




Article

Improving Radar Human Activity Classification Using Synthetic Data with Image Transformation

Rodrigo Hernangómez ^{1,*} , Tristan Visentin ¹, Lorenzo Servadei ^{2,3} , Hamid Khodabakhshandeh ¹ and Sławomir Stańczak ^{1,4} 

¹ Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany; tristan.visentin@hhi.fraunhofer.de (T.V.); hamid.khb.89@gmail.com (H.K.); slawomir.stanczak@hhi.fraunhofer.de (S.S.)

² Infineon Technologies AG, 85579 Munich, Germany; lorenzo.servadei@infineon.com

³ Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany

⁴ Faculty IV, Electrical Engineering and Computer Science, Technical University of Berlin, 10587 Berlin, Germany

* Correspondence: rodrigo.hernangomez@hhi.fraunhofer.de

Abstract: Machine Learning (ML) methods have become state of the art in radar signal processing, particularly for classification tasks (e.g., of different human activities). Radar classification can be tedious to implement, though, due to the limited size and diversity of the source dataset, i.e., the data measured once for initial training of the Machine Learning algorithms. In this work, we introduce the algorithm Radar Activity Classification with Perceptual Image Transformation (RACPIT), which increases the accuracy of human activity classification while lowering the dependency on limited source data. In doing so, we focus on the augmentation of the dataset by synthetic data. We use a human radar reflection model based on the captured motion of the test subjects performing activities in the source dataset, which we recorded with a video camera. As the synthetic data generated by this model still deviates too much from the original radar data, we implement an image transformation network to bring real data close to their synthetic counterpart. We leverage these artificially generated data to train a Convolutional Neural Network for activity classification. We found that by using our approach, the classification accuracy could be increased by up to 20%, without the need of collecting more real data.

Keywords: radar; machine learning; deep learning; human activity classification; image transformation; domain shift



Citation: Hernangómez, R.; Visentin, T.; Servadei, L.; Khodabakhshandeh, H.; Stańczak, S. Improving Radar Human Activity Classification Using Synthetic Data with Image Transformation. *Sensors* **2022**, *22*, 1519. <https://doi.org/10.3390/s22041519>

Academic Editors: Akanksha Bhutani and Mario Pauli

Received: 10 December 2021

Accepted: 11 February 2022

Published: 16 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Motivation

The fast advances in Machine Learning (ML) research help to solve more and more important problems. Lately, sophisticated ML methods found their way into radar signal processing and analysis [1–3]. However, generating high-quality datasets that adequately represent reality to its full extent is still highly demanding. Methods like domain adaptation, transfer learning and Generative Adversarial Networks (GANs) assist the improvement of classification or regression tasks from incomplete datasets [4]. In this paper, we focus on human activity classification with radar due to the research interest that it has currently awakened [5–8], for which we count on the help of millimeter-wave radar sensors operating at 60 GHz. Problems with datasets in this context can originate from several reasons:

1. Incomplete dataset, meaning that a limited number of subjects executed the activities.
2. Insufficient measurement time, meaning that just a short time frame of the activity for each person is captured for the dataset.
3. Different radar sensor settings or parameters, e.g., bandwidth, measurement time, repetition time, etc.

4. Inconsistent measurement times, meaning the same person performing a specific activity task executes the activity differently at different times, e.g., after a coffee break or at a different distance to the sensor, etc.

From these given problem examples, 1 and 2 can only be solved adequately by gathering more data of different people for a longer time frame. Problems 3 and 4 are promising cases to be investigated though, as they lend themselves to be solved by ML methods. Our following evaluation showed that the same data being measured with equal radar sensors but different settings (example 3) did not lead to a large decline in classification accuracy [9]. Thus, in this paper, we focus on problem 4.

In the dataset originating from our measurement campaign (outlined in [9]), the data also showed a statistical discrepancy for different measurement time intervals (different subjects, disparate activity execution, etc.). Our idea to tackle this issue is to investigate whether we can utilize synthetic data to enhance our dataset. The proposed method uses a radar reflection model of the recorded subject that is based on a human motion model (also referred to as kinematic model) extracted from stereo video camera data. This was possible through the simultaneous recording of the activities with radar sensors and such cameras. Thus, this motion model provides the information to generate radar data from moving humans outlined by Chen [10]. Using this common model, we obtain synthetic radar data to augment our dataset. Furthermore, this approach offers the chance to augment the real radar data with synthetic data originating from human motion data created somewhere else and/or in future measurements. This makes the radar model universally applicable.

In this paper, we introduce an approach that we name Radar Activity Classification with Perceptual Image Transformation (RACPIT), proceeding in the following way:

- We first show that data from a source time and a target time interval in our dataset differ significantly.
- As a baseline, we train a Convolutional Neural Network (CNN) with only real data from one source time interval and test the trained CNN with data from a target time interval.
- We demonstrate the approach of perceptual loss with Image Transformation Networks [11] to show that we can increase classification accuracy by only using synthetic radar data from the target time (generated by taking the human motion data from the target time and using it as input for the human radar reflection model from Chen [10]).
- We propose improvements of this method for future research.

1.2. Related Work

Among ML algorithms, deep learning has gained popularity over the years as a technique to classify human activities using radar features [5,6,9,12–14]. High-quality public radar data is still hardly available, however, despite the great efforts that exist in this regard [15]. Yeo et al. [16] and Zhang et al. [17] have gathered remarkable datasets for tangible interaction and road object detection, respectively; in both cases, they have opted for Frequency-Modulated Continuous-Wave (FMCW) radar to retrieve information about the targets' range, velocity, or angle. In the area of human activity classification, Sevgi Z. Gurbuz et al. [14] have collected a comprehensive dataset that has been acquired with FMCW and Ultra-wideband (UWB) radar sensors for different frequency bands. Nevertheless, it remains unclear whether their FMCW data allows range detection.

