previous work, we demonstrate that our approach is comparable to vision-based HAR in terms of accuracy, enabling privacy-preserving HAR tasks.

II. Related Works

A. Human Activity Recognition based on mmWave

HAR based on mmWave technology can be broadly categorized into two approaches: 3D radar echo-based recognition and 2D radar echo-based recognition.

Range-Doppler is a type of 3D radar echos that reveals moving properties and micro-Doppler properties of targets, which provides comprehensive information about activities compared to 2D echoes. 3D Convolutional Neural Networks (3D-CNN) and Long Short Term Memory (LSTM) are the two most commonly used models for HAR based on 3D radar echos. For example, researchers[5] employ 3D-CNN and CNN-LSTM models for gesture recognition. However, our approach adopts the method proposed in work [6] to synthesize the 3D echo information into 2D spatial-temporal heat maps. With spatial-temporal maps, we can extract temporal and spatial features only using 2D-CNN without LSTM. This method can simplify the model training process while retaining high accuracy in HAR.

As for 2D radar echoes, micro-Doppler has emerged as the most prevalent choice. Given that the features of 2D echo data are usually images, they are commonly transformed into image classification tasks. 2D-CNNs are the most widely employed models for processing such radar data. Previous works [7, 8] have successfully utilized CNNs to accomplish classification tasks. Compared to the works with 2D echoes, the features extracted by our method contain information in three dimensions, which has the advantage of one more dimension and can be better used for performing HAR.

B. Synthesize mmWave Data

As for mmWave data synthesis, existing work can be broadly categorized into synthesis from real mmWave data and synthesis from non-mmWave data. A common approach to synthesizing from real mmWave data is Generative Adversarial Networks (GANs), such as the work [9]. As for synthesis from non-mmWave data, existing works usually synthesize sensing features of mmWave from non-radar data, such as MoCap [10], camera point clouds [11] and video [12, 13]. Our approach focuses on synthesizing data from RGB video.

The existing works for synthesizing mmWave data from video are Vid2doppler [12] and SynMotion [13]. There is a lack of methods to synthesize range-Doppler data from video, which is the innovation of our work. Vid2doppler synthesizes micro-Doppler data based on human body mesh and uses an encoder-decoder to make it into realistic micro-Doppler data, which is used to train the HAR model. However, our synthetic range-Doppler offers the advantage of an additional dimension compared with micro-Doppler, and can provide the range and speed information of the target.

As for SynMotion, apart from synthesizing sensing features, researchers synthesize mmWave at the signal level using video data and process the synthetic signals into micro-Doppler for HAR. Compared to synthesizing signals and then processing the raw data into sensing features (e.g., micro-Doppler and range-Doppler) for HAR, our method has advantages. We synthesize range-Doppler data directly from video, eliminating the complex computational process of generating sensing features from raw signals using the Fast Fourier Transform (FFT). And saving range-Doppler data requires less storage space than raw signals.

Name	Source	Training data	HAR Model
3D echos [5]	Radar	range-Doppler etc.	3D-CNN+LSTM
2D echos [7, 8]	Radar	micro-Doppler	2D-CNN
Lin et al. [10]	MoCap	micro-Doppler	2D-CNN
Erol et al. [11]	Kinect	micro-Doppler	2D-CNN
Vid2doppler [12]	Video	micro-Doppler	2D-CNN
SynMotion [13]	Video	micro-Doppler	2D-CNN
Our approach	Video	range-Doppler	2D-CNN

TABLE I: Comparison of related works

III. System Overview

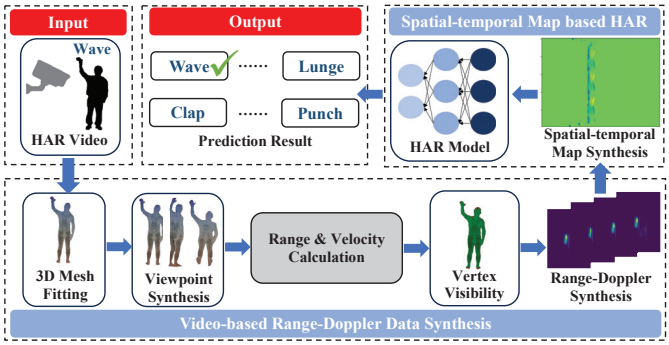


Fig. 1: System overview

Figure 1 illustrates the overall architecture of our approach. And it should be noted that we only employ synthesized range-Doppler data for HAR, without utilizing micro-Doppler data.

