Generating micro-Doppler Maps of patient activities with Generative Adversarial Network

Adnan Kerem Aksoy†, Burak Karabulut†, Arda Ertanhan†

***Abstract*—Nowadays Human Activity Recognition (HAR) is used in many areas. In the hospital field, we can use HAR for patients. But for privacy reasons cameras cannot be used in this field. As result, for this field, they started to use radar systems. With raw radar data, we can create Range Doppler (RD) maps and from RD maps we can create Micro Doppler maps (MD). From the MD maps, we can understand the activity of the patient in the rooms with neural networks. But to train neural networks and get better results we need to gather more data. Currently there is a way to generate data from the already existing data. GAN networks can generate artificial data by being trained by the original data. In the scope of the project the main purpose to use GAN to generate Micro Doppler maps in this field is, because, collecting real micro-Doppler examples is expensive, it requires access to restricted environments (hospitals), and the generation of realistic samples with GAN can reduce the amount of real data to collect. With up-sampling, down-sampling, and the activity function we have used in our GAN, we were able to create micro-Doppler data that is similar to the micro-Doppler data we had. Using the provided GAN architecture with a hardware that is a better fit than ours, the results can be improved.**

***Index Terms*—Human Activity Recognition, Supervised Learning, Optimization, Generative Adverserial Networks, Neural Networks, Convolutional Neural Networks**

I. Introduction

The aging population and the acute scarcity of medical professionals are two significant issues for today's healthcare systems [1]. Patients are more frequently observed these days utilizing equipment ranging from IOT sensors to various medical sensors. One of the key components of numerous intelligent surveillance systems, from smart homes to patient health monitoring tools, is the identification of indoor human activity. Video cameras are frequently used as the main sensors in these surveillance systems. Wide viewing angles, high-definition resolution, and cost-effectiveness are some benefits of video-based surveillance systems. However, they exhibit fundamental flaws such as being ineffective in bad weather and low-light conditions, making it harder to spot people wearing disguises, etc. Moreover, video-based sensors are invasive and might not be a viable option in situations where privacy is vital. Radars, on the other hand, can function in a variety of hazardous situations, including rain, fog, dust, darkness, smoke, and heat. Radar devices are also non-intrusive and privacy-preserving, which makes it easier for them to be used in settings where privacy is highly valued. Additionally, they have the ability to detect through obstacles like walls or other obstructions [2]. The radar sensors are an obvious solution for indoor human activity detection where privacy is important because of these characteristics of the radar sensors.

An electromagnetic radio signal is transmitted by a radar instrument in its line of sight, and targets and objects reflect this signal. After a brief interval, the receiver then picks up the signal that was reflected. The range and angle of the target are determined using the signal that was received. The frequency shift in the received signal can be utilized to track a moving target in order to calculate the moving object's speed. The superposition of all the reflected signals can be visualized as a Micro-Doppler (MD) signature when there are numerous independently moving objects. In the scope of this project, public radar data generated in both hospital and homelab environment which is mentioned as an innovative 240 square meter independent residential test environment for IOT,ss used to train the GAN network to generate data, thus further labeled the generated data.

The main contribution provided by this research is Human micro-Doppler data generation for patient activity recognition by using public data provided by X: Collection of real human micro-Doppler data is expensive in terms of cost and time which requires access to restricted environments such as hospitals which are almost always actively used for serving purposes. Generating human micro-Doppler data by GANs reduces the need to collect real data up to some extent. Moreover, being an active research topic, it is contributed to academia for further research. Instead, what can be considered as a secondary contribution of our research is 'Classification of the Data Generated by GAN': There is not much work conducted, if not any, to generate human micro-doppler data further used to classify patient activity. The data generated by GAN is further classified by a label that belongs to 14 different activity classes gathered by two different radars with respect to timestamps. More information regarding the different type of activities can be found in the paper X.

The organization of this report (document) is as follows. In section II, we discuss previous research in the literature and contrast it with our own results. The processing pipeline is displayed in Section III. Information about signals and features is provided in Section IV. The project's used learning framework is presented in Section V, and Section VI evaluates its effectiveness. The concluding remarks are found in Section VII.

II. Related Work

Human activity recognition consists of many different human behaviors, including walking, running, sitting, sleeping, standing and other anomalous behaviors. HAR has a variety of applications. It can be used to diagnose illnesses and keep track of old folks. It can also aid in keeping an eye on illicit activity [1].

HAR is applied using supervised and unsupervised models. A comprehensive deep learning algorithm was developed by Alazrai et al. to identify human-to-human interactions (HHI) from Wi-Fi signals. A well-known CSI dataset comprised of 13 human-to-human encounters with 40 different patient couples was used to assess this model. The planned model's average accuracy in all of this was 86.3 percent.Through the use of an amalgamation of many convolutional neural networks, Zehra et al. developed HAR by using trained and evaluated several ensembles and CNN models. Comparatively speaking, the proposed ensemble model is more accurate than the established models. The primary benefit of this approach is the ability to extract model-relevant features. Because this model does not require pre-processing, training and testing are faster using it [2].

Sikder et al. generated a dataset with labels that depicts human actions including standing up, standing down, and lying down using an attached smartphone that has accelerometer and gyroscope. In order to enable the computer to comprehend a variety of human actions, they additionally collected frequency and other information from human action signals using a two-channel convolutional neural network [3].