Due to this limitation in data availability, some authors resort to simulated data to train their deep-learning algorithms [7,8]. The core of the problem is then shifted to the choice of a suitable model to simulate human motion; this can be either analytically tailored to specific movements [18] or rely on MOTion CAPture (MOCAP) data [19].

Regarding the different methods to bridge the discrepancies between real and simulated radar data, visual domain adaptation techniques constitute a sensible choice [7,8]. In this work, however, we focus on ML-aided image-to-image translation. Among the different approaches of image-to-image translation in the Computer Vision (CV) community, GANs remains one of the most popular ones [20,21]. Originally conceived for style

transfer, perceptual-loss approaches have emerged in recent years as an alternative strategy to tackle image-to-image translation, especially in scenarios such as that of super-resolution images [11]. Here the image translation focuses on the upsampling of a low-quality picture into a bigger, high-definition version of it.

2. Radar Sensor and Dataset

We use the dataset from [9], which consists of the following five human activities:

- (a) Standing in front of the sensor.
- (b) Waving with the hand at the sensor.
- (c) Walking back and forth to the sensor.
- (d) Boxing while standing still.
- (e) Boxing while walking towards the sensor.

In other words, the considered activities belong to a set \mathcal{A} with $|\mathcal{A}| = 5$. For each activity, data were acquired simultaneously with four FMCW radar sensors in a series of non-contiguous recordings. The duration of a single recording ranges from 30 s to 2 min and the recorded data per activity adds up to roughly 10 min, with an overall number of recordings per sensor of $\mathcal{M} = 69$. In this recording environment that can be seen in Figure 1, two male adults took turns to perform the activities individually.

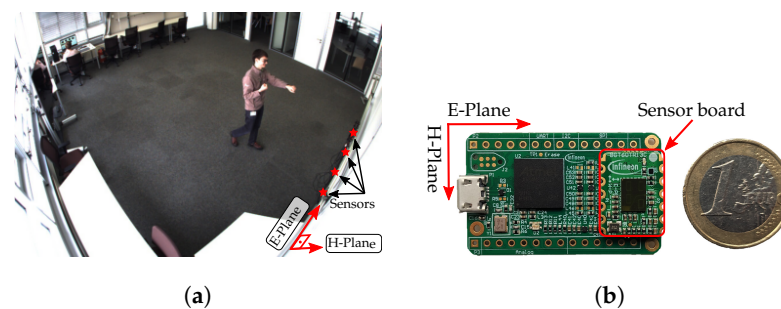


Figure 1. Experimental setup. (a) Overview of the measuring room. (b) Detail of the BGT60TR13C 60 GHz radar system.

All sensors are based on Infineon’s BGT60TR13C chipset, a 60 GHz FMCW radar system (c.f. Figure 1b). Each sensor was previously configured individually with one of the four different settings (I–IV) listed in Table 1.

Table 1. FMCW settings.

Configuration Name		I	II	III	IV
Chirps per frame	n_c	64	64	64	128
Samples per chirp	n_s	256	256	256	256
Chirp to chirp time	[μ s]	250	250	250	250
Bandwidth	[GHz]	2	4	4	4
Frame period	[ms]	50	32	50	50
Range resolution	[cm]	7.5	3.8	3.8	3.8
Max. range	[m]	9.6	4.8	4.8	4.8
Max. speed	[m/s]	5.0	5.0	5.0	5.0
Speed resolution	[m/s]	0.15	0.15	0.15	0.08

Since each activity was recorded for around 10 min, we collected up to 50 min of data per configuration. These 50 min translate to around 60,000 or 93,000 frames depending whether the configuration’s frame period is 50 ms or 32 ms, respectively. We express the

relation between the total number of frames per configuration, L , with the number of recordings per configuration, \mathcal{M} , through the following expression:

$$L = l_1 + l_2 + \dots + l_{\mathcal{M}}, \quad (1)$$

where l_i for $i \in \{1, \dots, \mathcal{M}\}$ equals the number of frames for the i -th recording.

Preliminary data exploration has shown statistical and qualitative differences between recordings that lead to poor generalization across them. In order to explore this phenomenon, we split our recordings into a source domain \mathcal{S} of length $L_{\mathcal{S}}$ frames and a target domain \mathcal{T} of length $L_{\mathcal{T}}$ frames. We do so by assigning the first m recordings to \mathcal{S} and the last $\mathcal{M} - m$ recordings to \mathcal{T} . The splitting point m is chosen so that:

$$L_{\mathcal{S}} = l_1 + \dots + l_m \simeq L_{\mathcal{T}} = l_{m+1} + \dots + l_{\mathcal{M}} \simeq L/2. \quad (2)$$

The FMCW sensors emit a train of n_c linear frequency-modulated electromagnetic pulses (so-called chirps), as outlined in Figure 2. The reflected and returned signal is then mixed down with the emitted one to an Intermediate Frequency (IF) and sampled with n_s samples per chirp. In that way, we obtain the raw data, which we further rearrange in an $n_c \times n_s$ matrix called *frame*. On a frame, we refer to the horizontal direction ranging from 1 to n_s as the *fast time* in contrast to the vertical direction from 1 to n_c , which we refer to as *slow time*. Frames are acquired periodically at a fixed frame rate and pre-processed with the following steps, which revolve around the Fast Fourier Transform (FFT) algorithm:

