

Department of Electrical, Computer, and Software Engineering

Part IV Research Project

Final Report

Project Number: 77

Millimeter Wave Radar based Human Activity Recognition

Beck Busch

Samuel Mason

Kevin I-Kai Wang

Akshat Bisht

13/10/2023

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

A handwritten signature in black ink, appearing to read "B Beck Busch". The "B" is a large, stylized initial, followed by "Beck" and "Busch" in a cursive script.

Name: Beck Busch

ABSTRACT: Human activity recognition is an essential area of study, as it allows for many advances in how technology can interact with the people who use it. Millimeter wave radar (mmWave) is an up-and-coming solution to this problem as it has several advantages over other technologies. Unlike optical sensors, mmWave is not reliant on external lighting conditions and has vastly reduced privacy concerns. To expand the usage of this technology, we aim to develop methods for detecting multiple activity sequences. Current mmWave solutions struggle to identify activity patterns of multiple actions in sequence. We aim to remedy this with a novel frame selection and sliding window approach.

Acknowledgements: I would like to thank my supervisors, Kevin I-Kai Wang and Akshat Bisht, for their invaluable support and guidance during this project, and my project partner, Sam Mason, for their continued commitment to the project and high standard of teamwork. I would also like to thank my family for their overwhelming support during not only my honours study but my entire degree.

Table of Contents

| | |
|--|-----------|
| 1. Introduction..... | 1 |
| 2. Literature Review..... | 2 |
| 2.1. HAR Technologies..... | 2 |
| 2.1.1. Analysis of physical sensors and CSI..... | 2 |
| 2.1.2. Advantages of mmWave FMCW radar..... | 3 |
| 2.2. Data recording..... | 4 |
| 2.2.1. Alternative pre-processing..... | 4 |
| 2.3. Data segmentation..... | 4 |
| 2.4. Classification algorithms..... | 5 |
| 2.5. Real-time applicability..... | 6 |
| 3. Methodology..... | 7 |
| 3.1. Radar technology..... | 7 |
| 3.1.1. FMCW Radar Chirp..... | 8 |
| 3.2. Data extraction..... | 8 |
| 3.2.1. Radar Data Cube..... | 9 |
| 3.3. Sliding window..... | 10 |
| 3.4. Frame selection..... | 11 |
| 4. Experimental Process..... | 12 |
| 4.1. Data extraction..... | 13 |
| 4.2. Data collection..... | 13 |
| 4.3. Processing script development..... | 14 |
| 5. Discussion..... | 15 |
| 6. Conclusion..... | 16 |
| 6.1. Future works..... | 16 |
| 7. References..... | 1 |
| 8. Appendix..... | 2 |
| 8.1. Literature review sources table..... | 2 |

Tables and Figures

| | |
|--|----|
| Figure 1: Full System Pipeline..... | 7 |
| Figure 2: FMCW Chirp Diagram, Texas Instruments..... | 8 |
| Figure 3: Radar Data Cube, Kang, Sung-wook..... | 9 |
| Figure 4: Sliding window diagram..... | 10 |
| Figure 5: Frame selection Diagram..... | 11 |
| Figure 6: Radar Setup..... | 12 |
| Table 1: Radar Configuration..... | 12 |

1. Introduction

Research into human activity recognition is extremely valuable for the current technology industry. There are a vast number of applications where identifying and responding to human behaviour is essential, and the more HAR technology improves, the better these applications can excel. Some of the key areas of application focus on protecting and monitoring humans, such as detecting incidents in aged care facilities or anticipating accidents in the workplace. Other areas focus on the analysis of actions for other means. Surveillance technology would be able to alert authorities of violent actions or trespassing. At the same time, smart home control could fully automate the functions of IOT devices while anticipating the needs and behaviour of the user. All of these applications have been, to some extent, approached by existing HAR techniques such as visual cameras. We already have computer vision algorithms running in areas such as banks or stores that detect when someone holds a weapon. The main problem facing HAR today is surrounding privacy. Customers and consumers are already weary and resistant to camera surveillance, and people resent the idea of a smart speaker listening to their conversations, let alone putting a “Google Camera” in their homes and living spaces. This is where mmWave radar can be most effective. Millimetre wave radar works by reflecting radio waves off objects in the environment and then analysing the returning waves' time delay and Doppler pitch shift. This approach prioritises user privacy since the data being recorded is far less invasive than any visual medium.

The main focus of our research is to accurately identify actions of varying speeds and lengths, essentially removing or abstracting the time dimension from the detection process. Most machine learning algorithms that identify video recordings or event sequences process them like humans do, as a chain of events that follow each other over a certain period. This can cause problems when the events being analysed take place at a slower speed or are spread out over a long period of time. Our approach will ideally be able to reduce a sequence of events in the time domain to a unique and identifiable pattern that can be processed with more accuracy and speed.