As shown in Figure 1, this part includes HAR Video Input, 3D Mesh Fitting, Viewpoint Synthesis, Range and Velocity Calculation, Vertex Visibility, and Range-Doppler Synthesis. Throughout these processes, we extract relevant human activity information from the RGB video and transform it into range-Doppler data, effectively representing the underlying human activities. Following the Range-Doppler Synthesis, we perform HAR using the synthesized data. This part includes Spatial-temporal Map Synthesis, HAR Model, and Prediction Results. We generate 2D spatial-temporal map features from the synthesized range-Doppler and then use a classification model to learn the features and output the prediction results.

IV. Main Design

We outline the key points of this work: video-based range-Doppler data synthesis and spatial-temporal map based HAR.

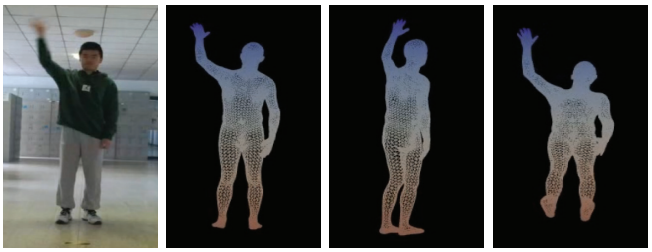
A. Video-based Range-Doppler Data Synthesis

1) 3D Mesh fitting Synthesizing range-Doppler data from 2D video frames necessitates obtaining 3D information regarding the user's body. Fortunately, there have been significant advancements in computer vision that combine 3D meshes with human images. Therefore, our approach first employs the VIBE model [14], which leverages an adversarial learning framework to estimate realistic human pose meshes. Given a video, the VIBE model generates a 3D human mesh output for each frame in the video.

The output of VIBE corresponds to the SMPL (Skinned Multi-Person Linear) model [15]. This statistical human

model, which defines a total of $N = 6890$ vertices comprising the human body surface, plays a key role in the reflection of mmWave radar signals. The vertices are positioned in a 3D coordinate system, with the human chest serving as the origin. Each vertex contains three coordinate parameters: x, y, z. After obtaining the vertices, we improve the stability of the vertex positions by means of frame complement and multi-frame averaging. Then, we will use these vertices in Step *3)* for synthesizing range-Doppler.

2) Viewpoint synthesis After obtaining the 3D mesh, we position a virtual mmWave radar, referred to as a viewpoint, within the 3D coordinate system of SMPL. This viewpoint allows us to observe the user's action from a specific perspective for synthesizing the range-Doppler sensing features. By varying the coordinates and angles of the viewpoint, we can obtain different views of the SMPL model from distinct perspectives, as shown in Figure 2. In theory, it is feasible to position any observation viewpoint within the user-centered SMPL coordinate system. In this study, we specifically select the frontal viewpoint for further use, which aligns with the video acquisition viewpoint.



(a) Input video (b) Front view (c) Side view (d) Top view

Fig. 2: SMPL mesh in different views of the waving action

3) Range and velocity calculations Since the video has no range calibration and depth information, we must estimate and calculate the relative range and velocity of each vertex relative to the virtual radar under the frontal viewpoint in the SMPL coordinate system. Both the range and velocity will be used for the range-Doppler synthesis in Step *5)*.