Similar to Alazrai et al., Moshiri meat. al, tried to find correlation between WiFi signals and human movement to classify human activities using synthetic data and GAN. However, they used synthetic data to reduce the cost of channel state information. They applied principal component analysis on amplitude of data and applied short time Fourier transform for feature extraction [4].

The methods listed above demonstrate many approaches to human activity recognition by using neural networks. In contrast to previous projects, ours uses classified data and can generate realistic samples that reduces the amount of real data to be collected. Comparing to smart phone, radars can also provide patients using aids. Finally, our project is less costly in terms of equipment comparing to CSI based human activity recognition projects [4].

III. Processing Pipeline

The Multiple Input Multiple Output (MIMO) mode of the Texas Instruments (TI) Millimeter (Mm) Wave FMCW radars is utilized. Specifically, the sensors xWR14xx (77 GHz) and xWR68xx (60 GHz), created originally for the automotive industry. These radars also have the benefit of being able to be built inexpensively and with good power efficiency. The high power efficiency comes at the expense of having a poor Signal to Noise Ratio (SNR), which frequently presents substantial difficulties in data analysis. The FMCW radar is an active sensor that continuously produces electromagnetic signals through transmitting antennas. The target then reflects these signals, which are then picked up by a collection of receiving antennas. Then, using the time delay or phase shift, significant information about the targets (such as range, angle, and speed) is retrieved from the reflections (i.e., the Doppler

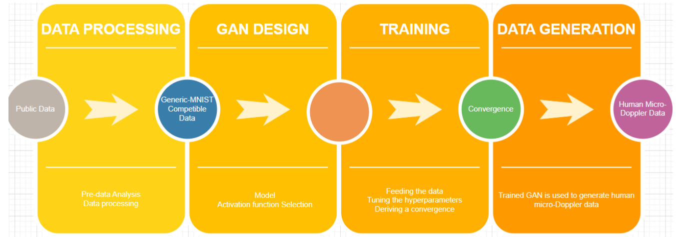

Fig. 1: Followed Processing Pipeline

effect [52]). The 2D Fourier transforms are applied to the reflected signals to produce the RD maps, which provide data on the target's range and velocity [5]. By summing over the range dimension and concatenating over the time dimension, RD maps can provide an MD signature. The patients in a hospital room carry out these activities, which the sensors used in this work record. Additionally, we just used the video sensor for annotation and validation. With the data gathered by mentioned sensors, the pipeline followed to achieve the contribution is as follows:

- Data processing : The data is worked on for interoperability purposes. Also on this part, we have arranged the structure of the data that we prepared according to MNIST dataset. The logic behind this is the MNIST data is one of the most common test data types used in neural networks. With this step, we had opportunity to use different GAN approaches on the internet.
- GAN design: In GAN, we have Generator and Discriminator networks. Our structure depends on up-sampling in the Discriminator network and down-sampling in the Generator network. We have given input of micro doppler maps that have been generated by the optimal frame rate (40 frames for each map) to the GAN.
- Training: With the dataset we created according to the optimal frame rate, we train our Discriminator and Generator networks simultaneously. We have used the optimal parameters to train Discriminator and Generator in balance. As result we have gathered the artificial images that out Generator network creates. In this process, according to results get, via trial and error, hyperparameters, such as dropout rate, epoch size, are tuned. Model is trained number of epochs. Softmax activation function is used.
- Data generation: Upon completion of training process, GAN is used to generate artificial data, generated data further visualized and compared with the real data. Also the generated micro doppler maps had noise in it, and we denoised them with using Scipy library.

IV. Signals and Features

Dataset

The two settings used for data collecting are a hospital and a home lab. Each space is roughly 30 m2 (Fig. 5). As was previously mentioned, they are keeping an eye on the patients in the hospital room by discreetly observing various

	Total samples	Avg. time (s)
In-room activities		
Walk to room	2053	4.0
Fall on the floor	1782	3.6
Stand up from the floor	1772	4.0
Walk to chair	2102	3.4
Sit down on chair	1854	2.5
Stand up from chair	1820	2.5
Walk to bed	2384	4.1
Sit down on bed	921	3.0
Stand up from bed	895	2.5
Bed activities		
Get in bed	1300	3.5
Lie in bed	1153	2.6
Roll in bed	1148	5.0
Sit in bed	1145	2.5
Get out bed	1240	3.6

Fig. 2: Activities Table.

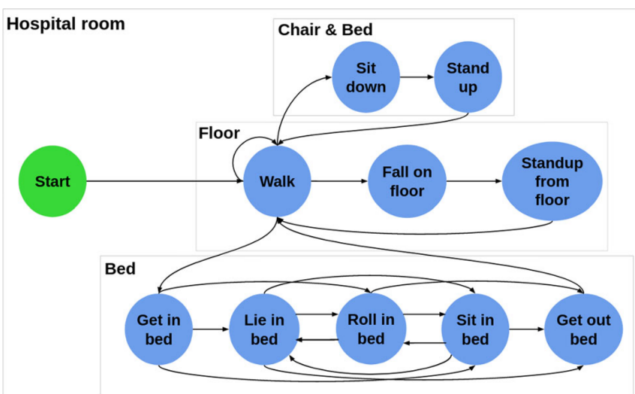

Fig. 3: ;Connection between activities.

actions and assuring their comfort. Due of this, they use two different FMCW radar sensors, a webcam, and two different surroundings to gather data for various patient activities.

