Bitlis Eren Üniversitesi Fen Bilimleri Dergisi
BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 13 NO: 3 PAGE: 578-586 YEAR: 2024
DOI:10.17798/bitlisfen.1457065

Enhancing Radar Image Classification with Autoencoder-CNN Hybrid System

Kürşad UÇAR[1*]

[1]*Department of Electrical-Electronics Engineering, Faculty of Technology, Selcuk University, Konya, Turkey*
(ORCID: 0000- 0001-5521-2447)

Keywords: Autoencoder Reconstruction, Behind-the-Wall Monitoring, CNN Classification, Human Movement Tracking, Radar Systems.

Abstract
The tracking, analysis, and classification of human movements can be crucial, particularly in areas such as elderly care, healthcare, and infant care. Typically, such tracking is done remotely with cameras. However, radar systems have emerged as significant methods and tools for these tasks due to their advantages such as privacy, wireless operation, and the ability to work through walls. By converting reflected radar signals from targets into images, human activities can be classified using powerful classification tools like deep learning. In this study, range-Doppler images of behind-the-wall human movements obtained with a radar system consisting of one transmitter and four receiver antennas were classified. Since the data collected from the four receiver antennas are in different positions, the collected reflection signals also differ. The signals collected with the range-time matrix content were divided into positive and negative parts, resulting in eight images from the four antennas. Instead of using all the data in CNN training, the images were first subjected to a reconstruction process with an autoencoder to reduce differences. The reproduced images were then classified with CNN. Moreover, the classification success is increased by 8.50% with the proposed method compared to classification only with CNN. In conclusion, it was observed that the classification success of radar images can be increased by using a hybrid system with an autoencoder to reconstruct the images before classification with CNN.

1. Introduction

Electronic devices have begun to play an active role in daily life, such as machines, robots, and cameras, due to their more effective operation compared to humans in various fields [1]-[3]. These devices are advantageous due to their speed, continuity, lower energy consumption, and reduced errors. Additionally, they can prioritize important values such as privacy, confidentiality, and security over humans [4]. Given all these advantages, leveraging technology in critical tasks such as monitoring the elderly, children, and patients is highly appealing. It requires constant monitoring of individuals who may be in dangerous situations or require immediate intervention. Continuous monitoring implies observing every moment of these individuals, potentially violating their private spaces. Therefore, radar devices and systems have started to be used in such tasks as an alternative to devices like cameras that directly display people.

Radar (Radio Detection and Ranging) is a system that uses electromagnetic waves to determine the position and movement of objects [5], [6]. Radars are commonly used in various fields such as aircraft, ships, vehicles, and airports. Radar begins its operation by emitting electromagnetic waves at a specific frequency. These waves are typically radio waves or microwaves. When these emitted waves encounter target objects, reflection occurs. The reflected waves travel back from the object and are received by the radar. The collected waves are

*Corresponding author: kucar@selcuk.edu.tr
Received: 22.03.2024, Accepted: 18.07.2024

processed by the radar system. This process involves analyzing the timing, frequency, and power of the waves to determine the position, distance, and speed of the target object [7], [8]. Additionally, inference can be made based on the converted signals into images [9]-[11].

Radar systems, by converting radar signals into images, have become an area of interest for researchers in classifying human movements [11]-[14]. Since reflections in collected signals differ based on movements, motion analysis can be performed from the generated images [15]-[17]. However, due to the different semantic relationships between neighboring pixels in such images compared to classical images, feature extraction or image enhancement cannot be performed using classical image processing methods [18]. Although these images may be meaningless to humans, deep learning methods can establish meaningful relationships between pixels thanks to their strong structures. Therefore, deep learning tools developed for image processing play a critical role in achieving effective results on radar images [15]. Among these algorithms, autoencoders have become preferred due to their powerful features such as noise reduction and image reconstruction. Autoencoders, with their deep learning structure, have the ability to reproduce an image similar to the target images.