3.1) Range calculation The calculation of range is actually to calculate the relative distance between the vertex coordinates and the virtual radar coordinates. However, since the origin of the SMPL coordinate system is located in the user's chest, when the user moves extensively back and forth, the origin will shift within the global coordinate system. As a result, the relative z coordinate of the virtual radar in the SMPL coordinate system will change. Thus, to calculate the range, we first estimate the variations in the z coordinate based on the principles of camera imaging, as depicted in Figure 3.

In Figure 3, O represents the focus point, which corresponds to the camera, while f denotes the focal length. The initial position of the person in the first frame is recorded as $Z1$, the actual height at that time is recorded as $h1$, and the height of the bounding box is recorded as $B1$. For any frame other than the first frame, the range of the person is denoted as $Z2$, the

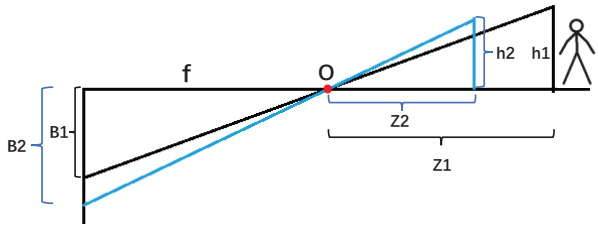


Fig. 3: Camera imaging principle

actual height at that time is recorded as $h2$, and the height of the bounding box is recorded as $B2$.

Considering the fitted SMPL mesh size approximates a 1:1 equivalence with the users, the difference between the maximum and minimum y coordinates of the mesh can represent the real human body height. The bounding box, represented by $bboxes(nframes, 4)$, is one of the outputs of the VIBE, with the width w and height h being equal. Applying the camera imaging principle and the similar triangle principle, we can derive the following formulas: $\frac{B1}{f} = \frac{h1}{Z1}$ and $\frac{B2}{f} = \frac{h2}{Z2}$. With the initial range $Z1$ of the first frame, the camera range $Z2$ of other frames can be obtained from the above formulas:

$$Z2 = \frac{h2 \times B1}{h1 \times B2} Z1 \tag{1}$$

The range $Z2$ calculated using Formula 1 represents the new estimated value of the virtual mmWave radar's z coordinate after the user's movement. Formula 1 effectively eliminates the effect of human posture on range estimation. For instance, when the user stands and squats in place, as shown in Figure 4, the actual height and bounding box height differ between Figure 4a and Figure 4b, but the range remains unchanged. In Figure 4a, the actual height of the human mesh is 1.72 and the bounding box height is 247.71 when standing. When squatting, the mesh height is 1.19 and the bounding box height is 167.13. According to Formula 1, $\frac{1.19 \times 247.71}{1.72 \times 167.13} \approx 1.025$. Thus, we can consider that $Z1 \approx Z2$, which eliminates the effect of human posture successfully.

(a) Bounding box when standing (b) Bounding box when squatting

Fig. 4: Different actions affect the size of the bounding box

With Formula 1, we can obtain the changed z and calculate the range of the vertices relative to the radar, which is used for the synthesis of the range-Doppler.

3.2) Velocity calculation Once the range is obtained, we can calculate the relative radial velocity of each vertex in the front view. This is achieved by determining the motion direction

vector for each mesh vertex and its corresponding direction vector with respect to the virtual radar.

Then we smooth out the relative range and velocity of each vertex to ensure the continuity and stability between frames.

4) Vertex visibility In the previous steps, the assumption is that all SMPL vertices are visible and will contribute to the synthetic Doppler signal. However, some vertices are not visible with respect to the virtual radar and they do not reflect radar signals, such as the vertices located at the back of the human body. Since these vertices do not contribute to the reflection, they must be removed from the full body mesh. We retain the visible vertices through back-face culling [16].

