High-Resolution Radar Sensors for Human Gait Classification

KHADIJAH HABIB & KUAN TEH WAN
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

High-Resolution Radar Sensors for Human Gait Classification

Khadijah Habib
Kuan Teh Wan

Department of Electrical and Information Technology
Lund University

Supervisor: Daniel Sjöberg

Examiner: Michael Lentmaier

June 17, 2025

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Video cameras are widely used for gait-based surveillance systems. However, they raise privacy concerns in sensitive environments such as private homes or restricted areas. As a result, radar-based methods are being explored as a privacy-preserving alternative. These methods are particularly promising with the emergence of high-resolution radar sensors capable of operating effectively in indoor conditions.

This thesis investigates a classification pipeline that uses radar-based gait data to identify different walking patterns relevant to surveillance scenarios. The input data is preprocessed into radar RGB spectrograms of standardized size and fed into a Convolutional Neural Network (CNN) architecture. Three classes are considered: walking, walking with hands in pockets, and walking while carrying a box. Multiple CNN architectures were explored and optimized, including experiments with different input channels, convolutional depths, and pooling methods. The performance of the trained models is evaluated using separate training, validation, and test datasets.

The final model achieved high validation accuracy but showed a drop in test performance, suggesting signs of overfitting. Results indicate that while CNN-based classification is feasible for real-world gait analysis from radar data, careful attention must be paid to model complexity and dataset quality to improve generalizability.

Keywords

FMCW radar, Gait classification, CNN, Machine Learning, Deep Learning

Declaration

We hereby affirm that this Master thesis was composed by ourselves, that the work herein is our own except where explicitly stated otherwise in the text. This work has not been submitted for any other degree or professional qualification except as specified; nor has it been published. Where other people's work has been used (either from a printed source, internet, or any other source), this has been carefully acknowledged and referenced.

During the preparation of this thesis, we have used ChatGPT 3.5 to assist us in the writing process to improve language, flow, and readability. After using these tools/services, we have reviewed and edited the content as needed and take full responsibility for the content of the whole thesis.

Acknowledgements

I'm grateful to everyone who's been part of this journey, in ways seen and unseen.

To my supervisors, Daniel and Sebastian — thank you for your guidance, insight, and patience. To those I've worked around during this time — thank you for the quiet encouragement that comes simply from being alongside others. To my family and friends — thank you for holding me up in all the ordinary ways that matter most.

Whatever clarity or direction has come, has come through the unfolding of many moments, many people, and something larger that moves beneath it all. I can only be grateful.

Khadijah Habib

We would like to thank our industry supervisors, Anders Mannesson and Sebastian Heunisch, for all the support and guidance throughout this thesis.

We also want to thank our supervisor at Lund University, Daniel Sjöberg, for the support in technical questions and comments on the report.

Kuan Teh Wan

Popular Science Summary

In this thesis, a radar-based gait classification system was developed to distinguish subtle arm movements between different walking patterns relevant to surveillance and security. Using machine learning techniques and range-Doppler signatures, the system identifies motion behaviors such as walking, carrying, and concealing actions. Exploring the challenges of real-world gait recognition, from data collection to classification accuracy, and proposes an approach that is both robust and adaptable to varied operational environments.

The work involved collecting radar data using a synchronized camera-radar setup to capture real human motion sequences. Each frame will then be processed into range-Doppler images—frequency-distance representations that encode the dynamic features of body movement. Convolutional neural network (CNN) architectures were designed and trained on these images in a sliding-window fashion to automatically extract features and classify the gait type. The networks were tuned using different preprocessing strategies, architectural variants, and performance metrics to achieve robust classification accuracy.

One of the most interesting findings was the sensitivity of classification performance to noisy transitional frames. After removing these and optimizing the CNN structure, the model achieved improved reliability and interpretability. Suggesting that, when processed correctly, the radar data carry enough distinctive features to differentiate these slight variations between gait types. Results also show that the CNN models performed well with acceptable accuracy given the small amount of self-collected data in limited time.

The thesis addresses a growing need in modern surveillance systems: the ability to detect and respond to specific human behaviors without relying on cameras, which often raise privacy concerns. Traditional vision-based methods, while powerful, may be unsuitable in low-light or occluded environments and can be intrusive. Radar-based approaches, in contrast, are anonymous, robust to lighting conditions, and can operate unobtrusively in both indoor and outdoor settings. By focusing

on behavior rather than identity, this method supports ethical monitoring while preserving individual privacy.

The relevance of this work lies in its potential to enable intelligent, context-aware surveillance systems that can flag unusual activity without human oversight. In environments such as airports, border checkpoints, or secure facilities, early detection of abnormal motion—such as concealed carrying or hesitant movement—can enhance response times and prevent incidents before they escalate.

Beyond security, the system can be adapted for healthcare monitoring, such as mobility issues in elderly patients, where privacy and non-intrusiveness are also key. The modular design of the pipeline allows for future extension to more complex actions or integration with other sensors.

Overall, this thesis contributes to the growing field of radar-based human activity recognition and presents a framework that balances technical feasibility with ethical awareness.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Human gait analysis has long been a subject of interest due to its numerous applications in security surveillance and biomedical domains [1]. In the computer vision community gait recognition techniques have made significant progress but they often face challenges under real-world conditions. Factors such as varying lighting, changing camera viewpoints and occlusions can degrade the performance of vision-based gait recognition systems [2]. Moreover video-based approaches may raise privacy concerns since they capture identifiable visual information of individuals. These limitations motivate the exploration of alternative sensing capabilities for gait classification.

Radar-based gait classification has emerged as a compelling complementary approach to vision. Radars are largely insensitive to lighting and weather conditions and can even penetrate certain occlusions (like clothing) allowing operation in scenarios where optical systems would fail [1]. A radar system transmits electromagnetic waves and analyzes the returned signals to infer target motion. In particular, the Doppler effect is used to measure the velocity of a moving object by the frequency shift of the reflected waves [1]. For a walking person, which is a multi-jointed target, different body parts (legs, arms, etc.) introduce additional, smaller Doppler frequency modulations around the main body Doppler frequency. These modulations produce sideband frequencies known as micro-Doppler signatures [1].

A micro-Doppler signature encapsulates the characteristic movements of various limbs; for example, the swinging of arms and legs induces distinct time-varying frequency patterns in the radar return. By capturing a person's micro-Doppler signature, a radar can obtain a sort of fingerprint of the gait dynamics.

Modern radar hardware makes it feasible to record these signatures at high resolution. In this work, we use a Texas Instruments IWR6843AOP millimeter-wave radar, which is a Frequency-Modulated Continuous Wave (FMCW) radar sensor. FMCW radars transmit periodic chirped signals and measure the frequency

difference between transmitted and received signals over time. This enables the simultaneous estimation of target range (distance) and radial velocity.

The core problem addressed in this thesis is the difficulty of accurately classifying human walking styles using radar data, especially when dealing with limited resolution and noisy environments. While radar spectrograms provide detailed representations of motion, including both Doppler and micro-Doppler signatures, many existing approaches either do not fully utilize this information or fail to generalize across different walking patterns.

To address this, we use a Texas Instruments IWR6843AOP millimeter-wave radar to collect high-resolution time–Doppler data from multiple walking styles. The goal is to extract meaningful representations that enable reliable classification of normal walking, walking with hands in pockets and walking while carrying objects. Our method involves generating 3-channel Doppler spectrograms by segmenting the signal into upper, central and lower regions, then training convolutional neural networks (CNNs) to classify the resulting spectrograms.

Recent advances in deep learning have significantly influenced radar signal classification, including gait recognition. Instead of manually crafting features, modern approaches leverage convolutional neural networks (CNNs) to automatically learn features from the radar spectrogram data. Kim and Moon [3] were among the first to introduce deep learning for human micro-Doppler analysis, using a deep CNN directly on raw micro-Doppler spectrograms. Their method achieved high accuracy (around 90.9% for classifying multiple human activities) without the need for explicit feature extraction, as the network learned the distinguishing patterns from the spectrogram images themselves.

This demonstrated the power of image-based learning in radar applications: the same deep learning architectures successful in computer vision could be repurposed for classifying human motions via radar. Subsequent works have extended this idea to more challenging tasks like identifying individual persons by their gait. For example, Papanastasiou et al. [2] investigated radar-based gait biometrics – using a person's walking micro-Doppler signature as a unique identifier. By training deep learning models (including CNNs on time-frequency representations), they showed that subtle gait differences between individuals can be recognized by a radar system, achieving over 93% identification accuracy on a test with 22 subjects. Such results confirm that human gait carries distinctive signatures in radar data that can be learned and exploited by modern algorithms.

These developments in radar sensing and learning algorithms form the backdrop and motivation for our work. High-resolution mmWave radar sensors (like the TI IWR6843AOP) offer the possibility of capturing fine-grained gait dynamics and image-based deep learning models provide a powerful tool to classify those dynamics. In this thesis, we aim to combine these technologies to perform robust human gait classification. We are also interested in how reduced radar resolution might retain the separability of gait patterns.

1.2 Purpose

The main objective of this thesis is to develop and evaluate a radar-based gait classification system that leverages high-resolution radar sensing and image-based machine learning. We focus on the TI IWR6843AOP FMCW radar as our sensing platform and convolutional neural network models for classification. In particular, we intend to:

- Capture and exploit high-resolution radar data for gait classification. This involves configuring the radar to its high-resolution modes and developing a signal processing pipeline to produce time-frequency spectrograms of walking targets. By doing so, we aim to capture subtle motion features (e.g., leg swing velocity profiles, arm movement patterns) that lower-resolution systems might miss.
- Apply image-based deep learning for gait classification. We plan to design or adapt a deep learning model (such as a CNN) that takes the radar spectrogram as input and classifies the gait. The classification task encompasses categorizing the gait type/condition (e.g., normal vs. carrying object). The key objective is to let the model learn discriminative features from the radar images, rather than relying on manual feature engineering.

The research question we aim to answer is:

- How accurately can a high-resolution mmWave radar sensor classify human gait patterns using image-based deep learning techniques?

We make the following contributions:

- Data was collected using the TI IWR6843AOP radar under diverse environmental conditions.
- A processing pipeline was developed to extract and structure micro-Doppler spectrograms building on existing methods while adapting them to our range-region segmentation approach.
- An RGB spectrogram input was constructed by segmenting the micro-Doppler image into upper, central and lower Doppler regions.
- A convolutional neural network was trained to classify three gait types using the segmented spectrogram input.

1.3 Related Work

1.3.1 Vision and Inertial Methods

Human gait classification has been traditionally approached through vision-based systems and wearable inertial sensors. Vision-based methods rely on video sequences to capture the motion patterns of individuals. Techniques such as silhouette extraction [4], optical flow analysis and model-based pose estimation have

been widely used. Although successful in controlled settings, vision-based systems often face challenges in varying lighting conditions, occlusions, and background clutter. Furthermore, they raise privacy concerns as they capture identifiable images of individuals [4–6].

Wearable inertial sensors [7], such as accelerometers and gyroscopes, have also been employed to capture gait dynamics. These systems provide reliable data independent of environmental lighting but require individuals to carry or wear devices, which limits their applicability in public or non-cooperative settings. Given these limitations, there has been growing interest in contactless sensing modalities, with radar emerging as a promising alternative for human gait analysis.

1.3.2 Radar-Based Gait Recognition

Tivive et al. [1] demonstrated that Doppler spectrograms could be used to distinguish between different arm motions during walking. Their method relied on image processing techniques to extract statistical and structural features from time-frequency representations of the radar returns.

With the advancement of machine learning, researchers shifted towards using data-driven approaches. Kim and Moon [3] were among the first to apply deep convolutional neural networks (CNNs) directly to radar micro-Doppler spectrograms, bypassing the need for manual feature extraction. Their results highlighted that CNNs could effectively learn discriminative features from raw spectrograms, achieving high classification performance.

Subsequent works explored radar-based person identification. Papanastasiou et al. [2] trained CNNs on spectrograms to recognize individuals based on their unique walking patterns. Other studies, such as those by Gokaraju et al. [8], extended radar classification to broader domains like distinguishing humans from birds using micro-Doppler signatures.

1.3.3 Spectrogram-Based Human Activity Classification

Spectrograms provide a compact and informative representation of radar returns, mapping time against Doppler frequency shifts. Human walking activities generate characteristic micro-Doppler patterns: the torso typically induces a dominant, relatively stable Doppler shift, while swinging arms and legs create periodic sidebands.

Many early works treated the spectrogram as a grayscale image input to machine learning models. For instance, Kim and Moon [3] successfully applied CNNs on grayscale spectrograms without additional segmentation. Other researchers [1, 9, 10] experimented with handcrafted feature extraction from spectrograms, such as dominant frequency tracking or energy-based statistics, to classify activities.