While a human can understand real-world images by examining them and easily identify shapes, activities, or objects in the image, radar or frequency domain images showing different features of the environment do not make sense to humans. It is very difficult for individuals to make an inference from such images at first glance. Even non-experts would perceive these images as completely meaningless. However, with powerful classification algorithms, it is possible for machines to understand such images. In this context, many studies have been conducted to classify radar images using machine learning algorithms [11], [19], [20]. Radar signals reflected from the human chest have been used to perform tasks such as detecting living beings or counting pulses [21], [22]. Radar signals have also been used to detect instances of individuals in need of care, such as monitoring breathing and detecting falls [23]-[25]. Additionally, it has been shown that different movements can be detected using images created with radar signals [26], [27].

In [11], it was shown that images of signals from different antennas for human activities could be classified with CNN. It was reported that interference and noise differences in signals received with 4 antennas were suitable for data augmentation for CNN, known to provide better results with more data, and the amount of data was increased eightfold with the structure created. In this study, instead of using all the data for CNN training, an autoencoder-CNN hybrid system was proposed to improve images by reconstructing them with an autoencoder [28]. Three different combinations were created for training and testing the autoencoder, and to compare the results, the CNN structure used in [11] was employed.

The contributions of this study are as follows.

• Classification of images obtained with a multi-antenna structure by increasing the similarity to each other in the classification of human movements behind walls,

• Improving classification performance with fewer images.

The rest of the paper is organized as follows: Section 2 provides the autoencoder and dataset. Section 3 presents the experiments and results. The last section concludes the study.

2. Material and Method

2.1. Autoencoder

Autoencoders, unsupervised learning techniques that utilize neural networks for representation learning [29]. Essentially, it is a type of artificial neural network that attempts to learn the original representation of input data. Autoencoder can be used in many applications such as data compression, noise reduction, dimensionality reduction, and feature learning.

Autoencoder consists of two main components:

Encoder: The part that transforms the input data into a lower-dimensional representation.

Decoder: The part that reconstructs the original input data from the lower-dimensional representation generated by the encoder.

During the training of the autoencoder, the model first transforms the input data into a lower-dimensional representation with the encoder and then attempts to reconstruct the original data using the decoder. In this process, the model tries to minimize the difference between the input and output. Thus, the model learns the original representation of the input data.

Autoencoder is used to compress multi-dimensional data into hidden space first and then reconstruct the compressed data from the compressed hidden space [30]. The network

architecture consists of a neural network that creates a compressed representation of the original input and then recreates it. When input features are independent of each other, this compression and subsequent reconstruction become a very challenging task. However, if there is a kind of learnable structure in the data, this structure can be learned and used to force the input through the bottleneck of the network.

The network takes an unlabeled dataset as input and can be summarized as the reconstruction of the original input x in the framework of a supervised learning problem that produces the output x^. The training of the network can be achieved by minimizing the reconstruction error L(x,x^), which measures the differences between the original input and the reconstruction [31]. The bottleneck is a crucial step in the network; with the bottleneck, the input cannot directly pass to the output, thus preventing memorization.

The ideal autoencoder model should be sensitive enough to the inputs when reconstructing the outputs based on inputs. At the same time, sensitivity should not be too excessive for problems such as memorizing or overfitting the training data. This balance forces the model to preserve variations in the data required to reconstruct the input without retaining redundancies in the input. In most cases, this involves creating a loss function that encourages the model to be sensitive to the inputs (i.e., reconstruction loss L(x,x^)) and another term that discourages memorization/overfitting (i.e., an additional regularizer).

Figure 1 shows the autoencoder structure used. A 104x40 pixel input image passes through two convolution layers with 3x3 kernel sizes to reach the fully connected layer. Then, the convolution operations are repeated in reverse, and the image is reconstructed. Here, the training process aims to obtain images similar to the target image from the input image. Two hidden layers are used in autoencoder. Layer sizes consist of 8 and 16 neurons, respectively. This structure was determined based on trial and error method and cross-validation results.