They have given the subjects audio of the activity to be performed, along with a display mentioning the current activity (a timer), as they will as the next upcoming activity, in order to make it simple for them to record the activities. This way, the subjects always know exactly which activity to perform at any given time. 14 distinct activities that they tracked are listed in Table. They categorize related activity as follows:

- walk to bed, walk to room, walk to chair: [walk]
- sit down on bed, sit down on chair: [sit down]
- stand up from bed, stand up from chair: [stand up]

As a result, the 14 original activities are divided into 10 different activity groups. Additionally, as seen in Figure, the 10 groups of activities are sequential. For example, the phrase ”walk: walk, fall on the floor, sit down, get in bed” refers to

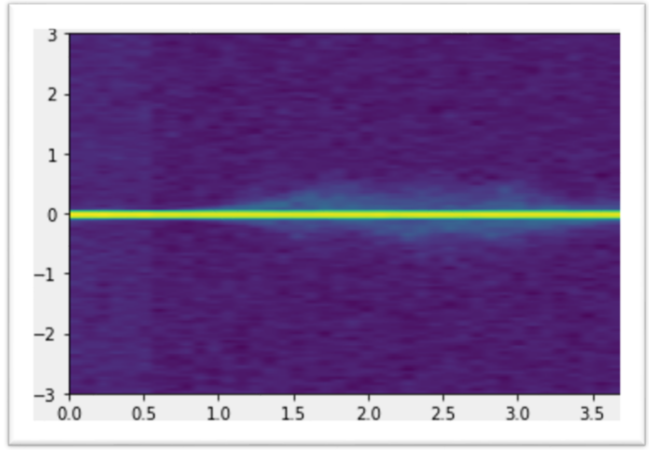

Fig. 4: Example MicroDoppler map.

the following sequential actions: if a person is now walking, their next activity might be either walking, falling on the floor, sitting down, or getting into bed. The recordings was done in four sessions, each lasting ten minutes, and the activities they are selected at random for both environments.

Pre-procecssing

A sample is a group of related frames that includes all of the target’s activity. By segmenting the radar frames based on the activity audio and the radar timestamps, the recordings are given labels. Subject .h5 files contains three type of data. ”cube”, ”RD map” and ”microDoppler map”. In this project we used only ”microDoppler” data in the subject .h5 files. The main radar data is ”cube” data. They processed this data and created ”RandeDopller map” and they processed ”microDoppler map” out of ”RangeDoppler” map. The microDoppler map is summation of the RD maps over the range dimension. MicroDoppler maps are two dimensional. Each row in the microDoppler map corresponds to 90 milliseconds. We get the timerange of each activity from ”Timestamp.csv” datas for each subject. Based on study of various sample lengths, 40 frames, or 3.7 seconds, is the recommended sample length. For visualized example, the figure above visualized version of 40 frames of micro doppler data between random range. Additionally, we take into account a zero-padding method where, if the sample length is less than a predetermined k seconds, we fill the remaining frames with zeros, also we take into account overlapping if the sample length is more than predetermined k seconds, we pad with the last frame to fill the rest of the frames.

Post-processing

After gathering information about the dataset we download the data of a subject and it’s timestamps. For first step we gathered timestamp data and turn these timestamp data to milliseconds to find out how much seconds past between each

$$ELU : \mathrm{f}(x;\alpha) = \begin{cases} x, & \text{if} \quad x > 0 \\ \alpha * (e^x - 1), & \text{if} \quad x \leq 0 \end{cases}$$

Fig. 5: ELU Activation Function.

activity. We gathered these activity time ranges in milliseconds. With respect to these time ranges in milliseconds we created our microDoppler maps. Because as we mentioned before, each row in microDoppler data, it is equals to 90 miliseconds. Each row's length in microDoppler dataset is 128. Our recommended sample length is 40 frames. As result of these two conditions, our visualized microDopller data is 40x128. Especially for the GAN we will use, we tried to turn our dataset's structure to MNIST dataset structure. With this implementation we were able to use different approaches on the internet that used MNIST data with GAN.