1.3.4 Machine Learning Approaches in Radar Signals

Traditional radar signal classification methods relied heavily on manual feature design. Typical features included mean and variance of Doppler frequencies, energy distributionsand duration of micro-Doppler events. While these methods offered interpretability, they often required extensive domain knowledge and did not generalize well to new datasets.

The introduction of deep learning, particularly CNNs, changed the landscape. Instead of manually designing features, models could learn hierarchical feature representations directly from raw or minimally processed spectrogram data. Kim and Moon [3] achieved over 90% accuracy on human activity classification using deep CNNs. Later works expanded on this success by exploring deeper architectures and hybrid models.

Despite these advances, many studies either focus on broad activity categories (e.g., walking, sitting, running) or on individual identification. There remains relatively limited work specifically addressing subtle variations in gait style under diverse environmental conditions.

1.3.5 Limitations of Prior Studies and Our Focus

A review of prior works reveals several common limitations:

- Many studies collect data under often limited environmental variation and subject diversity.
- Most radar-based classification research either categorizes broad activities or identifies individuals, rather than recognizing nuanced differences between gait styles.
- Spectrograms are often used as raw grayscale images without structured segmentation to highlight different motion parts.
- The role of range resolution in micro-Doppler pattern clarity and classification accuracy is underexplored.

Our work is designed to address these gaps. We construct a dataset involving five participants walking under different environmental conditions, capturing diverse radar returns. Our target is gait type classification, not person identification. We preprocess spectrograms to explicitly encode upper, central and lower Doppler motion beginning the investigation into the effects of radar range resolution, laying the groundwork for future real-time, high-performance radar-based gait recognition systems.

Chapter 2

Theory

2.1 Radar

Radio Detection and Ranging (Radar) is a technology that uses radio waves to detect and track objects in various environments. It works by transmitting radio waves, which reflect off targets and return to a receiver. This process makes it possible to calculate the range, velocity, and angle of an object [11, 12].

Radar has widespread applications, such as air traffic control, military defense, automotive collision avoidance, and weather monitoring. Its ability to operate in diverse conditions, for example, low visibility or in rain, makes it a robust alternative to video cameras. In this thesis, we use Frequency Modulated Continuous Wave (FMCW) radar to extract motion characteristics of walking individuals.

2.1.1 FMCW Radar

An FMCW radar transmits a signal called a *chirp*. A chirp is a sinusoidal wave whose frequency increases (or decreases) linearly with time, as shown in Fig 2.1. The rate of frequency change is called the slope S and the total frequency span is the bandwidth B.

FMCW radar captures the reflections of these chirps from objects and by comparing the transmitted and received signals, the system can estimate the distance and motion.

2.1.2 Range

The round-trip time is denoted as τ, distance between the object and radar is d. As shown in Fig 2.2, when the frequency changes with slope S and the speed of light is denoted as c, $3 \times 10^8 m/s$, we have the frequency F that an object produces as

$$F = S \cdot \tau = S \cdot \frac{2d}{c} \tag{2.1}$$

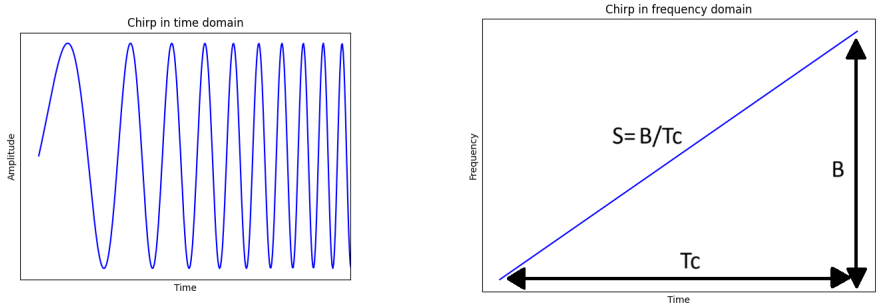


Figure 2.1: Linear frequency chirp in time and frequency domains. S, B, T_c denotes slope, bandwidth and the duration of the signal respectively.

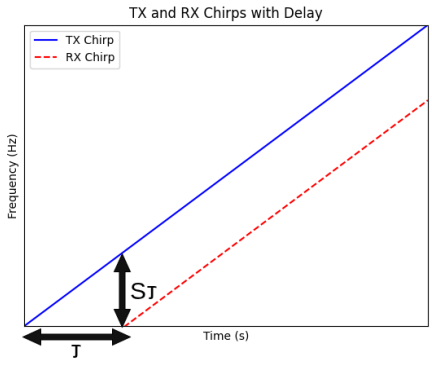


Figure 2.2: Radar transmitter and receiver delay in time. τ denotes the round-trip time.

Let T_c denote the duration of the signal. Then for two objects to show up as distinct peaks, their frequency difference Δf, must fulfill the condition that

$$\Delta f > 1/T_c \tag{2.2}$$

Substituting these two equations, we have that

$$S \cdot \frac{2d}{c} > \frac{1}{T_c} \tag{2.3}$$

With the slope $S = B/T_c$, where B is the bandwidth, and rearranging, we have that the range resolution

$$d_{res} > \frac{c}{2ST_c} = \frac{c}{2B} \tag{2.4}$$

2.1.3 Velocity

For moving objects, velocity causes a Doppler frequency shift Δf_v. The radial velocity v_r is computed as:

$$v_r = \frac{\Delta f_v \cdot \lambda}{2} \tag{2.5}$$

where λ is the wavelength of the transmitted signal. Multiple chirps are needed for Doppler estimation [12].

2.1.4 Fast Fourier Transform

The Fast Fourier Transform (FFT) is a computational method used to convert signals from the time domain into the frequency domain. In radar signal processing, FFTs are applied along multiple dimensions to resolve information about distance, motion, and spatial direction.

Each radar chirp is sampled over time by an analog-to-digital converter (ADC), which converts the continuous-time reflected signal into discrete digital values. These samples are known as *ADC samples*, and they form what is referred to as the fast time axis. Performing a 1D FFT along this axis provides range information, allowing us to estimate how far objects are from the radar.

The radar also sends out a sequence of chirps over time. The phase shift between these consecutive chirps is analyzed using a second FFT applied along what is called the slow time axis. This resolves the Doppler frequency, which corresponds to the radial velocity of the moving object.

Lastly, the radar captures signals using multiple receiving antennas. Applying FFT across these channels enables estimation of the angle of arrival (AoA). While AoA is not used in our classification pipeline, this dimension completes the three-dimensional data cube.

These three dimensions form a 3D range–Doppler–antenna cube with shape:

`ADC samples (fast time)` $\times$ `chirps (slow time)` $\times$ `antennas (channels)`

In our system, all FFT operations are performed internally by the radar hardware. The radar outputs amplitude images—2D slices of the data cube representing Doppler vs. range intensity at a given frame. These are saved as Python `.pkl` (pickle) files.

A `.pkl` file is a serialized Python object file. In our case, it contains the amplitude data as a NumPy array along with associated metadata. Using pickle files makes it easy to load the processed spectrogram frames into Python for further handling, such as thresholding, slicing, and classification.

2.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a class of models that learn a mapping between inputs and outputs by composing layers of weighted linear transformations followed by nonlinear activation functions [13]. These models are inspired by simplified representations of neurons and synapses, but are primarily used for their empirical performance rather than biological plausibility.

An ANN typically consists of an input layer, one or more hidden layers, and an output layer. Each layer contains a set of units (neurons), where each unit computes a weighted sum of its input, adds a bias term, and passes the result through an activation function. This transformation produces a new representation of the data, often called an embedding.

For classification tasks, the most commonly used activation functions are the Rectified Linear Unit (ReLU) for hidden layers and the softmax function for the output layer. These are defined as follows:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2.6}$$

$$\mathrm{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.7}$$

In the ReLU function, x is the input to a neuron, and the function outputs either 0 or x, whichever is greater. This introduces non-linearity while keeping computations efficient and simple.

In the softmax function, x_i refers to the i-th element of the input vector $\mathbf{x}$, which contains the raw output scores (logits) from the last layer of the neural network. The softmax normalizes these values into a probability distribution across all classes, where each output lies between 0 and 1 and the sum across all outputs equals 1.

The model is trained using gradient descent to minimize a loss function. For multi-class classification, the categorical cross-entropy loss is used. It is defined as:

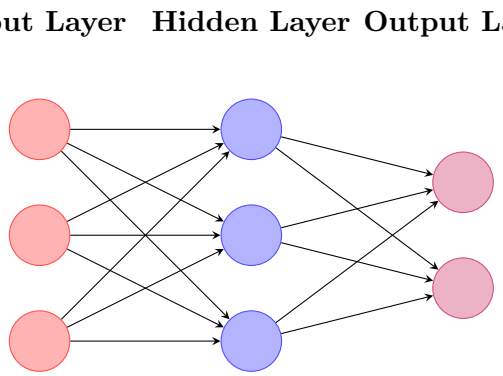


Figure 2.3: Illustration of a simple feedforward neural network with one hidden layer.

$$L = -\sum_{i} y_i \log(\hat{y}_i) \tag{2.8}$$

Here, y_i is the true label for class i, expressed as a one-hot encoded vector, where the correct class has value 1 and all others have value 0. The term $\hat{y}_i$ represents the predicted probability for class i, as output by the softmax function. The loss function measures the dissimilarity between the true and predicted distributions. Lower values indicate better alignment between prediction and ground truth.

Figure 2.3 shows a simple example of a feedforward neural network (also known as a perceptron), consisting of an input layer, a hidden layer, and an output layer, where each layer is fully connected to the next.

2.3 Convolutional Neural Network(CNN)

Convolutional Neural Network (CNN) is a class of deep learning models designed primarily for analyzing *spatial relations*, making it especially effective for image analysis tasks. It is also known for its *shift invariant* specialty, based on the shared-weight architecture of the convolution kernels or filters that slide along input features. Thus, it is widely used in fields such as computer vision, medical imaging and object recognition.

The key to its success in analyzing spatial data lies in the multiple convolutional layers it has and the *feature detection* ability. For example, the first layer can detect edges, textures and shapes. While further layers learn more complex and abstract representations of the image.

This is based on *convolution*,

Original	Gaussian Blur	Sharpen	Edge Detection
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
			

Figure 2.4: Image after using different kernels. EyeQ Tech (2018, Sep 20) Convolutional Neural Network (CNN) overview. https://www.medium.com/@eyeq/convolutional-neural-network-cnn-overview-33026f15dd28

$$H[m] = (I * K)[m] = \sum_n K[m-n]I[n]$$

The convolution can, of course, be multi-dimensional. If the input I is a vector itself (perhaps representing an RGB image), then the kernel (filter) K is a matrix, creating a new vector H as the convolution result.

However, in deep learning, the kernels in the convolution are not reversed. So, essentially, is *cross-correlation*

$$R_{IK}[m] = \sum_n K[m+n]I[n]$$

With CNN's advantages in exploring spatial relations, we still need to consider its lack of explicit temporal awareness. Since a CNN operates independently on each input (e.g., a single image) at a time, meaning that it does not have an inherent mechanism to track temporal relations over time, in our case, like joint positions or step sequences, which are closely related to what we want to investigate. Therefore, a CNN model with a modified structure for a better fit to temporal data should be considered.

2.3.1 Convolution Kernels

A convolution kernel, also known as a filter, is a small matrix that slides over the input image and performs element-wise multiplication, summing the results to produce a single value in the output feature map. Fig. 2.4 is an example of applying various filters to an image.

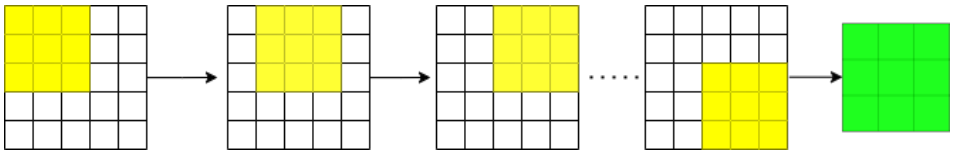


Figure 2.5: 5x5 image and the filter (yellow) is 3x3. Moving the filter throughout the image, with stride = 1 and no extra border pixels padded, will produce a 3x3 filtered images (green).

To emphasize, we do not need to consider which filter we will use for the CNN, as this is part of the CNN training, where the filters are initialized with random values at first and then further tuned by back-propagation.

The filter is a square matrix ($M \times M$), where the size M depends on the input image's shape and architecture of the network. Filters move in a certain way, starting from the top left corner of the input image and move left to right, top to bottom in a sliding-window fashion, as shown in Fig. 2.5.