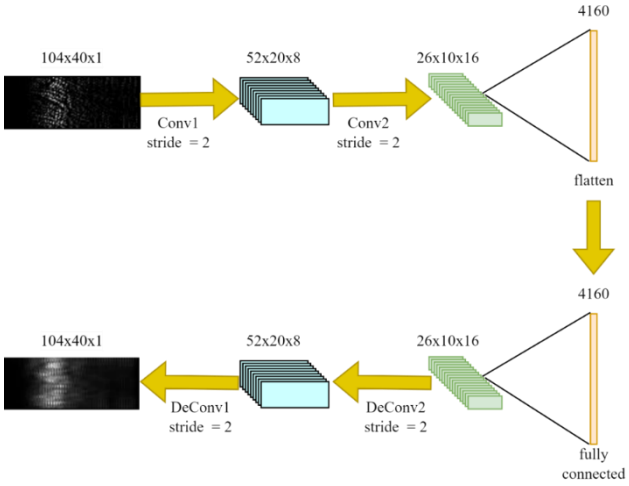

Figure 1. Autoencoder structure

2.2. CNN

Convolutional Neural Network (CNN) is a deep learning architecture widely used especially in image and video recognition, image classification, object detection and similar tasks. CNNs, unlike classical artificial neural networks, are much more successful in learning spatial and temporal relationships within data. Convolutional layer, activation function, pooling layer, fully connected layer are the basic components of CNN. The convolutional layer is applied by shifting filters (kernels) of certain sizes on the input data. Each filter is used to recognize certain features in the image, such as edges, corners. Each application of the filter produces an output called a feature map. Activation function is generally used to produce the output of the layer after convolutional layers. Pooling layer reduces the size of feature maps. Thus, computational cost and memory usage are reduced. The most commonly used type of pooling is Max Pooling, this type of pooling selects the maximum value in a particular region. The last layer, the fully connected layer, connects all neurons to each other, as in classical artificial neural networks. A flattened vector of feature maps is given as input to this layer. This layer is used as the last layer in classification tasks and provides probability distribution between classes with the softmax activation function [32]. The CNN architecture used in [11] is as shown in Figure 2. After 2 convolutional and pooling layers, output estimation is performed in the fully connected layer. The numbers of 3x3 sized kernels in the convolutional layers are 32 and 64, respectively.

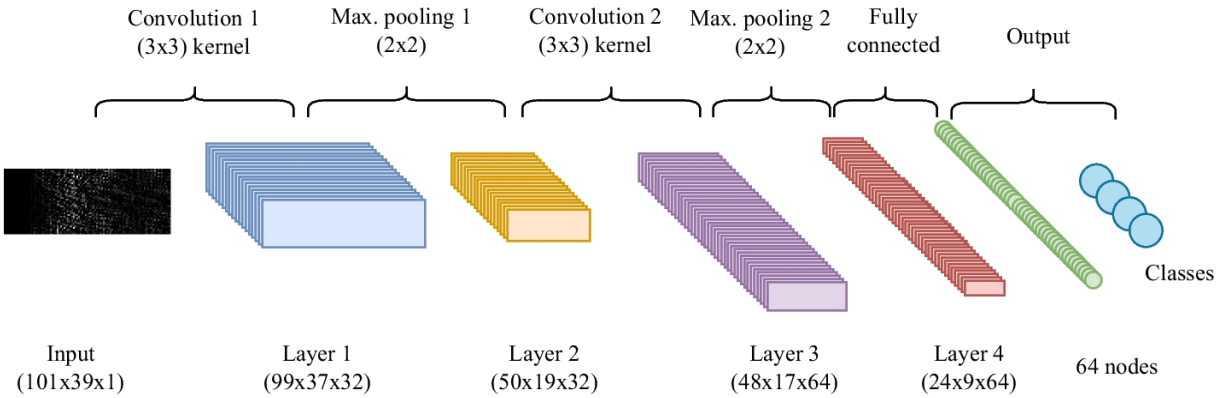

Figure 2. Used CNN model [11]

2.3. Dataset

The dataset used [11] consists of four human activities: running, walking with swinging arms, waving, and walking with steady arms. 50 samples were collected for each activity using 4 antennas. After obtaining grayscale images from the 4 antennas, the method described in Figure 3 [33] was used to double the signals using the negative and positive parts. Thus, there are a total of 1600 images in the dataset, with 400 images for each class. While the image dimensions are 101x39 pixels, they are resized to 104x40 pixels with zero padding to fit the input of the autoencoder. Due to the structure of the autoencoder, the images are reconstructed to be of size 104x40 pixels. Therefore, the original images are given to the autoencoder with zero padding to match the output size. Figure 4 shows the input and output images of the autoencoder for each class. Although the generated images may appear different from the original images, they will only be used in the training and testing of the classifier (CNN) in the study, hence they will not have a detrimental effect on the classification.

