Short Range Gesture Sensing and Classification Using Pulsed Millimeter-Wave Radar and Convolutional Neural Networks

HANNES DAHLBERG
ANTON EVERTSSON

MASTER´S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Short Range Gesture Sensing and Classification Using Pulsed Millimeter-Wave Radar and Convolutional Neural Networks

Hannes Dahlberg
tna14hda@student.lu.se
Anton Evertsson
tna14aev@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisors: Lars Ohlsson Fhager, Sebastian Heunisch

Examiner: Mats Gustafsson

July 11, 2019

Printed in Sweden
Tryckeriet i E-huset, Lund

Acknowledgments

We would like to thank our supervisors Lars and Sebastian for all their help along the way. We greatly appreciate your advice and all the time you have spent helping us in the lab.

Thank you to Adrian, Axel, Cornelia, Erika and Gustaf for helping us acquire additional test data.

Finally, thanks to the Nanoelectronics departement for an interesting and welcoming workplace, and for providing much needed coffee.

Abstract

Human hand gestures present a novel way of interacting with electronic devices. A millimeter-wave radar setup utilizing a pulsed Resonant-Tunneling Diode signal generator in the 60 GHz ISM band is used to measure 12 different hand gestures. The data is used to train and validate Convolutional Neural Networks (CNNs). The measurement setup utilizes a real time sampling oscilloscope and down-mixing of the received radar signal. Three data types are under test: data without processing (Range-Time), Fourier-transformed data (Range-Doppler) and windowed Fourier-transformed data divided into three frames (WFT).

A pre-defined and pre-trained CNN `ResNet50` is initially used for data classification. The validation accuracy for 12 gestures with 180 measurements each and a training-validation quota of 60%-40% was 98% (Range-Time), 85% (Range-Doppler) and 91% (WFT). Additionally, a proposed CNN architecture with less complexity named `SimpleNet` is investigated, showing a validation accuracy (for the same data and training-validation quota as for `ResNet50`) of 95% (Range-Time), 85% (Range-Doppler) and 93% (WFT). For `SimpleNet`, the presented results are the average values of 25 different training sessions.

Additionally, measurements from an independent test group were classified using above trained networks, with results that indicated relatively weak generalization for the classifying networks under test.

Popular Science Summary

Träffsäker geststyrning med höghastighetsradar och maskininlärning

Att styra elektronik med handgester kommer att revolutionera sättet vi interagerar med teknologi. Med en höghastighetsradar kan man tillsammans med maskininlärning avläsa 12 olika hangester med över 95% träffsäkerhet. Detta är en snabb och energisnål metod som fungerar oberoende av ljusförhållanden.

Kontaktlös styrning med handgester kräver en hög noggrannhet om det ska kunna ersätta eller utöka dagens existerande touch-skärmar och knappar. Elektronik som är relevant för geststyrning är till exempel smartphones, smarta klockor och infotainment-system i bilar.

Radar har klassiskt varit stora maskiner som använts för att upptäcka och spåra stora fordon, raketer, och flygplan. Radar idag är annorlunda! Dagens teknik har gjort det möjligt att bygga extremt snabba och noggranna radarsystem inte större än en enkrona, som drivs av bråkdelen av den energi som en radar i ett flygtorn behöver. Idag används sådana typer av radarsystem i bland annat självkörande bilar. Nästa steg kan vara att bygga in en radar i användarelektronik som till exempel smartphones. Här skulle man då med handgester kunna höja volymen, starta en app, ta en bild, eller till och med skriva ett sms. Gestmätningar med radar har fördelen att inget ljus behövs, som för en kamera till exempel. Istället används elektromagnetisk strålning för att känna av handen. Då spelar det ingen roll om det är kolsvart eller om radarsensorn är täckt av smuts heller för den delen, den fungerar ändå.

För att klassificera de olika handgesterna används maskininlärning. Maskininlärning är en typ av artificiell intelligens (AI) där datorn själv hittar mönster och viktig information från radarmätningen, mönster och kopplingar som en människa aldrig hade klarat av att göra. Det maskininlärnings-system som används i denna studie kan särskilja mellan de 12 olika handgesterna med över 95% träffsäkerhet. På hundra gester blir alltså bara max fem stycken feltolkade. Mer avancerad AI kan öka träffsäkerheten, och ett system utvecklat av forskare och ansett vara en av världens bästa dataklassificerare tolkade gesterna rätt till 99%.

Mätningarna från handgesterna tycktes dock vara relativt personliga, alltså var det en utmaning för AI-systemet att generellt tolka blandad data från ett flertal personer. En utveckling för att göra radarmätningar av gester från många olika personer kräver alltså ett mer utvecklad och ”generell” AI-system.

Denna studie använder handgester som passar till användarelektronik, du kan till exempel svepa med handen åt höger eller vänster för att byta sida eller snurra ditt pekfinger i en cirkel för att höja eller sänka volymen i din bilstereo. Allt utan att behöva hitta en knapp i mörkret eller att ens kolla bort från vägen när du kör. Möjligheterna för att implementera olika gester är nästan oändliga och kan anpassas till de behov som användaren har.

I takt med att tekniken utvecklas kommer fler och fler användningsområden att uppenbara sig. Avancerad gestmätning skulle kunna användas till VR-miljöer, medicinska operationer på distans, teckenspråkstolkning, och mycket mer. Framtiden ser i vilket fall ljus ut för geststyrning med radarteknik. Snart är kanske störande fingeravtryck på mobilen ett minne blott!

Nomenclature

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CR	Cross-Range
DR	Down-Range
EM	Electromagnetic
ET	Equivalent Time
FMCW	Frequency-Modulated Continuous-Wave
FoV	Field of View
FT	Fourier Transform
IF	Intermediate Frequency
IQ	In-phase-Quadrature-phase
LO	Local Oscillator
PRF	Pulse Repetition Frequency
PRI	Pulse Repetition Interval
`ResNet50`	Pre-defined and pre-trained CNN
RF	Radio Frequency
RT	Real Time
RTD	Resonant Tunneling Diode
Rx	Receiving antenna
`SimpleNet`	Own, less complex CNN architecture
Tx	Transmitting antenna
WFT	Windowed Fourier Transform
WG	Wavelet Generator

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Radio Detection and Ranging (RADAR) have historically been used as means of object detection in large-scale applications, with technological development initially driven by the military sector [1]. The operating wavelength for early radar systems was typically in the order of MHz, and had a large implementation size and high energy demands. But as technology matured and electrical components decreased in size, as well as the move towards solid-state integration, radars have started operating with higher frequencies with wavelengths in the order of millimeters [2]. The term "millimeter-wave radar" includes radars operational at 30-300 GHz. Today, radars together with sophisticated computing are used in many applications such as astronomy, weather detection, radar imaging, object tracking, sensors, and hand-held instruments for measuring vehicle speed [2, 3, 4].

A big driving factor behind millimeter-wave radar today, aside from military applications [2], is the automotive industry, with a key use in autonomous vehicles [5]. Here there exists a need for radars with high accuracy sensing both outside and inside of the vehicle. This industrial need is contributing to the interest and development of short-range radar detection. While it is mainly the automotive industry that drives the development, novel applications are investigated along the way as the technology becomes more available. A key benefit with radar is its independence of light (as for optical systems) and sound (as for audio or voice-detecting systems) [6], it only relies on reflected electromagnetic (EM) waves.

As the usage of operating frequency and bandwidth is heavily controlled around the world, it is attractive for researchers to investigate frequencies which have less convention. A so-called unlicensed band presents the opportunity for novel standards and operations. One such band is the 60 GHz ISM (Industrial, Scientific and Medical) band, with a center frequency of 60 GHz and an open 7 GHz bandwidth [7]. With no strict demands and a relatively wide bandwidth, this frequency band is of interest to both the industry and the scientific community. The 60 GHz ISM band also has limited propagation range compared to lower frequencies due to a higher EM attenuation in the atmosphere [7], which allows for re-usage of the spectrum.

As modern electronic devices develop both in their performance and applications, a need for novel ways of interaction is presented. Especially contactless or touchless interactions, which enables more convenient control of smaller devices or devices out of reach. An example of an intuitive way of touchless interactions with a device would be by using hand gestures. This could prove to be a future way of handling upcoming types of consumer electronics [8, 6, 9].

The demands of touchless interactions lies both in reliability, power consumption and device size [8]. The need for low power and low cost implementations is crucial when it comes to battery driven consumer devices. Implementations relying on touchless interactions needs to meet the standards of today's market, thus creating a demand for research in the field.

In contrast to for example mechanical interactions such as pressing a button, where the intention of an input is clear, complex inputs with many variables need capable methods for providing a correct output. This would be the case for gesture recognition, as the measurement of a single gesture could contain large amounts of data, for example in the form of finger position and velocity. Classification of information is a large and active field of research, and as computational speed becomes more available the use of data recognition methods such as Machine Learning and Deep Learning becomes more feasible [10]. Methods such as Convolutional Neural Networks (CNN) are extensively used for image classification, showing an unprecedented performance in its ability to recognize images belonging to hundreds or thousands of categories, or classifying hand-written text [10]. The field of Deep Learning is in no way saturated and novel applications are investigated all around the world.

The goal of this thesis is to evaluate millimeter-wave radars for touchless interactions, specifically hand gesture sensing, and how well the gestures can be classified. This is a field of interest for future modern electronic applications. Current research of gesture classification using millimeter-wave radar systems relies on frequency-modulated continuous-wave (FMCW) radar to a large extent [6, 8, 11, 12]. Although FMCW implementations perform well in terms of resolution and detection, the power consumption is relatively high due to continuous EM emission compared to pulsed emission. As FMCW provides convenient Doppler processing, most research is directed towards utilizing and classifying Doppler processed data.

This thesis will define a radar setup and use it to measure human hand gestures with varying complexity. The radar setup utilizes an in-house developed and manufactured resonant tunneling diode (RTD), which acts as a pulsed millimeter-wave signal generator. It displays a large bandwidth due to pulse lengths in the picosecond scale [13]. The data will be classified using different CNNs. A network with an existing architecture found in literature, which is pre-trained on various image types, will be trained with supplied radar data using transfer learning. Additionally, a network of own design and less complexity is presented in this thesis, only relying on the supplied radar data and no pre-training. A comparison of networks with varying architectures will give insight in the importance of network complexity. While the radar setup will not fulfill demands regarding power consumption, size, and cost discussed above, it will work as a proof-of-concept for implementing gesture sensing using a low-power and small size RTD pulsed signal generator.

Both reciprocal (frequency-domain) and spatial data will be investigated in regards to classification performance in this thesis. The measurements from the radar setup will be processed using methods first shown on simulated data, different degrees of signal processing will be tested and evaluated. Classifying and comparing these methods will be useful for discussing the need of signal processing for deep learning. The generated waveform is pulsed contrary to using the in literature more popular FMCW approach. A number of 12 gestures, a relatively larger number relative to what is often found in studied literature, are tested to investigate limitations and/or capabilities of the proposed methods and difference in response between the gestures.

This thesis is structured as follows. Chapter 2 introduces theory for understanding relevant physics, radar operations and signal processing, as well as deep learning classification. Chapter 3 presents methods for choosing a suitable implementation of the radar setup, along with two alternatives. The chapter also presents measurement-and signal processing methods, as well as to how classification will be performed. Additionally, a simulation process is introduced where the proposed signal processing is visualized and tested. In chapter 4, results from the measurements and classification are presented along with analysis and discussion. Finally, chapter 5 summarizes the experiments, results, and discussions. Chapter 6 provides an outlook for further research as well as work that this thesis could be complemented with. At the end, an Appendix is presented with additional material, including additional tests and data visualizations.

Chapter 2

Theory

In order to better understand the methodology and results in this thesis some theoretical background in electromagnetic field theory, radar operations, signal processing, and artificial neural networks is presented.

2.1 Electromagnetic Field Theory

Electromagnetic (EM) waves are a combination of electric field (**E**) and magnetic field (**B**) waves. For plane waves the two fields exist in planes orthogonal to each other and propagate with the speed of light , $c = 3 \cdot 10^8$ m/s, in the direction orthogonal to both **E** and **B** [14] (p.5-8).

The amplitude A of **E** and **B** can be mathematically described as sinusoidal waves propagating in z-direction with time t, oscillating with a frequency f Hz:

$$A = A_0 cos(kz - \omega t + \phi_0) \tag{2.1}$$

where A_0 is the peak amplitude and ϕ_0 the initial (fixed) phase. The angular frequency ω rad/s and the wavenumber k rad/m are both related to frequency as:

$$\omega = 2\pi f, \quad k = \frac{2\pi}{\lambda}, \quad \lambda = \frac{c}{f} \tag{2.2}$$

where λ is the wavelength in meters.