2. Literature Review

Analysis of prior works is vital for undertaking a project of this scope. With this literature review, we aimed to expand our knowledge of the field while also identifying an area of research that would be beneficial. A decent amount of research exists in this field, with various papers focusing on different aspects of the solution, such as recognition technology, recording and segmentation methods, and especially the algorithms that analyse and classify the data. We reviewed a selection of these research outputs focusing on real-time classification and sequence detection. Appendix 9.1 contains an index of the papers I will reference, with relevant information on how they conducted their research.

2.1. HAR Technologies

There are several different base technologies at the forefront of HAR. Optical methods like visual and infrared cameras excel at pattern recognition and motion tracking, but this approach has several concerns. The most apparent concern is that placing a constantly recording camera in an area with frequent human activity can be an invasion of privacy. Consumers will never be comfortable with third-party camera systems installed in private places like hospital rooms or houses. Hence, an effective HAR system needs to operate without the ability to record such detailed information. Another problem with optical sensors is environmental lighting. While infrared cameras can circumvent this with infrared LEDs, regular cameras rely on the lighting conditions of the environment they are monitoring to record quality data. This leads us to non-optical measurements such as accelerometers, channel state information, and mmWave radar.

2.1.1. Analysis of physical sensors and CSI

Physical sensors like accelerometers and gyroscopes offer several advantages over vision-based sensors, but they come with their own limitations. While physical sensors are often able to capture cleaner data with much less noise, [6] they require that the sensor itself be attached to the subject. This works well for situations where the activity monitoring is temporary and involved, such as

physical exercise. However, in more passive monitoring environments, such as a bedroom or office, wearable devices are an unwanted burden. Channel State Information monitoring refers to the process of observing the changes in wireless signals created by objects moving in an environment. As human bodies move between a signal source and receiver, they create variations in how the signal propagates through the environment. Analysis of this change can be accurate enough to identify the posture and motion of a human form, allowing for a non-invasive form of HAR. One of the most significant advantages of CSI detection is that wifi signals can be used as the signal source. Since wifi signals are already highly prevalent in our everyday lives, signal sources are already readily available. The downsides to wifi CSI are related to the quality and accuracy of the received information. Due to the wavelength of the wifi signal, the data lacks granularity, making small-scale motion challenging to detect. On top of this, CSI experiments require precisely placed signal sources and receivers that cannot be used for regular data transmission to achieve the best result [5]. This contradicts the notion that CSI is advantageous since wifi signals are already available in the environment.

2.1.2. Advantages of mmWave FMCW radar

FMCW radar is a technique that can identify moving objects in the environment by analysing their effect on an emitted signal. Signal bursts with a constant linear change in frequency, known as chirps, are emitted by the radar, then the reflection of this signal is received. When the signal reflects off an object, the object's velocity creates variations in the signal that can be analysed through comparison to the initially emitted chirp. This technique allows us to gather more information points than CSI without optical sensors. The larger wavelengths of FMCW radar also provide more accurate information due to the granularity of the received data, [8] making it our preferred technology for HAR.

2.2. Data recording

There are several different methods for recording and interpreting the FMCW chirp, each with its own advantages and disadvantages. The two most commonly used methods are range-doppler image and range-angle image. Range-doppler imaging works by interpreting the variations in frequency created by the signal reflecting off a moving object. The Fourier transform of this received signal can be used to estimate the velocity of a moving object, providing data for activity recognition. Range-angle imaging compares the reflected signal received by two antennas. It uses the phase difference along with the distance between antennas to calculate the range and angle of objects in the environment. While range-doppler imaging is often preferred for its fast generation time and depth of information [8], it can often fall short when movement occurs parallel to the radar transmitter [9]. Horizontal movements are still detectable by range-doppler image due to the change of velocity of the object, but since the distance from the object to the radar (range) stays relatively constant, RDI can struggle to define the gesture. Range-angle imaging does not suffer from these problems and generally performs better at detecting motion than RDI, especially horizontal motion. For these reasons, most approaches utilise a fusion of RDI and RAI as the inputs to their systems, which achieve excellent experimental results compared to only one data system [8].

2.2.1. Alternative pre-processing

In [7], Peijun Zhao et al. investigate a new preprocessing technique they call CubeLearn. This novel approach to FMCW radar interpretation aims to train a machine learning model based on conventional Fourier transform techniques to achieve a similar processed signal without the costly Fourier transform operations. This results in accuracies of up to 95% with a processing time of only 25ms.

2.3. Data segmentation

When analysing radar data for classification of events, data segmentation is necessary to reveal the relevant parts of the data that contain movements or behaviour that we aim to detect. This can be

performed in several ways, with varying results from each method. Several studies manually segment the data into the separate activities and transitions before beginning analysis. This can be fine when researching a smaller part of the HAR pipeline, but for our purposes, it is unsatisfactory. Another common approach is the usage of sliding windows. ‘Windows’ of a fixed length are slid over the data recording, and the data within the window is analysed. Based on this analysis, the section of the recording containing the most useful information can be identified. This approach can cause issues when the fixed window size is either too small or too large for the relevant action, so the sliding window process can be performed multiple times with windows of varying lengths [2]. Alexandros Ninos et al. [9] proposed a novel solution to data segmentation that uses an event detection algorithm to identify events taking place in real-time. By analysing the radar image for a velocity change above a certain threshold, they are able to identify the start of an action. Although this approach still relies on analysing a data segment of a fixed length, the event detection allows it to start the analysis in real-time instead of investigating pre-recorded data.