1. Moving Target Indication (MTI): In order to filter out the clutter from static objects, we calculate the mean across the slow time to obtain an n_s -long vector that we subtract from every column of the frame [22].
2. Range FFT: We perform an FFT across the fast time for every chirp, obtaining thus n_c *range profiles* [22]. Prior to that, we apply a Hanning window [23] and zero-padding, so that the range profiles have a length of 128 regardless of the value of n_s .
3. Doppler FFT: We also perform an FFT across the slow time, turning the range profiles into a Range-Doppler Map (RDM) that conveys information on the returned power across range and Doppler, i.e., distance and velocity (c.f. Figure 3c). Here we use a Chebyshev window with 60 dB sidelobe attenuation [23] and the same zero-padding strategy as for the range FFT. As a result, the dimensions of the RDMs are 128×128 for all configurations.
4. Spectrogram extraction: Similar to [12], we stack the pre-processed frames into an RDM sequence and we sum this over the Doppler and range axes to get range and Doppler spectrograms, respectively (c.f. Figure 4). Here the magnitude of both spectrograms is converted to decibels.
5. Slicing and normalization: As the last step, we slice each long recording into 64-frame long spectrograms with an overlap such that the last 56 frames of one spectrogram coincide with the first 56 frames of the next sliced spectrogram. Furthermore, we shift the decibel values of each sliced spectrogram so that the maximum value equals 0 dB and we subsequently clip out all values below -40 dB.

As a result of this preprocessing and the domain split described in Equation (2), datasets comprise between 3000 and 6000 data points, depending on the configuration. A single data point corresponds to a tuple containing one range spectrogram and one Doppler spectrogram; the spectrogram size in either case is $M \times N$ for $M = 64$ and $N = 128$.

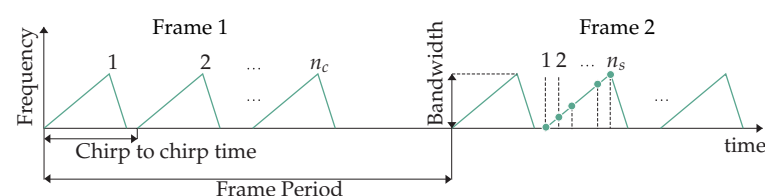


Figure 2. FMCW modulation and sampling pattern.

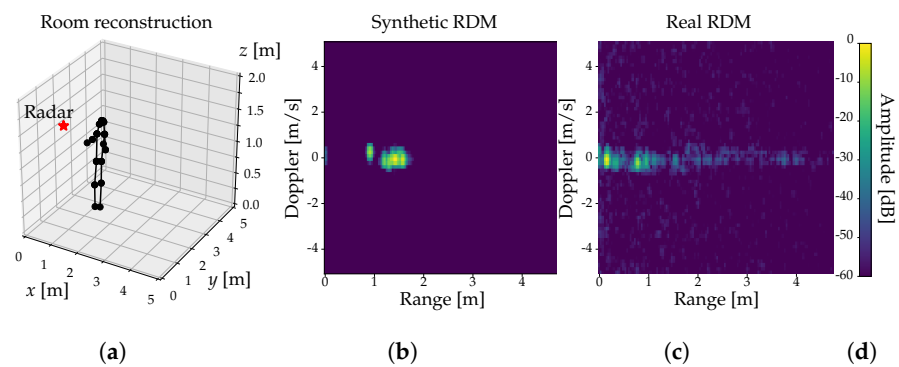


Figure 3. Radar data synthesis (a) CV-generated skeleton keypoints. (b) Synthetic radar data. (c) Real radar data. (d) Color bar.

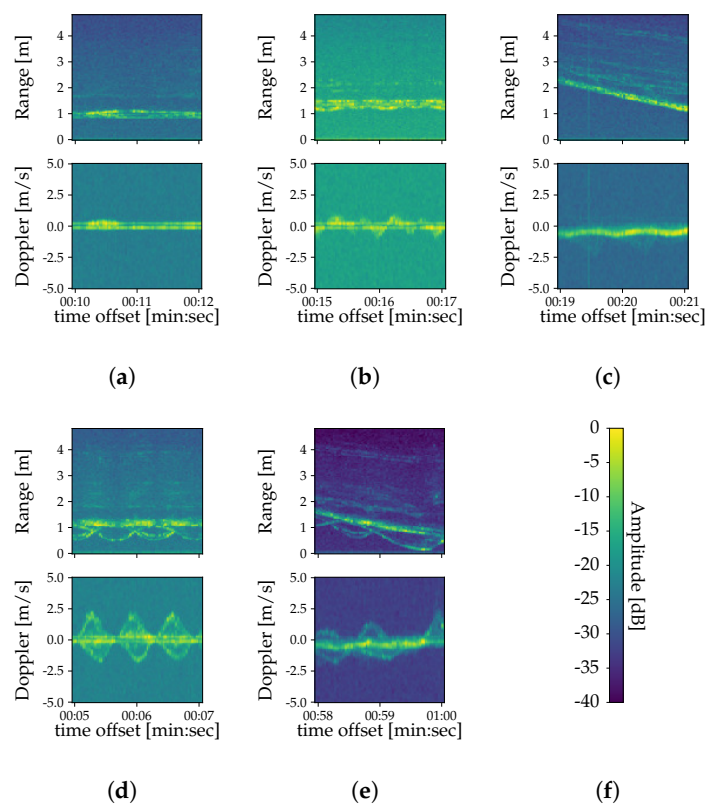


Figure 4. Exemplary range and Doppler spectrograms showing human activities (a–e) for configuration II. (a) Standing. (b) Waving. (c) Walking. (d) Boxing. (e) Boxing and walking. (f) Color bar showing amplitude levels in dB.