The wavelength correlates to the propagation of an EM wave at a certain time t, where it is the closest distance between two corresponding points with equal amplitude on the sinusoidal **E** or **B** wave. Frequency can similarly be attributed to the number of oscillations or cycles per second for the propagating EM wave at a certain position z. The time between equivalent points on the EM wave is called the period T, which is equal to the inverse of f. Figure 2.1 illustrates the propagation of EM waves in free space, in the z-direction, at a certain time.

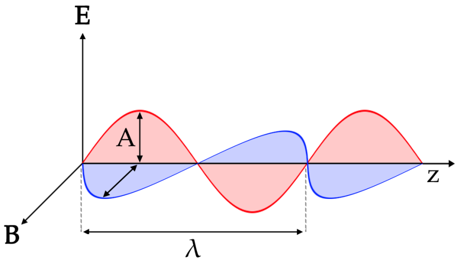


Figure 2.1: EM wave propagating in free space (z) at a set time (t).

The phase of the **E** and **B** waves at $t = z = 0$ is set by the initial phase ϕ_0 (2.1). The phase as the waves propagate is then determined by the total argument of (2.1). The phase at a specific time and distance with a certain initial phase can then be described as a function $\phi = f(z, t, \phi_0)$.

The intensity Q of an EM wave can be described as the power per unit area of the wave, which is the same as power density (W/m^2) [14] (p.10). In the theoretical case with a singe point in space acting as an EM-transmitting antenna, with an assumed isotropic radiation (equal power in all directions), the EM waves will propagate so that the amplitude (and thus the power) is identical at all points in a sphere around the antenna. The intensity Q is then the transmitted power divided by the surface area of the sphere at radius r (from the antenna):

$$Q = \frac{P_t}{4\pi r^2} \tag{2.3}$$

The intensity of the wave is thus decreasing $\propto 1/r^2$ [14] (p.10).

When an EM wave hit a physical object, it induces electrical charge on the surface of the object, which in turn radiates an EM wave back towards the origin of the incident wave. This process is called scattering or reflection of the incident wave. If the material is a conductor it means that charge can move freely in the matter, and thus all of the EM wave energy is reflected. If not, some or all of the wave energy will be absorbed in the matter, which results in a weaker or non-existent reflection.

2.2 Radar Operations

By measuring the reflection of transmitted EM waves, information about the range, relative position, velocity and direction of movement for one or multiple objects is acquired through radar operations.

2.2.1 Basic Radar Principles

A radar consists in its simplest form of a transmitter that emits an EM signal and a receiver which detects the transmitted and, upon contact with an object, reflected signal. A basic schematic is illustrated in figure 2.2. The transmitter consists of an electrical system that generates a signal and uses a transmitting antenna (Tx) for EM emission. The receiver often consists of an amplifier and sometimes utilizes signal mixing (section 2.2.3) together with an analog-to-digital converter [14] (p.4-5) for further digital signal processing. A receiving antenna (Rx) receives the reflected EM waves.

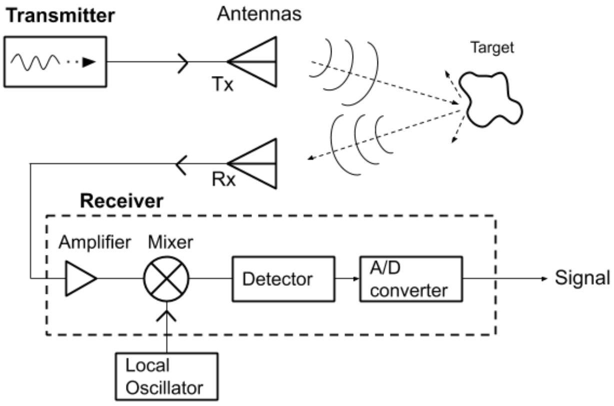


Figure 2.2: Schematic of basic radar setup with mixing.

When using the same antenna for both transmitting and receiving EM signals the radar is defined as monostatic. Bi-static radars use separate Tx and Rx antennas (figure 2.2). A setup that is physically bistatic but with the angle between the directed antennas close to zero can be seen as quasi-monostatic [14] (p.18-20).

In basic radar operations, there are two dimensions of spatial interest: cross-range (CR) and down-range (DR), illustrated in figure (2.3). DR is the direct path between the object of interest and the radar, and DR motion is movement along that path (radial movement). CR is the directions normal to DR. A movement in CR would not result in any movement in DR.

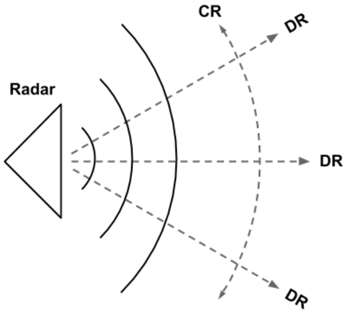


Figure 2.3: Down-range and cross-range illustrated with a monostatic radar antenna as reference.

Periodic waves can be classified as different waveforms depending on what shape and characteristics they have [15] (p.160-161). EM waves used for radar operations can generally be divided into two types of waveforms: continuous and pulsed waveforms [14] (p.20-21).

Continuous waveforms are transmitted and received continuously, i.e. the transmitter and receiver are always on. To determine a reflection of the signal in time, the wave needs to have some detectable change in its characteristics. This change is called modulation and can be in amplitude, frequency or phase. A modulation would then introduce a reference in the signal that can be detected, for example a section of increased amplitude or a sudden phase-change.

Pulsed waveforms are transmitted as pulses with a certain pulse duration τ. These pulses are called wavelets. The receiver does not need to be on then the transmitter is, as it detects the reflected pulsed signal with a delay ΔT, dependent on the one-way distance (range R) to the reflecting object. ΔT and R relates to each other as:

$$R = \frac{c\Delta T}{2} \tag{2.4}$$

where a factor of 2 is introduced as ΔT is the total delay of a pulse travelling to and from the reflecting object.

A pulsed monostatic radar with a single Tx/Rx can not differentiate between objects depending on their angle towards the radar: the only observed difference is for different DR distances (see figure 2.3). Multiple objects at the same distance in DR but separate locations in CR would be indistinguishable from each other in range.

2.2.2 Radar Measurements

The pulsed radar receiver will detect reflected pulses at a certain time after the initial transmission. This time (seen as ΔT in equation (2.4)) is called "Fast Time" and represents range information. Assuming the signal consists of pulsed waveforms, the pulse repetition frequency (PRF, in Hz) is the number of transmit/receive cycles performed or measured per second. The pulse repetition interval (PRI seconds) is the time between pulses and it relates to PRF as $PRI = 1/PRF$. By recording the fast time with a time PRI between recordings (thus PRF can be seen as rate of measurements or acquisitions) for a certain number of pulses each recording can be seen as one point in the total acquisition time, also called "Slow Time" due to it having a relatively slower sampling interval than the fast time [14] (p.502-503). This creates a 2D matrix of measurement data, seen in figure 2.4, called the radar datacube, or a Fast-Slow Time matrix.

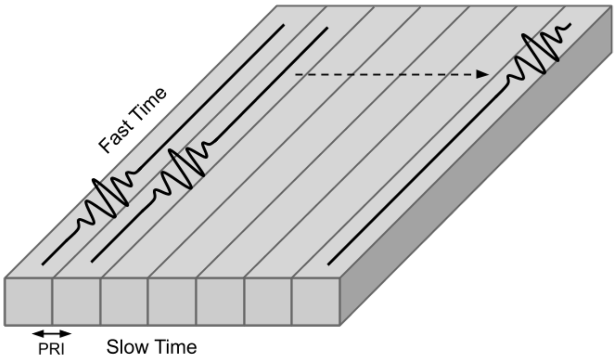


Figure 2.4: A radar datacube. Fast time measurements are "stored" in each slow time cell, as pulsed wavelets.

2.2.3 Frequency Mixing and Coherence

For practical reasons, radars must often downconvert the transmitted signal to an intermediate frequency (IF) that is low enough for detection in the limited bandwidths of the receiving system.

The original radio frequency (RF) that is transmitted is received at the receiving antenna after reflection. By using a non-linear device called a mixer, illustrated in figure 2.5 (a), the RF signal can be downconverted with a controlled and known signal coming from a local oscillator (LO) with a certain frequency [16]. This produces an IF (|RF-LO|) which is effectively downconverted in frequency (figure 2.5 (b)).

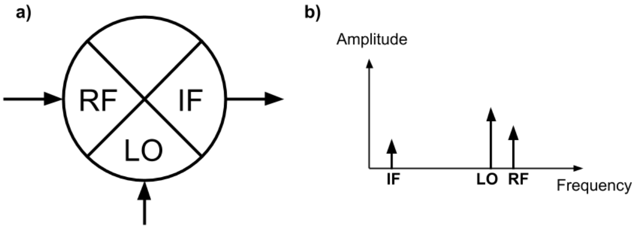


Figure 2.5: a) Schematic symbol of a frequency mixer. b) Frequency spectrum with the two mixed frequencies (LO and RF) and the resulting frequency (IF).

By using two mixers in parallel where the local oscillators supply the same frequency signal but with a relative phase difference of 90°(a sine and cosine wave respectively), a two-channel output called "In phase-Quadrature phase" or simply "I" and "Q" can be produced. Figure 2.6 shows a simple schematic for a an IQ mixer.

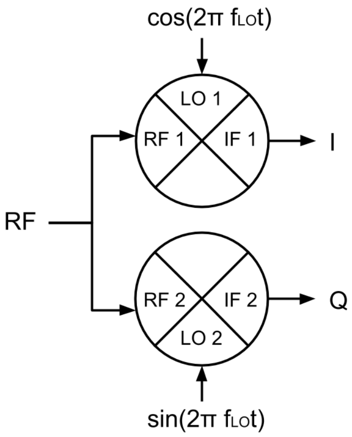


Figure 2.6: Schematic symbol of an IQ mixer.

A signal with I and Q components will in a unit circle produce both an amplitude and phase response $I + iQ$, where $i = \sqrt{-1}$. In radar systems, this enables coherent detection [14] (p.288-289), as the amplitude of the IQ signal is calculated as

$$A = |I + iQ| \tag{2.5}$$

where $I = A_I \cdot cos(\phi)$ and $Q = A_Q \cdot sin(\phi)$. As the phase determines the radial vector angle in the unit circle, it is possible to separate positive and negative phase changes as different rotating directions in the circle.

2.2.4 Sampling

When digitally measuring a signal, the sampling time T or sampling rate $R_s = 1/T$ is an important metric. To represent a signal $f(t)$ completely, the sample rate R_s needs to fulfill the Nyquist criteria [14] (p.497-499) of unambiguously representing a signal with frequency f_0:

$$R_s \geq 2f_0 \tag{2.6}$$

If the criteria is not fulfilled a signal could suffer from aliasing, where sampling points comes to far between to show the true oscillation pattern. Aliasing creates an incorrect wave representation, illustrated in figure 2.7.

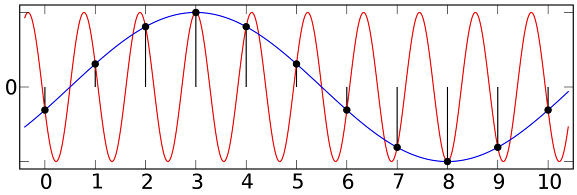


Figure 2.7: Sampling of a sinusoidal signal (red) where sample rate R_s is below the Nyquist criteria of $2f_0$ samples/second, resulting in an aliased representation (blue) of the original signal.

Thus, measuring and sampling a signal in real time (called Real Time (RT) Sampling) requires a sample rate that is at least twice that of the signal frequency. This limits measurements of high frequency signals to equipment that have a sufficiently high R_s, creating a "digital bandwidth" that limits the highest frequency detectable regardless of the "analog bandwidth" (hardware limitation). While mixing can be implemented (discussed in section 2.2.3) to achieve real time detection of signals with a R_s lower than the limit, a method called Equivalent Time (ET) Sampling can be used to sample a raw high-frequency signal with the only limitation being the analog bandwidth [17, 18]. The method is based on sampling a signal multiple times, each time at a slightly different trigger delay (which creates the demand for a repeating signal). Eventually, enough sample points exist so the signal can be represented unambiguously. As this method relies on multiple triggers for one detection, it comes with a price of more required time per measurement: effectively a lower measurement frame rate.