When identifying an activity is not dependent on a period of time, individual frame analysis can be used to classify data. Arindam Sengupta et al. [11] created a CNN model to estimate the posture of the human frame. Since their desired outcome does not involve multiple data frames at once, they can analyse each frame to identify human posture. In this way, they do not need to utilise data segmentation.

2.4. Classification algorithms

The core functionality of a HAR system is the algorithm that can receive the radar input and determine which action is being performed. Best results are often achieved through the use of a machine learning model such as convolutional neural networks, sometimes in tandem with long short-term memory networks. In [3], Geethika Bhavanasi et al. perform a comparison of various ML models for mmWave FMCW HAR. They compared a CNN model to LSTM, CNN&LSTM, and other models. They found that the CNN model performed the best at 97% accuracy in a clinical

environment and still performed the best in other environments. In [1], Chengxi Yu et al. propose a novel HAR pipeline involving a dual-view convolutional neural network. They found that this approach greatly outperforms other similar models, with an accuracy of 97%, compared to the similar models with a max accuracy of 70%.

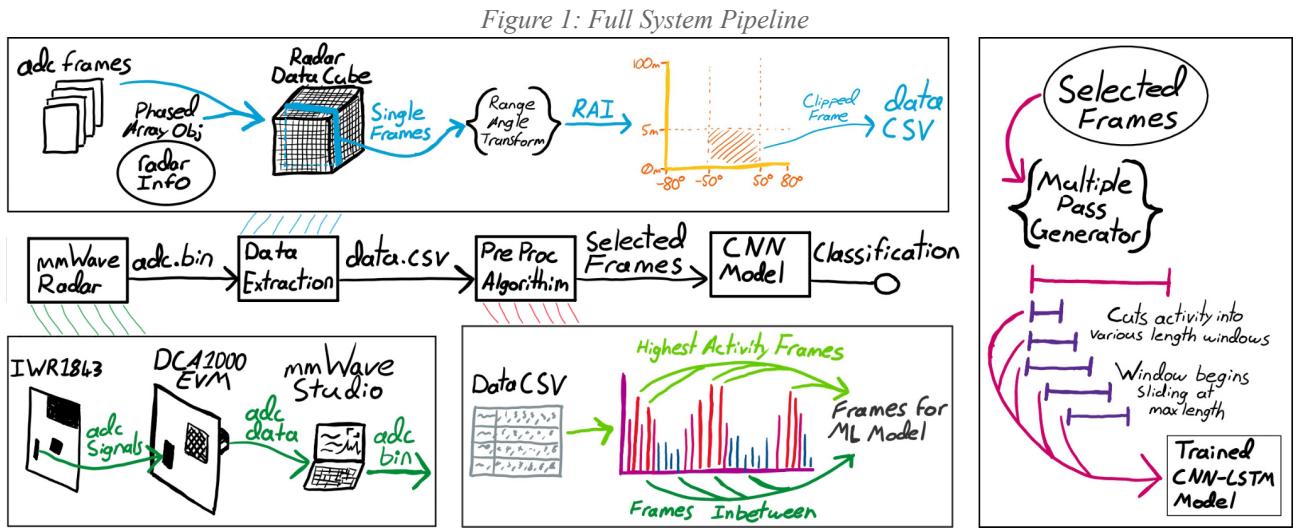
2.5. Real-time applicability

Our research has been focused on an HAR system that can function in real-time. To this end, we have investigated the various experimental attempts at behaviour classification in real-time. In [10], Haihua Xie et al. devised a lightweight classification algorithm to maximise the running speed of the system. They did this through the use of a PointNet algorithm to efficiently identify spatial features in the data. PointNet is a deep net architecture that allows for the processing of point clouds, a format of radar data that is less refined than micro-doppler formats like range-doppler images. By working directly with point clouds, the system can avoid the costly data calculations that are required to generate range-doppler or range-angle images. This is a great advantage for real-time systems, as the PointNet classifier was able to achieve higher accuracy than conventional algorithms such as LSTM while only taking 81 milliseconds to complete.

In [4], Renyuan Zhang and Siyang Cao design an HAR system that uses a constant false alarm rate (CFAR) algorithm running on the radar board to retrieve the relevant data points from the radar, and construct a point cloud. This removes unnecessary information, allowing for faster data transmission. The point cloud data is then processed by the host computer into a micro-doppler image for the classification algorithm. The faster transmission rates allow for data to be processed in real-time. To allow the algorithm to detect human behaviour in real-time, they create a constantly updated buffer of micro-doppler frames to be fed into the algorithm. This allows the algorithm to detect sequential changes in motion and identify behaviour.