Data Synthesis

For the measuring campaign, we have also recorded video data from four different cameras. Moreover, we have extracted the pose of the subject from the video data using Detectron2, a CV library provided by Meta Platforms Inc. (formerly Facebook Inc.) [24]. The extracted pose is given by a skeleton composed of 17 different keypoints, each of them labeled with x , y , and z coordinates.

We use these skeleton keypoints (c.f. Figure 3a) to create synthetic radar data using the analytical FMCW radar model from Chen [10], which in turn is based on a human model proposed by Boulic et al. [25]. As depicted in Figure 5, This human model represents subjects with $K = 14$ ellipsoids, each of which spans over two keypoints and represents a human limb or body part (head, torso, forearm, etc.). For each ellipsoid k we calculate its

distance $d_{k,t}$ to the radar sensor and its corresponding Radar Cross Section (RCS) $A_{k,t}$ at any given point in time t , which we use to obtain the returned and mixed-down signal for a single chirp as

$$s(t) = \sum_{k=1}^K \sqrt{\frac{A_{k,t}}{L_{k,t}}} \sin(2\pi f_{k,t}t + \phi_{k,t}), \quad (3)$$

with a frequency $f_{k,t} = 2d_{k,t}B/T_c c$, an initial phase $\phi_{k,t} = 4\pi f_c d_{k,t}/c$ and a free-space path loss $L_{k,t} = (4\pi d_{k,t}f_c/c)^2$ for given bandwidth B , chirp time T_c and lower frequency f_c . To determine $A_{k,t}$ we use the same model as in [10].

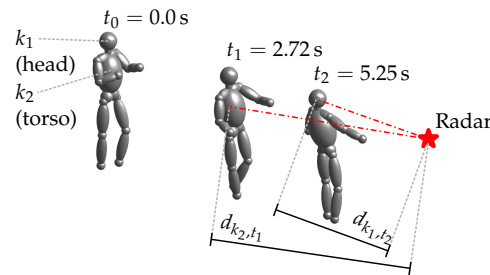


Figure 5. Human posture with respect to the radar at different times, as modeled by Boulic et al. [25].

We apply this procedure throughout all FMCW chirps, thus producing a synthetic IF signal that ideally corresponds to the sampled signal from the radar sensor. We then process the synthetic signal in the same way as the real one to extract its RDMs (c.f. Figure 3b), range, and Doppler spectrograms respectively.

3. Method

As explained in Section 2, our pre-processed real data x comprises a tuple of range and Doppler spectrograms, x^R and x^D , i.e.,:

$$x = (x^R, x^D), \quad x^R, x^D \in \mathbb{R}^{M \times N}. \quad (4)$$

The same holds for our synthetic data y , so that $y = (y^R, y^D)$ with identical dimensions. In order to leverage both range and Doppler data for classification, we make use of the multi-branch CNN from [9], whose implementation details are presented in Figure 6. This CNN passes x^R and x^D through the convolutional branches and feeds the result of both branches into its fully connected layers. We write $\phi^R : \mathbb{R}^{M \times N} \mapsto \mathbb{R}^P$ and $\phi^D : \mathbb{R}^{M \times N} \mapsto \mathbb{R}^P$ to indicate the transformations from spectrogram to features that the range and Doppler convolutional branches respectively apply; the resulting range and Doppler feature vectors have a length of $P = 2304$. Likewise, we use $f : \mathbb{R}^{2P} \mapsto \mathcal{A}$ to denote the transformation from range and Doppler features to the predicted activity \tilde{a} through the fully connected layers. In summary, the relationship between x^R , x^D and \tilde{a} is given by

$$\tilde{a} = f(\phi^R(x^R), \phi^D(x^D)), \quad \tilde{a} \in \mathcal{A}. \quad (5)$$

The architecture of our proposed Radar Activity Classification with Perceptual Image Transformation (RACPIT) is completed by appending a pair of Image Transformation Networks (ITNs) to the input of the CNN, as depicted in Figure 7. The ITNs transform real data x into synthetic-like data $\hat{y} = (\hat{y}^R, \hat{y}^D)$, where $\hat{y}^R = \psi^R(x^R)$, $\hat{y}^D = \psi^D(x^D)$ and the functions $\psi^R : \mathbb{R}^{M \times N} \mapsto \mathbb{R}^{M \times N}$ and $\psi^D : \mathbb{R}^{M \times N} \mapsto \mathbb{R}^{M \times N}$ represent the image transformations that the range and Doppler data undergo, respectively. For the ITNs we use residual convolutional autoencoders ([26], Chapter 14) with the same architecture details as [11].