2.2.5 Doppler, Velocity, Range and Resolution

The physical phenomena of Doppler shift is when a transmitted frequency is observed as higher if the transmitter is moving towards the observer and vice versa [14] (p.274-276). The Doppler shift frequency (f_d, difference between observed and transmitted frequency in Hz) is expressed as

$$f_d \approx \frac{2v}{\lambda} \tag{2.7}$$

where v is the relative velocity of the transmitter and the observer.

Suppose the data from pulsed radar operations measuring a moving scatter object exists in a fast-slow time matrix as in figure 2.4. Measuring the signal phase at a specific sample point in fast time for each of the pulses in slow time produces a standing wave if the wavelets vary in range for different points in slow time. This standing wave yields a spatial Doppler frequency. If IQ mixing is performed, described in section 2.2.3, both negative and positive shifts can be detected due to coherent detection [19]. The Doppler frequency gives the relative velocity of the scatter object as:

$$v \approx \frac{f_d \cdot \lambda}{2} = \frac{f_d \cdot c}{2 \cdot f_{wavelet}}. \tag{2.8}$$

The Doppler frequency shift is sampled at the pulse repetition frequency (PRF). This creates a dependence on PRF as to not create frequency ambiguities [14] (p.628-629), it sets a limit of the highest velocity V_{max} that is possible to calculate, according to the Nyquist criteria (2.6) the maximum frequency that can be measured is half the sampling rate (PRF/2):

$$v_{max} \approx \frac{c \cdot PRF/2}{2 \cdot f_{wavelet}} = \frac{c \cdot PRF}{4 \cdot f_{wavelet}} \tag{2.9}$$

This is a contributing factor when determining the setups Doppler limitations.

If the PRI is shorter than the longest return time ΔT for a pulse, the pulse will be detected as part of another transmit/receive cycle ant thus produce a false detection. The condition for this not happening is called unambiguous range [14] (p.628-629) and is expressed as

$$PRI \geq \Delta T_{max} \tag{2.10}$$

and further, according to (2.4):

$$R_{max} \leq \frac{c \cdot PRI}{2} = \frac{c}{2 \cdot PRF} \tag{2.11}$$

The limitations in (2.9) and (2.11) both depend on the PRF, but one inversely to the other. This creates a radar ambiguity where a large PRF is good for the velocity limit and bad for the unambiguous range, and vice versa. The ambiguity can also be changed by altering the wavelength in 2.9. In both cases, if the ambiguity is surpassed, aliasing will occur along the range or velocity axis. This can be seen as a folding of the spectrum, where for example an increasing positive velocity becomes a decreasing negative one.

Further, the pulse length τ is important for range resolution. If two scattering objects are in close proximity, so that a reflected pulse from both will to the receiver look like a single but longer pulse, then the radar has reached its range resolution limit: its ability to discern multiple objects at different distances. The theoretical range resolution S_r (distance between two objects) should be that of a half pulse width, according to:

$$S_r = \frac{c \cdot \tau}{2} \tag{2.12}$$

2.3 Signal Processing

Radar data often undergoes processing in some way to achieve and interpret certain results. For many signal processing applications, including radar operations, Fourier transform is a powerful tool for signal processing [15] (p.24-27). This, together with the concept of sampling and non-periodic signal analysis will be covered in the following section.

2.3.1 Fourier Transform

Fourier transform (FT) is a mathematical operation that transforms a time-domain function into a frequency-domain function [20]. The operation can be described mathematically as:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \tag{2.13}$$

where $F(\omega)$ is the frequency-dependent function and $f(t)$ the time-dependent function. A constant $f(t)$ will in (2.13) produce a $F(\omega \neq 0) = 0$, as the time-domain function does not change over time, thus having no frequency. The component with $\omega = 0$ ($F(0)$) has a value separate from zero. For electronic signals, this can be seen as a bias: direct current (DC). Further, an infinitely small pulse in time-domain will translate to a infinitely broad frequency spectrum (constant $F(\omega)$). This follows from (2.13) where a $f(t)$ only has a non-zero value at a certain time t, which yields $F(\omega)$ = constant. The concept of frequency spectrum width is called bandwidth, and is expressed in Hz.

The operation can be performed in reverse, called inverse Fourier transform (IFT):

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega \tag{2.14}$$

The expression in 2.13 can be expressed as a Discrete Fourier Transform (DFT), to enable FT with a finite number of computations:

$$F(\omega) = \sum_{k=0}^{N-1} f[k]e^{-j\omega kT} \tag{2.15}$$

where $f[k]$ is a discrete sample of $f(t)$ at $t = k$, N the total number of samples, and T the time that separates each sample (sampling time). The Discrete FT can thus be performed with a finite number of computations for all ω. Here, $\frac{1}{NT} = 1$ Hz, and by setting $\omega = 2\pi \cdot 1n = \frac{2\pi}{NT}n$, where n is an integer $n = 0$ to $N-1$, the equation 2.15 can be written as

$$F[n] = \sum_{k=0}^{N-1} f[k]e^{-j\frac{2\pi}{N}nk} \tag{2.16}$$

where $F[n]$ is the Fourier transform at $\omega = \frac{2\pi}{NT}n$. The process of DFT is used by computers when digitally performing an FT.

2.3.2 Non-Periodic Frequency Analysis

Even though FT is ideal to analyze periodic signals, it is less suited for analysis of non-periodic signals: signals with a time-dependent change of frequency. In those cases, a FT of a signal would result in an integration over the total time and thus the total frequency response without the distribution of different frequencies at different times [21] (p.44-45).

A solutions for this is Windowed Fourier transform (WFT) [21] (p.44-50). This will perform a frequency-domain analysis with time-dependence as $F(\omega, t)$, and is implemented by performing a FT on discrete and finite time-windows of the signal. These windows in the time-domain signal can be defined by multiplying the signal with a windowing function the size of the desired window. The ”Hamming” windowing function used in this thesis takes its form after the equation:

$$w(n) = 0.54 + 0.46cos(2\pi \frac{n}{N}), \quad 0 \leq n \leq N \tag{2.17}$$

where N is the total number of discrete sampling points of the function. The Hamming windowing function then suppresses the signal intensity at the ”edges” of the window to decrease spectral leakage [22]. The function is illustrated in figure 2.8.

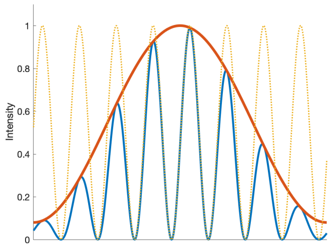


Figure 2.8: ”Windowed” sine wave (Blue) by multiplication with a Hamming window function (Red). Sine wave displayed as dotted line (Yellow).

2.4 Artificial Neural Networks

Artificial neural networks (ANN) is a very powerful tool for categorizing and understanding data. In a basic sense, an ANN is a set of algorithms that are able to process the input data given to the network and find connections and similarities in order to predict a specific output [23]. ANNs are vaguely inspired by the way the human brain implements a large set of connected neurons to process the inputs it gets via stimulated senses [10]. This section will cover the fundamentals of artificial neural networks, including the topics of network structure, training, optimization and overfitting. The basic operations of Convolutional Neural Networks (CNN) will also be covered here.

2.4.1 Network Structure and Components

ANNs is a collection of nodes organized in layers, the nodes are often referred to as neurons and are connected to each other via several inputs and an output. Each connection transmits information to other nodes and has adjustable weights that the network can vary in order to change how much any given input will affect the likelihood of generating a specific output. The general structure of an individual neuron can be seen in figure 2.9.

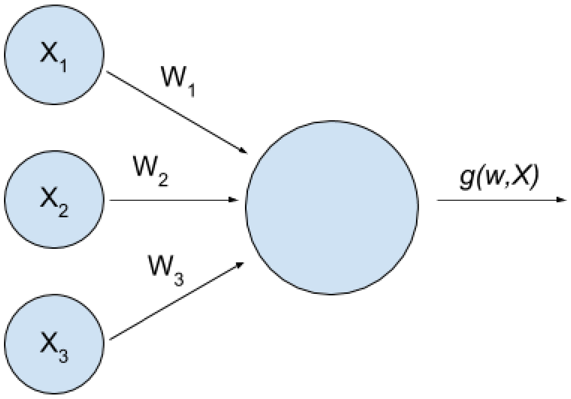


Figure 2.9: General structure of a neuron with three inputs.

The neuron can have multiple inputs denoted as n number of inputs $X = X_1, X_2, ..., X_n$ where each input is an independent variable. The neuron also has n number of weights $w = w_1, w_2, ..., w_n$, it can be useful to see each weight w as connected to an input X [24](p.171). The output of the neuron $g(w, X)$ is calculated as the sum of the inputs and their weights according to equation 2.18

$$g(w, X) = \sum_{i=1}^{n} w_n X_n \tag{2.18}$$

where w is the adjustable weights and X is the inputs to the neuron. The output from the neurons is then normally processed by an activation function: the activation function modifies the networks output value and is another layer of control for adjusting how the neuron behaves. There are different types of activation functions, for example linear functions ($f(x) = Const \cdot x$) or step-functions ($f(x) = 0$ if $x < 0$, $f(x) = Const$ if $x > 0$). Some popular activation functions used in ANNs are the Sigmoid, ReLu and the Tanh activation functions [25, 26].

An neural network can have many shapes, the simplest form of a neural network would just have a single input layer and an output layer. Most neural networks however has one or more hidden layers in between the input and output layers that adds complexity to the network which allows it to find more complex patterns in the data [24](p.165). An example of an ANN can be seen in figure 2.10

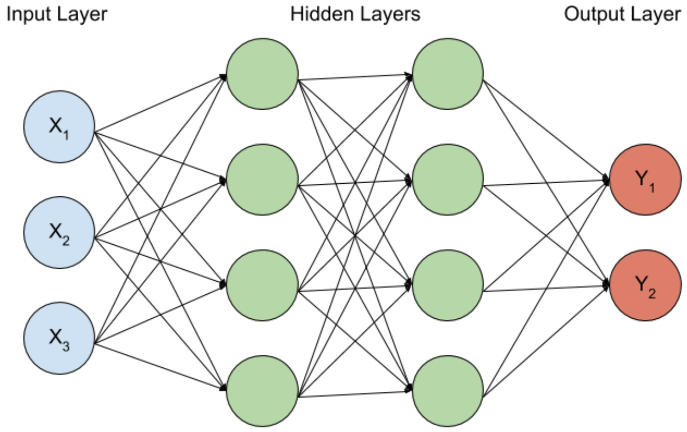


Figure 2.10: A neural network with three input nodes, two hidden layers with four nodes each and a output layer with two nodes.

The input layer is the first layer in the typical ANN an it is where the input data enters the network. The input layer consists of an array of input nodes that each represent an input value. The size and layout of the input layer varies depending on the size and form of the input data. When working with fully connected layers each input node is connected to the nodes in the proceeding hidden layer, the input nodes forwards the input data to the hidden nodes where each neuron receives the data according to the process previously described in equation 2.18. This process in then repeated for the second hidden layer but now the first hidden layer is treated as the input values. The number of hidden layers as well as the size of each layer varies between different networks, extensive research has been made in order to find a method for predicting the optimal number of hidden layers and nodes but no reliable way has been found [27]. The last layer in the network is called the output layer or the classification layer, this is the layer where the data is retrieved after being processed by the network. Just as the other layers the output layer may look very different depending on what kind of problem the network is designed to solve. In classification problems the output layer usually has one output node for each class with a value specifying the calculated probability of a class. Other problems might only have one output node returning a single value that is decided by the weights in the network.

2.4.2 Learning and Optimization

Tasks for machine learning networks are usually divided into two categories, supervised and unsupervised learning. Unsupervised learning is often used for very complex tasks where the goal might be to extract the underlying causes for problems where the right answer is uncertain or might not be available [28]. When using supervised learning the task that the network needs to solve is usually well known, a data set with labeled data is available and allows for training the network. When training a network using supervised learning the goal is to shift the weights into values that create a mapping where the desired output is predicted when the correlated input is given to the network [29]. The network is trained by providing it with a set of training data that is labeled with the correct answer, the data is processed by the network and a guess at the right answer is provided. If the answer is wrong the weights are slightly adjusted and the same process is repeated again, this process is done in an iterative fashion until the network manages to solve as many of the problems as possible.

In order to measure how well the network solves a given task a measure of accuracy is needed. A simple and common way to do it is to measure the networks accuracy when predicting labels, this metric does not however include the factor of how certain the network is on its guess. Instead networks are often trained using a cost function where the goal is to measure how far away the current weights are from an optimal solution [29]. For a network where the function f is the networks transfer function, x is the input vector and y is the target class the cost function can be described by using a mean square error function as can be seen in equation 2.19

$$C = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 \tag{2.19}$$

where C is the networks cost. By summing the squared difference between the predicted class and the correct class we can express a measurement of the accuracy of the network with a single mean squared error value which includes how close or far away the network was from an optimal answer. The goal of the training is now to minimize the systems cost [29].