2.3.2 Pooling

Pooling is a downsampling operation used in convolutional neural networks (CNNs) to reduce the spatial resolution of feature maps while preserving the most important information. The most commonly used pooling method is **max pooling**, which operates by dividing the input feature map into non-overlapping (or sometimes overlapping) rectangular regions and retaining only the maximum value from each region.

This process serves several purposes. First, it significantly reduces the number of parameters and computations in the network, leading to lower memory usage and faster training. Second, pooling introduces a level of spatial invariance, meaning that small translations or shifts in the input (such as a feature moving slightly to the left or right) do not drastically affect the pooled output. This improves the model's robustness to variations in the input data. Finally, by eliminating less dominant activations, pooling helps prevent overfitting and encourages the model to learn more generalized representations [13].

While max pooling is widely used, other pooling strategies also exist, such as average pooling (which computes the mean of each region) and global pooling (which condenses an entire feature map to a single value). However, for most computer vision tasks, max pooling remains the default due to its ability to preserve the strongest activations, which often correspond to the most relevant features.

2.3.3 Residual Networks (ResNet)

Residual networks introduce shortcut connections that skip one or more layers, allowing the network to learn residual mappings instead of directly learning complex functions. This helps address the vanishing gradient problem in deeper architec-

tures and improves training stability. A residual block typically adds the input of a layer to its output before applying the activation function. Residual connections have been shown to improve performance in image classification tasks and are increasingly used in radar-based activity recognition [14].

2.3.4 Attention Mechanisms

Attention modules help neural networks focus on the most informative parts of the input by assigning weights to different channels or spatial locations. In convolutional networks, attention mechanisms like the Convolutional Block Attention Module (CBAM) apply both channel attention and spatial attention sequentially, enhancing the representational power of the model. These mechanisms can help radar-based models distinguish between subtle differences in human motion by emphasizing important features and suppressing irrelevant ones [15].

Chapter 3

Data

3.1 Overview

This project uses high-resolution radar data collected using the Texas Instruments IWR6843AOP FMCW radar sensor. The goal is to classify human gait into three distinct categories:

- Walking normally (baseline)
- Walking with hands in pockets
- Walking while carrying an object

The dataset was designed to capture subtle differences in upper and lower body motion patterns. Radar was chosen because of its robustness to lighting conditions and ability to preserve privacy while capturing micro-motion details [16].

3.2 Radar Setup

The radar used in this project was the Texas Instruments IWR6843AOP, a mmWave FMCW radar equipped with 3 transmitting (TX) and 4 receiving (RX) antennas. Key configuration parameters of the radar are summarized in Table 3.1. It was mounted on a tripod approximately 2.0 meter above the ground and tilted downward at approximately 15 degrees from the horizontal axis, allowing the radar to better capture motion across the walking path. It stayed fixed during all sessions.

A large industrial camera was mounted in approximately the same position as the radar. Its purpose was not for input to the model but to allow visual confirmation of radar readings during later labeling. It provided synchronized visual footage alongside the radar amplitude images, helping us remember what class was recorded in hindsight.

Table 3.1: TI IWR6843AOP Radar Configuration

Parameter	Value
Model	IWR6843AOP (3 TX, 4 RX)
Chirps per Frame	228 (per TX), total 684
ADC Samples per Chirp	304
Bandwidth	2 GHz (61.25 – 63.25 GHz)
Chirp Duration	35.611 s
ADC Sampling Rate	10 MHz

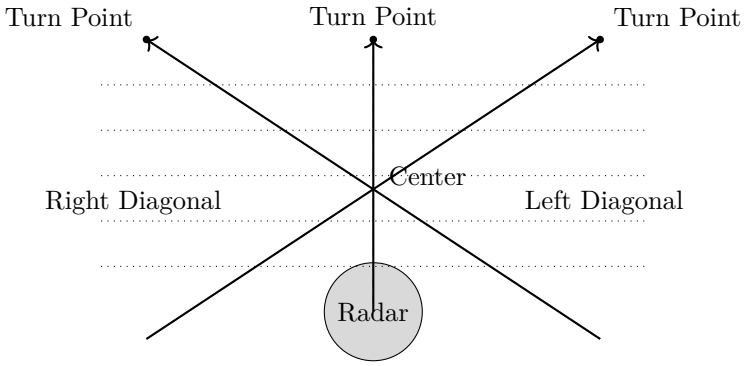


Figure 3.1: Diagram of walking paths used for each session. Paths include center, left-diagonal (crossing to top-right)and right-diagonal (crossing to top-left), each with labeled turn points.

3.3 Recording Setup

Radar data was collected indoors and outdoors, including office corridors, empty rooms, and a tunnel outside. Each session began with careful alignment of the radar field of view and defined walking paths. The subjects walked in three directions: straight toward the radar, diagonally from the right, and diagonally from the left. From each start point, the subject walked forward for about 12 steps (around 10 – 12 meters), then turned and walked back. This forward–backward motion was repeated 5 times per session. The walking paths are illustrated in Figure 3.1.

There were markers on the floor to guide step lengths and turn-around points. In most recordings, two participants were recorded taking turns after one other between each of the classes. They walked naturally, without being told how fast to go or how to behave.

Each session lasted about 15 minutes. Some sessions had two people alternating, and each person performed all three classes: normal walk, walk with hands in pockets, and walk while carrying.

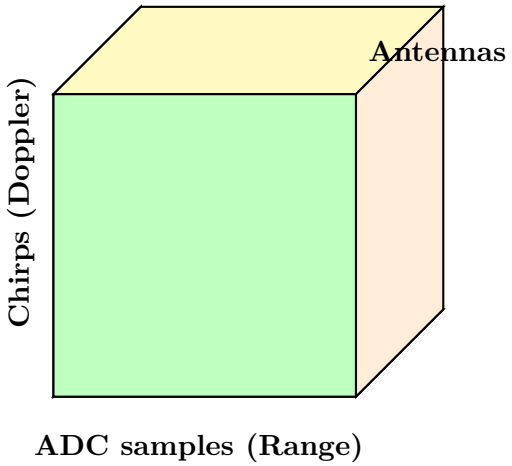


Figure 3.2: Illustration of raw radar data cube with dimensions: ADC samples (fast time), chirps (slow time) and receiving antennas.

3.4 Raw Data Collection

The radar system was operated in CHIRPS mode, where each transmit antenna sequentially sends multiple chirps per frame. This configuration enables the construction of a higher-resolution data cube by combining multiple chirp returns across all transmitter-receiver antenna pairs. More details about CHIRPS mode and related acquisition settings can be found in the Texas Instruments IWR6843AOP datasheet [17]. Each frame contains 684 chirps and 304 ADC samples per chirp, forming a 3D data cube (ADC samples $\times$ chirps $\times$ receiver channels).

3.4.1 Raw Frame-Level Statistics

Before applying any preprocessing, we analyzed the raw radar frame data to understand its structure and limitations. Each frame contains a number of detected objects, with associated radial velocities, signal-to-noise ratios (SNR), and Doppler bin indices.

The average velocity in each frame was computed as a simple arithmetic mean of the radial velocities provided in the radar's `object_list`. No weighting was applied. These velocities are calculated by the radar based on Doppler shifts. SNR values were also taken directly from the radar output and represent the signal-to-noise ratio per detection, estimated internally by the device.

Figure 3.3 shows the variation in average velocity across frames. Transitions between motion states are visible, along with unstable regions likely caused by low detection counts or turning.

Figure 3.4 shows that the velocity data is multimodal, with peaks near zero and

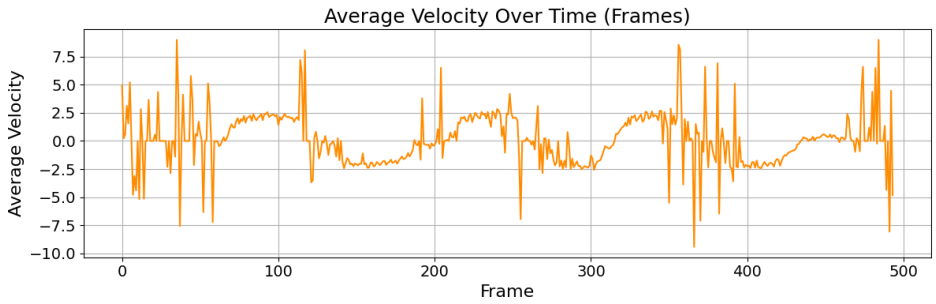
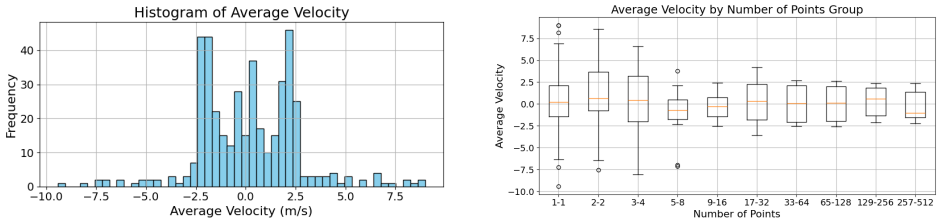


Figure 3.3: Average velocity per frame over time (raw data).



(a) Velocity histogram

(b) Velocity by point group

Figure 3.4: Velocity distribution in raw frames.

±2.5 m/s, consistent with back-and-forth walking. The boxplot confirms that frames with very few points produce unstable velocity estimates. Based on this, we later filtered out low-point frames during preprocessing.

While Doppler bin indices were also available, we relied on the radar-provided velocity values directly, as they are already resolved and more interpretable.

These statistics motivated the first steps of our pipeline, including frame filtering by detection count and later separation of steady-state versus transitional motion.

3.5 Radar Processing Pipeline

The radar system performs all primary signal processing steps internally. Specifically, it applies range and Doppler FFTs to extract distance and velocity information, then stores the amplitude images in .pkl files. These pre-processed frames were the starting point for this project.

3.6 Spectrogram Generation and RGB Encoding

Each resulting amplitude image is initially a Range-Doppler frame, where:

- The vertical axis represents Doppler velocity (i.e., radial motion).
- The horizontal axis represents range (i.e., distance from the radar).
- Brightness indicates the intensity of radar reflections (i.e., how strongly a target reflects the signal at a specific range and Doppler bin).

To capture motion over time, we construct spectrograms by extracting Doppler profiles from each frame (typically by selecting the range bins with higher reflections) and stacking these profiles across consecutive frames. This forms a Doppler-time representation, where:

- The vertical axis represents Doppler velocity (as before).
- The horizontal axis now represents time.
- Brightness continues to indicate the strength of radar reflections at each Doppler bin over time.

The sliding window approach involves grouping a fixed number of consecutive frames into a single spectrogram slice. In our case, we use a window length of 50 frames, with a step size of 5 between windows. This means each spectrogram covers a temporal span of approximately 5 seconds, assuming a frame rate of 10 frames per second. The window captures short sequences of motion, allowing the network to learn local temporal features while maintaining a manageable input size. A longer window would capture more of the walking sequence but reduce the number of training samples, while a shorter window would yield more samples at the cost of losing temporal context.

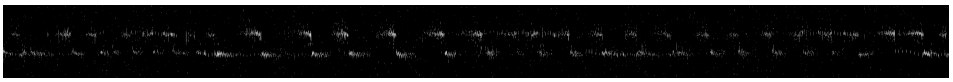
To improve feature separation, we performed slicing on the Range-Doppler images before spectrogram construction. We focused on a narrow Doppler region of interest— around 10 bins with the strongest reflections—and divided it vertically along the range axis into three segments:

- Higher range bins (upper region)
- Mid-range bins (central region)
- Lower range bins (lower region)

These segments were averaged and stacked to form an RGB-style input. This structure helped preserve subtle variations across the range dimension while keeping the input compact. Figure 3.5 illustrates the full spectrogram pre-processing pipeline, showing how the Doppler range is divided and mapped to RGB channels.

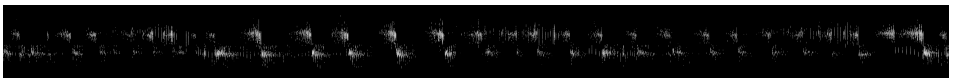
3.7 Dataset Construction

The processed spectrogram slices were organized into three labeled classes stored in directory-based format for image classification. Each image slice has shape ($200 \times$

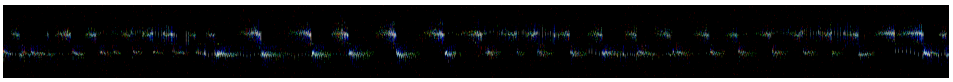


(a) Spectrogram section highlighting higher Doppler range to be mapped to the red channel.

(b) Spectrogram section corresponding to central Doppler values to be mapped to the green channel.



(c) Spectrogram section with lower Doppler range to be mapped to the blue channel.