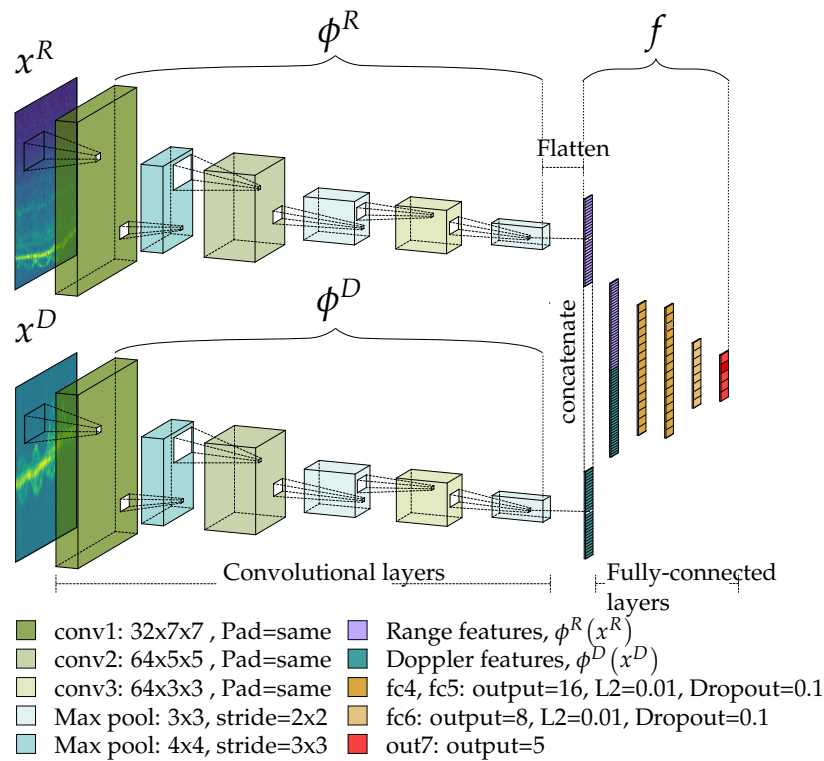


Figure 6. CNN architecture details.

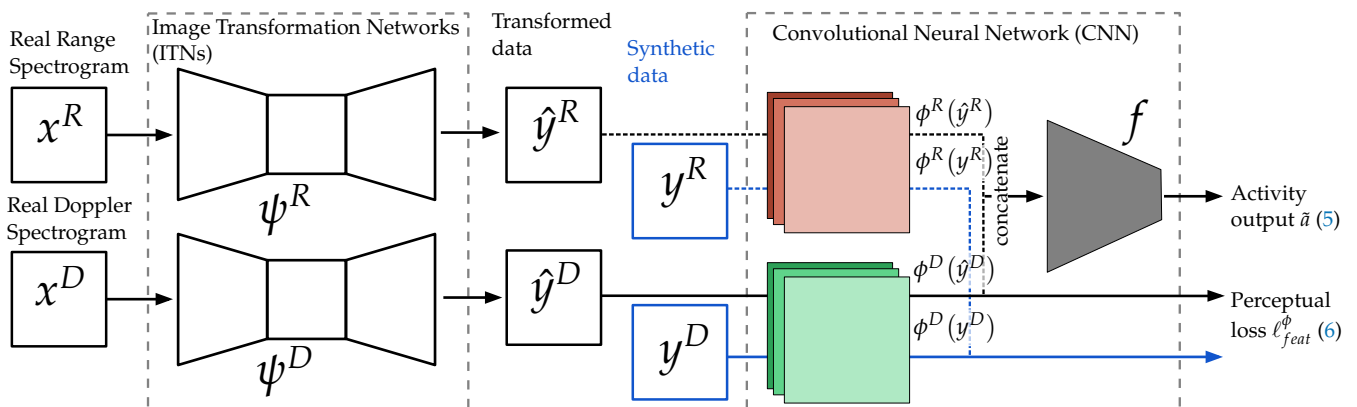


Figure 7. Architecture of RACPIT.

Here, we first train the CNN using labeled synthetic data y from source and target recordings. This is required since we apply perceptual loss for the training of ψ^R and ψ^D [11]. Perceptual loss uses the pre-trained CNN to compare the high-level perceptual and semantic features of the transformed data \hat{y} to the synthetic data y , which here serves as the ground truth. Once the CNN is trained, we freeze all of its layers and use the convolutional branches ϕ^R and ϕ^D to train the ITNs ψ^R and ψ^D with the following objective function:

$$\ell_{feat}^{\phi} = \frac{1}{2P} \left(\left\| \phi^R(\hat{y}^R) - \phi^R(y^R) \right\|_2^2 + \left\| \phi^D(\hat{y}^D) - \phi^D(y^D) \right\|_2^2 \right). \quad (6)$$

In that sense, instead of enforcing pixelwise resemblance, the ITNs try to generate similar feature representations extracted by the perceptual network, e.g., the CNN [11].

Implementation and Training

RACPIT has been written on PyTorch [27] from Daniel Yang's implementation of perceptual loss. The code is publicly available at <https://github.com/fraunhoferhhi/>

[racpit](#) (accessed on 14 February 2022). We have trained the full pipeline (CNN and ITNs) independently for every configuration in Table 1 and we have repeated each training experiment 5 times. The ITNs have been trained for 500 epochs; other than that we have retained most of the hyperparameter values in [11], including a batch size of 4 and Adam optimization with a learning rate of 1×10^{-4} . As their recommendation, we have kept the total variation regularization with a strength of 1×10^{-7} . Prior to that, we have pre-trained the CNN for 100 epochs with a batch size of 32, using early stopping and the same optimization method and learning rate.

Besides the training of RACPIT, we have also trained just the CNN with real data as a baseline. Likewise, these baseline experiments were repeated 5 times for every different configuration in Table 1. In total, we have performed 40 training experiments, which we have run as parallel tasks on an NVIDIA Tesla V100 SXM2 GPU with CUDA 11 and 4 GB allocated per task. The training duration of RACPIT on this hardware adds up to about 10 h per experiment on average.