V. Learning Framework

We create GAN neural network architectures using the PyTorch machine learning framework. In our GAN there are two neural network architectures. One which accepts the 2D-MD signatures and decides if the image is fake or not (Discriminator), and one which accepts seed data and creates fake data(Generator). The discriminator is made up of strides, convolution layers, batch norm layers, and ELU activations. The generator is comprised of convolutional-transpose layers, batch norm layers, and ELU activations. The Discriminator network extracts features from Doppler and the time dimensions by using two-dimensional convolutional layers with 64,128 and 128 convolutional filters, respectively. Further, to reduce the input data dimensions, each convolutional layer is a down-sampling of size 5x5 kernel with ELU activation function. To prevent the domination of the Discriminator network over the Generator network, dropout was performed to all three layers. ELU is used in every step at Generator, and again ELU is used in every step in Discriminator except the last step. Elu is defined as follows: with x 2 R represents the input and a predefined parameter greater than zero. We also used softmax activation with same architecture. Parameters We used a different range of parameters. We used Adam Optimizer in our project. With Adam Optimizer, to find the global minimum and to make the loss function not noisy, different values between a range of 0.00001 to 0.001 have been tried. Between 0.00001 and 0.000099 our loss neural network was not learning properly. But between 0.001 to 0.00011, our loss function was getting noisier. The best result was making the learning rate 0.0001. With this learning rate, our loss function was more stable, and also our neural network can learn at the proper speed. After this step, we did several attempts with the dropout function. Because our Discriminator network was starting to dominate the Generator network as time passed. Domination means that the Discriminator learns faster than Generator and when Discriminator learns more than Generator, the Discriminator starts to tell every image generated by Generator is fake, as result, the loss function of the Generator gets higher and Discriminator's loss function gets lower with time passes. So, we had to make the dropout higher. It was 0.2 before, and it was dominating. We did several tests of dropout values between 0.1 to 0.6. If we set the dropout value lower than 0.4, than the Discriminator generally dominates the Generator as time passes, if we set it to more than 0.4 (to 0.6) the Discriminator does not learn properly, therefore Generator does not generate proper data. The best result we get is with a dropout value of 0.4. We also did several attempts with batch size. We used the Batch Size value between 4 to 32. When it is higher than 8 (values 16 and 32) our network's loss function is becoming so noisy therefore our results are not improving after some point. If we set it to 4 or 8, we got the best results, but if we have to compare the loss function results of 4 and 8 batch size values, we can see from the loss function graph, setting the Batch Size value to 4 is more optimal. For the loss function, we used a loss function that combines a Sigmoid layer and the BCELoss in one single class. It is called BCEWithLogitsLoss() function. And lastly, we had to arrange the Epoch value. We saw that with the information we got and with the neural network we work with, it can be more accurate to set the Epoch value between 200 to 400. We can set the Epoch value maximum of 400 because our GPU RAM gets full and stops the learning process. To prevent this we generally worked with a 350 Epoch value. We run our GAN network with GTX 1650(mobile) with 350 Epochs (generally).

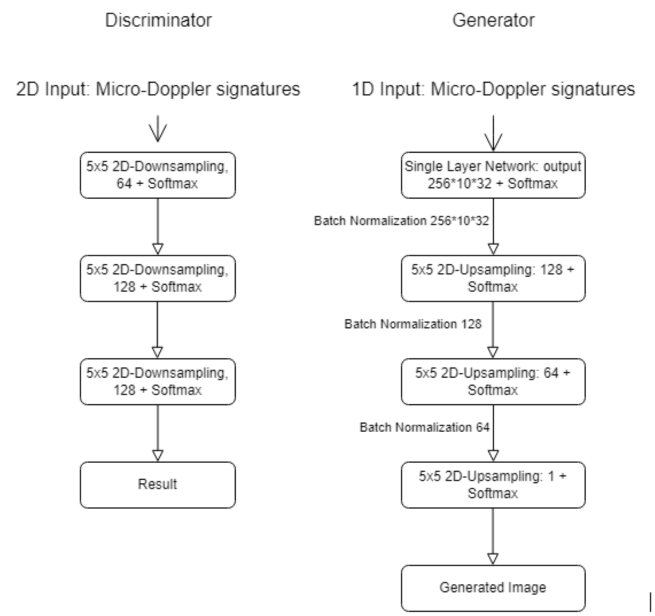

Fig. 6: Discriminator and Generator Diagrams.

VI. RESULTS

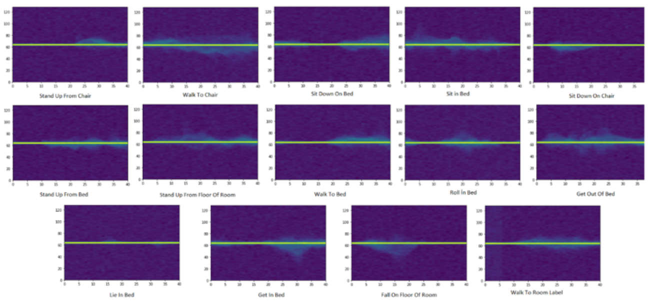

Fig. 7: Micro Doppler maps of 14 activities.

You can see every activity's micro doppler map sample figure above. With our GAN network, we tried to create similar maps with the data above. With the optimal parameters we found, started to teach both our Discriminator and Generator neural network. With Epoch count 360, you can see the 32 grayscale predictions that our GAN generated below.

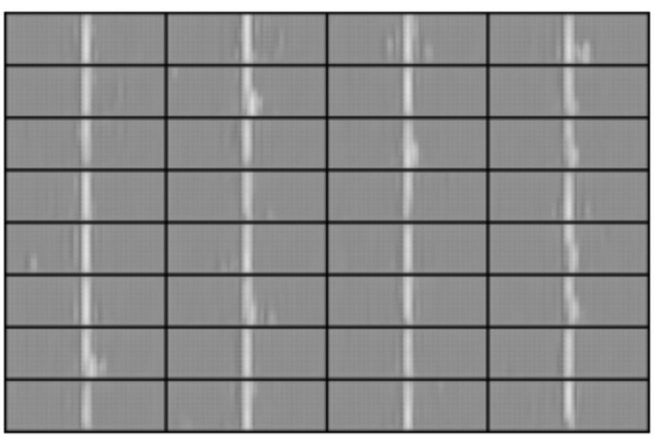

Fig. 8: 32 Generated Predictions.

To make it understandable and comparable, we have separated and transposed the 32 predictions you see above. In the picture below you can see some of these images that we have created with our GAN network. With our Discriminator's prediction, we have labeled the generated images.