(d) Final RGB spectrogram constructed from the three Doppler segments, illustrating the input format used for classification.

Figure 3.5: Spectrogram pre-processing pipeline: selected Doppler segments are mapped to RGB channels based on frequency range, producing structured inputs for model training.

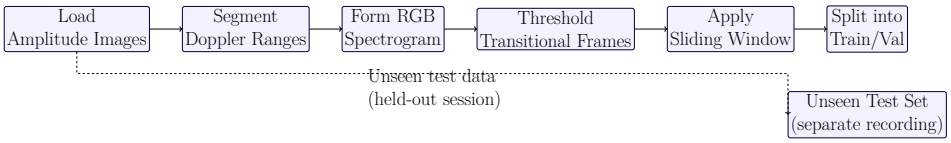
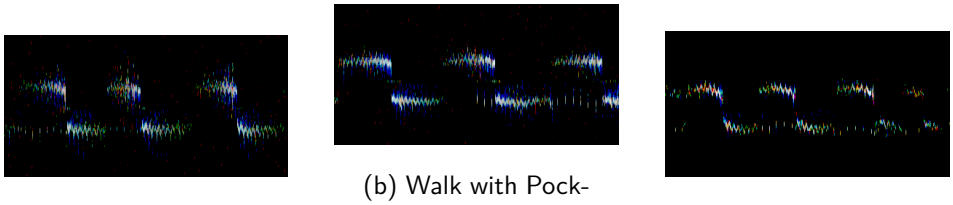


Figure 3.6: Data preprocessing pipeline. The main pipeline prepares RGB spectrogram samples for training and validation. A separate set of amplitude images, collected under different recording conditions is reserved as an unseen test set for final evaluation.



(a) Walk (b) Walk with Pockets (c) Walk Carrying

Figure 3.7: Sample RGB spectrograms for each gait class. Each image encodes upper, middle and lower Doppler reflections in RGB.

50×3). A sliding window of 50 frames with a stride of 5 was applied to extract temporal slices from the full-length recordings. This window length corresponds to approximately 5 seconds of motion, capturing multiple gait cycles within each slice while preserving enough temporal context for classification. The 5-frame stride (i.e., 90% overlap) allows for dense sampling, increasing the number of training examples while maintaining continuity across adjacent slices. This setup was chosen to balance temporal resolution with dataset size and to ensure that key gait transitions—such as arm swings or leg motion patterns—are adequately represented within each input sample.

The overall data preprocessing workflow is summarized in Figure 3.6, showing how amplitude frames are transformed into structured spectrogram slices.

The dataset was split into:

- 70% training
- 15% validation
- 15% test (unseen set)

Transitions (e.g., turning) were excluded via Doppler-based thresholding.

Table 3.2: Recording Sessions by Gait Class

Gait Class	Number of Sessions	Subjects
Normal Walking	10	5
Hands in Pockets	10	5
Walking While Carrying	10	5

3.8 Data Sets

There were 30 sessions recorded in total. Each of the 5 subjects performed all three classes. Session variation existed but each class had wide enough coverage.

Table 3.2 summarizes the number of high-resolution and low-resolution sessions recorded for each gait class.

3.9 Considerations

Several practical issues were addressed during data preparation:

- Thresholding was used to suppress low-reflection/background clutter/noisy frames.
- Class boundaries (e.g., *pockets* vs. *carrying*) may blur depending on clothing fit, hand placement, and object visibility. To mitigate this, data were manually reviewed during labeling to ensure consistency. Ambiguous or low-confidence samples—such as when hands were partially in pockets or the carried object was not visible—were excluded from the training and validation sets to improve inter-class separation.
- Ensuring test data is from unseen sessions helped validate generalization.

Transitions such as turning, starting and stopping were labeled separately for possible future use.

Chapter 4

Methods

4.1 Overview of Implementation

This chapter describes the full methodological pipeline used in this thesis—from radar data acquisition and signal interpretation to spectrogram generation and classification using convolutional neural networks. The methods were developed incrementally over months and involved extensive trial-and-error, particularly in designing a robust and interpretable preprocessing pipeline. At each stage care was taken to balance data fidelity, model tractability and practical feasibility for deployment. Crucially, the processing responsibilities are shared between the built-in radar system and our own implementation and we describe this separation clearly.

Figure 4.1 shows a simplified classification pipeline. Data from the scene shown is stored as a pickle file, a Python module that makes it easier to serialize variables and load them when needed. The range-Doppler images are then accessed, and peaks are stacked across all frames to generate a spectrogram. Finally, classification is performed based on the spectrogram.

A more detailed working pipeline is shown in Figure 4.2. It shows that the signals from all frames are further split into three channels to facilitate stacking into a single RGB spectrogram. With the spectrogram constructed, the data is then separated into motion (walk, carrying, or hands in pockets) and (de)acceleration, to allow the network to learn better and achieve higher accuracy.

4.2 Data Acquisition and Internal Signal Processing

The sensing device used is the Texas Instruments IWR6843AOP, a 60 GHz millimeter-wave FMCW radar with 3 transmit and 4 receive antennas. It was configured with 304 ADC samples per chirp and 228 chirps per frame, yielding 10 frames per second. The radar was mounted on a tripod at approximately 1.9 meters and angled downward by 15 degrees from the horizontal axis to maximize Doppler visibility from chest and leg motion during human gait.

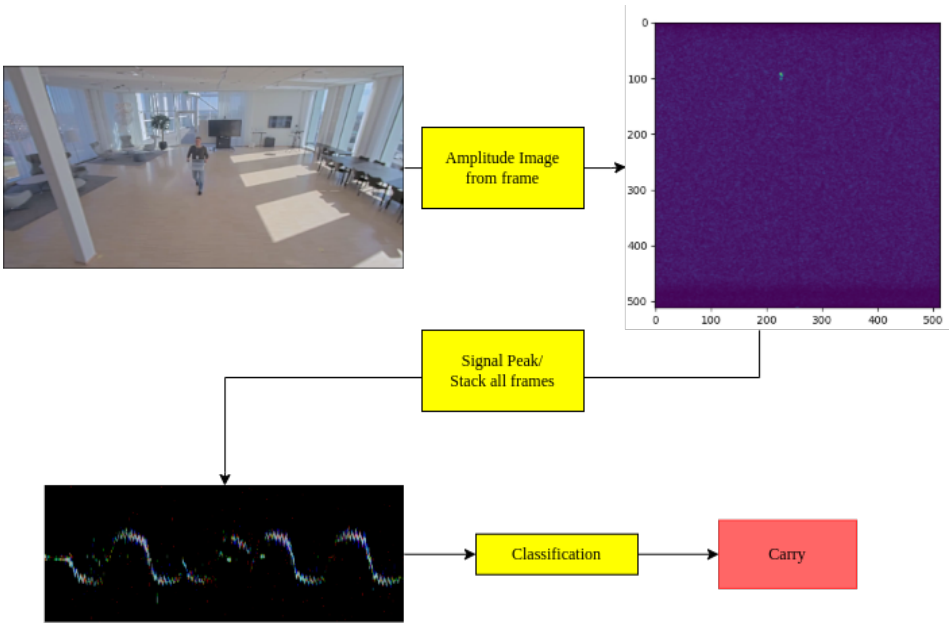


Figure 4.1: Simplified version of the working pipeline.

All radar signal-level processing is handled internally by the radar system itself. This includes:

- Generating FMCW chirps
- Capturing analog reflections from the scene
- Performing ADC conversion
- Executing 1D (range) FFT and 2D (Doppler) FFT
- Applying log-magnitude conversion to yield 2D amplitude images

These amplitude images—each of size 512×512—are output from the radar and saved to `.pkl` files. They represent Doppler (vertical) versus range (horizontal) intensity. Although the raw radar data contains 304 ADC samples per chirp (range bins) and 684 chirps per frame (across all TX antennas), the final image resolution is standardized to 512×512 after the radar's internal signal chain applies FFT operations with zero-padding and/or interpolation. This ensures consistent square-shaped outputs for downstream applications such as visualization or neural network input. Specifically, the radar applies a 1D FFT along the ADC samples to generate range bins and a second FFT across chirps to extract Doppler bins. These outputs are interpolated to 512 bins in each direction as part of the radar SDK's default configuration.

An overview of the full signal pipeline, from radar chirp capture to CNN input generation, is shown in Figure 4.3.

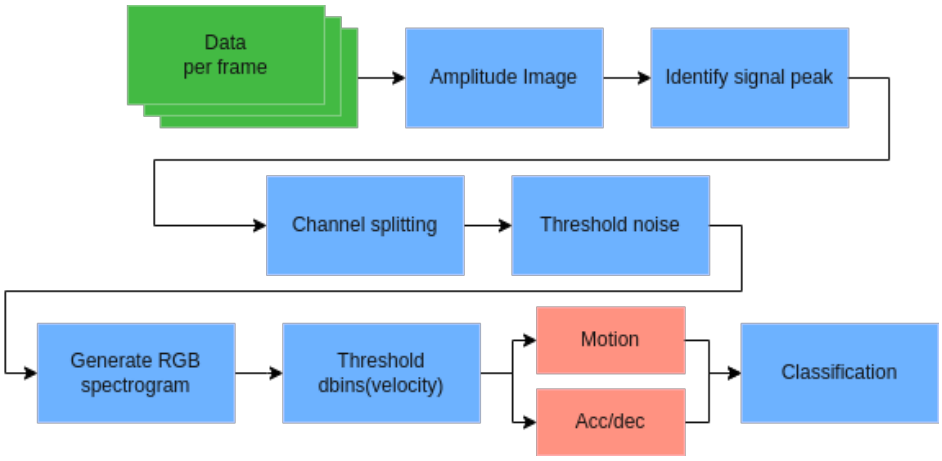


Figure 4.2: Diagram showing how RGB spectrogram is generated and categorized into motion and (de)acceleration. Each blocks are further explained in the next section.

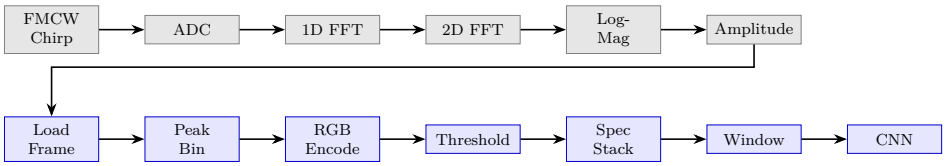


Figure 4.3: Compact system pipeline showing radar signal flow and custom spectrogram processing. Gray boxes represent radar operations; blue boxes represent thesis contributions.

We will now explain how these blocks work in depth.

4.3 Data Processing

Raw data often contains noise, missing values, or inconsistencies that can mislead the learning process. Data processing not only improves accuracy but also speeds up training and reduces overfitting, making it a foundation for successful machine learning.

We begin by looking at the data processing part of the pipeline, namely the "Identify Signal Peak", "Channel Splitting", "Threshold Noise", and "RGB Spectrogram / Threshold dbins" blocks in Fig. 4.2.

4.3.1 Signal Peak

FMCW radar transmits continuous waves and receives them back to detect moving objects. However, even with no one in the radar's range, weaker signals will still be present in the range-Doppler image—for example, as shown in Fig. 4.4.

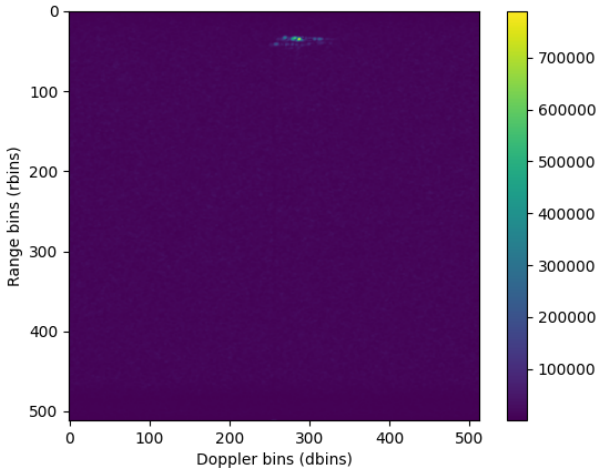


Figure 4.4: Example of a range-Doppler image. x-axis: Doppler (velocity) bins; y-axis: Range bins. Purple region indicates lower intensity.

Our aim here is to classify a person's gait, thus finding an individual's signal should be the first step. Thinking of a human as an object, the torso has a larger radar cross-section compared to other body parts [9, 10]. Therefore, identifying the target signal in a range-Doppler image means locating the torso's signal, which is the signal peak. We can achieve this by inspecting the range-Doppler image as

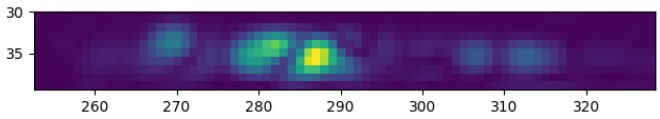


Figure 4.5: Region of interest in the range-Doppler image.

a matrix, then looking for the highest intensity in the matrix and recording its position at the same time.