4. Results

Once trained, the ITNs have a clearly visible denoising effect (Figure 8b,e) on their real inputs (Figure 8a,d). This is to be expected, since its target is the synthetic data that we create according to the noiseless model in Equation (3) (Figure 8c,f).

We test the full pipeline on our target domain \mathcal{T} . In order to assess the resulting accuracy in a quantitative way, we compare it to a baseline consisting of the CNN in Figure 6 trained only using real source data and tested only on the real target data. Figure 9 shows that RACPIT improves the accuracy for all configurations by 10% to 20%.

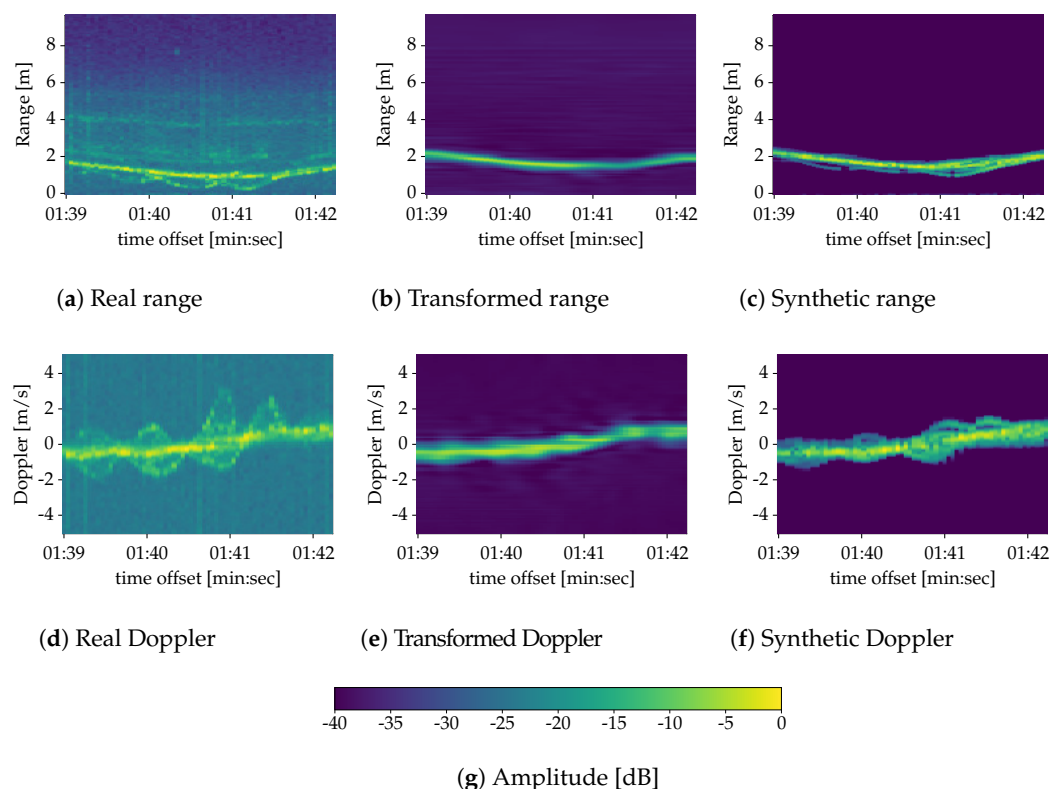


Figure 8. Comparison of range (a–c) and Doppler (d–f) spectrograms for real (a,d), transformed (b,e) and synthetic (c,f) data. The spectrograms display an instance of the activity item (e) in Section 2 Boxing and walking.

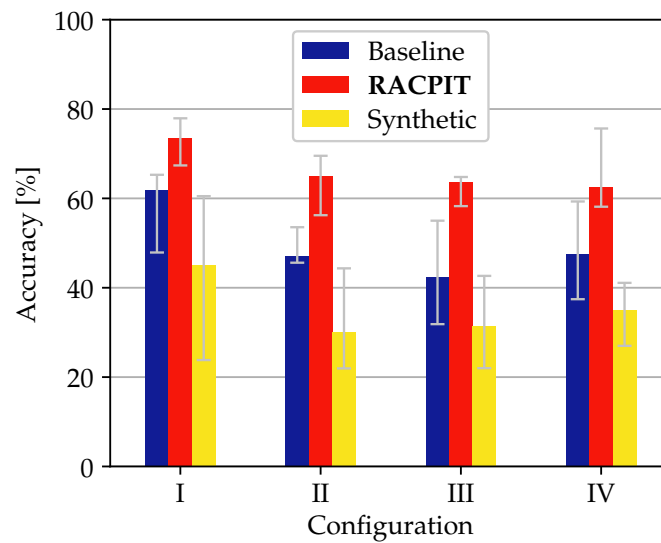


Figure 9. Experiment results. Each bar represents the median value out of five experiments, with the error bars indicating the minimum and maximum observed value. Besides the accuracy for the baseline and RACPIT, we also present the obtained accuracy when we directly feed synthetic data to the baseline.

Despite this improvement, the accuracy does not reach the 90% figures of previous works [9,12,13]. A reason for this can indeed be found in the denoising behavior of the ITNs, which not only filter out background noise but also some micro-Doppler features, as it can be seen in Figure 8. In any case, the results suggest that real to synthetic transformation is an interesting strategy to explore with room for improvement. Besides the accuracy, we have also computed the average F1 score [28] and average balanced accuracy [29] per configuration in Tables 2 and 3, respectively. These metrics indeed confirm the performance increment of RACPIT respect to the baseline.