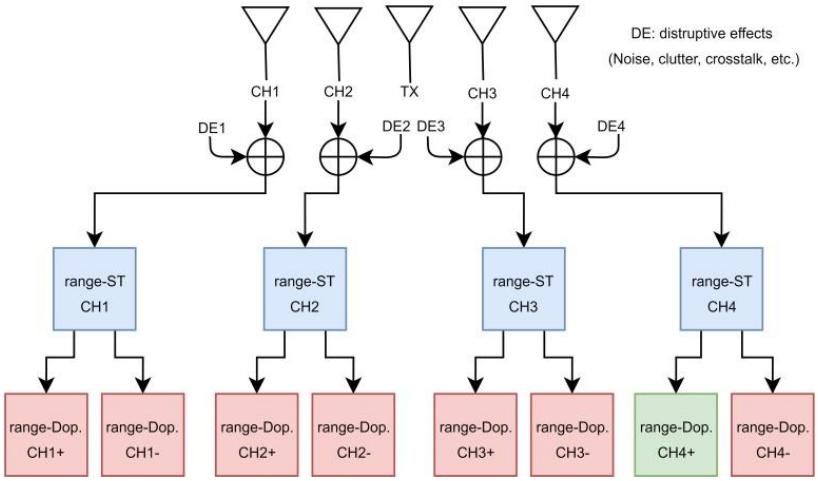

Figure 3. Antenna array [11]

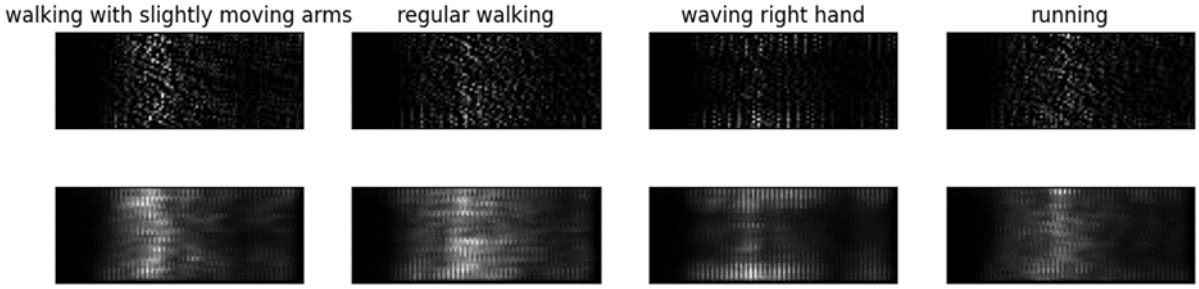

Figure 4. Original images (first row) and images created with autoencoder (last row)

2.4. Proposed Method

The proposed autoencoder-CNN architecture is as shown in Figure 5. The system reconstructs the images with Autoencoder before classification. The output of the Autoencoder is fed into CNN for classification, and as a result, the classes of the movements are obtained.

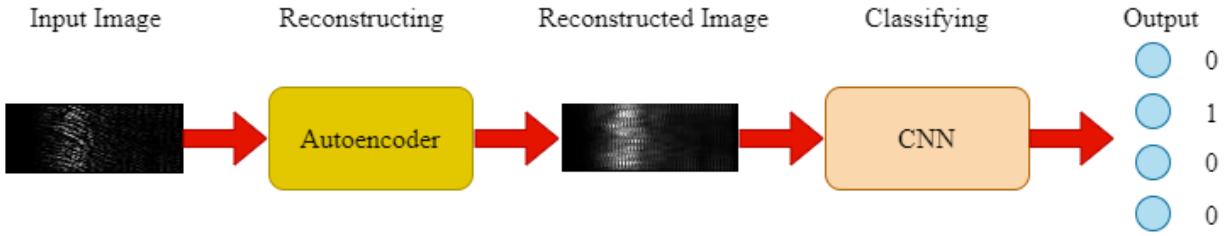

Figure 5. Proposed autoencoder-CNN hybrid model

3. Experimental Result and Discussion

For both the training and testing of the autoencoder, the dataset is further divided into two groups: one for reconstructing the input images and the other for the target images. To investigate the effect of image reconstruction by the autoencoder on CNN classification, the dataset [11] is divided into different groups and three approaches are tried. In [11], it is stated that during classification, ch4+ (the positive signal part of the 4th receiver antenna) yields better results compared to other channels. From this statement, it can be inferred that positive channels may perform better for classification compared to negatives. Therefore, in the experiments, the positive parts of the channels are used as target images for the autoencoder, while the negatives are used as input images to be reconstructed. In other words, the negative region images are attempted to be made similar to the positive ones. Additionally, care has been taken to ensure that the target and input images are taken from the same sample. Table 1 provides the images used for autoencoder training and testing for the three experiments. Table 2 presents the image channels and numbers classified with CNN in these experiments.