The weights of the network are usually initialized as random small values before the training procedure begins, after being initialized the weights are systematically shifted into slightly smaller or larger numbers in order to find the best mapping of weights to minimize the cost function. The process of updating weights in this manner is called back propagation with stochastic gradient descent and is a very popular method for training networks [30]. There are several different optimization algorithms for updating the weights in a network, the one used in this thesis is called Adam and is a more advanced version of stochastic gradient descent [31]. The basics of both optimization algorithms work in a similar manner where the initial weight is updated by a small fraction Δw over many iterations t according to equation 2.20.

$$w(t+1) = w(t) + \Delta w(t) \tag{2.20}$$

This process is done for each weight in the network in order to find the optimal weights for minimizing the cost function and finding the correct values for each output $f(x_i)$ according to 2.21.

$$\frac{\delta C}{\delta w_i} = \frac{\delta C}{\delta f(x_i)}\frac{\delta f(x_i)}{\delta w_i} \tag{2.21}$$

The iterative process of updating weights is continued until a minima is found, however sometimes the network converges and gets stuck in a local minima where small nudges to the weights does not yield a better result even though a better allocation of the weights is possible at the global minima, an example of this problem can be seen in figure 2.11.

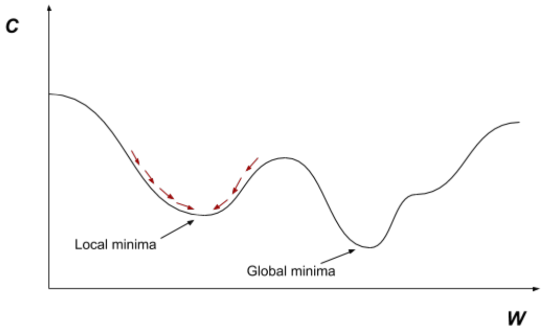


Figure 2.11: Visualization of a local and global minima of the cost function at different weight values.

2.4.3 Generalization

The networks ability to solve problems that has not been a part of the training data is referred to as the networks generalization and is an important factor in judging the success of the network. When training a network it is important that the the network performs well not only on tasks that was used during training but also tasks that it has never seen before. Strategies that are used to improve this ability are collectively known as regularization techniques. A common problem in machine learning is that the network after many iterations becomes very good at solving the specific tasks in the data set but completely fails at solving very similar problems, this problem is known as overtraining or overfitting and results from having a limited data set or a lack of regularization in the network [24](p.224). It is common practice to split the available data into a training set and a validation set, the training set is used to train the network using the iterative weight updating process and then the network is tested against the validation data in order to measure its performance[32]. The networks generalization is often discussed in terms of bias and variance, a network structure with high variance is prone to be very good at finding a solution for the training data but will often make the wights too specific and lose out in terms of generalization while a high bias will not be able to reach a high accuracy [24](p.127). In the context of a regression problem where the task is to fit a line to a collection of data points it is easy to see the results of the bias-variance trade off. In figure 2.12 we can see how the fit of the curve in the second example hits all of the data points but if the trained model would be used to classify other similar data the high variance network would not longer perform as well.

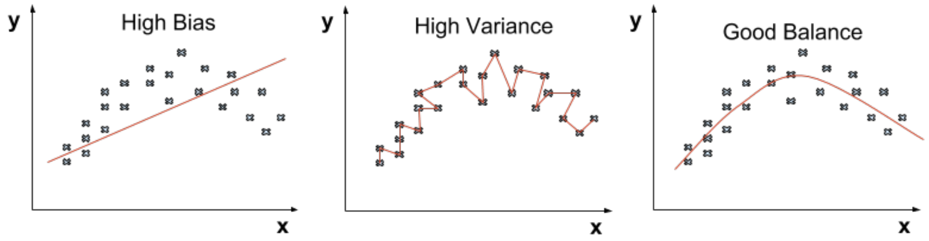


Figure 2.12: Representation of the Bias-Variance trade off in a network.

One regularization technique that has been rising in popularity since it was introduced is the dropout technique [33]. When using the dropout method each node in the network has a predefined probability of being removed from the network during that iteration of weight updates. This creates an ever changing network structure that will work similarly to an ensemble of several different networks. The ever changing network structure will prevent the network from overfitting to a specific detail and keep the network generalized.

L2 regularization is another popular technique that is used to prevent overfitting [34]. The goal of the L2 regularization technique is to keep the weights small and not let any single weight grow too dominant. This is done by adding a regularization term to the cost function described in equation 2.19 and thus penalizing big weights by having them directly contribute to the cost function by the power of two. The modified cost function can be seen in equation 2.22

$$C = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 + \lambda \sum W^2 \tag{2.22}$$

where λ is called the regularization parameter and is a hyper-parameter that has to be adjusted to a balanced value for each network.

2.4.4 Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a type of neural network that is very popular in the field of image analysis. Just like other neural networks they have an input layer, an output layer and fully connected layers with hidden nodes, in addition to these layers they also have convolutional and pooling layers [35] that will be introduced in this section.

To understand the convolutional layers the concept of filters must first be introduced. The filters used in CNNs are made to extract features in a matrix that consists of scalar values (a gray-scale image would result in a AxBx1 matrix). The filter systematically scans the image and performs a series of matrix operations called convolutions to produce a new image [24](p.327). The general process of a filter scanning an image is illustrated by the following matrix operation where an 4x4 image is convoluted by a 2x2 filter to produce a new 3x3 image:

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} * \begin{bmatrix} x & y \\ z & w \end{bmatrix} =$$

$$\begin{bmatrix} (xa+yb+ze+wf) & (xb+yc+zf+wg) & (xc+yd+zg+wh) \\ (xe+yf+zi+wj) & (xf+yg+zi+wj) & (xg+yh+zk+wl) \\ (xi+yj+zm+wn) & (xj+yk+zm+wn) & (xk+yl+zo+wp) \end{bmatrix}$$

By using different values in the filter matrix the image can be scanned for a certain feature. For example by using an $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ filter the image can be scanned for features along a diagonal axis.

Convolutional layers use this mathematical process to extract features by sliding the filter along an image according to figure 2.13 and thus creating a new image that indicates or highlights a specific feature [24](p.328), the new image is referred to as a feature map and is usually downsized compared to the original image. The filters size and its stride (number of moved steps along the input image between operations) is manually set before training, thus treated as a hyper-parameter.

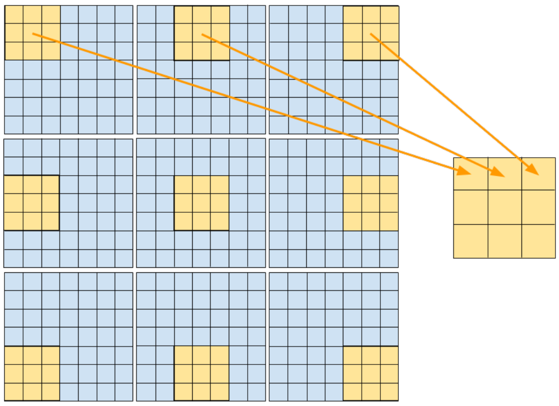


Figure 2.13: A 3x3 filter with stride 2 scans an 7x7 image.

In a convolutional layer the image is scanned several times with different filters to create several feature maps. Each feature map contains information highlighting where a specific feature or pattern is present in the image.

Due to the vast amount of parameters being created when using multiple filters the convolutional neural networks often include pooling layers. A pooling layer is usually located after a convolutional layer in the network. The pooling layer downsizes the image reducing the amount of parameters that is being used in the model, the progressive reduction in parameters helps to reduce the amount of needed computations and thus making the network faster while at the same time reducing over-fitting in the training process [24](p.336). There are different pooling operations available but the most common one in CNNs is the max-pooling operation. Just like the filters the max-pooling operation scans the image like in figure 2.13, the max-pooling operation however returns only the largest value in the window.

After the last convolutional and pooling layer the CNN ends with a set of fully connected layers that will process the information extracted by the convolutional layers, see image 2.14. In order for the fully connected hidden layers to process the information in the last feature map a flattening operation is performed where the feature map matrix is transformed into a 1D array of values [24](p.352). This array is then treated as an input array for the hidden layer network. The processed information is finally forwarded to the output layer that returns a value with the calculated probability of each class.

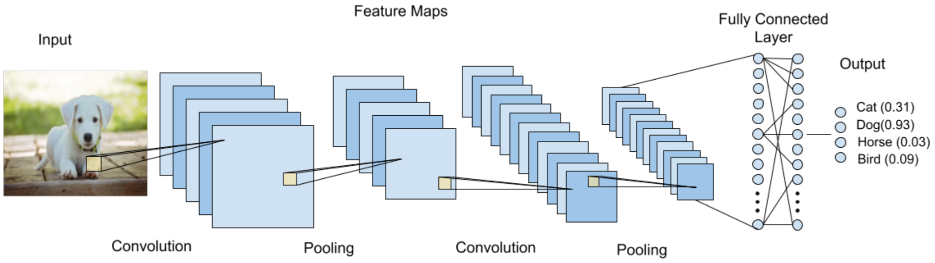


Figure 2.14: General structure and operations of a convolutional neural network.

Chapter 3

Method

The aim of this thesis is to explore the capabilities of an in-house radar for measuring hand gestures, and using convolutional neural networks (CNNs) to categorize the measured data.

This chapter will introduce two different setups used for measurements, as well as present the methodology for evaluating and choosing one of them. These two setups are centered around the use of a certain wavelet generator (WG). Further, the gestures under test will be presented, as well as the methods of measuring a large data set. The signal processing used on the data will also be discussed and presented along with radar simulations. Finally, a pre-trained and pre-designed CNN will be presented, as well as methodology for designing and optimizing a less complex CNN to be used for additional classification.

3.1 Millimeter-Wave Setups

For radar signal transmission, a coherent Resonant Tunneling Diode (RTD) WG is used. The RTD emits an electromagnetic (EM) signal at a certain frequency, which is dependent on the electronic bias as well as the device design. The device can produce wavelets with pulse lengths in the picosecond scale [13]. Distinct wavelets (signal pulses) are created by a switching transistor (MOSFET). The WG is prefabricated in-house and positioned on a silicon wafer die, and is contacted using a three-point probe. Here, a center frequency of 63GHz (in the 60 GHz ISM band) was emitted and used for gesture measurements.

For the following radar experiments in this thesis, two types of millimeter-wave radar setups were investigated and tested individually. The setups are based on the setup used in [36]. The central part of these setups is the use of the WG described above. The differing aspect of the setups is the method used for sampling the received signal when performing measurements.

One variant uses a ”real time” (RT) sampling oscilloscope with a limited input signal bandwidth, and is illustrated in figure 3.1. This setup utilizes IQ down-mixing (thus coherent) with a local oscillator to bring the signal down to a center frequency that fits the oscilloscope bandwidth (see section 2.2.3 and 2.2.4). Even with a low enough center frequency, the signal needs to fit in the limited oscilloscope bandwidth, thus a longer signal pulse length τ is needed as bandwidth $\propto 1/\tau$. This is due to the fact that a transform of a time domain pulse to frequency domain results in a larger spectrum of frequencies the shorter the pulse is (see section 2.3.1).

The other variant utilize an oscilloscope capable of ”Equivalent Time” (ET) sampling. This setup is illustrated in figure 3.2. ET sampling enables measurements without any downmixing step. Thus, the received signal is recorded at native frequency. This also enables the use of shorter pulse lengths, as the signal bandwidth is of less concern. The main difference of the setups is thus the frequency down-mixing that occurs for the RT setup in figure 3.1.

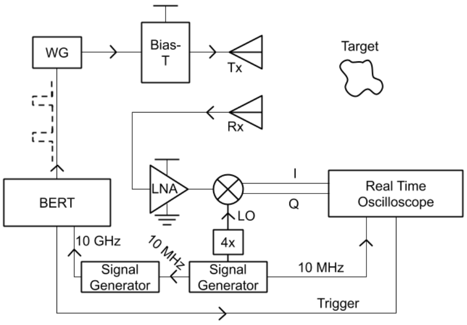


Figure 3.1: Real Time (RT) setup.