Table 2. F1 score of the baseline and RACPIT for all different configurations.

Configuration	I	II	III	IV
Baseline	0.49	0.41	0.37	0.47
RACPIT	0.65	0.55	0.56	0.55

Table 3. Balanced accuracy of the baseline and RACPIT for all different configurations.

Configuration	I	II	III	IV
Baseline	0.53	0.44	0.40	0.48
RACPIT	0.67	0.57	0.57	0.58

Figure 9 also includes the results when synthetic data are fed to our baseline, which we had trained with real data. The poor accuracy, which goes as low as 20%, provides a quantitative confirmation of the dissimilarities between real and synthetic data and justifies the use of ITNs to bridge those.

5. Conclusions

In this paper, we have focused on improvements of deep learning methods for human activity classification on radar images. We have presented our own architecture, Radar Activity Classification with Perceptual Image Transformation (RACPIT), to mitigate the dependency of classification accuracy (and thus increase generalization) of Convolutional Neural Networks (CNNs) on the dataset recorded for the training process. RACPIT tackles this using Image Transformation Networks (ITNs) in combination with synthetic radar data,

which have been generated from a radar model using human motion data. Five different activities performed by two different test subjects have been recorded in an office-like lab. The measurements have been acquired by four Frequency-Modulated Continuous-Wave (FMCW) radar sensors located at the same position, operating at around 60 GHz with different configuration settings like bandwidth, chirp time, etc. For each sensor, about 50 min of radar and video footage recordings have been taken.

We have observed that the test subjects performed the activities quite differently across recordings, considering that these were interleaved with short pauses. Hence, the data deviated largely depending on the recording time. Accordingly, the resulting radar data and the pre-processed radar images diverge also largely from each other. The recordings were thus split into a source and a target dataset with different recording times. A large decrease in classification accuracy was observed for the CNN that had been trained with the source dataset upon testing it with the target dataset.

We have tried to solve this with RACPIT in the following manner. First, we have moved on from real to synthetic data for the training of our CNN to also incorporate the target recordings into it. Since we use a lightweight radar model for data synthesis, synthetic data present a large domain shift with respect to real data, which has been confirmed in our experiments. This has inspired us to include a next step where we have investigated the impact of ITNs on the real-synthetic domain shift. As such, we have used real and synthetic data from the source domain to train two ITNs that take real data on the input and transform them so that they resemble synthetic data. By including these synthetic data from the target domain in the training, we have found an increase in classification accuracy by up to 20%. Still, the classification accuracy performance remains about 15% lower than if the CNN had been trained using both source and target datasets with real radar data.

In future research, a further in-depth investigation could show if more advanced radar models can further increase classification accuracy. For this, Shooting and Bouncing Rays (SBR) or even full 3D solver techniques could be used to generate more sophisticated radar data from human motion models. These kinematic models could be improved by using methods like dense pose algorithms [30] to generate a fine mesh of the moving human body. Finally, RACPIT should be tested and verified further by using other datasets and activities, thus human motion models that were generated purely from camera data taken in another environment, e.g., another lab with different test subjects.

Supplementary Materials: The supporting information can be downloaded at: <https://github.com/fraunhoferhhi/racpit>.

Author Contributions: Conceptualization, R.H. and T.V.; methodology, R.H. and T.V.; software, R.H. and H.K.; validation, L.S.; formal analysis, R.H. and S.S.; investigation, R.H. and H.K.; resources, S.S. and L.S.; data curation, R.H. and H.K.; writing—original draft preparation, R.H., T.V. and H.K.; writing—review and editing, L.S., S.S., R.H. and T.V.; visualization, R.H. and H.K.; supervision, T.V. and S.S.; project administration, T.V. and S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the non-critical nature of the disclosed data and the compliance and agreement of all involved actors with Infineon's internal General Data Protection Regulation.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data analyzed in this study are property of Infineon Technologies AG and partially available on request from the corresponding author. The data are not publicly available due to privacy issues with Infineon and due to its internal General Data Protection Regulation (in the case of video data). Code and Supplementary Material are publicly available at <https://github.com/fraunhoferhhi/racpit>, accessed on 14 February 2022.

Acknowledgments: We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla V100 SXM2 used for this research. Likewise, we would like to thank Avik Santra from Infineon Technologies AG for his support and wise advice throughout this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
CV	Computer Vision
GAN	Generative Adversarial Network
IF	Intermediate Frequency
ITN	Image Transformation Network
FMCW	Frequency-Modulated Continuous-Wave
FFT	Fast Fourier Transform
ML	Machine Learning
MOCAP	MOtion CAPture
MTI	Moving Target Indication
RACPIT	Radar Activity Classification with Perceptual Image Transformation
RCS	Radar Cross Section
RDM	Range-Doppler Map
SBR	Shooting and Bouncing Rays
UWB	Ultra-wideband