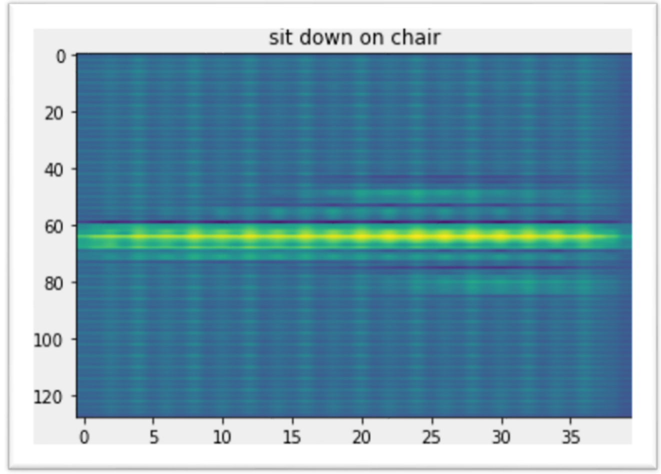

Fig. 9: Sit down on chair Micro Doppler Map.

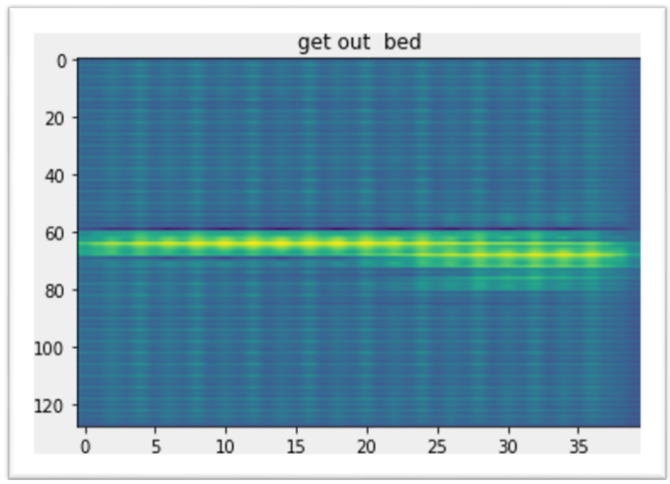

Fig. 10: Get out bed Micro Doppler map.

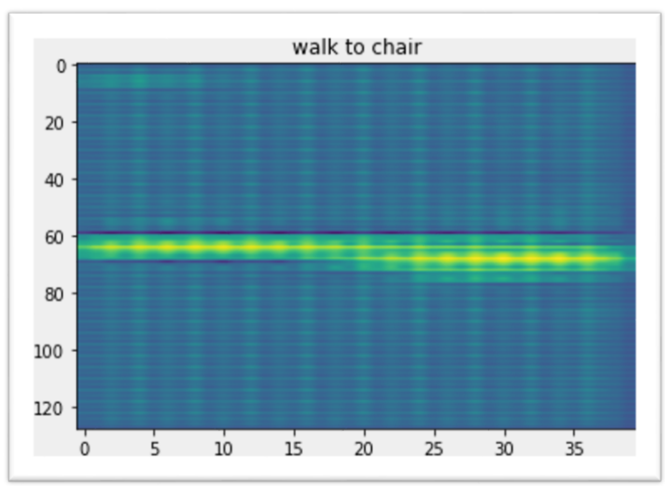

Fig. 11: Walk to chair Micro Doppler map.

You can see the GAN network generated micro doppler maps above. With using our Discriminator network, we have labeled the GAN network generated images at the end of training.

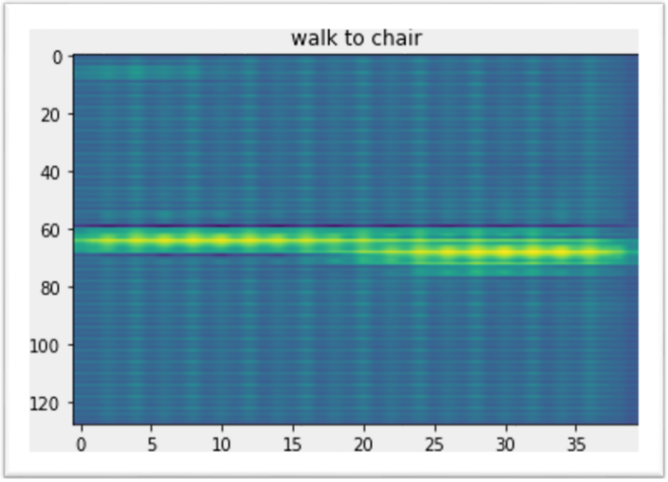

Fig. 12: Walk to chair Micro Doppler map 1.

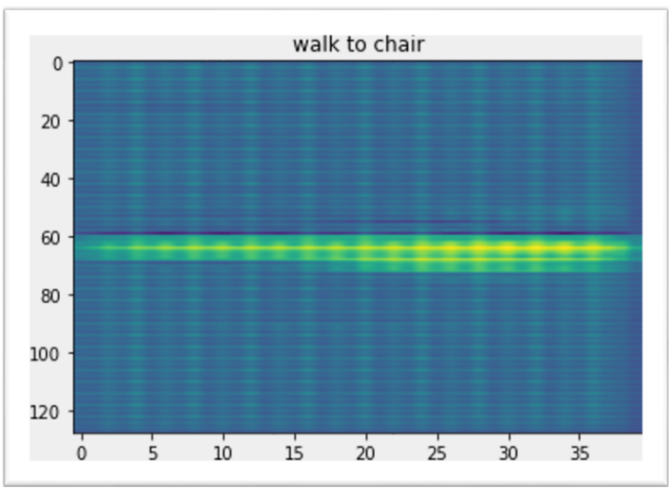

Fig. 13: Walk to chair Micro Doppler map 2.