Next, after finding the signal peak, instead of using the whole image—which includes lots of irrelevant background signal and also increases running time—we select only 5 rows above and below the signal peak, since most meaningful data seems to be in this region, as shown in Fig. 4.5.

4.3.2 Channel Splitting

A majority of available studies on CNN-based natural image classification operate on three-channel RGB images. Although this approach is somewhat optional, results from a similar topic on human activity classification [18] show that using RGB spectrograms leads to faster convergence time, while test accuracy remains the same as using grayscale spectrograms. Thus, we decided to adopt this method to shorten the training process.

Within the 10 rows of interest, we split them into 3 channels with a ratio of 4:2:4. The second channel contains the signal peak; therefore, it has a smaller range. The other two channels have an equal number of rows for the range bins.

Finally, we sum the data within each region along the range axis to form a 1D vector for each channel. By doing this, we extract local spatial context and compress the spatial information, while discarding range bins that are not needed.

4.3.3 Thresholding Noise

From Figure 4.4, in the targeted 10 rows, around the purple region (background weak signal), there is a faint blue area as well, which is close to the target signal. Because of its lower intensity, it will create a lot of noise in the spectrogram later on. Thus, having a thresholding mechanism that filters out low-intensity signals should be considered.

Since the intensity of the range-Doppler image can range from 10^4 (purple region) to 10^7 (brightest, highest intensity), setting a threshold based directly on these values would not be ideal. In the context of range-Doppler image intensity, researchers often use the dB scale (decibel scale), which involves taking the logarithm of the value and multiplying by ten. Ultimately, we empirically set the threshold to 51 dB; any values lower than this are set to 40 dB and considered background signal.

Table 4.1: Radar Spectrogram Dataset Breakdown

Category	Slices	Percent	Description
Steady-State Motion (used)	5007	50%	Walking at constant speed
Transitional Motion (excluded)	4978	50%	Turning, accelerating, decelerating
Total Slices	**9985**	100%	All labeled spectrogram windows

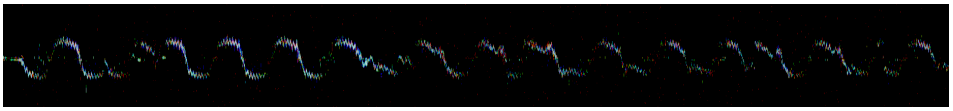


Figure 4.6: RGB spectrogram for a recording.

This thresholding is performed automatically during preprocessing. Frames that fall below this threshold are treated as transitional or noisy (e.g., turning, starting, or stopping), and they are separated from the spectrogram pipeline used for training. In effect, two spectrogram sets are produced: one containing stable walking sequences, and another holding excluded, unstable motion. This filtering step cut the usable dataset approximately in half. Table 4.1 summarizes the distribution.

4.3.4 RGB Spectrogram

With the three channels in hand and after denoising, we can create an RGB spectrogram for each recording. To do this, we stack the three channels depthwise to create a multi-dimensional array with shape: (number of frames, number of velocity bins, 3), as shown in Fig. 4.6.

For all collected recordings, it is crucial to extract only the "unique" information from each class and group shared information into a different class. This is done to avoid confusing the network during training for the classification task.

To elaborate, during the recording, the individual walks both away from and towards the radar. When changing direction, it is expected that the person will slow down in order to turn—this occurs in all classes. The same applies to transitions, such as switching from walking straight to walking diagonally. To address this, we set an adaptive threshold for each recording by inspecting the maximum velocity per frame and the median velocity of the recording. If the difference between them is smaller than one standard deviation, the frame is labeled as a (de)acceleration or transition frame, as shown in Fig. 4.7.

We also attempted to use the (de)acceleration or transition frame slices as a sep-

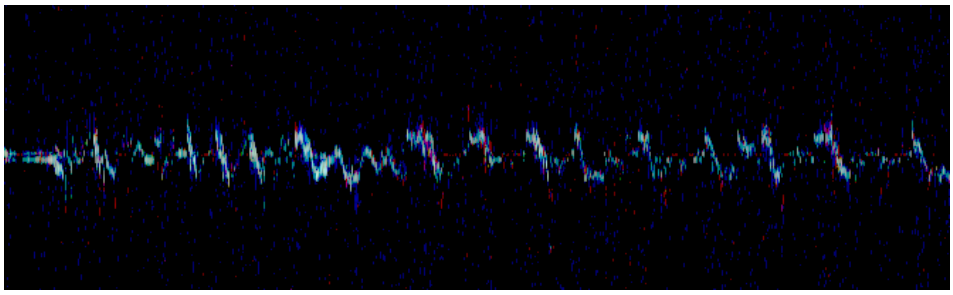


Figure 4.7: Separated (de)acceleration or transition frames from Figure 4.6. Slope-like curve compared to flat lines when walking at a constant speed.

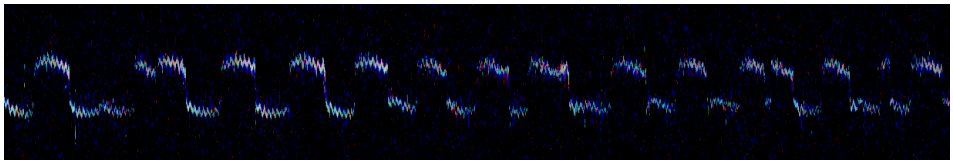


Figure 4.8: RGB spectrogram after filtering out (de)acceleration and transition frames, containing only motion frames.

arate class, turning the task into a 4-class classification problem. However, as mentioned earlier, these frames appear in every class, which created a significant class imbalance. The number of training slices per class was as follows: acc/dec/-trans – 8231 slices, carry – 1724 slices, hands in pocket – 1774 slices, and walking – 1509 slices.

Class imbalance can cause the CNN model to favor the class with significantly more training data, leading to biased decision boundaries—in this case, favoring the (de)acceleration and transition class.

The model may prioritize accuracy on the majority class at the expense of the minority classes, even if it means overfitting to the training data. Under-representation of the minority classes can also prevent the model from learning robust and generalizable features. As a result, the model tends to memorize the training data instead of learning underlying patterns, leading to overfitting. Ultimately, it performs poorly on unseen test data.

This was exactly what happened when we tried solving the problem using the 4-class approach. Despite achieving 100% accuracy in the (de)acceleration and transition class, the other classes performed poorly. Therefore, implementing a mechanism to filter out (de)acceleration and transition frames proved more effective and did not harm overall classification performance. Figure 4.8 shows the RGB spectrogram after discarding the necessary frames.

4.3.5 Sliding Window

Training a CNN-based model that generalizes well and classifies accurately requires more than just a single large spectrogram of the recording. In papers on CNN-based classification models for human activities, using windowing techniques or functions such as the sliding window [18] or Hamming window [19] is common practice in the fields of spectral analysis and time-series analysis.

Slicing one large spectrogram into a greater number of smaller windows is mainly done to increase the amount of data available, so the model is trained with a reasonable number of samples. The average time for an adult to complete a gait cycle is 1.0–1.2 seconds. In this thesis, we use the sliding window method for its simplicity, with a 50-frame (5-second) window. A larger window contains more information, and this also helps compensate for the limited amount of self-collected data.

4.4 CNN Classification

4.4.1 Model 1 – Baseline CNN

Convolutional Neural Networks (CNNs) are commonly used for radar-based human activity recognition, particularly for classifying micro-Doppler spectrograms. When applied to 2D representations, CNNs can extract spatial motion features that correspond to patterns of body movement [9, 10]. In many prior studies, deeper CNNs such as ResNet-18 or modified GoogLeNet have been used to learn these discriminative features [20, 21].

While such models are effective, they are often computationally heavy and contain millions of trainable parameters. Their performance depends heavily on data diversity and volume, and they tend to overfit when trained on smaller or noisier datasets. For this reason, we begin with a lightweight baseline CNN model for comparison.

The model input has shape $200 \times 50 \times 3$, corresponding to the RGB spectrogram slices. The architecture consists of three convolutional layers with max pooling, batch normalization, and ReLU activation. This is followed by global average pooling and two dense layers, with a final softmax output for the three classes.

The detailed architecture is shown in Figure 4.9 and summarized below:

- **Input:** RGB spectrogram slice of size $200 \times 50 \times 3$
- **Conv2D (32 filters)** $\rightarrow$ MaxPooling
- **Conv2D (64 filters)** $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling
- **Conv2D (64 filters)** $\rightarrow$ BatchNorm $\rightarrow$ MaxPooling
- **GlobalAveragePooling**
- **Dense (64 units)** $\rightarrow$ Dropout

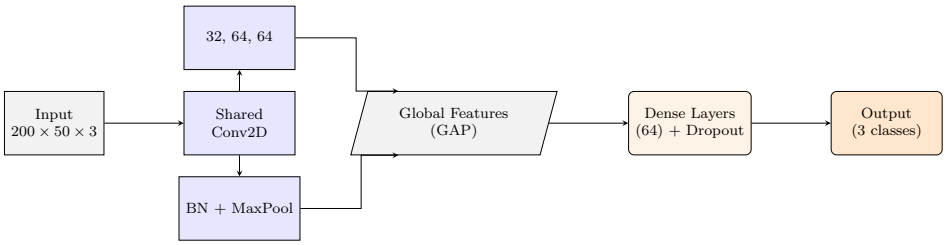


Figure 4.9: Baseline CNN architecture used for spectrogram classification. The model applies three convolutional blocks followed by global average pooling and dense layers.

- **Dense (3 units)** → Softmax

The total number of trainable parameters in this model is 61,189.

This architecture was chosen as a lightweight baseline to serve as a point of comparison for more complex models. It is intentionally shallow, with a limited number of layers and parameters, making it suitable for small datasets where overfitting is a concern. The use of global average pooling reduces the number of parameters while still preserving important features.

Hyperparameters such as the number of filters, kernel sizes, and dense layer dimensions were selected based on initial testing and adjusted empirically through validation performance. We used early stopping and learning rate scheduling to improve training stability and avoid overfitting.

4.4.2 Model 2 – Residual CNN

Residual networks (ResNets) were introduced by He et al. [22] to enable the training of very deep neural networks by addressing the vanishing gradient problem. Instead of learning a direct mapping $H(x)$, ResNets learn a residual function $F(x) = H(x) - x$, reformulated as $H(x) = F(x) + x$. This is achieved through the use of *skip connections*, which bypass one or more layers by directly adding the input x to the output of the stacked nonlinear transformations. Figure 4.10 contrasts a standard CNN block with a residual block that adds the input to the output via a skip connection.

In our case, the goal was not to build a deep model, but to test whether adding skip connections would help preserve features across layers and improve generalization. Since we are working with a small dataset, deeper models are likely to overfit, but shallow networks may struggle to learn effectively. A lightweight residual CNN offered a compromise between the two.

The architecture includes three residual blocks, each composed of two convolutional layers with batch normalization. After each block, max pooling reduces the spatial dimensions. The final part of the network includes global average pooling, a dense layer, dropout, and a softmax output for classification.

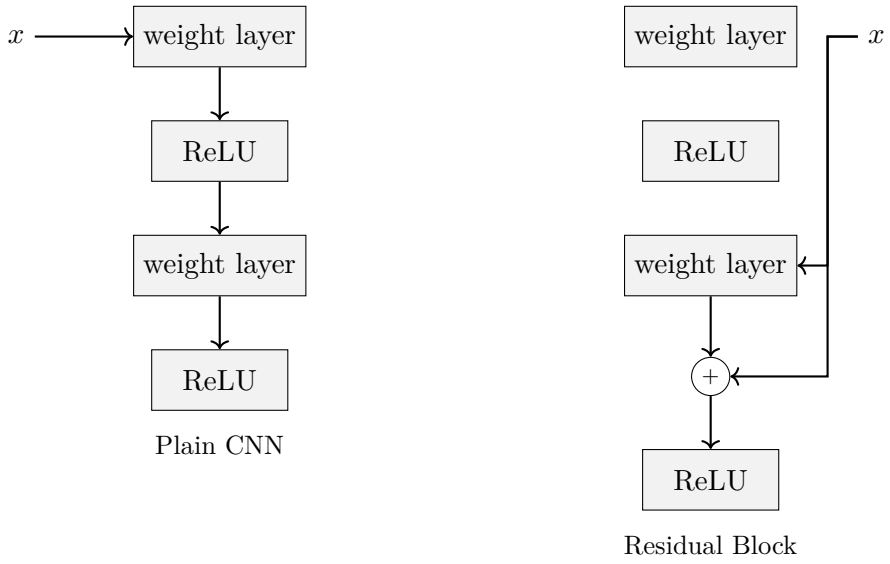


Figure 4.10: Comparison of a plain CNN block (left) and a residual block (right). The residual block includes a skip connection that adds the input x to the output of the stacked layers.