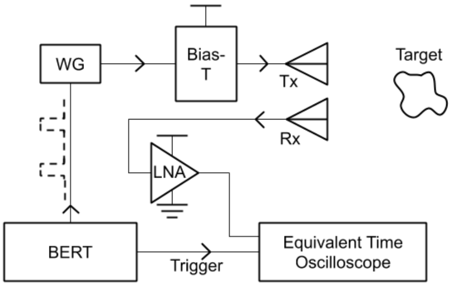


Figure 3.2: Equivalent Time (ET) setup.

The RT setup uses a `Rohde & Schwarz RTO 1044` oscilloscope with a bandwidth of 4 GHz and sampling speed of 20GSa/s (Gigasamples per second), while the ET setup uses a `Lecroy Waveexpert 100H - SE70` oscilloscope with a nominal bandwidth of 70 GHz. For the RT setup, the pulse length is set to 600 ps. For the ET setup, the pulse length is set to 80 ps.

For the RT setup, a `Agilent Technologies E8257D PSG` signal generator with $f = 15.3$ GHz is after multiplication (with a `Millitech AMC-RFH00` 4x multiplier) used as an local oscillator (LO) signal at $LO = 61.2$ GHz. An identical signal generator is connected to the BERT with a 10 GHz internal clock reference signal. A 10 MHz reference signal ensures that the phase of the LO and the 10GHz control signal is coherent. The IF output from the mixer (a `Millitech MIQ-15-01900` mixer) comes as an I-and Q-signal and is recorded by the oscilloscope in two separate channels. The LNA, mixer and horn antennas have a ”WR-15” waveguide interface, enabling a series connection between them without intermediate connectors or converters. The ET setup is more simple in the aspect of number of components. No mixer, LO or multiplier is used.

Both setups are bistatic (alternatively quasi-monostatic, due to closely spaced antennas) with `Flann 25240-20` horn antennas as transmitter (Tx) and receiver (Rx) antennas. They display a gain of 20 dBi [37]. A bias-T is used to supply power to a wavelet generator which is connected to the setup using a probing station with three-point probes. The generator is in turn excited with digital pulses from a `Agilent Technology N4906B` Serial Bit Error Rate Tester (BERT). The pulse length from the BERT decides the length of the WG from the generator and transmitted by the horn antenna. The BERT is also used to trigger the oscilloscope for detection. To amplify the received signal, a `HXI HLNAV-383` Low Noise Amplifier (LNA) is used at the receiving antenna. It presents a small-signal gain of 30 dB (at 50-65 GHz) with a noise figure of 5.2 dB (at 52-67 GHz) [38]. The setup is mounted on an optical table for reduced vibrations at the probing, and in fixed positions to ensure no cable bending.

A generated wavelet with a 100 ps pulse length is measured for reference and illustrated in figure 3.3 as the detected signal in the Rx horn antenna. It is sampled by the ET oscilloscope, and displays a center frequency of 62.5 GHz. Observe that multiple wavelet generators were used, and the one used for later measurements displayed a center frequency of 63 GHz.

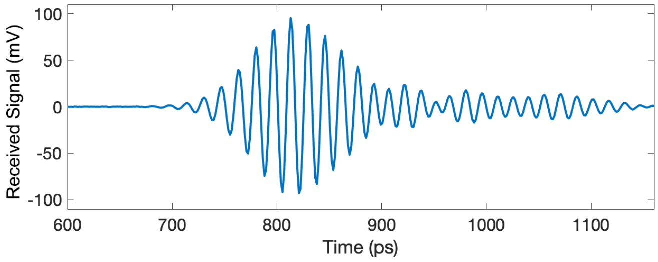


Figure 3.3: A measurement of a transmitted wavelet with a set 100 ps pulse length.

3.1.1 Choice of Setup

To evaluate the two setups, certain factors are taken into account.

The acquisition speed of the different oscilloscopes determines how many times the signal is measured during a performed gesture. This can effectively be seen as the pulse repetition frequency (PRF), which then can be used to calculate limitations according to section 2.2.5.

The smallest possible pulse length differs between the setups, which will create differences in the measurements.

Finally, the external control of the oscilloscope will be taken into account. This would include saving of data to external storage and initialization of measurements. The speed of data transfer, measurement saving and sending commands is of importance for convenient experiments.

Initial measurements of a small gesture set (6 gestures) is performed on both setups to relate the result with the differing factors under consideration.

3.2 Large Data Set Measurements

The final data set that is used for classification consists of 12 different gestures. This section describes the different gestures as well as methods of measurement and the precautions taken in order to create a valid data set.

3.2.1 Gestures

During the process of evaluating the capabilities of the radar setup, a set of gestures was tested and progressively expanded. Some gestures were modified compared to their initial implementation while some gestures were removed completely. The final set of gestures consists of the 12 different gestures that can be seen in figure 3.4.

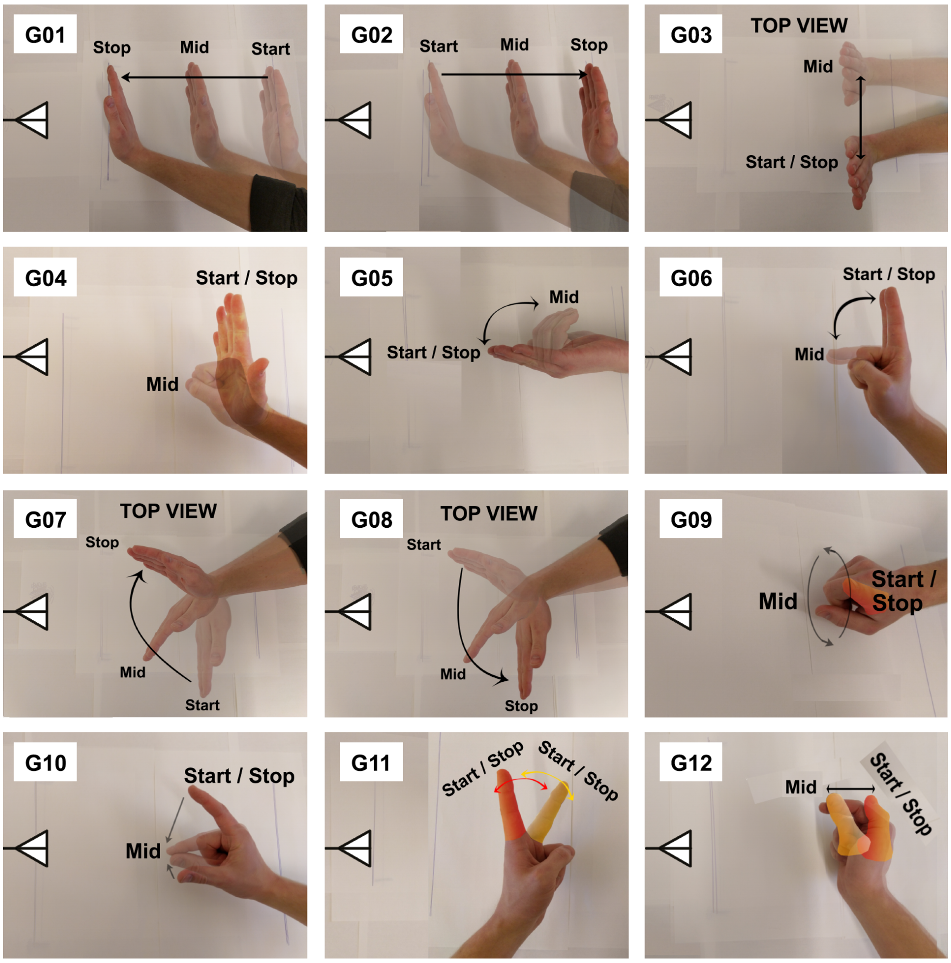


Figure 3.4: G01: Hand Forth, G02: Hand Back, G03 CR Wave, G04: Close and Open Fist, G05: Come Here, G06: Fold two Fingers, G07: Hand Swipe Right, G08: Hand Swipe Left, G09: Spin Finger in Circle, G10: Click with thumb and index finger, G11: Wiggle two Fingers, G12: Slide thumb back and forth on index finger.

G01 and G02 are two very basic motions where a hand is moved toward and away, respectively, from the radar in the down range (DR) direction (see section

2.2.1, figure 2.3). G03 is a waving motion performed in the cross range (CR) direction where the hand moves back and forth one time in front of the radar. In G04 the hand is closed into a fist and then opened again. For G05 the fingers are folded towards the palm and then opened again in a ”come here” motion. In G06 the index and middle finger are folded together towards the radar and then back again. G07 and G08 is two swiping gestures, in G07 the hand swipes towards the right and in G08 the hand swipes to the left. G09 starts with a closed fist with only the index finger pointing towards the radar, the finger is then rotated in a circular spinning motion. In G10 a click motion is performed with the thumb and the index finger, most of the movement is performed by the index finger. In G11 the index and middle fingers are wiggled back and forth, the fingers start separated and then perform a back and forth motion twice thus passing each other a total of four times. In the last gesture G12 the thumb slides along the index finger pointing towards the radar; the thumb slides towards the radar and then back again.

For most gestures the hand enters and exits the radars field of view (FoV) in a CR motion at the start and ending of each gesture. The exceptions to this is G01, G02 and G09. In G01 and G02 the hand enters the radars vision in the DR motion at the beginning or ending of the gesture. During G09 the hand never exits the radars FoV, the gesture is performed until the end of the measurement period.

As described in section 2.2.1 the radar setup is only capable of measuring distances in the DR direction, which affected the choice of suitable gestures. The different gestures have a varying amount of movement in the DR and CR directions, for instance G01 and G02 moves only in the DR direction while G03 practically only moves in CR. The gestures were chosen to have varying levels of difficulty and some gestures were intentionally designed to have similarities to each other ,such as G06 and G12 that include a small movement in the DR direction, or G09 and G11 that include a repeating periodic movement.

3.2.2 Method of Measurement

All of the measurements for the final data set were performed using the same setup. The gesture measurements were performed and measured in an distance-interval of 30 cm (corresponding to a 2 ns fast time interval on the oscilloscope). The time period for a single measurement was approximately 5 seconds per gesture. The distance from the Rx and Tx antennas to the closest point of a performed gesture was approximately 10 cm.

The measurement environment is depicted in figure 3.5. Figure (a) shows the setup in the direction of the radar emission, while (b) shows the opposite direction (looking ”into” the antennas). Figure (b) also shows an on-going gesture measurement, and the rough hand placement that was used. The antennas were facing an open area where the person performing the gestures was standing slightly to the side outside the radars FoV: only the hand performing the gesture was visible to the radar setup. EM-shielding barriers were placed down the direction of the radar emission (figure 3.5 (a), indicated in green) from the antennas to prevent any unwanted reflections from the background. Shielding was also installed around the antennas (figure 3.5 (b), indicated in green), to minimize any reflections in the forward direction.

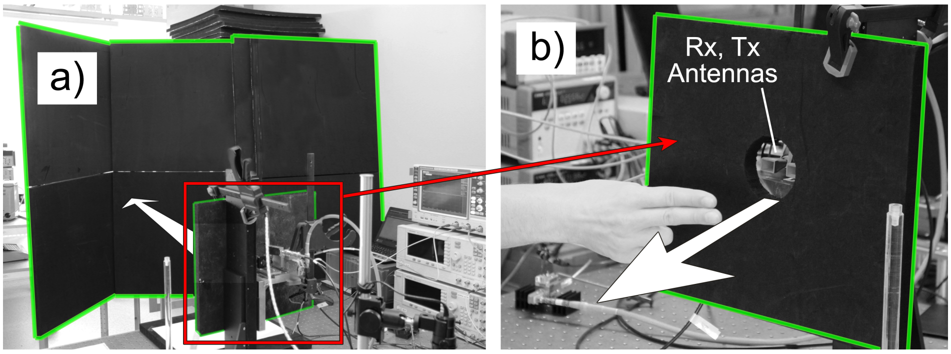


Figure 3.5: Measurement environment. Green lines indicate the EM shielding barriers. White arrows indicate direction of EM transmission. a) Setup shown in the direction of radar emission. b) Antennas and an on-going gesture. Antennas protrude through an EM shield plate.

3.2.3 Creating a Valid Data Set

To produce classification results with a high fidelity from deep learning models, a large amount of data is needed. A data set consisting of 180 measurements for each of the 12 gestures was recorded, thus resulting in a total of 2160 measurements. The measurements were conducted by two people (the authors) performing 90 measurements per gesture each, with measurements progressing through the gesture set instead of multiple measurements of the same gesture back to back. The measurements were initialized with a laptop running MATLAB that was connected to the oscilloscope. The laptop provided information to the user with start and stop times as well as what gesture to perform, thus no visual feedback from the oscilloscope was required. This increased the focus on performing the gesture as defined, instead of relying on visual feedback to modify the gesture and get an optimal radar response by looking at the signal plotted on the oscilloscope. When training the network, the training and validation data was chosen randomly and in approximately equal amounts from both authors.