You can see the "Walk To Chair" labeled two generated images above. We can see these two generated images that labeled "Walk To Chair" is pretty similar to each other. With human eyes, we can see the similarity between the two, but if we look closely we can see that these images are not same. From this information, we can understand that our Discriminator is trained to tell the difference between labels, and not labeling randomly.

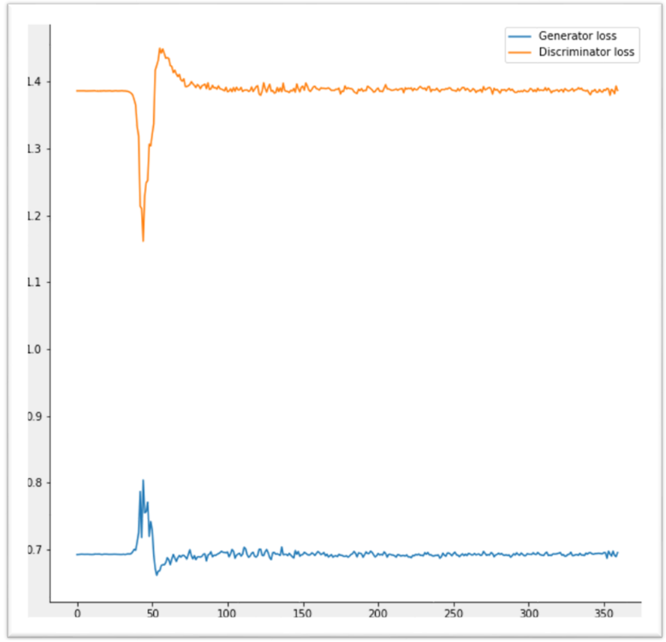

Fig. 14: Loss function Graph.

From the Loss Function graph, you can see above, our network starts to converge around 50 to 70 Epochs. According to our loss function graph and the similarity of the images produced by the GAN network with the original micro-Doppler maps, we can see that our Discriminator network is well trained, and thus the Discriminator network can train without dominating the Generator network. Also, from the generated images, you can see that there is still some noise remains. To remove this noise (denoising) and create more clear micro doppler maps, we have done Image denoising by FFT using the Scipy library. We have denoised the generated images and reconstructed images.

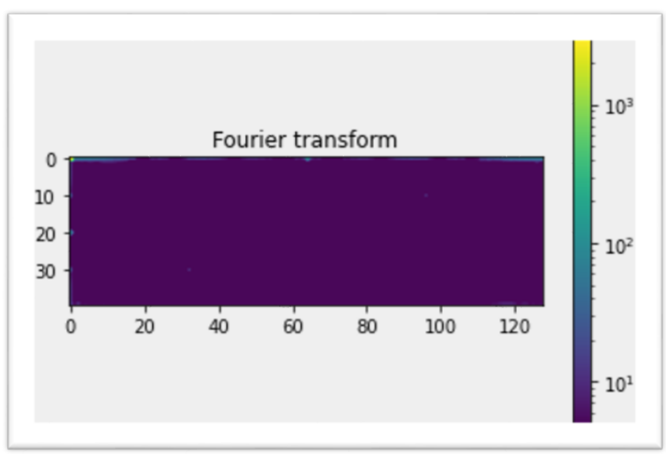

Fig. 15: Fourier Transform.

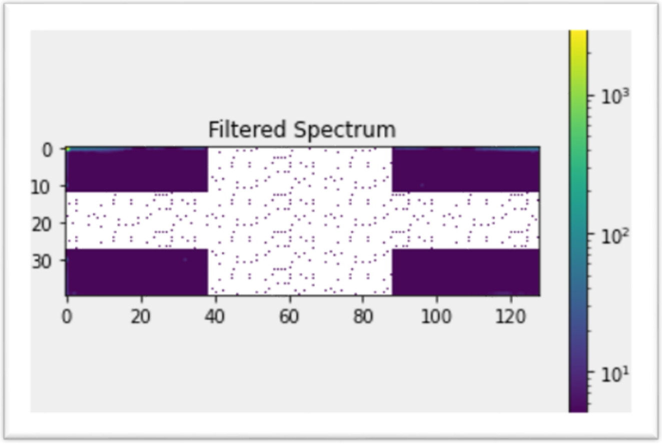

Fig. 16: Filtered Spectrum.

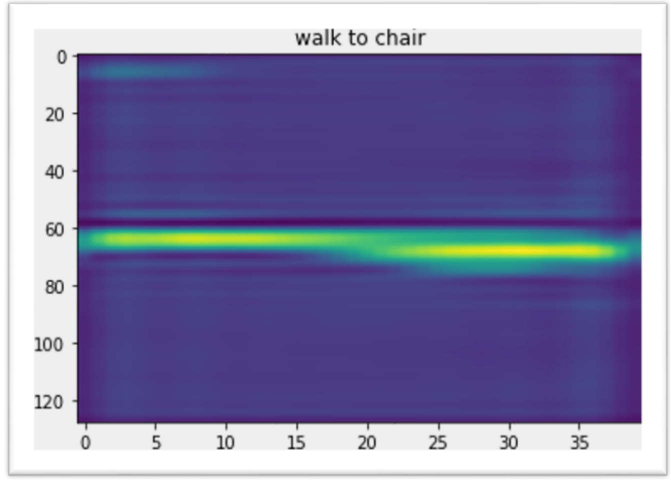

Fig. 19: Denoised Walk to chair Micro Doppler map 1.