3. Methodology

Our proposed solution involves a novel combination of a frame selection algorithm and a multiple pass generator that, together, are able to identify complex activity sequences of any length and number without compromising on efficiency. The multiple pass generator acts as a novel sliding window implementation that rapidly sections the radar data while continually responding to the CNN-LSTM model's classification. This continual usage of the model is facilitated by a frame selection algorithm that dramatically reduces the size of the input data without disfiguring the contents. These two components, along with several other stages of data manipulation, form our approach, which saw an accuracy of above 90% in preliminary testing.



3.1. Radar technology

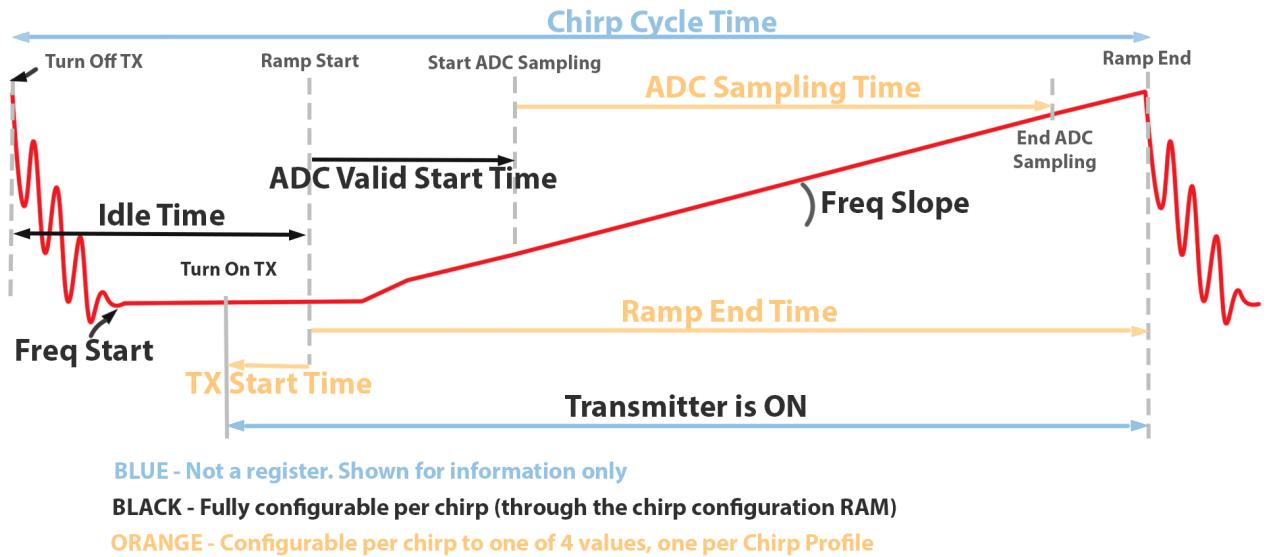
The entire system must start with the recording of radar data to be used in training and evaluation. Our radar technology is a combination of two boards, a radar module and a data streaming board. The IWR1843 emits radio waves from a transmission antenna, which are picked back up by four receiver antennas. The data from these antennas is retrieved by several ADC modules before being made available to the data streaming board via a sixty-pin ribbon cable. This board, the DCA1000EVM, exists to communicate these raw ADC readings directly to the host device. The IWR1843 is able to transmit data directly, but the limits on data processing and streaming mean that

any data output must first be compressed and transformed. The DCA1000 is able to provide our system with the uncompressed data, giving us a large amount of freedom with how we choose to process and filter it. The setup and control of both radar boards is facilitated by the mmWave Studio software provided by Texas Instruments. This software communicates via SPI to configure the parameters of data collection and manages the recording of data.

3.1.1. FMCW Radar Chirp

The radar format we are working with is known as frequency modulated continuous wave, or FMCW. This format emits the radar waves in “chirps”, where the frequency of the waves rapidly increases over a short period of time. This provides a wide range of information as we are able to analyse how the objects in the environment affect different frequencies.

Figure 2: FMCW Chirp Diagram, Texas Instruments



3.2. Data extraction

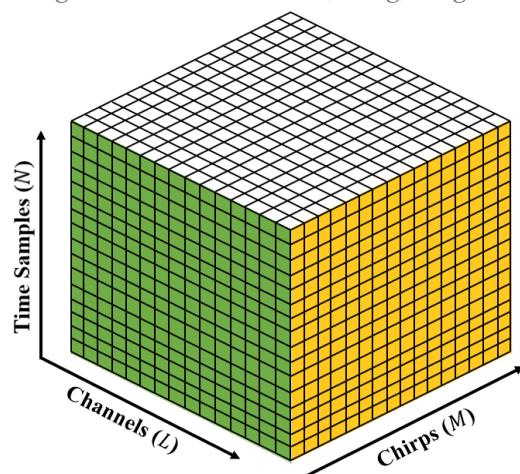
Once the radar data has been collected by mmWave Studio, we need to convert the ADC binaries into frames of Doppler values that we can use in our system. This is done by a Matlab script making use of the Phased Array System toolbox. The first step is processing the binary file with a script provided by Texas Instruments. The job of this script is to convert the raw input from the radar, which can sometimes contain missing or out-of-order data frames, into a clean and usable array.