5) Range-Doppler synthesis In real scenarios, radar signals need to go through steps such as Range-FFT and Dopper-FFT to extract range and Doppler velocity information of the target. For the synthesis of range-Doppler, we can bypass such steps and directly obtain the required features using the range and velocity calculated in previous steps. We obtain the synthesized range-Doppler matrix for the visible vertices by overlaying their range and velocity using a two-dimensional histogram. In the synthesized range-Doppler matrix, the Y coordinate represents the radial velocity, ranging from -2 to 2, divided into 32 bins. The X coordinate represents the relative distance range, ranging from 0 to 10, divided into 224 bins. We synthesize range-Doppler by arranging different frames' range-Doppler matrices in the time dimension.

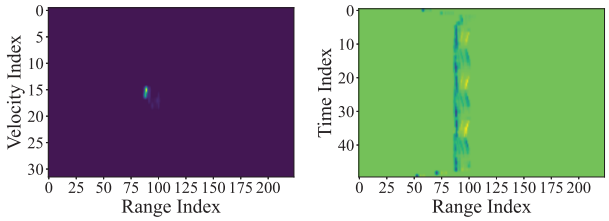
B. Spatial-temporal Map based HAR

1) Spatial-temporal map synthesis After obtaining the synthetic range-Doppler, we convert it into spatial-temporal heat map features. As described in *Related Work*, this method can simplify the model training while making good use of the three-dimensional features in the range-Doppler. Specifically, the work calculates and superimposes the velocity at each range in the range-Doppler by weight:

$$V_{q,t} = \sum_{p=1}^{D} (I_{p,q,t}) \times v_{p,t}, p \in [1, D], q \in [1, R] \qquad (2)$$

In Formula 2, $I_{p,q,t}$ refers to the intensity of the velocity in range-Doppler, and p refers to the velocity index, and q represents the range index, and t represents the frame index. $v_{p,t}$ is the velocity corresponding to velocity index p at frame t. D represents the bin number of velocity, in this paper $D = 32$. R represents the bin number of range, in this paper $R = 224$. In Formula 2, an array containing single-frame range and velocity information is generated after the velocity superposition, and a two-dimensional feature can be obtained by supplementing the array with the time dimension.

The action in Figure 5 is waving, as shown in Figure 2a. The range-Doppler map synthesized based on a certain frame of the video is shown in Figure 5a. The spatial-temporal map synthesized based on the range-Doppler of 50 frames of input video is shown in Figure 5b, where the features generated by the repetitive waving motion can be seen.



(a) Range-Doppler map (b) Spatial-temporal map

Fig. 5: Spatial-temporal feature synthesis

2) HAR model For the classification model, we choose the well-known VGG-16 network. The initial size of the spatial-temporal map is 224×50. To prepare the data for input into the VGG-16 model, we map the matrix data in the spatial-temporal map to RGB color values ranging from 0 to 255 and make the map multi-channeled. The model will be trained to classify different activities in the following section.

V. Evaluation

A. Data Collection

First, we design seven common actions for recognition, as shown in Figure 6.

(a) (b) (c) (d) (e) (f) (g)

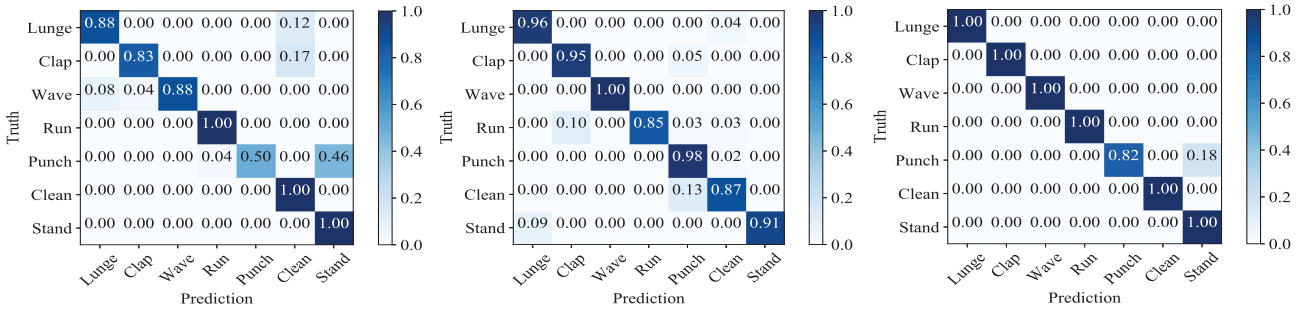
Fig. 6: Seven actions for classification. Action (a) to (g) refers to Lunge, Clap, Wave, Run, Punch, Clean and Stand.