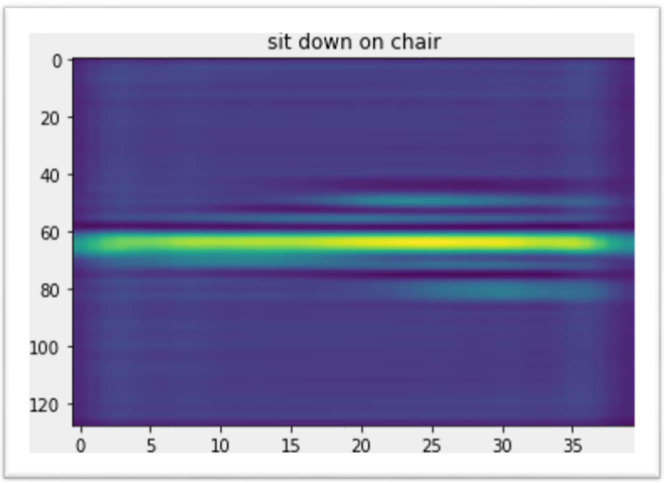

Fig. 17: Denoised Sit down on chair Micro Doppler map.

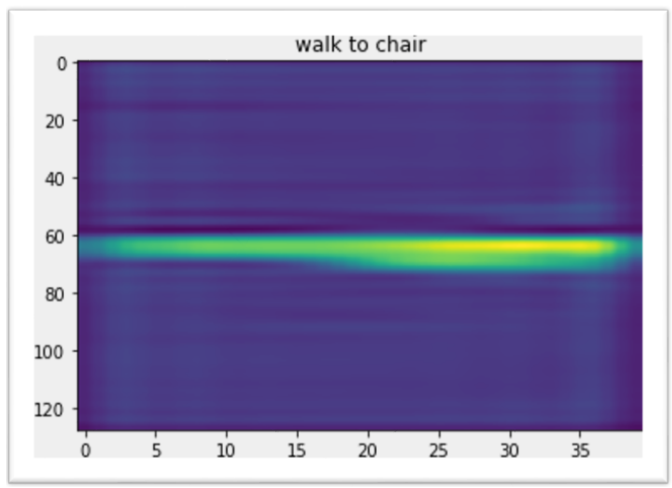

Fig. 20: Denoised Walk to chair Micro Doppler map 2.

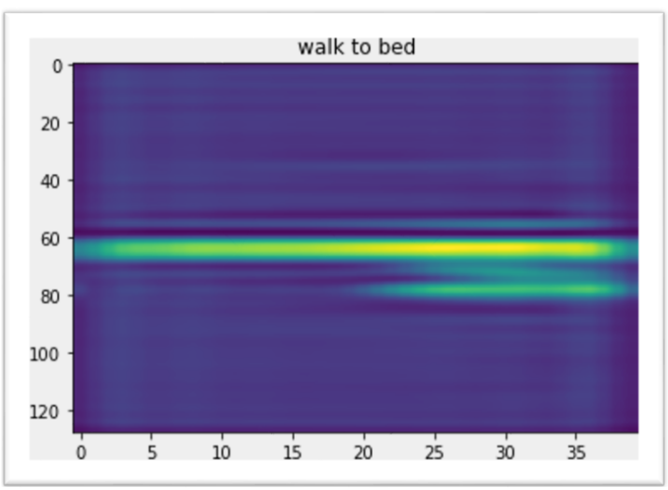

Fig. 18: Denoised Walk to bed Micro Doppler map.

As a result of denoising with FFT, you can see that the images look more clearly, and human eyes can see that most of the noise has been cleared from the images. We also used another activation function with our GAN network. We used ELU, but with the system we have we were not able to train it completely. The generated images were not good and the loss function graph was too noisy. You can see one generated image and a filtered version of it below.

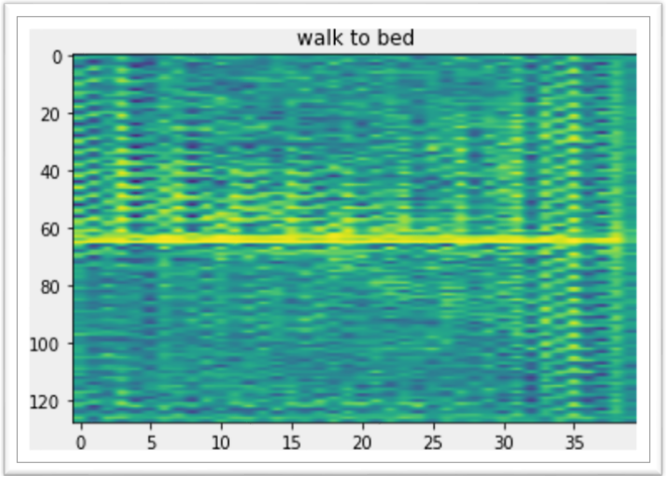

Fig. 21: Walk to bed Micro Doppler Map Generated with ELU.