- **Input:** $200 \times 50 \times 3$ RGB spectrogram slice
- **Conv2D (32)** $\rightarrow$ MaxPooling
- **Residual Block 1:** Conv2D $\rightarrow$ BN $\rightarrow$ Conv2D $\rightarrow$ BN $\rightarrow$ Add $\rightarrow$ ReLU
- **Conv2D (64)** $\rightarrow$ MaxPooling
- **Residual Block 2:** same structure (64 filters)
- **Conv2D (64)** $\rightarrow$ MaxPooling
- **Residual Block 3:** same structure (64 filters)
- **GlobalAveragePooling**
- **Dense (64)** $\rightarrow$ Dropout
- **Dense (3)** $\rightarrow$ Softmax

The model was tuned by adjusting filter sizes, block structure, and dense layer dimensions while keeping the total number of parameters close to the baseline CNN. We used early stopping and a learning rate scheduler to control overfitting. Most tuning was done manually by observing validation performance and trying one change at a time.

This model has 228,165 trainable parameters.

4.4.3 Model 3 - CNN with Attention

Convolutional neural networks enhanced with attention mechanisms aim to improve feature selection by assigning varying importance to different regions of an input. These mechanisms help guide the network to focus on more relevant features, particularly useful when working with limited data.

Attention can be applied in different forms—most commonly as spatial attention (which highlights important spatial locations) and channel attention (which emphasizes relevant feature maps). In this work, we implemented a CBAM (Convolutional Block Attention Module) variant with channel and spatial attention applied sequentially. This choice was motivated by prior studies showing improved performance in image classification tasks without a large increase in model size [15].

The CBAM block first applies channel attention. It uses global average and max pooling, each followed by shared dense layers to produce a channel attention map. This map is used to modulate the input via element-wise multiplication. The output is then passed to the spatial attention sub-module, which computes average and max projections across the channel dimension, concatenates them, and applies a 7×7 convolution to produce a spatial attention map. This map is again applied via element-wise multiplication.

We placed the CBAM block after the second convolutional stage in the network, where intermediate features are rich enough for attention to have a meaningful effect. This position was selected after empirical testing of different placements. The full architecture is shown in Figure 4.11.

This model contains approximately 102,894 trainable parameters.

4.5 Evaluation

4.5.1 Evaluation Metrics

To evaluate the effectiveness of classification models, several metrics are commonly employed, including precision, recall and the F1-score. These metrics help in understanding how well the model performs for each individual class and can also be averaged across all classes to assess the model's overall performance. Additionally, accuracy is computed to measure the proportion of correct predictions out of the total number of samples.

Precision indicates the proportion of predicted positive cases that are actually positive. It reflects how reliable the model's positive predictions are. It is defined as:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{4.1}$$

Recall (also known as sensitivity) measures the proportion of actual positive cases

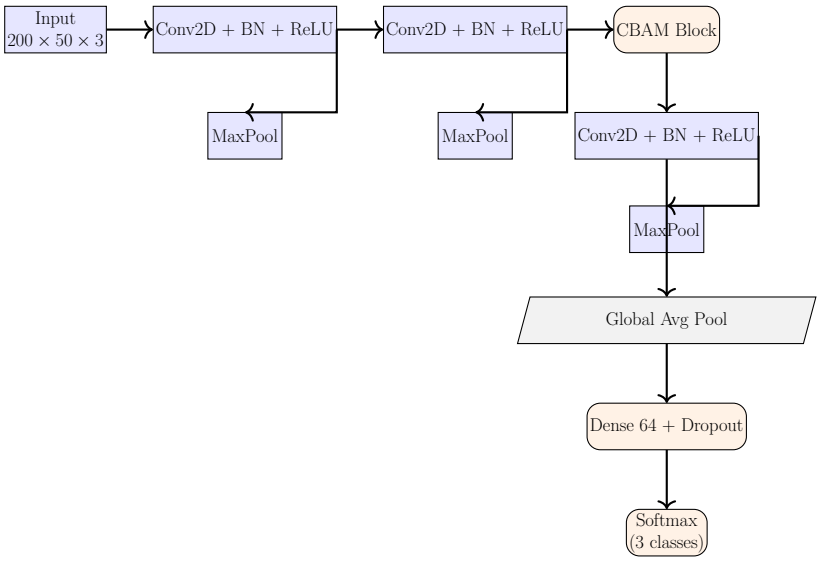


Figure 4.11: CNN with Attention: Each convolutional stage is followed by MaxPooling. The CBAM module is inserted after the second convolutional stage.

that are correctly identified by the model. It shows how effectively the model captures positive instances. The recall formula is:

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{4.2}$$

The **F1-score** is a single metric that combines precision and recall using their harmonic mean. This is particularly useful for balancing the trade-off between precision and recall, especially when the class distribution is imbalanced. It is calculated as:

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.3}$$

In this thesis, although some tasks may benefit from prioritizing either precision or recall, the F1-score and accuracy are primarily used for comparing model performance across different classes and scenarios.

Chapter 5

Results

5.1 Classification per Window

Each RGB spectrogram slice used for classification was generated using a sliding window approach. The full-length Doppler spectrogram of a sample was divided into shorter segments of size 200×50, with a step size of 5 frames. Each segment was treated as an independent input to the model. This method allowed us to capture variations within a motion sequence, especially in cases where gait patterns temporarily changed during walking or when subjects turned.

By using a sliding window, we increased the number of training samples and captured the temporal evolution of the motion. This was especially important given the relatively small size of our dataset. Instead of relying on entire sequences, we used compact windows to represent local behavior. Each window could be associated with steady-state motion, making it easier for the model to learn consistent patterns.

The choice of window size and stride was based on trial and error. A width of 50 frames captured roughly 5 seconds of motion per slice. The overlap ensured that each new sample did not start from a completely unrelated point in time, which helped maintain coherence between slices and improved classification consistency.

5.2 Higher Resolution

Our spectrograms were generated using higher Doppler resolution than in typical low-resolution FMCW radar setups. This was made possible by the IWR6843AOP radar, which allowed for 228 chirps per frame per transmit antenna. As a result, each frame provided a Doppler axis of 512 bins. This higher Doppler resolution gave a finer velocity representation and made it possible to isolate subtle differences between walking motions.

This level of detail was particularly important for distinguishing between nor-

mal walking, walking while carrying, and walking with hands in pockets. These classes often differ more in where motion is concentrated than in overall speed. For instance, carry samples often showed higher Doppler energy in faster-moving regions—commonly associated with leg movement—while pocket samples tended to exhibit reduced motion in slower-moving regions, suggesting restricted arm swing. A lower-resolution spectrogram would have made these differences harder to detect.

To improve input consistency, we used amplitude spectrograms instead of raw phase or complex data, as they were found to be more stable for this classification task. Each spectrogram was divided along the Doppler (vertical) axis into three equal segments. These segments were used to heuristically represent motion at different velocity bands, which we interpreted as loosely corresponding to lower-, middle-, and upper-body movement based on typical human gait dynamics [23]. This segmentation was then encoded into RGB format, with each channel representing one velocity band.

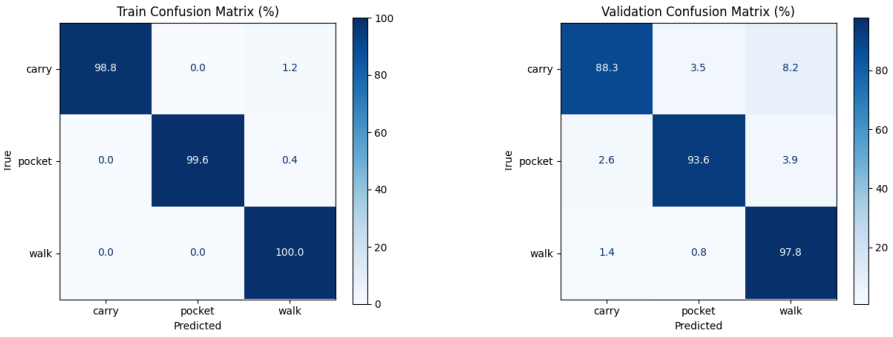
Although this method does not directly localize body parts in space, it provides a structured representation of how motion is distributed across velocity ranges. Combined with the high resolution, this velocity-band segmentation allowed the classifier to capture class-specific patterns even with a relatively shallow network. Without this preprocessing step, more overlap between classes would have remained in the input data, making classification more difficult.

5.3 Baseline CNN

This model serves as the baseline for evaluating Doppler spectrogram classification using convolutional neural networks. As described in Chapter 4, CNNs in radar-based motion classification are typically deep and parameter-heavy, often relying on hierarchical feature learning to resolve subtle motion differences. However, in this project, the total training data available consisted of 6261 samples. The validation and test sets each contained around 1,200 samples. This constrained size led us to adopt a lightweight model to avoid overfitting while still extracting useful temporal-velocity features from spectrogram slices.

The baseline architecture consists of three convolutional layers, each followed by max pooling. This is followed by a global average pooling layer, a dense layer with 64 units and a softmax output layer for the three gait classes. The model had a total of 61,189 parameters and was trained using the Adam optimizer [24] with the following settings (see Figure 4.9 for an overview of the model structure):

- Learning rate: 2.5×10^{-5}
- $\beta_1 = 0.9$: exponential decay rate for the first moment estimate (mean of gradients)
- $\beta_2 = 0.999$: exponential decay rate for the second moment estimate (uncentered variance of gradients)



(a) Training (b) Validation

Figure 5.1: Confusion matrices for training and validation sets using the baseline CNN model.

- $\epsilon = 1 \times 10^{-7}$: small constant added for numerical stability to avoid division by zero

The input to the model was a Doppler spectrogram slice of shape $200 \times 50 \times 3$, with each channel corresponding to a segmented Doppler region: upper, middle and lower. These were obtained by slicing the Doppler range dimension, as detailed in Section 5.1. Before training, we applied an adaptive thresholding strategy to remove low-SNR and transitional frames. This step was based on SNR statistics and Doppler bin distributions, which showed that clutter and transitional motion (e.g., turning, starting, stopping) could dominate radar returns and degrade training quality. As a result, the usable data was reduced, but more consistent.

5.3.1 Training and Validation

The model achieved 99.44% training accuracy and 92.98% validation accuracy. The training confusion matrix (Figure 5.1) showed near-perfect classification across all classes. Validation performance was similarly high, with walk achieving the highest recall (97.8%) and carry showing the most confusion, mostly with walk.

5.3.2 Test Evaluation

The test accuracy was 60.24%, highlighting a substantial drop in generalization. As shown in Figure 5.2, carry was frequently confused with walk or pocket. Walk maintained the strongest performance, consistent with its distinct Doppler motion profile.

The classification performance of the baseline CNN on the test set is summarized in Table 5.1, showing precision, recall, and F1-score for each class.

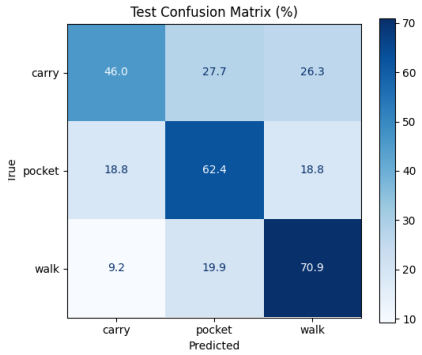


Figure 5.2: Test confusion matrix (Baseline CNN).

Table 5.1: Classification performance on the test set (Baseline CNN).

Class	Precision	Recall	F1-score
Carry	0.61	0.46	0.52
Pocket	0.56	0.62	0.59
Walk	0.64	0.71	0.67

5.3.3 Discussion

Although the model is shallow by design, it effectively learned to separate the gait classes in cleaned data. The performance drop on the test set indicates sensitivity to intra-class variability and unseen motion styles. This is particularly evident in the carry class, where motion in the lower Doppler bands—often associated with slower-moving regions such as the torso or arms—may resemble other categories depending on how the object is held [23].

Overall, the model provides a stable baseline and a reference point for comparing more complex models such as the residual and attention-based CNNs, discussed in the next sections.

5.3.4 Residual CNN

To complement the baseline CNN, we implemented a deeper residual architecture with skip connections across convolutional layers. The model is inspired by the principles of ResNet [22], where identity mappings are introduced to preserve gradient flow and enable training of deeper networks. Our implementation is lighter and tailored for the radar spectrogram slices used in this work (input shape: $200 \times 50 \times 3$), keeping the total number of parameters to 102,894.