Video capture format We use Microsoft Kinect to capture monocular camera RGB images with a resolution of 1280×720 pixels. After that, the captured images are synthesized into RGB video with a frame rate of 10 fps.

Data collection We invite 6 participants (5 males and 1 female, average age of about 22 years) to collect data in two different scenarios. In scenario 1, we collect video data of approximately 2.4 h in total for the participants (3 males and 1 female). In scenario 2, we collect video data of approximately 0.8 h in total for the participants (3 males and 0 females). It should be noted that we do not strictly correct the participants' actions during data acquisition. Thus, the data collected retained the participants' habits and the final accuracy will be relatively robust.

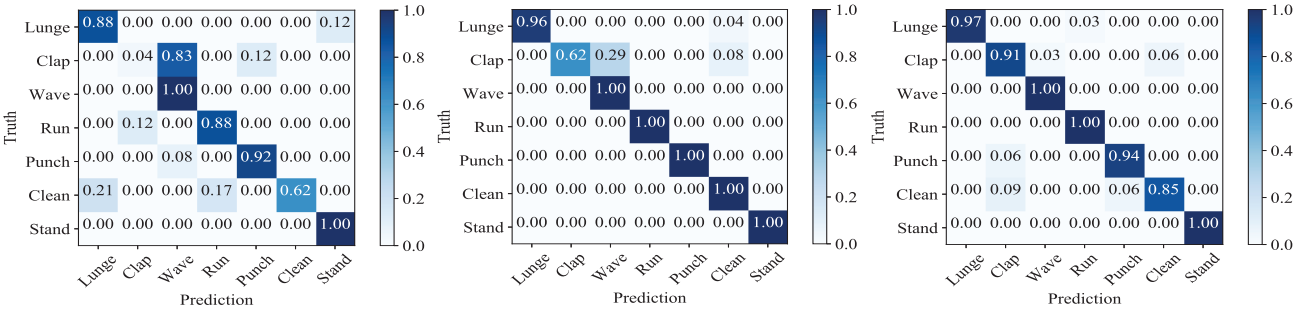
B. Experimental Platform and Environment Related Settings

The hardware platform is the GeForce RTX 4070Ti GPU. The software environment is built on Anaconda3, with Python version 3.7.16. The specific software packages used include



(a) Female 1 leave-one-out result (b) Male 2 leave-one-out result (c) Female 1 user calibration result

Fig. 7: Confusion matrices with models trained on the usage scenarios data



(a) No calibration result (b) User calibration result (c) Environmental and user calibration result

Fig. 8: Confusion matrices with models trained on the non-usage scenario data and calibration data

cudatoolkit version 11.6.0, cudnn version 8.8.0.121, pytorch version 1.12.1, tensorflow version 2.4.1, and other necessary packages required by the VIBE model.

We train the classification model VGG-16 with pre-training loading weights, the cross-entropy loss and Adam optimizer with a learning rate of 0.0002 for 150 epochs.

C. Experimental Results and Analysis

1) Recognition accuracy: data from usage scenarios. We select Scenario 1 as the usage scenario and train and test the HAR model with Scenario 1 data. We first take leave-one-out experiments, meaning that for 4 participants, select the data from 3 of them for training, and then use the data of the remaining one participant for testing (except for Male3). In this configuration, our model achieves an average accuracy of 80.9%. The results are shown in Figure 9 and the two test confusion matrices are shown in Figure 7.