Table 1. Autoencoder train and test image channels

Experiment no	Train input	Train target	Test input	Test target
Experiment 1	Ch1-, Ch4-	Ch1+, Ch4+	Ch2-, Ch3-	Ch2+, Ch3+
Experiment 2	Ch1-, Ch4-	Ch1+, Ch4+	Ch2-, Ch3-	Ch2+, Ch3+
Experiment 3	Ch1-, Ch2-	Ch1+, Ch2+	Ch3-	Ch3+

Table 2. CNN train, test images channel and number

Experiment no	Used channels	Number of images
Experiment 1	Ch1+, Ch2+, Ch3+, Ch4+, Ch1-, Ch2-, Ch3-, Ch4-	1600
Experiment 2	Ch2-, Ch3-, Ch2+, Ch3+	800
Experiment 3	Ch4+	200

In the three experiments for the autoencoder, the parameters listed in Table 3 have been utilized.

Table 3. Autoencoder parameters

Parametre	Chosen parameter
Epoch	15
Optimizer	Adam
Loss func	Mean Squerad Error (MSE)
Padding	Same
Activation func.	ReLu

In Experiment 1, the classification success of the entire dataset, as observed in [11], was monitored. The dataset was divided into four parts for both training and testing of the autoencoder. According to the antenna arrangement depicted in Figure 3, the images from the two antennas farthest from each other were used for training, while the images from the other antennas were used for testing. After 15 epochs, the training error was 0.0060, and the test error was 0.0085. Following the training of the autoencoder, the

images from the entire dataset were reconstructed. These reconstructed images were randomly distributed and used for training and testing of the CNN.

In Experiment 1, since the first and fourth channels were used for training the autoencoder, when fed back into the autoencoder for classification in the CNN, there might be more similarity compared to the second and third channels. In other words, after successful autoencoder training, the channels used in training would be more similar to the ones used in testing. This similarity could lead to inconsistencies in the data. Therefore, performing classification in the CNN using data not used in the autoencoder training ensures balanced similarity across all data. In Experiment 2, the aim was to classify only the images used in the testing of the autoencoder. Thus, the success of classifying images not included in the training data of the autoencoder was monitored. After 15 epochs, the training error was 0.0047, and the test error was 0.0070. Since [11] did not present the classification results using only these images, the CNN from [11] was trained and tested using the original images.

In the final experiment, the effect of reconstructing the ch4+ images with the autoencoder, which exhibited the highest classification success in [11], was demonstrated. In this experiment, images from channels other than the fourth channel were used for both training and testing of the autoencoder, while the CH4- images were not used in either the autoencoder or the CNN. After 15 epochs, the training error was 0.0036, and the test error was 0.0065. The results of all experiments are presented in Table 4, based on the classification results of the images used in [11].

Table 4. Classification accuracy of CNN

Experiment no	Original images	Reconstructed images
Experiment 1	90.25% [11]	91.87%
Experiment 2	85.88%	91.12%
Experiment 3	84.50% [11]	93.00%

Upon examination of Table 4, it is observed that as the amount of data decreases in the original images, the classification success also decreases. Conversely, with the proposed method, the situation is reversed. Overall, in all experiments, it is evident that the images reconstructed with the autoencoder improve the classification success. In Experiment 1, where classification was performed with all images from [11], the accuracy rate increased by 1.62%. In Experiment 2, where only the test images reconstructed by the autoencoder were used, significantly higher accuracy rates were achieved compared to the original images. With the proposed autoencoder-CNN hybrid method in this study, approximate values were obtained with less data in terms of accuracy. Since the training target images were also included in the CNN training and test data in Experiment 1, it is natural to observe higher classification success. The effect of the proposed method is very clear in Experiment 3. With only 200 ch4+ images, 8.50% higher accuracy was achieved compared to the original images. The accuracy value achieved in Experiment 3 is quite high compared to the results presented in [11]. It has been demonstrated that high accuracy can be achieved with the autoencoder without requiring a large amount of data.