The main data set used for training the network was created using only data with gestures from the authors. To investigate the generalization of the data and the classifying networks, two additional data sets were recorded. These two data sets were created by two additional persons (referred to as ”Test group”) who were not familiar with the project beforehand. The data sets were smaller, consisting of 10 measurements per person for each of the 12 gestures for a total of 240 gestures for the two data sets. The test group performed the gestures with no other instructions regarding how the gestures were performed than the predefined instructions in figure 3.4 with some initial feedback from the authors and pointers regarding the radar setups FoV. These smaller sets were used for validation purposes and not for inclusion in the training process. This will be discussed in section 4.3.5.

3.3 Signal Processing

Two domains are of interest for the measured data: time-domain and frequency-domain. The measured data is in its simplest form represented as varying range over time (Range-Time), where no processing is needed. To extract the range information, equation 2.4 is used.

Performing a Fourier transform (FT) of the slow time data (section 2.2.2) in the time dimension yields the frequency domain as a Range-Doppler spectrum, as described in 2.2.5. The DC component of each fast time column is removed according to:

$$S = S_0 - mean(S_0); \tag{3.1}$$

where S_0 is an arbitrary time-varying signals, and S represents the same signal but after removing the DC component.

The Doppler shift can directly be related to velocity, and thus the processed data shows the ranges where reflecting (scattering) objects have a certain velocity, which in most cases is not constant for the whole gesture range interval. Although this yields an additional dimension of information it also removes the dimension of time. To represent a Range-Doppler spectrum with an aspect of time, a ”windowed FT” (WFT) can be performed (section 2.3.2). By dividing the measurement time in a number of frames N and performing a FT on each frame, a time-dependence in N points can be shown for the Range-Doppler data. A window filter (Hamming, see 2.17) is applied for the Range-Doppler at each frame to decrease spectral leakage.

Figure 3.6 illustrate the numerical propagation of this ”sliding frame” WFT technique, where a frame size and frame overlap is shown. Overlapping frames will result in more possible frames for a set frame size, but is not implemented in the WFT method for this thesis.

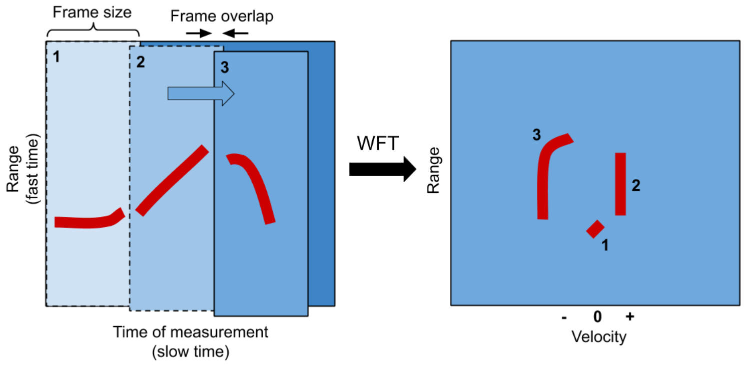


Figure 3.6: Illustration of a WFT process.

Before performing WFT, the Range-Time spectrum is cropped to exclude time intervals where no gesture is taking place. This is to ensure that the frames contain gesture radar data. The spectrum is cropped by defining a signal level where no gesture is taking place (noise level) and iterating through the columns in the fast-slow time matrix (along the slow time axis) to find the closest column from the start and end respectively where the mean signal level exceeds the defined noise level with a certain factor. Here, the noise level for a certain spectrum is determined by calculating the mean signal level for a small square area in the top corners in the fast time-slow time data matrix. The threshold for start-and end cropping is defined as 20 percent larger than the defined noise level for a certain spectrum. If the threshold is not reached, the WFT is performed on the whole image.

3.3.1 Data Representation

In this thesis, three data representations were used for classification. One was Range-Time data, without any signal processing except for removal of the DC component (equation 3.1). The second was a Range-Doppler representation, with a FT over the whole slow time interval. The third was Range-Doppler with WFT signal processing. All data representations have a native size of 434x343 pixels.

There are different options for representing time-domain in a WFT Range-Doppler spectrum as an image. For example concatenation, where the images are added in succession to create a larger image. Alternatively, the frames can be added and color-coded to indicate the order. Implemented here is a three-frame WFT with a step size of zero where the frames are represented as RGB (red, green, blue) channels respectively in a single image. Each frame is separately normalized to values between 0 and 255.

When converting matrices to images a decision of color representation needs to be made. Matlab has a built-in color scale, or color map, called ”hot”. This color map represents the pixel intensity with a color between black (zero intensity) and white (maximum intensity) with intermediate colors of red and yellow. This color map is here used in images for classification, due to the fact that in the case of conversion to gray scale the low-high intensity colors are black and white respectively. This yields an intensity range from 0 to 255, where 0 and 255 are the respective pixel values for black and white pixels in a gray scale image.

3.4 Simulating Radar Operations

By simulating wavelets with a specified frequency and length, a pulsed radar operation can be simulated by defining the wavelet placement in time domain (section 2.2.2). This is used to verify theory for the systems, such as unambiguous velocity limit, and the signal processing operations, such as the WFT process.

For basic simulations, some assumptions are made. The scattering object at range $R(t_n)$ is defined as an infinitely small perfect electrical conductor (PEC). The path of propagation is in a non-loss environment and the EM waves do not decrease in intensity meaning that eq. (2.3) in section 2.1 is disregarded. The digitally generated wavelet is sinusoidal. Finally, the radar is assumed to be monostatic.

3.4.1 Fast-Slow Time Operations

By defining a function $R(t)$ where t is slow time and R the object range for pulsed radar operations, a fast time-slow time matrix can be acquired by calculating the wavelet time domain placement ΔT (fast time) according to (2.4) for each point t_n in slow time:

$$\Delta T(t_n) = \frac{2R(t_n)}{c} \tag{3.2}$$

A fast-slow time matrix can be built as column vectors for each point t_n in slow time where there exists a wavelet $w(\Delta T(t_n))$ at $\Delta T(t_n)$ with a constant pulse-width τ and frequency:

$$\begin{bmatrix} w(\Delta T(t_1)) & w(\Delta T(t_2)) & w(\Delta T(t_3)) & \dots & w(\Delta T(t_N)) \end{bmatrix}$$

In the case of multiple reflections and detections in the same PRI, the fast-slow time matrix can be expressed as a sum of wavelets $\sum_{k=1}^{M} w_k(\Delta T(t_{n,k}))$ for each slow time:

$$\begin{bmatrix} \sum_{k=1}^{M} w_k(\Delta T(t_{1,k})) & \dots & \sum_{k=1}^{M} w_k(\Delta T(t_{N,k})) \end{bmatrix}$$

A frequency-mixing process (section 2.2.3) can be replicated in simulations by multiplying the RF signal with an LO-signal to acquire IF (|RF-LO|). An I and Q signal can then be generated through down-mixing as:

$$I = w_n \cdot cos(2\pi f_{LO} t_f) \tag{3.3}$$

$$Q = w_n \cdot sin(2\pi f_{LO} t_f) \tag{3.4}$$

where w_n is a wavelet at slow time t_n, and t_f is time in fast time. Applying a low-pass filter will then result in the down-sampled w_n to IF=|RF-LO| (see figure 2.5 b).

3.4.2 Simulation of Scatter Points

The presented steps of signal processing is Range-Doppler and WFT data with three frames and no overlap. The WFT data is represented as three frames in RGB channels, as discussed in section (3.3). A Hamming window was used to perform signal windowing for each frame, reducing spectral leakage but in the process removing some signal intensity at the start and end of the range-interval (see figure (2.8)).

The parameters for the simulations are presented in table 3.1. A down-mixing of the signal is simulated. This motivates the use of a pulse length of 600 ps as a down-mixing process would occur for a setup with limited signal input bandwidth, thus requiring a longer pulse length (section 3.1).

PRF	140 Hz
$f_{wavelet}$	63 GHz
Pulsewidth	600 ps
f_{LO}	60 GHz
Lowpass filter cutoff	4 GHz

Table 3.1: Simulation parameters.

Observe that in figure 3.7, 3.8, 3.9, the division of frames is noted as colored bars under each corresponding time interval (x-axis) for the Range-Time data.

3.4.3 Linear Movement

A linear movement in DR is defined in figure 3.7 (a), where the Range-Time data in (b) is Fourier transformed in (c) yielding a velocity of $0.02 m/s = 2 cm/s$, which is in agreement with the defined range-time function in (a). The Doppler velocity is calculated according to (2.8). As the velocity is constant for the whole movement, the WFT results in (d) are similar to the result in (c) but for different range intervals. The simulations also show a correct interpretation of a fast time-slow time matrix in (b), according to previous theory.

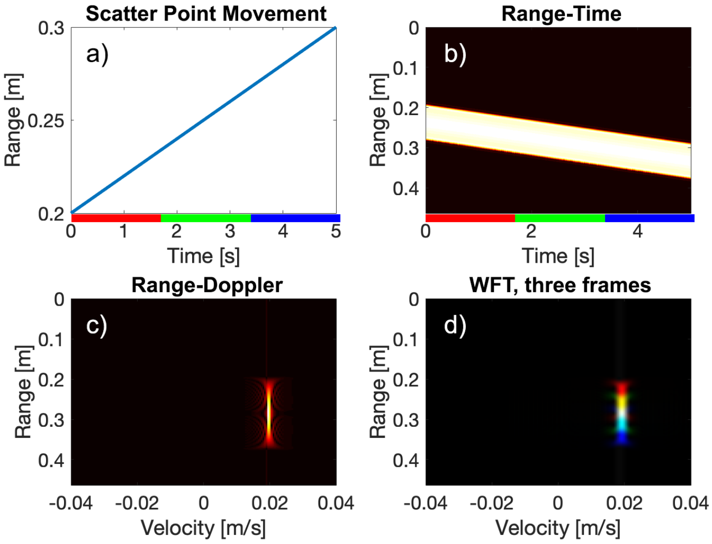


Figure 3.7: Linear continuous movement. a) Range as a function of time. b) Simulated fast time-slow time matrix (Range-Time). c) Range-Doppler over total measurement time. d) WFT with three frames (respective order in time: red, green, blue).

Figure 3.8 shows the simulation results for a linear but discontinuous movement divided into three parts. The object velocities is calculated in (a) to $5cm/s$, $0cm/s$

and $-11cm/s$ respectively. The Range-Doppler in (c) shows the expected response of all Doppler frequencies (velocities) present in the movement. By performing a WFT the three velocities are extracted for their time-dependent fashion, as presented in (d).

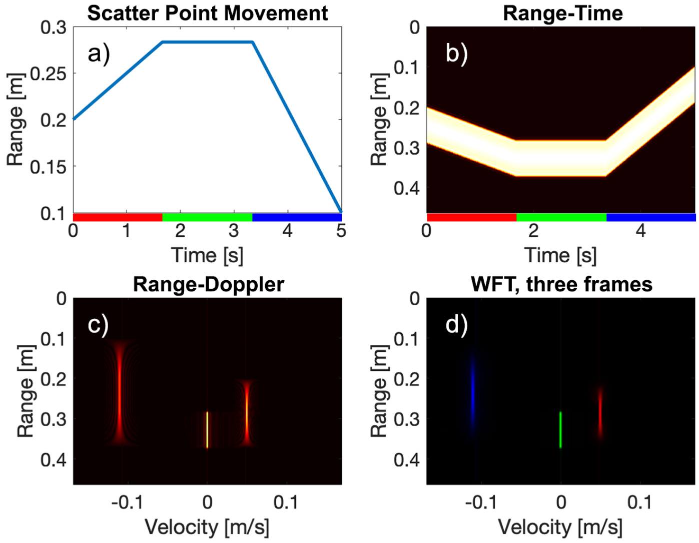


Figure 3.8: Linear discontinuous movement. a) Range as a function of time. b) Simulated fast time-slow time matrix (Range-Time). c) Range-Doppler over total measurement time. d) WFT with three frames (respective order in time: red, green, blue).