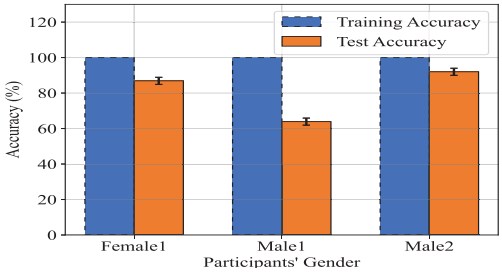


Fig. 9: Training results with leave-one-out methods

However, in practical scenarios, it is common for sensing systems to collect some training data from users for calibration. Therefore, based on the leave-one-out method for Female 1, we add half of the data from her for training and use the other half for testing (the training-to-testing data ratio is approximately 8:1). In this case, our model achieves the accuracy of 96.5% after user calibration. The confusion matrix is shown in Figure 7c. Compared with Fig 7a, it shows that user calibration will be effective in improving model accuracy.

2) Recognition accuracy: only data from non-usage scenario for training. To better evaluate our method, we choose Scenario 1 with more data as a non-usage scenario for training and choose Scenario 2 as the use scenario for testing. In the following experiments, we utilize the user calibration model in experiment 1 for testing. First, we test the model with data from the participants in Scenario 2, who did not contribute to the data collection in Scenario 1. The final accuracy rate is 76.2% and the confusion matrix is presented in Figure 8a. This result can be regarded as "out-of-the-box" accuracy, without any calibration to the local environment or user. The accuracy rate decreases compared to the leave-one-out method, possibly due to variations in the testing environment.

Subsequently, we test the model with the data from the participant in Scenario 2, who also participated in Scenario 1 data collection. With only user calibration, the final accuracy rate of this test is 94%, and the confusion matrix obtained from the test is shown in Figure 8b. The significantly higher accuracy compared to 76.2% indicates that user calibration has

played an important role in the model performance.

3) Recognition accuracy: data from non-usage scenario and usage scenario for training. We take Scenario 1 as the non-use scenario and Scenario 2 as the usage scenario. All of the data in Scenario 1 and half of the data in Scenario 2 are used for training, and the rest of the data are used for testing (training data to test data ratio is 5.8:1). This experimental setup is **closest to real-world applications**, where HAR models are typically trained with a large amount of non-use scenario data and calibrated with a small amount of data. In this configuration, our model achieves a final test accuracy of 95.7% and the confusion matrix is displayed in Figure 8c. The results demonstrate that after completing environment and user calibration, the model exhibits high recognition accuracy across all actions, yielding improved performance.

The above experiments show that our method performs better when trained with all scenario data. And the "out-of-the-box" accuracy is 76.2%. In the most realistic scenarios, it achieves 95.7% accuracy, indicating that our method is able to efficiently and accurately accomplish HAR.

4) Comparison with previous works. First, we compare our work with mmWave synthesis works in *Related Work*. Vid2dop [12] achieves a recognition accuracy of 95.9% for 12 human activities, incorporating user and environment calibration. SynMotion [13] achieved 94.1% accuracy across eight activities. Our work achieves 95.7% on 7 actions with calibrations, showcasing promising performance compared to previous studies. Second, we compare our work with vision-based HAR studies to assess the potential of our approach as a substitute for video data in HAR tasks. With larger datasets like UCF101, work [17] achieves 91.4% recognition accuracy with high efficiency. According to the work [18], state-of-the-art vision-based methods can achieve efficient and accurate multi-person pose estimation in complex scenes. Therefore, compared with the latest or state-of-the-art vision-based work, our work has limitations concerning datasets, the number of recognized action categories, and multi-object recognition. However, the recognition accuracy of our work reaches 95.7%, meeting the standard that the advanced deep learning recognition methods can attain accuracy rates exceeding 90% [19]. Therefore, the accuracy of our work is comparable to vision-based HAR, which means our approach can replace video data for the HAR task from the accuracy point of view.