The individual frames are then arranged into a radar data cube format so that they can be processed by the various functions provided by the Matlab libraries. Once the data is in the data cube format, we are able to transform the data using the range-angle response function. This function performs a fast Fourier transform on the data, and calculates the intensity of recorded motion along the axis of distance and angle from the radar. This provides a two-dimensional graph of complex values, which we then crop to a more usable range. For our purposes, this range is zero to five meters away from the radar, and one hundred degrees of visibility centred on the radar. Finally, these values are saved to a CSV file to be used by other parts of the pipeline. The data cube and FFT steps are repeated for each frame of the source recording, and the script is able to process several binary files in one run. This creates an output file with four dimensions; radar recordings, recording frames, range, and angle.

3.2.1. Radar Data Cube

The radar data is recorded over four dimensions that need to be arranged into a three-dimensional matrix of ADC values known as a radar cube. The largest dimension is the different frames in the recording. For our purposes, we were working with 150 frames, each of which creates its own unique data cube. The other three dimensions making up the data cube are the FMCW chirps, the receiving antennas, and the different adc samples taken within a single chirp. These make up the slow time, spatial, and fast time dimensions, respectively.

Figure 3: Radar Data Cube, Kang, Sung-wook

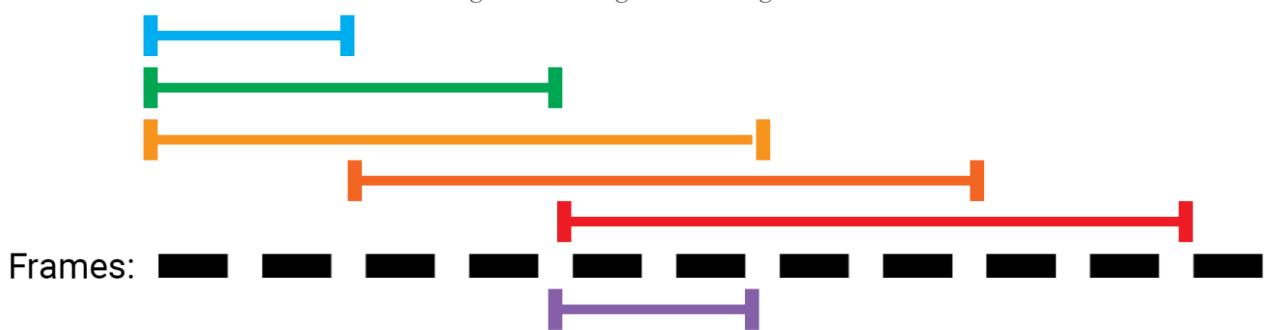


During a radar frame, several dozen chirps are emitted and recorded, creating the slow time dimension. During each chirp, the receive antennas rapidly record 256 ADC values to sample the effects of the different frequencies in the chirp, creating the fast time dimension. By performing computational algorithms along these different dimensions, we are able to accurately extract information about the motion of subjects. This process is called Beamforming. The data cube object also contains a Phased Array object, which is a Matlab structure that stores data about the physical arrangement of the radar, such as wave frequencies, antenna placement, and recording speed.

3.3. Sliding window

Before classification by the ML model, the data is clipped by a novel sliding window method that aims to facilitate activity sequence detection by analysing data samples of various lengths. The window begins at the start of the recording and gradually increases in size until it reaches a maximum length. The different windows are continually fed into the model for classification until the model fails to identify an action. When this happens, either immediately or part way through the recording, the sliding window anchors its starting point to the end of the current window and resets the window size to the minimum. If the model does identify an action, the window changes as normal until the action is no longer detected. At this point, we can identify the full length of the detected action and continue searching. This is how our system is able to identify activity sequences. By continually changing the sliding window, we are able to identify when one action begins immediately after another.