The architecture consists of three main residual blocks, each made up of two

Class	Precision	Recall	F1-score	Support
Carry	1.00	1.00	1.00	1729
Pocket	1.00	1.00	1.00	1751
Walk	1.00	1.00	1.00	1528
Overall	**1.00**	**1.00**	**1.00**	**5008**

Table 5.2: Residual CNN training classification report.

Class	Precision	Recall	F1-score	Support
Carry	0.92	0.93	0.93	427
Pocket	0.96	0.94	0.95	467
Walk	0.95	0.96	0.96	359
Overall	**0.94**	**0.94**	**0.94**	**1253**

Table 5.3: Residual CNN validation classification report.

convolutional layers (3×3 kernel), followed by batch normalization and ReLU activation. A skip connection is added between the input and output of each block, followed by an additional activation. Max pooling is applied after each residual unit to downsample the feature maps. The network ends with a global average pooling layer, a dense layer with 64 units, dropout (rate 0.3), and a softmax layer for three-class classification.

Training was done using the Adam optimizer with a learning rate of 1×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. The loss function was categorical crossentropy. The model was trained on the full training set without augmentation or additional regularization.

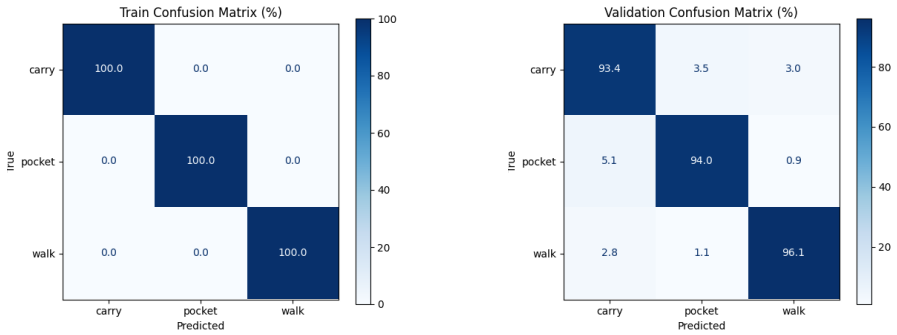
The training, validation, and test performances are given in Tables 5.2 - 5.4.

The model achieves perfect accuracy on the training set, suggesting overfitting. Validation accuracy remains high at 94.4%, indicating that the model generalizes well to unseen data from the same distribution. However, the test accuracy drops sharply to 50.0%, with the largest performance drop occurring for the *carry* and *pocket* classes. This confirms the model struggles with generalization when exposed to distribution shifts between training/validation and test data—a common issue in small or imbalanced datasets [25].

Confusion matrices for all three splits are shown in Figures 5.3, and 5.4.

Class	Precision	Recall	F1-score	Support
Carry	0.44	0.43	0.43	376
Pocket	0.42	0.54	0.47	388
Walk	0.70	0.53	0.60	423
Overall	**0.50**	**0.50**	**0.50**	**1187**

Table 5.4: Residual CNN test classification report.



(a) Training (b) Validation

Figure 5.3: Confusion matrices for training and validation data using the Residual CNN model.

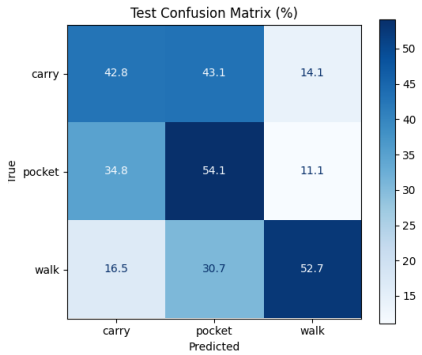


Figure 5.4: Confusion matrix for test data.

5.3.5 CNN with Attention

The CNN with attention model combines a lightweight convolutional architecture with a channel modulation mechanism, implemented via a dense attention mask. This mechanism is inspired by the Convolutional Block Attention Module (CBAM) [15], though only the channel attention branch was applied in our case. The motivation is to allow the model to selectively emphasize informative channels in the feature map before final classification, without introducing excessive computational complexity.

The network consists of three convolutional blocks with increasing filter sizes: 32, 64, and 128. Each block includes a 3×3 convolution, batch normalization, ReLU activation, and max pooling. After the second block, a CBAM (Convolutional Block Attention Module) is applied to enhance feature representation. For channel attention, both global average and max pooling are used to extract a descriptor from the feature map, which is passed through shared dense layers and reshaped to modulate the original feature map. The modulated output is then processed by the spatial attention module, which concatenates average and max projections across the channel dimension and applies a 7×7 convolution to generate a spatial attention map. The attention-weighted features are passed through the third convolutional block, followed by global average pooling, a dense layer with 64 units and dropout, and finally a softmax layer for three-class classification.

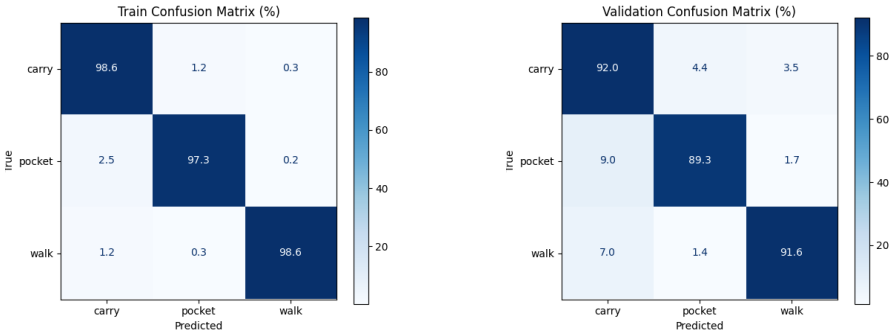
Model Summary:

- **Input shape:** (200, 50, 3)
- **Total parameters:** 115,285
- **Trainable parameters:** 114,835
- **Non-trainable parameters:** 448
- **Optimizer:** Adam
- **Learning rate:** 0.001
- **Beta values:** $\beta_1 = 0.9$, $\beta_2 = 0.999$
- **Epsilon:** 1×10^{-7}
- **Dropout rate:** 0.5

Performance Analysis:

On the training set (5008 samples), the model achieved an accuracy of 0.9996, with all three classes being classified almost perfectly. Precision, recall, and F1-score were all above 0.99 for each class, suggesting near-perfect fitting of the training data.

On the validation set (1253 samples), the model maintained strong performance, reaching an accuracy of 0.9920. The *pocket* class had perfect recall (1.00) but



(a) Training (b) Validation

Figure 5.5: Confusion matrices for the CNN with attention model on the training and validation sets.

Table 5.5: Test accuracy comparison of all three models.

Model	Test Accuracy	Params
Baseline CNN	60%	61,189
Residual CNN	50%	216,099
CNN + Attention	54%	115,285

slightly lower precision (0.98), while the *carry* and *walk* classes achieved high scores across all metrics.

However, on the unseen test set (1187 samples), overall accuracy dropped sharply to 0.5435. This discrepancy highlights significant overfitting, despite the inclusion of dropout and a modest model size.

Compared to the baseline CNN and residual CNN, it was the only model able to somewhat separate the *carry* and *pocket* classes, which were otherwise frequently confused. The residual CNN showed more stable test performance overall, likely due to skip connections that preserve transferable mid-level features. These findings suggest that while attention mechanisms improve feature selectivity, they may also increase the risk of memorization if not supported by stronger regularization or data augmentation [15, 22].

The confusion matrices are shown in Figures 5.5 and 5.6

Comparison with other models is shown in Table 5.5.

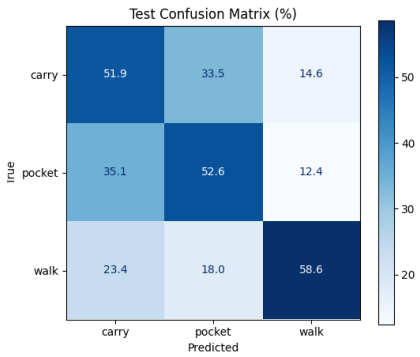


Figure 5.6: Confusion matrix for the CNN with attention model on the unseen test set.

Chapter 6

Discussion

6.1 Comparison with Prior Work

Several studies have investigated radar-based human gait classification using micro-Doppler signatures, often using Doppler spectrograms as the primary input representation. A widely cited early study by Tivive, Bouzerdoum, and Amin [1] proposed a pipeline based on handcrafted features extracted from Doppler spectrograms, followed by classification using algorithms like k-NN and SVM. While their results were reasonable, the method relied heavily on manual preprocessing such as noise filtering and dimensionality reduction. In contrast, our approach uses convolutional neural networks that learn directly from RGB spectrogram slices, removing the need for manual feature design.

Papanastasiou [2] explored the use of 1D CNNs on micro-Doppler time-series data for gait identification. While it demonstrated the viability of deep learning for radar-based tasks, it did not leverage time-frequency representations or exploit spatial structure. By using 2D RGB spectrograms, our method captures both temporal and frequency patterns, and incorporates spatial cues through body-region slicing. This allows for better discrimination of subtle gait differences.

Kim and Moon [3] applied CNNs to classify broad activities such as walking and running based on micro-Doppler data. Although effective, their task was coarser than ours. We focus on more nuanced gait differences, such as walking with hands in pockets or while carrying, which requires higher sensitivity to subtle variations. Our results suggest that even simple CNN architectures can perform well on such tasks, especially when combined with targeted preprocessing.

Lang et al. [18] used CNNs on grayscale spectrograms and showed promising results on small datasets. However, they did not include motion-state filtering or body-region separation. In our work, adaptive thresholding helps isolate steady-state walking, and the RGB mapping reflects motion in different body regions, which contributes to better localization of class-relevant features.

Mboyi, Oh, and Han [26] focused on clutter mitigation and transfer learning for

radar-based people counting. Although we do not apply explicit background suppression, we use adaptive thresholding and exclude transitional frames, which serves a similar purpose by reducing signal clutter and improving input quality.

Other studies such as Han and Bhanu [4], Hofmann et al. [5], Zhang et al. [6], and Gafurov et al. [7] have explored gait recognition using video, accelerometers, or other sensors. While these modalities are effective, they often require direct visibility or physical attachment. Our radar-based approach offers a contactless, privacy-preserving alternative suitable for constrained indoor environments.

Traditional micro-Doppler studies by Harmanny, De Wit, and Cabic [10] focused on handcrafted signal analysis. Our work moves beyond that by allowing the model to learn features directly from the raw spectrograms. Additionally, studies addressing clutter removal [27–30] highlight the importance of cleaning radar data. Our use of motion-state filtering, although simpler, contributes to a similar effect.

Lightweight and real-time models have also been explored in prior work [14, 20, 21]. We further extend these ideas with our Residual CNN and CBAM-inspired attention model [15,22], applying them to our structured RGB spectrogram inputs. Despite limited data, our models perform well on validation sets, although the test performance shows some overfitting, consistent with the challenges outlined by Zhang et al. [25].

Overall, our work combines several useful practices from existing literature—such as time-frequency representation, deep learning, and adaptive preprocessing—into a focused, reproducible pipeline for fine-grained gait classification using FMCW radar.

6.2 Limitations

Although the proposed models show strong performance on training and validation sets, there are notable limitations. Most significantly, test set accuracy drops considerably compared to validation accuracy, particularly for the attention-based model. This suggests potential overfitting due to the limited size and variability of the dataset. While motion-state segmentation and adaptive thresholding improve data quality, they may also discard ambiguous but informative frames that could aid generalization.

The dataset is also inherently constrained by the experimental environment. Subjects performed gait actions within a fixed room layout and radar positioning, which may limit the diversity of Doppler signatures. This restriction likely impacts the model's ability to generalize to new scenes or unseen individuals. Future work could explore data augmentation or domain adaptation strategies to mitigate this.

In addition, we employed lightweight CNNs, this also limits the depth and capacity of the models. A deeper network or pretrained backbone may capture more com-

plex spatiotemporal features, but would require larger datasets and more robust regularization strategies to avoid overfitting.

6.2.1 Lack of Data for CNN

One of the biggest challenges for this thesis is having a rather small quantity of self-collected data. CNN's effectiveness is highly dependent on the availability of large and diverse datasets. This requirement is rooted in several core principles:

First, CNN architectures inherently possess a large number of parameters distributed across multiple convolutional and fully connected layers. The learning of these parameters is data-driven, demanding significant volumes of quality training data to accurately capture the underlying patterns without overfitting to noise. Inadequate data leads to poor generalization, where the model performs well on training samples but fails on unseen data.