3.4.4 Non-linear Movement

Figure 3.9 presents a simulation of a non-linear movement, defined in (a) as an exponential range-time function. The velocities for this movement ranges from $\approx 0cm/s$ at the beginning of the measurement and $\approx 21cm/s$ at the end of the measurement (calculated with tangents at start-and endpoints in slow time). The unambiguous velocity limit (2.9) is here calculated to

$$V_{limit} = \frac{c \cdot PRF}{4 \cdot f_{wavelet}} \approx 17cm/s. \tag{3.5}$$

This indicates that the maximum velocity at the end of the gesture, $\approx 21cm/s$, should exceed the maximum velocity permitted by the unambiguous velocity limit with current parameters. This is the case as seen in (c) where the majority of velocities lies under the limit, but with clearly noticeable velocities also exceeding the limit. This can also be seen in (d) for the last frame (blue) and it confirms the expression in 2.9 for radar operations simulated as presented. This also presents a visualization of velocities exceeding the unambiguous limit: the Doppler response folds and appears on the other side of the velocity spectrum, now with reversed direction. This is known as aliasing, indicated by a red square in figure 3.9 (c) and (d), see section 2.2.4.

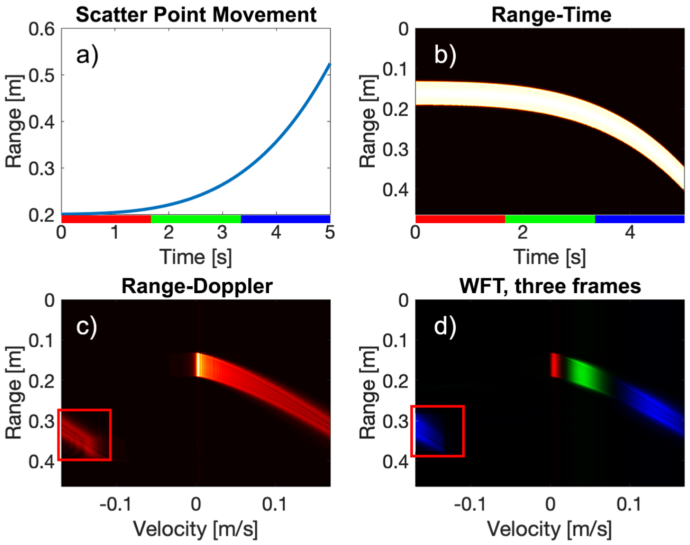


Figure 3.9: Non-linear movement. a) Range as a function of time. b) Simulated fast time-slow time matrix (Range-Time). c) Range-Doppler over total measurement time. d) WFT with three frames (respective order in time: red, green, blue). The red square in c) and d) indicates the observed aliasing of the signal.

Figure 3.9 (d) also shows the value of WFT (compared to a complete Range-Doppler such as in (c)) for a signal with a frequency that is time-dependent (non-constant velocities). Smaller frequency changes in time or smaller windows yields a sharper velocity-time dependence, as can be seen in frame one (red) and two (green) where the velocity is relatively constant. While in frame 3 (blue) the velocity interval is noticeably increased, due to the constant frame size but exponential velocity increase.

3.4.5 Multiple Scatter Points

A human hand will not act as a single scatter point: but as a surface that simplified can be seen as a collection of scatter points. To verify the proposed signal processing for radar measurements containing more than one reflected signal, a simulation containing all three above defined movements (figure 3.7, 3.8, 3.9) are combined and simulated as one measurement. The simulated response is seen in figure 3.10, which shows that the simulation and signal processing work as it did above for the single scattering objects. Here the product of signal processing is equal to a sum of the singular cases above. The WFT in (d) shows a closer grouping of Doppler velocities in the red and green frames, but more divergent in the blue. To note is that the figures in (c) and (d) has a post-simulation amplified intensity to better observe the aliasing as the image intensity is dependent on relative signal strengths. But aliasing is non the less present as expected for multiple scatter points.

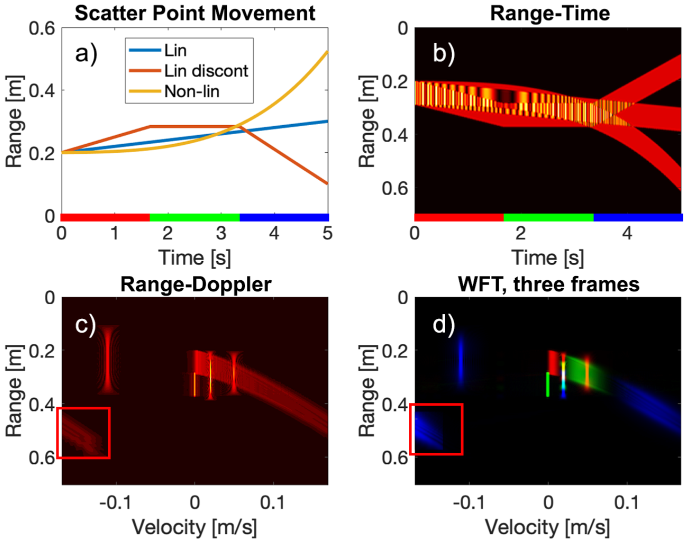


Figure 3.10: Combination of three previously defined movements. a) Range as a function of time. b) Simulated fast time-slow time matrix (Range-Time). c) Range-Doppler over total measurement time. d) WFT with three frames (respective order in time: red, green, blue). The red square in c) and d) indicates the observed aliasing of the signal.

3.5 Classification

Developments in the field of deep learning have made CNNs viable as computer-vision tools where the result of a classification comes from the input of an image [10]. Radar data as spectrum images in different executions can thus act as inputs to a CNN.

The viability of using CNNs to classify radar data is evaluated. This section presents an already existing network to be used for initial classification, called `ResNet50`. It also presents the methodology for designing and optimizing a new CNN to be used for radar data classification.

3.5.1 `ResNet50` - A Pre-Defined Network

As a first order of analysis, the radar measurement data will be classified using transfer-learning from a pre-defined network. Transfer-learning utilizes pre-trained networks that are extensively trained on image sets that often range in the order of millions with thousands of different classes. Instead of starting at zero or being randomly initialized, the weights in the CNN have a starting value dictated by the previous training. Training the network on new data continues to update the pre-allocated weights and thus adapting them to the new material.

A pre-trained and pre-defined network that is highly regarded in literature and which will be used in this thesis is called `ResNet50`[1] [39, 40]. The network architecture presents $25.6 \cdot 10^6$ trainable parameters. The network is pre-trained on the ImageNet[2] image set.

Although radar data might be a different sort of input compared to the data used when pre-training `ResNet50`, the network still carry an ability to recognize a range of shapes and details in the image. Fine-tuning the training with new data could yield a powerful model of classification, although probably after a very time-consuming training process due to the deep design of the network architecture.

3.5.2 `SimpleNet` - A Low-Complexity Network

In addition to `ResNet50`, an unique CNN architecture is developed and trained from scratch. It will henceforth be referred to as `SimpleNet`.

An important aspect to be taken into consideration when working on a project with a limited time-frame is the time needed for training. A network with a large number of trainable weights (parameters) will take longer to train than a network with fewer trainable weights. In the case of `ResNet50`, with 25.6 million parameters, the computational load and time will be significant. The training time is also of importance if hyperparameter optimization is of concern, which requires numerous training iterations (see section 3.5.3). Thus, a demand on the architecture of `SimpleNet` is a relatively low complexity. A comparison between `SimpleNet` and `ResNet50` would give insight in how complex the design of a network needs to be for a certain result.

Here, four different layouts were tested using two different image sizes as the inputs. Classification accuracy and training time is of concern when choosing network, and serves as the basis for the choice of network. The networks were varied with two conditions: one or multiple convolutional (Conv, or ”C”) layers, and one or multiple Fully Connected (FC) layers. The input sizes under test was 24 x 24 and 48 x 48, illustrated in figure 3.11.

[1] A detailed layout of the network architecture (accessed 2019-05-20) can be found at `http://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006.`

[2] Information about ImageNet (accessed 2019-05-20) can be found at `http://www.image-net.org.`

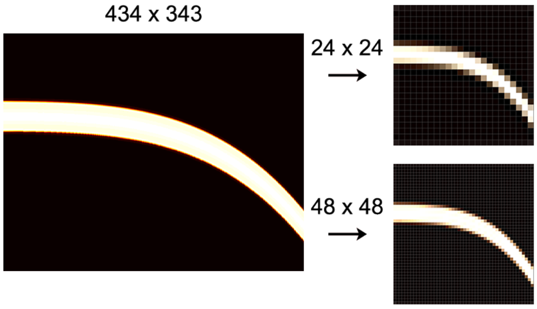


Figure 3.11: Example of simulated data downsized from its original size 434x343 to 24x24 and 48x48.

These networks were trained with one-channel (gray scale) Range-Time data with 60%-40% training-validation data. MaxPool layers (MP) was used to decrease the image size as it propagates through the network. Dropout (DO) and L2-regularization was utilized, and the training period was 200 epochs. A "Softmax" layer is used to normalize the output to a probability distribution, used for classifying the input. The network layouts were designed as presented in table 3.2 and the training options are denoted in table 3.3.

Single FC Multiple Conv	Single FC Single Conv
Multiple FC Multiple Conv	Multiple FC Single Conv

=

Net 1	Net 2
Net 3	Net 4

Table 3.2: Schematic of four networks with two changing conditions (left). Notation of different networks (right).

Optimization Algorithm	ADAM
Train/Validation	60%/40%
Epochs	200
Minibatch size	200
Learning rate	0.0001
L2-reg. constant	0.0005
Dropout chance	50%

Table 3.3: Training options for Net 1-4.

A detailed layer architecture for networks with 24 x 24 input size can be seen in table 3.4, a visual representation of a CNN can be found in figure 2.14. The networks with 48 x 48 input size is identical but all convolutional layers (filters) are of twice the size.

Net 1	Net 2	Net 3	Net 4
10 C(8 x 8)	20 C(12 x 12)	10 C(8 x 8)	20 C(12 x 12)
MP(2 x 2)	MP(2 x 2)	MP(2 x 2)	MP(2 x 2)
10 C(6 x 6)	FC(512)	10 C(6 x 6)	FC(100)
20 C(6 x 6)	DO(50%)	20 C(6 x 6)	DO(30%)
10 C(6 x 6)	FC(12)	10 C(6 x 6)	FC(100)
MP(2 x 2)	Softmax	MP(2 x 2)	FC(100)
20 C(4 x 4)		20 C(4 x 4)	DO(30%)
FC(512)		FC(100)	FC(100)
DO(50%)		DO(30%)	FC(12)
FC(12)		FC(100)	Softmax
Softmax		FC(100)	
		DO(30%)	
		FC(100)	
		FC(12)	
		Softmax	

Table 3.4: Layout of Net 1-4 with (24 x 24) input size. "C": Convolutional Layer, "MP": MaxPool Layer (stride 2), "FC": Fully Connected Layer, "DO": Dropout Layer. All "C" layers has ReLU activation function, normalization layer and stride 1.

For each network, the classification accuracy and total training time were recorded and the total number of trainable parameters were calculated. Observe that the purpouse of the "time" observations are to show the difference in speed between networks, and are not supposed to be reproducible. The networks were trained in Matlab on a single `Intel i7 3400 Sandy Bridge` CPU.

3.5.3 Hyperparameter Optimization

The hyperparameters of the chosen model are optimized using a process called "Bayesian Optimization" [41]. This process is here used to maximize network validation accuracy by varying hyperparameters in pre-determined intervals. The method builds a surrogate model of the optimization problem and moves the parameter values in a calculated "guess" of optimal direction. The operation needs no function derivatives. The relevant hyperparameters and their respective defined interval of interest were the learning rate: $1e-4$ to 1 in logarithmic scale, L2 regularization rate: $1e-6$ to $1e-1$ in logarithmic scale, mini batch size: 10 to 1296 in linear scale (integers), and dropout rate: 0 to 1 in linear scale. A session with 60 training iterations of `SimpleNet` were performed, with the data from the large data set.

Chapter 4

Results and Discussion

The results of this thesis includes a choice of radar setup, measurements of the chosen gesture set, classification results using transfer-learning on the pre-defined convolutional neural network (CNN) `ResNet50`, as well as the design and classification results from an additional and less complex CNN called `SimpleNet`. Analysis and discussion is presented regarding the results, as well as an evaluation of the trained models classifying data from a test group not involved in training the networks.

4.1 Setup Evaluation

This thesis presents two different millimeter-wave radar setups, a real time (RT) sampling setup and an equivalent time (ET) sampling setup, with two different oscilloscopes (section 3.1). In preparation for measuring the final data set a question of big importance was the decision of which of the two setups to use. The oscilloscopes have different properties in regards to what frequencies they can measure as well as acquisition speed and pulse length. In this section the results from cross-comparison of the setups is presented.

4.1.1 Difference in Performance

Initially, pros and cons of the two setups are presented in table 4.1. The setup acquisition rates were 140 Hz and 14 Hz for the RT and the ET setup respectively. This was the maximum implementable measurement speed on the instruments used.