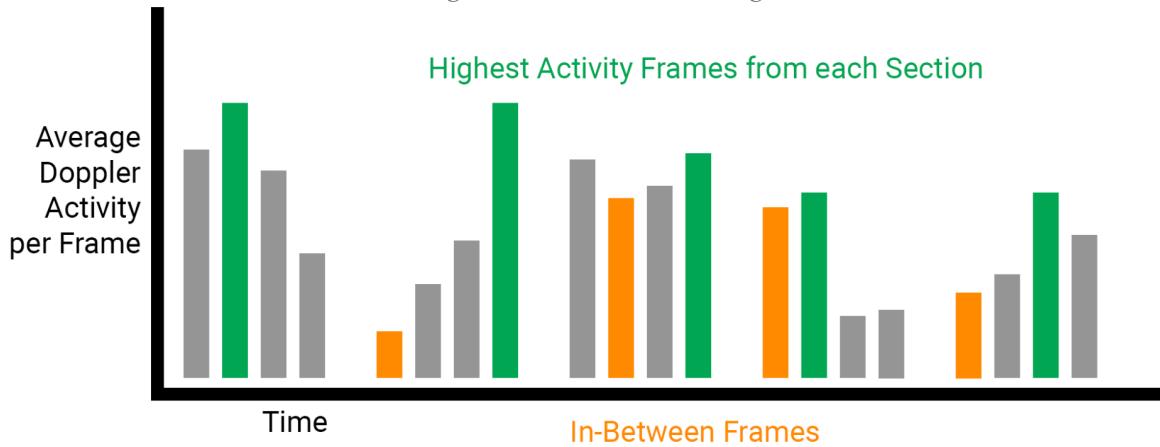
Figure 4: Sliding window diagram



3.4. Frame selection

The frame selection algorithm is the focus of our research contribution, as we believe that by isolating various frames from the radar recording, we will be able to increase the effectiveness of a machine learning approach. The algorithm first separates the radar data into several sections along the time dimension. This ensures that the selected frames will remain representative of the activity as a whole and prevents the selection from being biased towards areas of high movement. The algorithm then selects the frame with the highest Doppler values from each section. These frames represent the moment of highest motion within each section and will ideally be the most representative of the action being processed. Finally, the algorithm selects the frames directly between the already selected frames, which aims to reintroduce some detail by representing a wider variation of frames. This system is also in place to normalise the input size to the ML model.

Figure 5: Frame selection Diagram



4. Experimental Process

The majority of our practical work consisted of creating and refining the systems for data collection and extraction, and the systems for machine learning and classification. Before any technical research could begin, we identified the radar setup that would best suit our needs. Working with knowledge from the literature review, we compared the capabilities of various radar systems before deciding that we needed to purchase a data streaming board to avoid built-in compression. After procuring a radar setup that met our requirements, we created a setup for using the devices. This consisted of a tripod and a 3D printed bracket, as well as clearing an open space to record consistent data. For the data collection, we focused on researching and testing different radar configurations until we identified a setup that best suited our needs. We arrived at the values displayed in table 1.