VI. Conclusion

Vision-based HARs have now achieved significant results in IoT applications, but the privacy issues they pose have been controversial. How to better protect the privacy of users who have installed related devices is an urgent problem to be solved. In this paper, we propose a solution by first synthesizing range-Doppler data containing human motion information from human activity video, and then implementing activity classification using a deep learning network based on the synthesized data. The experimental results show that the model's human activity recognition accuracy is 95.7% in the most realistic scenarios, which is a good performance relative to previous work on synthesized mmWave data. Compared to vision-based work, from the accuracy point of view, our method is comparable and can replace video data to accomplish the task of HAR. This work contributes to the advancement of privacy-preserving HAR systems, providing an accurate approach while mitigating privacy concerns associated with vision-based approaches.

References

[1] P. Song, L. Mei and H. Cheng, "Human Semantic Segmentation using Millimeter-Wave Radar Sparse Point Clouds," *CSCWD 2023*, pp. 1275-1280.

[2] W. Jiang, F. Li, L. Mei, R. Liu and S. Wang, "VisBLE: Vision-Enhanced BLE Device Tracking," *SECON 2022*, pp. 217-225.

[3] Wang, S., Mei, L., et al.'End-to-End Target Liveness Detection via mmWave Radar and Vision Fusion for Autonomous Vehicles", *ACM Transactions on Sensor Networks*, Just Accepted.

[4] F. Fereidoonian, F. Firouzi, and B. Farahani, 'Human activity recognition: From sensors to applications', in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, 2020, pp. 1–8.

[5] Z. Zhang, Z. Tian, and M. Zhou, 'Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor', *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3278–3289, 2018.

[6] Y. Xie, R. Jiang, X. Guo, Y. Wang, J. Cheng, and Y. Chen, 'mmFit: Low-Effort Personalized Fitness Monitoring Using Millimeter Wave', in *2022 International Conference on Computer Communications and Networks (ICCCN)*, 2022, pp. 1–10.

[7] R. P. Trommel, R. I. A. Harmanny, L. Cifola, and J. N. Driessen, 'Multi-target human gait classification using deep convolutional neural networks on micro-Doppler spectrograms', in *2016 European Radar Conference (EuRAD)*, 2016, pp. 81–84.

[8] G. Li, Z. Zhang, H. Yang, J. Pan, D. Chen, and J. Zhang, 'Capturing human pose using mmWave radar', in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6.

[9] B. Erol, S. Z. Gurbuz, and M. G. Amin, 'GAN-based synthetic radar micro-Doppler augmentations for improved human activity recognition', in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–5.

[10] Y. Lin and J. Le Kernec, 'Performance analysis of classification algorithms for activity recognition using micro-Doppler feature', in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 2017, pp. 480–483.

[11] B. Erol and S. Z. Gurbuz, 'A kinect-based human micro-doppler simulator', *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 5, pp. 6–17, 2015.

[12] K. Ahuja, Y. Jiang, M. Goel, and C. Harrison, 'Vid2doppler: Synthesizing doppler radar data from videos for training privacy-preserving activity recognition', in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–10.

[13] X. Zhang, Z. Li, and J. Zhang, 'Synthesized Millimeter-Waves for Human Motion Sensing', in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 377–390.

[14] M. Kocabas, N. Athanasiou, and M. J. Black, 'Vibe: Video inference for human body pose and shape estimation', in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5253–5263.

[15] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, 'SMPL: A skinned multi-person linear model', *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[16] H. Zhang and K. E. Hoff III, 'Fast backface culling using normal masks', in *Proceedings of the 1997 symposium on Interactive 3D graphics*, 1997, pp. 103-ff.

[17] S. Yu, L. Xie, L. Liu, and D. Xia, 'Learning long-term temporal features with deep neural networks for human action recognition', *IEEE Access*, vol. 8, pp. 1840–1850, 2019.

[18] H.-B. Zhang et al., 'A comprehensive survey of vision-based human action recognition methods', *Sensors*, vol. 19, no. 5, p. 1005, 2019.

[19] A. Shenoy and N. Thillaiarasu, 'A survey on different computer vision based human activity recognition for surveillance applications', in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, 2022, pp. 1372–1376.