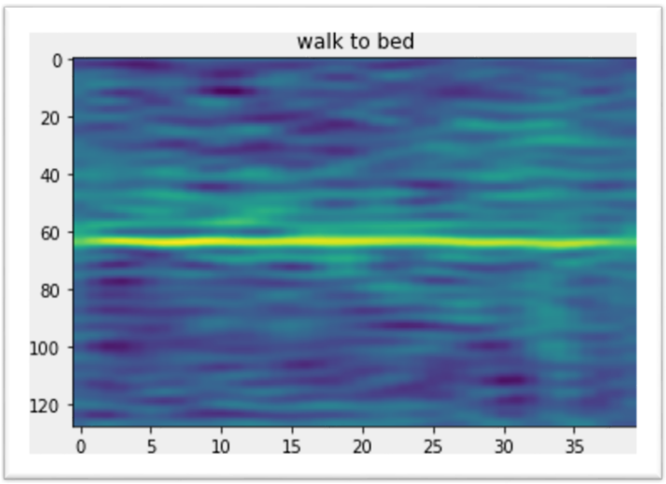

Fig. 22: Denoised Walk to bed Micro Doppler Map Generated with ELU.

As you can see, it is too noisy and not understandable. And from the Loss function graph, we can say that it is not stable. We tried to decrease the learning rate, but with decreased learning rate our GPU's memory couldn't enough the epoch count that we needed. With the maximum Epoch count, we were getting bad results.

VII. Concluding Remarks

With this project we have created a GAN network that gets micro doppler maps of 14 different activities that have been collected from home lab and hospital environments to generate similar data with these activities. If we look at the results we got, we can see that our network can generate similar data with the data that have been generated by a radar. By generating this data we can prevent the expense of collecting real micro-Doppler data, access to restricted environments like hospitals or patients in the houses. At the Results part you

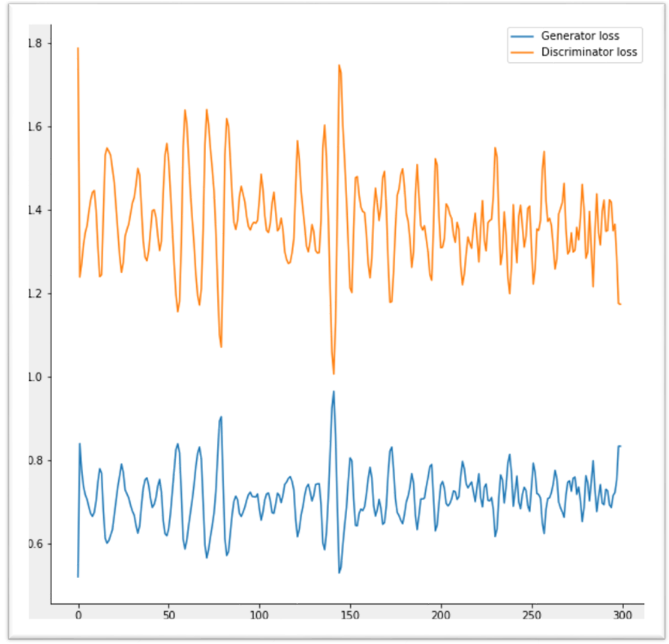

Fig. 23: ELU Loss Function.

can see that our results are similar to the data that used as an input with our GAN network, but not perfect. To solve this problem we can use better systems to run our GAN. For example we can run our GAN with some computer that has more GPU RAM. With better system we can increase the number of Epochs and reduce the learning rate. With this step we can reach the global minimum with learning rate (Adam optimizer) slowly but more accurately. With better system we can also test our discriminators accuracy with the real data that we didn't use. If we test simultaneously with the other real data in our systems while our GAN network learning, we have to reduce the number of Epoch between 180 to 200. With these value ranges we cannot get proper created micro doppler maps. Also with this infrastructure and little bit of manipulation with our data, we could turn this GAN to cGAN to create specific activities result. For example, data that we prepared, we could divide it by the activities and before GAN starts we could choose the activity set that we want to create and generate it.

We have learned what is GAN, how to use it. We also learned how to set the parameters according to the situation. For example, because the loss function was too noisy, we decreased the value of learning rate. Another example is because the Discriminator starts to dominate Generator, we increased the value of dropout in the Discriminator. We learned what is system dependency. For example, we wanted to test the discriminator with the other datasets but if we do that we saw that we can't get proper results because the system we use to run our GAN. If we do that we have to decrease the Epoch value almost half of it size, and that could impact to our final generated data.

References

[1] C. Jobanputra, J. Bavishib, and N. Doshic, "Human Activity Recognition: A Survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019.

[2] V. Sarveshwarana, T. I. Josephb, M. M, and K. P, "Investigation on Human Activity Recognition using Deep Learning," *In Procedia Computer Science* , vol. 204, pp. 73–80, 2022.

[3] N. Sikder, M. S. Chowdhury, A. S. M. Arif, and A.-A. Nahid, "Human Activity Recognition Using Multichannel Convolutional Neural Network," in *5th International Conference on Advances in Electrical Engineering*, 2019.

[4] P. F. Moshiri1, H. Navidan, R. Shahbazian, S. A. Ghorashil, and D. Windridge, "Using GAN to Enhance the Accuracy of Indoor Human Activity Recognition," 2020.