Secondly, the hierarchical structure of CNN allows for the automatic extraction of features—ranging from low-level edges and textures to high-level object parts. This multi-layered learning process benefits from diverse data to comprehensively understand different variations and nuances present in the input data. As CNN attempts to learn invariant features, the necessity for large datasets becomes evident to prevent the model from misrepresenting unseen variations.

6.2.2 Sliding Window Length

The sliding window length is the horizontal length of a spectrogram, namely, the length of a recording snippet. We have also trained and tested on different sliding window sizes. First, we tried using a smaller window, training the model with a 20-frame (2-second) snippet and a stride of 5 frames, then decided to train with a slightly bigger window, 50-frame (5-second) since we noticed that using a shorter snippet, the test accuracy for each class decreased by about 20%.

Suggesting that training based on a larger sliding window yields to better performance in testing, which is intuitive because it will contain more gait information than a shorter recording snippet would have. However, this does not necessarily mean that the longer window will always yield higher test accuracy, and the optimal length of the sliding window for this specific classification task should be further studied.

Note that, if one wishes to deploy the model on a device that does classification in a real-time system manner, one should avoid using long sliding windows due to the fact that this will result in longer time delay when performing classification, which defeats the purpose of real-time. This could be conquered by, again, having a lot more training data so that it allows the sliding window to be smaller, which at the same time does not affect the performance much.

Figure 6.1: Images of a person walking in a spacious court. The individual was walking away from the radar.

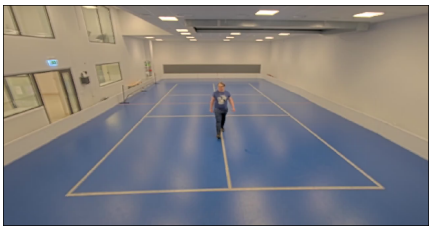


Figure 6.2: Frames dropped, which created a delay in time between the actual frame. When it started to catch up with the next frame, the individual has already started walking towards the radar.

6.2.3 Frame Drop

Another issue that we faced was frame drop during the recording sessions. Frame drops can occur when a system is unable to process fast enough, this might happen for several reasons: Overload the CPU with too many background processes running; limited CPU or GPU resources.

Since we are using a laptop instead of a personal computer with more resources while recording gait data, frame drops are inevitable. This will also cut down on the amount of some data that contains certain information from the classes we aim to classify.

For example, in Fig6.1, 6.2 show two consecutive frames. In Fig6.1, the individual was walking away from the radar and will only turn and walk back when he reaches the center of the court; thus the next frame, the individual should still be facing away from the radar. However, the system froze at this moment, resulting to a frame drop. Lost some gait radar data, shown in Fig6.2, as the person had already started walking towards the radar in the next frame.

6.2.4 Background Clutter, Thresholding and Data Volume

During data inspection, several spectrograms revealed persistent background clutter, often appearing as low-level horizontal lines across Doppler bins (see Figure 6.3). These artifacts, while stationary or near-stationary in Doppler space, are not associated with the classes of interest. Rather, they result from static or reflective objects in the room such as walls, furniture, or the radar housing itself, a common issue in indoor radar deployments [26, 29].

To mitigate this, we employed an adaptive thresholding technique based on the distribution of dominant Doppler bins across frames. This helped remove not only the background clutter but also frames that represented ambiguous or transitional

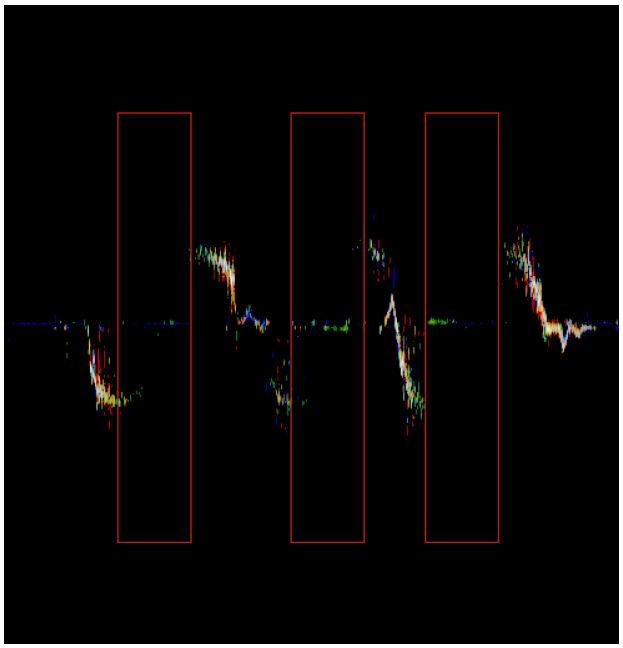


Figure 6.3: Sample spectrogram of jogging showing horizontal bands of background clutter highlighted by red boxes.

motion states, such as acceleration, turning, or minor hand gestures unrelated to the defined gait classes. An example of a spectrogram containing only such filtered-out frames is shown in Figure 6.4. This spectrogram corresponds to a single subject recording, highlighting how nearly half the recorded frames were discarded after thresholding.

While effective in increasing the purity of training data, this process significantly reduced the total dataset size. Frame statistics across recordings showed that for many sessions, less than 55% of the original frames were retained as clean steady-state walking. This drop directly impacts the training volume, especially for models requiring large sample sizes to generalize well. It also illustrates the trade-off between data quality and quantity in real-world radar classification tasks.

Several studies have noted that clutter mitigation improves classification accuracy but often at the expense of reduced signal availability [27, 28, 30]. Our work supports this, demonstrating that clutter suppression is not merely a preprocessing choice but a design constraint that affects the learning dynamics and must be balanced accordingly.

6.2.5 Lower Range-Resolution

In this study, we did not explore the effect of lower range-resolution in a systematic way. Our pipeline was based on sectioning the range axis into three horizontal

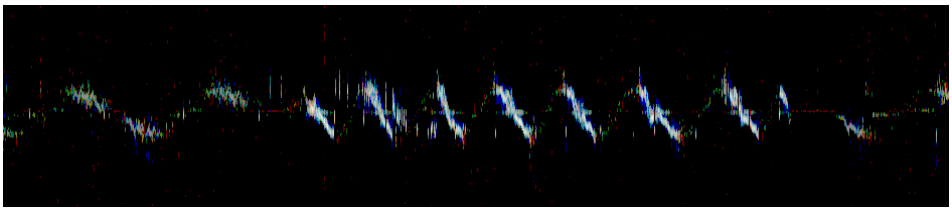


Figure 6.4: Filtered spectrogram segment containing only clutter and transitional motion frames, excluded from final dataset.

regions, which were then mapped onto the RGB channels. A natural baseline to compare against would have been to collapse the range dimension entirely by averaging over all range bins, producing a single-channel spectrogram. This could help assess whether the additional structure introduced by slicing is actually useful for classification. While this remains a possible direction for future work, we can only speculate on the impact such a simplification would have had. It might have reduced spatial redundancy, but also risked losing discriminative features related to body-region-specific motion [3].

Chapter 7

Conclusion

In this thesis, a CNN-based classification pipeline was proposed for human gait on radar spectrogram slices. It is capable of both finding the individual moving from the range-Doppler image in various environments and further classify their gait into three classes: walking, walking while carrying an object and walking with hands in pocket. The model was then evaluated based on an unseen outdoor environment, suggesting that the model was trained properly and generalized well with good performance.

The proposed model is able to classify the two categories: walking and walking with hands in pocket adequately. However, for class walking while carrying an object seems to have lower accuracy. This could be due to the confusion between the two other classes. We believe there are three potential reasons combined why this happened. First, with limited number of spectrogram slices, the network failed to learn generalizable features and had a hard time distinguishing classes with subtle arm movements. Second, class walking while carrying simply does not have enough or any robust features for the model to perform classification task satisfactorily. Lastly, the slices from the carrying class highly resembles the other two classes thus the higher misclassifications rate.

If provided with a larger and more diverse dataset, we believe the CNN model will improve significantly and become more than reliable in classifying various human gaits. With this, it can be deployed in security surveillance and biomedical domains where tracking a person's gait is necessary without violating one's privacy or collecting one's personal information and data.

The results of this work indicate that radar-based human gait classification is not only feasible, but also promising under realistic conditions. The system had trouble with generalization, as test accuracy was much lower. This suggests that the model is sensitive to changes in the environment and that the dataset was likely too limited. While challenges remain—especially in distinguishing visually similar motion types—the pipeline developed in this thesis provides a strong foundation for future improvements. Further work could explore larger datasets, more robust feature extraction methods, and real-time deployment, ultimately contributing to

privacy-respecting sensing solutions in real-world settings.

Bibliography

[1] F. H. C. Tivive, A. Bouzerdoum, and M. G. Amin, "A human gait classification method based on radar Doppler spectrograms," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–12, 2010.

[2] V. S. Papanastasiou, "Deep learning-based identification of human gait by radar micro-Doppler measurements," Master's thesis, Delft University of Technology, 2019.

[3] Y. Kim and T. Moon, "Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, 2016.

[4] J.-Y. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[5] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, "Gait recognition in the presence of occlusion: A new dataset and baseline results," in *2011 19th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, IEEE, 2011.

[6] C. Zhang, L. Wang, and Y. Wang, "A comprehensive study on gait-based gender classification with deep convolutional neural networks," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3269–3284, 2019.

[7] D. Gafurov, E. Snekkenes, and P. Bours, "Gait authentication and identification using wearable accelerometer sensors," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 51–62, 2007.

[8] B. Gokaraju, S. Ghosh, and A. Hossain, "Human and bird detection and classification based on Doppler radar spectrograms and vision images," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, 2021.

[9] V. C. Chen, F. Li, S. S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, 2006.

[10] R. Harmanny, J. De Wit, and G. P. Cabic, "Radar micro-Doppler feature extraction using the spectrogram and the cepstrogram," in *11th European Radar Conference*, pp. 165–168, 2014.

[11] M. I. Skolnik, *Radar Handbook*. McGraw-Hill, 2008.

[12] M. A. Richards, *Fundamentals of Radar Signal Processing*. McGraw-Hill Education, 2014.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[14] D. Park, S. Lee, S. Park, and N. Kwak, "Resnet-sp: A lightweight model for real-time radar spectrogram classification," *Sensors*, vol. 21, no. 1, p. 210, 2021.

[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, Springer, 2018.

[16] H. D. Griffiths, *Principles of Radar and Sonar Signal Processing*. Institution of Engineering and Technology, 2017.

[17] Texas Instruments, "Iwr6843aop: Single-chip 60-ghz mmwave sensor datasheet." Datasheet, 2022. Available at `https://www.ti.com/lit/ds/symlink/iwr6843aop.pdf`.

[18] Y. Lang, C. Hou, Y. Yang, D. Huang, and Y. He, "Convolutional neural network for human micro-Doppler classification," in *Proc. Eur. Microw. Conf.*, pp. 1–4, 2017.

[19] T. Jordan, "Using convolutional neural networks for human activity classification on micro-Doppler radar spectrograms," p. 982509, 05 2016.

[20] D. Park, S. Lee, S. Park, and N. Kwak, "Radar-spectrogram-based UAV classification using convolutional neural networks," *Sensors*, vol. 21, no. 1, p. 210, 2021.

[21] S. Rahman and D. A. Robertson, "Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images," *IET Radar, Sonar & Navigation*, vol. 14, no. 4, pp. 653–661, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[23] V. C. Chen, F. Li, and S. Ho, "Micro-Doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, 2006.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[25] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *5th International Conference on Learning Representations (ICLR)*, 2017.

[26] M. M. G. Yowel, D.-H. Oh, and J.-H. Han, "Hybrid DCNN–transfer learning model coupled with background clutter mitigation for FMCW radar-based people counting improvement," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–12, 2025.

[27] J. Yin, C. Unal, M. Schleiss, and H. Russchenberg, "Radar target and moving clutter separation based on the low-rank matrix optimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4765–4778, 2018.

[28] J. Yin, M. Schleiss, and X. Wang, "Clutter-contaminated signal recovery in spectral domain for polarimetric weather radar," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, 2022.

[29] C. Liu, J. Dong, P. Lang, M. Li, X. Fu, and X. Qi, "Ground background clutter recognition based on fully convolutional neural network," in *2021 CIE International Conference on Radar (Radar)*, pp. 1850–1855, IEEE, 2021.

[30] Y.-M. Yang, C. G. Lee, J. S. Jao, N. Rodriguez-Alvarez, W. Majid, and K. Oudrhiri, "Investigation of background clutter removal and impacts on cis-lunar target detection and tracking using DSN open-loop tracking measurements," in *2024 IEEE Aerospace Conference*, 2024.

Printed by Tryckeriet i E-huset, Lund 2025

LUND
UNIVERSITY

Series of Master´s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2025-1072
http://www.eit.lth.se