Autoencoders can perform data cleaning by reducing noise. This can improve classification performance by reducing interference and noise in radar images. The hybrid model provides the combination of both low-dimensional feature representation and spatial features. This allows for a more powerful and effective data representation. Because hybrid models can learn both low-dimensional features and spatial layout, their generalization ability is generally high. This can be effective in recognizing different types of human movements.

Hybrid models generally require more computing power and memory. Training both Autoencoder and CNNs can be computationally intensive. The complexity of the hybrid model can create challenges during the design and training of the model. This requires careful adjustment of the model's settings and hyperparameters. Hybrid models generally require more data. Without a sufficient amount and variety of data, the model's ability to generalize may be limited.

Autoencoder training took approximately 10.45 seconds with NVIDIA Tesla K80 with 12GB of VRAM GPU. Memory usage was 1543.55 MB. Reconstructing an image with the autoencoder took approximately 0.55 μs. Classifying the reconstructed image with CNN takes 0.83 μs.

4. Conclusion and Suggestions

Autoencoders are deep learning methods used to reconstruct images. They take an input image, pass it through a convolution process, and then through a bottleneck before reconstructing the image through reverse convolution. In this study, an autoencoder-based approach is proposed to enhance the classification accuracy of radar images generated

with stepped-frequency continuous-wave (SFCWR) and Uniform Linear Array (ULA) structures. In the previous study [11], it was mentioned that when all images were used in CNN training and testing, the classification accuracy increased. However, in this study, unlike [11], the augmented data was used for training and testing the autoencoder. As a result, this paper demonstrates that the proposed autoencoder-CNN hybrid approach can achieve significantly higher accuracy with less data compared to the previous study.

Acknowledgment

I would like to thank Dr. Yunus Emre ACAR for sharing the dataset for this study.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

References

[1] W. Heng, S. Solomon, and W. Gao, "Flexible electronics and devices as human–machine interfaces for medical robotics," *Advanced Materials*, vol. 34, no. 16, p. 2107902, 2022.

[2] M. Javaid, A. Haleem, S. Rab, R. P. Singh, and R. Suman, "Sensors for daily life: A review," *Sensors International*, vol. 2, p. 100121, 2021.

[3] D. S. Nunes, P. Zhang, and J. S. Silva, "A survey on human-in-the-loop applications towards an internet of all," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 944-965, 2015.

[4] J. B. Awotunde, R. G. Jimoh, S. O. Folorunso, E. A. Adeniyi, K. M. Abiodun, and O. O. Banjo, "Privacy and security concerns in IoT-based healthcare systems," in *Internet of Things*, Cham: Springer International Publishing, 2021, pp. 105-134.

[5] A. E. Onoja, A. M. Oluwadamilola, and L. A. Ajao, "Embedded system based radio detection and ranging (RADAR) system using Arduino and ultra-sonic sensor," *American Journal of Embedded Systems and Applications*, vol. 5, no. 1, pp. 7-12, 2017.

[6] A. Biswas, S. Abedin, and M. A. Kabir, "Moving object detection using ultrasonic radar with proper distance, direction, and object shape analysis," *Journal of Information Systems Engineering and Business Intelligence*, vol. 6, no. 2, pp. 99-111, 2020.

[7] M. I. Skolnik, "Introduction to radar," *Radar handbook*, vol. 2, p. 21, 1962.

[8] A. M. Ponsford, L. Sevgi, and H. C. Chan, "An integrated maritime surveillance system based on high-frequency surface-wave radars. 2. Operational status and system performance," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 5, pp. 52-63, 2001.

[9] A. Reigber et al., "Very-high-resolution airborne synthetic aperture radar imaging: Signal processing and applications," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 759-783, 2012.

[10] S. Hazra and A. Santra, "Short-range radar-based gesture recognition system using 3D CNN with triplet loss," *IEEE Access*, vol. 7, pp. 125623-125633, 2019.