Figure 6: Radar Setup

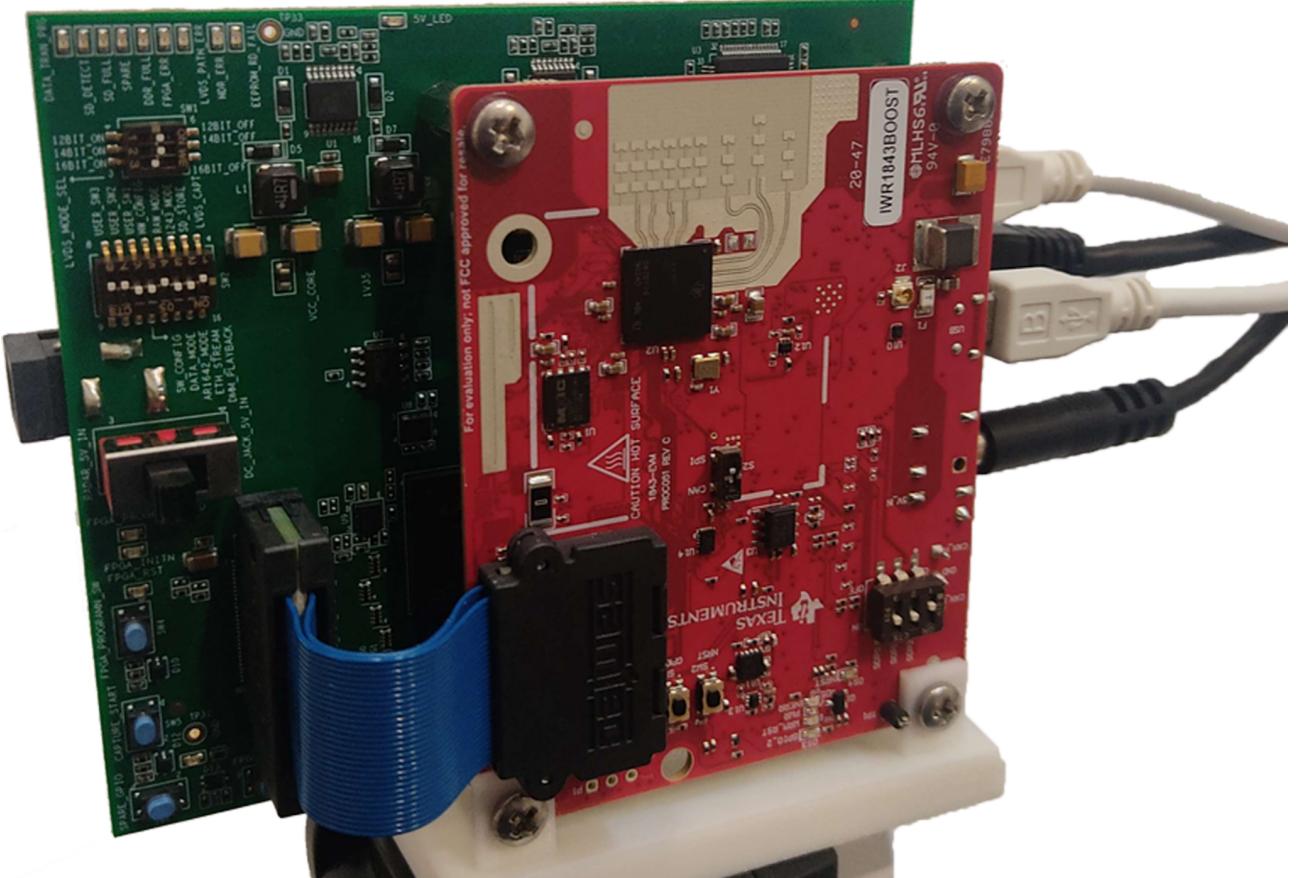


Table 1: Radar Configuration

| Setting | Value |
|-------------------|--------------------|
| ADC Sample Count | 256 |
| ADC Sample Rate | 10000 ksps |
| Frame Count | 150 |
| Chirp Count | 128 |
| Periodicity (FPS) | 40ms (25 FPS) |
| RX Antennas | 4 |
| ADC Format | 16 Bit, 1x Complex |
| Packet Delay | 50us |

4.1. Data extraction

With an adequately configured radar setup, we began developing the data extraction scripts. While there were several inconsequential decisions in the process, the most important part was selecting the best beamforming function for our needs. With the data in a radar cube format, we can use beamforming to calculate information about the recording subjects. Different methods of beamforming are used to calculate different information, such as the presence of objects at different ranges and angles or the Doppler shift present at a certain range. For our system, we decided to use the representation of Doppler shift across range and angle, called the range-angle response. This was chosen since the axes of the data are spatial dimensions, which is suited to the CNN-LSTM model that we plan to use.

4.2. Data collection

Recording data for training and validation was a difficult process due to the involved nature of the subjects. The data that we needed to collect involved a human participant actively carrying out different actions, so completing the recordings was a time-intensive task. To speed this up as much as possible, we created a simple Python script that would organise and sort the recordings for us.

The steps used in our data collection are as follows:

1. Recording triggered through mmWave Studio
2. A participant performs actions for the six-second period
3. Recording finishes and the binary file is saved
4. Python script used to relocate and rename binary file

This process was then repeated twenty times per activity, with each participant performing five different activities. Since this was the initial attempt using our system, we chose relatively simple actions to be performed. Standing motionless, pacing back and forth, hand waving, jumping jacks, and clapping were chosen to represent a wide range of different types of motion, with enough similarities present to validate the success of our model. These activities were captured alongside recordings of the environment with no participant present, to serve as the null classification.

4.3. Processing script development

Compared to existing approaches for activity sequence recognition, our approach is more efficient due to the lack of unique classes. Current approaches rely on classifying the boundaries between any permutation of actions, whereas our approach is simply able to report the detection of one action after another. Combined with a simple post-analysis system, we are able to identify sequences just the same, without the need for dozens of boundary classes. This massively simplifies our model and vastly improves its scalability.

5. Discussion

Our experimental procedures were able to return some encouraging results, and while we were not able to analyse a full system pipeline for the detection of multi-activity sequences, the available results suggest that our system is able to see success in the future. Since we were only able to train on single activities and not activity sequences, we are looking to compare our sliding window and frame selection approach to a baseline implementation of a CNN-LSTM model.

We saw successes as our model presented decreasing losses, and after 120 epochs, it resulted in an accuracy of 90%. Most importantly, the novel implementation was able to match the baseline implementation in accuracy. Since our activity sequence solution does not take place within a neural network, it is sufficient to simply match the baseline in effectiveness. This proves that our solution is no worse than existing approaches in accuracy and efficiency.

6. Conclusion

While we have made a good start on our research goal, it is far from finished. Our process saw acceptable results when tested on individual activities; however it has yet to prove its usefulness for the classification of multi-activity sequences. Based on the robustness of our sliding window method, it can be assumed that we will see similar results when applying our system to multi activity sequences. I am also interested in continuing to research the possibility of real-time detection, which is something that we investigated in our literature review before deciding on a different direction for our studies.

6.1. Future works

One of the biggest non-technical roadblocks to this research is the collection of training data. Since we are using a unique technology, there are no existing large datasets that can be used. The process of data collection is also more time-consuming and laborious than other use cases for machine learning. One way this process could be improved is by developing a novel pipeline for data recording, bypassing the mmWave Studio GUI. Since both radar control and data transmission are performed over existing communication protocols (SPI and TCP), it is entirely reasonable to create this process from the ground up. Implementing such a system would dramatically increase the efficiency of data collection, making it feasible to gather more than enough data for a serious attempt at improving this technology. This is an approach that we attempted to implement, but we found that the time investment would not be worthwhile considering the small scale of our experimental approach.