References

- Hazra, S.; Santra, A. Robust Gesture Recognition Using Millimetric-Wave Radar System. *IEEE Sens. Lett.* **2018**, *2*, 7001804. [[CrossRef](#)]
- Stephan, M.; Santra, A. Radar-Based Human Target Detection using Deep Residual U-Net for Smart Home Applications. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 175–182. [[CrossRef](#)]
- Stephan, M.; Stadelmayer, T.; Santra, A.; Fischer, G.; Weigel, R.; Lurz, F. Radar Image Reconstruction from Raw ADC Data using Parametric Variational Autoencoder with Domain Adaptation. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 9529–9536. [[CrossRef](#)]
- Redko, I.; Morvant, E.; Habrard, A.; Sebban, M.; Bennani, Y. *Advances in Domain Adaptation Theory*; Elsevier: Amsterdam, The Netherlands, 2019. [[CrossRef](#)]
- Kim, Y.; Moon, T. Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 8–12. [[CrossRef](#)]
- Shah, S.A.; Fioranelli, F. Human Activity Recognition: Preliminary Results for Dataset Portability using FMCW Radar. In Proceedings of the 2019 International Radar Conference (RADAR), Toulon, France, 23–27 September 2019; pp. 1–4. [[CrossRef](#)]
- Du, H.; Jin, T.; Song, Y.; Dai, Y. Unsupervised Adversarial Domain Adaptation for Micro-Doppler Based Human Activity Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 62–66. [[CrossRef](#)]
- Li, X.; Jing, X.; He, Y. Unsupervised Domain Adaptation for Human Activity Recognition in Radar. In Proceedings of the 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020; pp. 1–5. [[CrossRef](#)]
- Khodabakhshandeh, H.; Visentin, T.; Hernangómez, R.; Pütz, M. Domain Adaptation Across Configurations of FMCW Radar for Deep Learning Based Human Activity Classification. In Proceedings of the 2021 21st International Radar Symposium (IRS), Berlin, Germany, 21–22 June 2021; pp. 1–10. [[CrossRef](#)]
- Chen, V.C. *The Micro-Doppler Effect in Radar*; Artech House: Norwood, MA, USA, 2011.
- Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *arXiv* **2016**, arXiv:1603.08155.
- Hernangómez, R.; Santra, A.; Stańczak, S. Human Activity Classification with Frequency Modulated Continuous Wave Radar Using Deep Convolutional Neural Networks. In Proceedings of the 2019 International Radar Conference (RADAR), Toulon, France, 23–27 September 2019; pp. 1–6. [[CrossRef](#)]
- Hernangómez, R.; Santra, A.; Stańczak, S. A Study on Feature Processing Schemes for Deep-Learning-Based Human Activity Classification Using Frequency-Modulated Continuous-Wave Radar. *IET Radar Sonar Navig.* **2020**, *15*, 932–944. [[CrossRef](#)]
- Gurbuz, S.Z.; Rahman, M.M.; Kurtoglu, E.; Macks, T.; Fioranelli, F. Cross-frequency training with adversarial learning for radar micro-Doppler signature classification (Rising Researcher). In *Radar Sensor Technology XXIV*; International Society for Optics and Photonics: Bellingham, WA, USA, 2020; Volume 11408. [[CrossRef](#)]

15. Gusland, D.; Christiansen, J.M.; Torvik, B.; Fioranelli, F.; Gurbuz, S.Z.; Ritchie, M. Open Radar Initiative: Large Scale Dataset for Benchmarking of micro-Doppler Recognition Algorithms. In Proceedings of the 2021 IEEE Radar Conference (RadarConf21), Atlanta, GA, USA, 7–14 May 2021; pp. 1–6. [CrossRef]
16. Yeo, H.S.; Minami, R.; Rodriguez, K.; Shaker, G.; Quigley, A. Exploring Tangible Interactions with Radar Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–25. [CrossRef]
17. Zhang, A.; Nowruzi, F.E.; Laganieri, R. RADDet: Range-Azimuth-Doppler based Radar Object Detection for Dynamic Road Users. In Proceedings of the 2021 18th Conference on Robots and Vision (CRV), Burnaby, BC, Canada, 26–28 May 2021.
18. Stolz, M.; Schubert, E.; Meinl, F.; Kunert, M.; Menzel, W. Multi-target reflection point model of cyclists for automotive radar. In Proceedings of the 2017 European Radar Conference (EURAD), Nuremberg, Germany, 11–13 October 2017; pp. 94–97. [CrossRef]
19. De la Torre, F.; Hodgins, J.; Montano, J.; Valcarcel, S.; Forcada, R.; Macey, J. *Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database*; Robotics Institute, Carnegie Mellon University: Pittsburgh, PA, USA, 2009; Volume 5.
20. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
21. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (ICCV), Venice, Italy, 22–29 October 2017.
22. Richards, M.A.; Scheer, J.; Holm, W.A.; Melvin, W.L. *Principles of Modern Radar*; Citeseer: Princeton, NJ, USA, 2010.
23. Manolakis, D.G.; Ingle, V.K.; Kogon, S.M. *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*; Artech House Signal Processing Library, Artech House: Boston, MA, USA, 2005.
24. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. Available online: <https://github.com/facebookresearch/detectron2> (accessed on 14 February 2022).
25. Boulic, R.; Thalmann, N.M.; Thalmann, D. A global human walking model with real-time kinematic personification. *Vis. Comput.* **1990**, *6*, 344–358. [CrossRef]
26. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; Adaptive Computation and Machine Learning; The MIT Press: Cambridge, MA, USA, 2016.
27. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
28. Wu, X.Z.; Zhou, Z.H. A unified view of multi-label performance measures. In Proceedings of the 34th International Conference on Machine Learning (ICML'17), Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 3780–3788.
29. Brodersen, K.H.; Ong, C.S.; Stephan, K.E.; Buhmann, J.M. The Balanced Accuracy and Its Posterior Distribution. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 3121–3124. [CrossRef]
30. Guler, R.A.; Neverova, N.; Kokkinos, I. DensePose: Dense Human Pose Estimation in the Wild. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7297–7306. [CrossRef]