[11] Y. E. Acar, K. Ucar, I. Saritas, and E. Yaldiz, "Classification of human target movements behind walls using multi-channel range-doppler images," *Multimedia Tools and Applications*, pp. 1-18, 2023.

[12] X. Li, Y. He, and X. Jing, "A survey of deep learning-based human activity recognition in radar," *Remote Sensing*, vol. 11, no. 9, p. 1068, 2019.

[13] H. T. Le, S. L. Phung, and A. Bouzerdoum, "Human gait recognition with micro-Doppler radar and deep autoencoder," *in 2018 24th International Conference on Pattern Recognition (ICPR), 2018*: IEEE, pp. 3347-3352.

[14] Y. Shao, S. Guo, L. Sun, and W. Chen, "Human motion classification based on range information with deep convolutional neural network," *in 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 2017: IEEE*, pp. 1519-1523.

[15] P. van Dorp and F. Groen, "Feature-based human motion parameter estimation with radar," *IET Radar, Sonar & Navigation*, vol. 2, no. 2, pp. 135-145, 2008.

[16] A. Sume et al., "Radar detection of moving targets behind corners," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2259-2267, 2011.

[17] X. Li, Z. Li, F. Fioranelli, S. Yang, O. Romain, and J. L. Kernec, "Hierarchical radar data analysis for activity and personnel recognition," *Remote Sensing*, vol. 12, no. 14, p. 2237, 2020.

[18] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern recognition letters*, vol. 119, pp. 3-11, 2019.

[19] M. Zenaldin and R. M. Narayanan, "Radar micro-Doppler based human activity classification for indoor and outdoor environments," *in Radar Sensor Technology XX*, 2016, vol. 9829: SPIE, pp. 364-373.

[20] F. Qi, H. Lv, F. Liang, Z. Li, X. Yu, and J. Wang, "MHHT-based method for analysis of micro-Doppler signatures for human finer-grained activity using through-wall SFCW radar," *Remote Sensing*, vol. 9, no. 3, p. 260, 2017.

[21] M. He, Y. Nian, and Y. Gong, "Novel signal processing method for vital sign monitoring using FMCW radar," *Biomedical Signal Processing and Control*, vol. 33, pp. 335-345, 2017.

[22] I. Seflek, Y. E. Acar, and E. Yaldiz, "Small motion detection and non-contact vital signs monitoring with continuous wave doppler radars," *Elektronika ir elektrotechnika*, vol. 26, no. 3, pp. 54-60, 2020.

[23] G. Diraco, A. Leone, and P. Siciliano, "A radar-based smart sensor for unobtrusive elderly monitoring in ambient assisted living applications," *Biosensors*, vol. 7, no. 4, p. 55, 2017.

[24] F. Fioranelli, J. Le Kernec, and S. A. Shah, "Radar for health care: Recognizing human activities and monitoring vital signs," *IEEE Potentials*, vol. 38, no. 4, pp. 16-23, 2019.

[25] K. Hanifi and M. E. Karsligil, "Elderly fall detection with vital signs monitoring using CW Doppler radar," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 16969-16978, 2021.

[26] S. Z. Gurbuz and M. G. Amin, "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16-28, 2019.

[27] S. Nag, M. A. Barnes, T. Payment, and G. Holladay, "Ultrawideband through-wall radar for detecting the motion of people in real time," *in Radar Sensor Technology and Data Visualization*, 2002, vol. 4744: SPIE, pp. 48-57.

[28] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval," *in ESANN*, 2011, vol. 1: Citeseer, p. 2.

[29] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3D shape retrieval," *Neurocomputing*, vol. 204, pp. 41-50, 2016.

[30] T. Liu, J. Wang, Q. Liu, S. Alibhai, T. Lu, and X. He, "High-ratio lossy compression: Exploring the autoencoder to compress scientific data," *IEEE Transactions on Big Data*, vol. 9, no. 1, pp. 22-36, 2021.

[31] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels," *Advances in neural information processing systems*, vol. 32, 2019.

[32] D. Bhatt et al., "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[33] Y. E. Acar, I. Saritas, and E. Yaldiz, "An experimental study: Detecting the respiration rates of multiple stationary human targets by stepped frequency continuous wave radar," *Measurement*, vol. 167, p. 108268, 2021.