7. References

- [1] C. Yu, Z. Xu, K. Yan, Y. -R. Chien, S. -H. Fang and H. -C. Wu, "Noninvasive Human Activity Recognition Using Millimeter-Wave Radar," in IEEE Systems Journal, vol. 16, no. 2, pp. 3036-3047, June 2022, doi: 10.1109/JSYST.2022.3140546.
- [2] J. -H. Li, L. Tian, H. Wang, Y. An, K. Wang and L. Yu, "Segmentation and Recognition of Basic and Transitional Activities for Continuous Physical Human Activity," in IEEE Access, vol. 7, pp. 42565-42576, 2019, doi: 10.1109/ACCESS.2019.2905575.
- [3] Bhavanasi, G., Werthen-Brabants, L., Dhaene, T. et al. Patient activity recognition using radar sensors and machine learning. *Neural Comput & Applic* 34, 16033–16048 (2022).
- [4] R. Zhang and S. Cao, "Real-Time Human Motion Behavior Detection via CNN Using mmWave Radar," in IEEE Sensors Letters, vol. 3, no. 2, pp. 1-4, Feb. 2019, Art no. 3500104, doi: 10.1109/LSENS.2018.2889060.
- [5] X. Cheng, B. Huang and J. Zong, "Device-Free Human Activity Recognition Based on GMM-HMM Using Channel State Information," in IEEE Access, vol. 9, pp. 76592-76601, 2021, doi: 10.1109/ACCESS.2021.3082627.
- [6] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie and J. Zhou, "TSE-CNN: A Two-Stage End-to-End CNN for Human Activity Recognition," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, pp. 292-299, Jan. 2020, doi: 10.1109/JBHI.2019.2909688.
- [7] P. Zhao, C. X. Lu, B. Wang, N. Trigoni and A. Markham, "CubeLearn: End-to-end Learning for Human Motion Recognition from Raw mmWave Radar Signals," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2023.3237494.
- [8] J. -T. Yu, L. Yen and P. -H. Tseng, "mmWave Radar-based Hand Gesture Recognition using Range-Angle Image," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9128573.
- [9] A. Ninos, J. Hasch and T. Zwick, "Real-Time Macro Gesture Recognition Using Efficient Empirical Feature Extraction With Millimeter-Wave Technology," in IEEE Sensors Journal, vol. 21, no. 13, pp. 15161-15170, 1 July 1, 2021, doi: 10.1109/JSEN.2021.3072680.
- [10] H. Xie, P. Han, C. Li, Y. Chen and S. Zeng, "Lightweight Midrange Arm-Gesture Recognition System From mmWave Radar Point Clouds," in IEEE Sensors Journal, vol. 23, no. 2, pp. 1261-1270, 15 Jan. 15, 2023, doi: 10.1109/JSEN.2022.3216676.
- [11] A. Sengupta, F. Jin, R. Zhang and S. Cao, "mm-Pose: Real-Time Human Skeletal Posture Estimation Using mmWave Radars and CNNs," in IEEE Sensors Journal, vol. 20, no. 17, pp. 10032-10044, 1 Sept. 1, 2020, doi: 10.1109/JSEN.2020.2991741.

Figure 2: mmWave Studio, Texas Instruments, recreated by Beck Busch

Figure 3: Kang, Sung-wook (2022). Autoencoder-Based Target Detection in Automotive MIMO FMCW Radar System. Sensors. 22. 5552. 10.3390/s22155552.

8. Appendix

8.1. Literature review sources table

| Paper | Activities | Segmentation Method | Technology | Real-time Implications |
|--------------------------------|---|--|--|--|
| Full-body Activity Recognition | | | | |
| Chengxi Yu et al. [1] | Walking, falling, sitting, standing, including transitions | Sliding window | mmWave Radar DVCNN | Real-time operation possible; detection not attempted. |
| Jun-Hua i Li et al. [2] | Walking, sitting, laying, including stairs and transitions | Sliding window | Physical sensors HMM | No attempt at real-time detection |
| Geethika Bhavani si et al. [3] | Walking, falling, sitting, laying, rolling, including chair, bed, and transitions | Recording sample | mmWave Radar CNN with various inputs | No attempt at real-time detection |
| Renyuan Zhang, Siyang Cao [4] | Walking, sitting, hand waving, including transitions | No mention | mmWave Radar CNN with Micro-Doppler | Real-time classification possible; no results provided |
| Xiaoyan Cheng et al. [5] | Walking, sitting, squatting, falling, boxing, pushing, waving | Recording sample | Channel State Information GMM-HMM | No attempt at real-time detection |
| Jiahui Huang et al. [6] | Walking, jogging, standing, sitting, ascending stairs, descending stairs | Sliding window | Wearable sensors. Two-stage end-to-end CNN. | No attempt at real-time detection |
| Peijun Zhao et al. [7] | Various motions | Recording sample | CubeLearn, CNN&LSTM | An attempt made at real-time implementation. |
| Arm Gesture Recognition | | | | |
| Jih-Tsun Yu et al. [8] | Various hand motions, including grabbing, tilting, turning, and swiping. | Recording sample | CNN&LSTM with Range-angle image | Real-time classification was not attempted. |
| Alexandros Ninos et al. [9] | Swiping, rotating, pattern, and random | Recording samples with event detection | MLP and custom algorithm | Capable of real-time recognition. |
| Haihua Xie et al. [10] | Lifting, lowering, pushing, pulling, circle motions, and combinations | Recording sample | PointNet classifier | Lightweight classifier for real-time operation. |
| Arindam Sengupta et al. [11] | Walking, swinging arms | Individual frame analysis | CNN with point cloud input | Capable of real-time operation. |