ET	**RT**
$+$ ET Sampling $\Longrightarrow$ Higher range res. (80 ps) Simpler setup	$+$ Faster acq. rate (140 Hz) $\Longrightarrow$ Higher velocity lim. Externally controllable, fast operations
$-$ Slower acq. rate (14 Hz) $\Longrightarrow$ Lower velocity lim. Inconvenient to implement external control, slow operations	$-$ RT Sampling with limited BW $\Longrightarrow$ Worse range res. (600 ps) More complex setup

Table 4.1: Pros and cons with the two different setups.

The complexity refers to the number of components in the setup.

As an initial setup evaluation, measurements of a small gesture set (six gestures) were performed. Of these six gestures, five exist in the final gesture set (see section 3.2.1: hand forth (G01), cross range wave (G03), fold two fingers (G06), hand swipe left (G08), wiggle two fingers (G11)). Initial classification using a simple CNN design yielded a relatively high validation accuracy (>80%) for a positive outlook of gesture classification using both setups. This classification was done with 21 data points for each class, thus it is hard to draw conclusions from the result except that it for both setups seems possible to perform classification using CNNs.

Also, several test measurements were performed where the radar pulse was reflected against a metal plate that was moving toward the radar at a fixed velocity. Figure 4.1 shows a measurement performed at 15 mm/s. Here it is clearly visible how the pulse lengths differ between the setups. According to equation 2.12, the ET and RT setup has a higher and lower range resolution respectively, due to the pulse lengths used (see section 2.2.5). The different range resolutions, unambiguous ranges (equation 2.11) and unambiguous velocities (equation 2.9) are presented in table 4.2. The acquisition rate is effectively seen as the pulse repetition frequency (PRF) in the fast-slow time matrix (see section 2.2.2).

	Unambig. range	Unambig. velocity	Range res. (S_r)
RT	1071 km	16.7 cm/s	9 cm
ET	10’710 km	1.67 cm/s	1.2 cm

Table 4.2: Unambiguous range/velocity and range resolution for the RT and ET setups.

It is evident that the unambiguous range is of no concern for either setup, as gestures will be performed a factor of 10^6 closer. Although the range resolution is larger for ET, the unambiguous velocity limit is higher for the RT setup due to a higher PRF - with a factor of 10. Figure 4.2 shows the Fourier transform (FT) of the images in figure 4.1, where the velocity limit is illustrated. As for range resolution, the RT setup shows an S_r only 13.3% of the ET S_r.

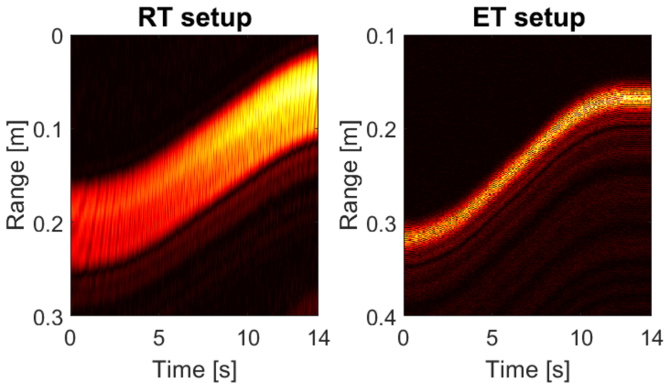


Figure 4.1: Range-Time diagram of a metal plate moving 15cm towards radar at 15mm/s with acceleration and deceleration at the start and end of the movement. RT setup (left) and ET setup (right).

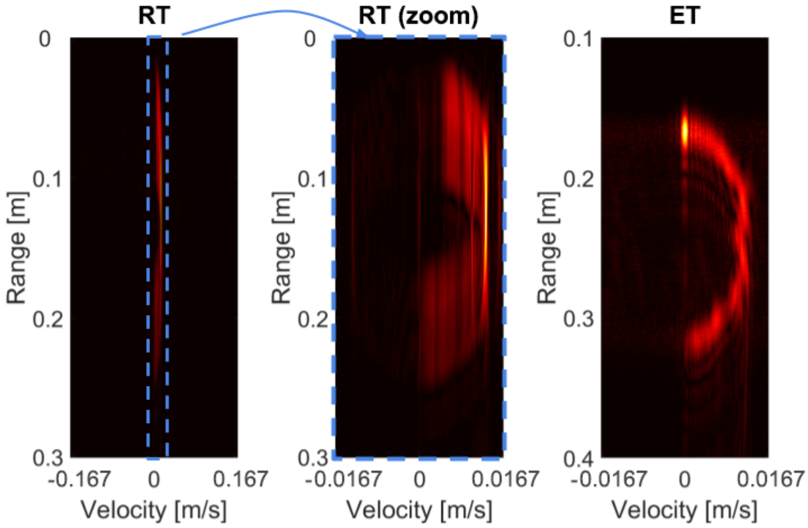


Figure 4.2: Range-Doppler diagram of data from figure 4.1. RT setup (left), zoomed RT (middle), ET setup (right). The X-axis in the left and right image represents the approximate velocity limit of the corresponding setups.

4.1.2 Choice of Setup

The scope of the thesis allowed for use of only one setup for the large data set measurements. Although initial classification experiments yielded a positive outlook for both setups, the data under test was limited. The setups differed in theoretical limitations, which will be the basis for the choice. The ET setup and the higher range resolution would be attractive for detailed finger movement, but the limitations in velocity measurements is limiting for frequency-based classification. The RT setup shows an acceptable unambiguous velocity limit, as well as range resolution still in the order of centimeters. A big factor behind the decision of setup was also how convenient the instruments could be controlled and recorded, as well as the time efficiency for the oscilloscope operations. With reasonable range resolution, good velocity limit, convenient and fast data recording and management, the RT setup was chosen for recording a big data set for classification.

4.2 Measurement Data

Measurements with Range-Time representation for each of the 12 gestures described in 3.4 can be seen in figure 4.3.

The hand forward and hand back motions (G01 and G02) were before the measurements regarded as the two easiest gestures to perform and measure, this assumption seems to be correct as they both produce easily distinguishable diagonal lines corresponding to the hand moving toward or moving away from the radar.

The movement in the waving motion (G03) is primarily in the cross range direction, the actual movement in the down range direction is negligible and the pattern produced is created from the hand moving in and out of the radars field of view (FoV) and thus creating two flashes where a pulse is reflected.

The pattern from closing and opening a hand into a fist (G04) is not as easily distinguishable. A continuous signal is observed for the entire length of the gesture with a clear variation in distance that can be seen as the fist is opened and closed since the fingers move closer to the radar in the fist position and the palm moves slightly away during the process.

The motions from folding two fingers (G06), clicking with thumb and index finger (G10) and sliding thumb back and forth on index finger (G12) all show similar patterns to that of G04. The hand movement when performing G06 has a resemblance to the one performed in G04, both gestures have fingers moving towards and away from the radar with a fist like stance in between. This similarity expresses itself in the radar patterns for the two gestures, G06 has a slightly larger variation in distance. G10 includes a movement separated in cross range but not in down range: we can not distinguish between the thumb and the index finger but rather their distance to the radar changing at the same time (see section 2.2.5 and range resolution). G12 shows a smoother curve where the thumb is slowly sliding back and forth along the index finger.

When performing the come here motion (G05) a recognizable U-shape pattern is created that originates from the motion where the fingers moves away and then returns towards the radar. Two flashes in signal intensity can be seen at the times the fingers are pointing vertically to the radar.

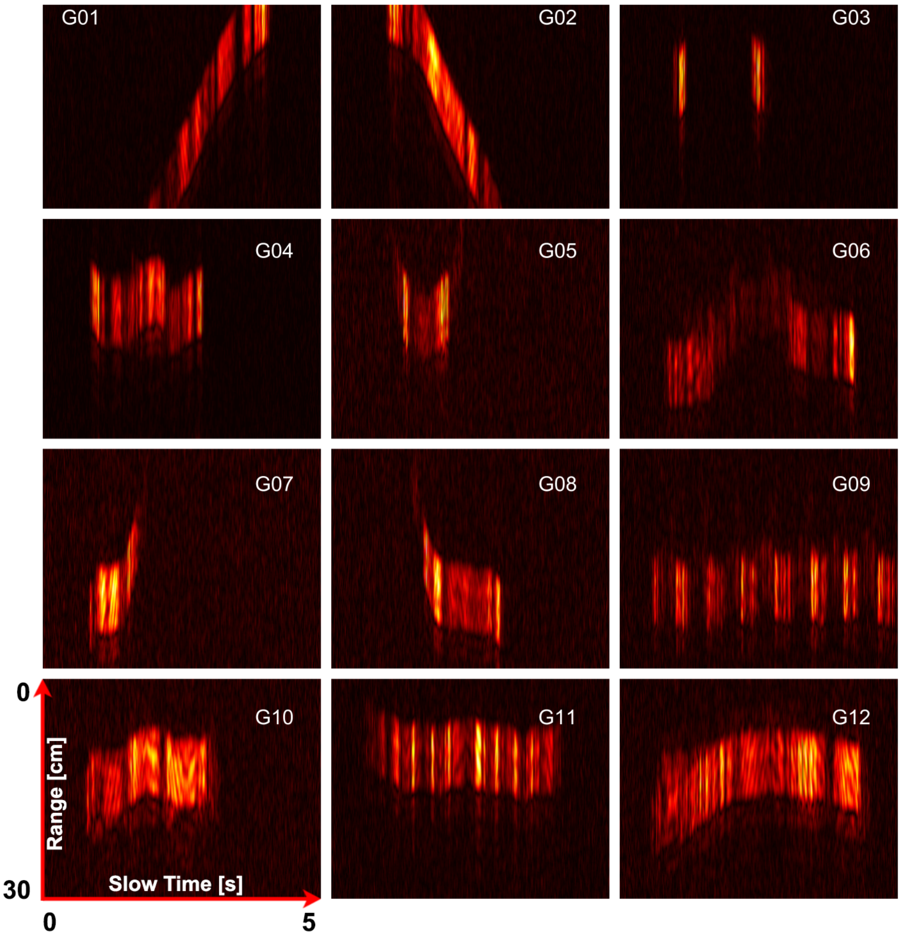


Figure 4.3: The 12 gestures represented as Range-Time data. G01: Hand Forth, G02: Hand Back, G03 CR Wave, G04: Close and Open Fist, G05: Come Here, G06: Fold two Fingers, G07: Hand Swipe Right, G08: Hand Swipe Left, G09: Spin Finger in Circle, G10: Click with thumb and index finger, G11: Wiggle two Fingers, G12: Slide thumb back and forth on index finger. The axis in the bottom left represents all measurements. Observe that the range interval 0-30 cm is relative for the gesture and not the actual distance from the radar antennas.

The gestures for swiping right and left (G07 and G08) show patterns that resemble each other but are inverted, as expected. Two stationary parts of the gesture from before the hand enters or exits the radars FoV and then a small portion similar to the ones seen in G01 and G02 where the hand moves towards or away from the radar during the swiping motion.

Spinning a finger in a circle (G09) and wiggling two fingers back and forth (G11) both produce a repeating pattern of periodic detection. For G09 it seems that at a certain point during the hands rotation there is a larger reflection and the rest of the time the comparative signal level is so weak that it is not visible. The same thing happens for gesture 11 where there are spikes in intensity each time the two fingers cross each other, the periodic flashes in signal intensity creates two patterns that look very similar to each other.

In appendix, figure A.3, A.4 and A.5, the WFT and Range-Doppler data representation for the measurements in figure 4.3 can be seen. The cropped Range-Time data used for acquiring the WFT data is also shown.

4.3 Classification

This section presents classification results for the gesture measurements. Initial classification was performed with `ResNet50`. The chosen architecture of `SimpleNet` is presented, along with discussion behind the choice as well as `SimpleNet` hyperparameter optimization results, followed by classifications with `SimpleNet`. The gestures are also evaluated by studying confusion matrices from `SimpleNet`. Additional analysis is performed with gesture measurements from a test group that were not involved in the network training.

4.3.1 Classification Using `ResNet50`

The pre-trained network `ResNet50` is trained via transfer-learning. The network was trained with Range-Time data, Range-Doppler data, and three-frame WFT data (section 3.3.1). Here, the results come from a single training session of the network and no averaging between multiple sessions, as training `ResNet50` is a relatively time consuming process. No sophisticated hyperparameter optimization was performed for `ResNet50` other than some manual optimization. Both data representations have an input size of 224x224x3 pixels (three channels, RGB) as this is the image input size of `ResNet50`. The model has a total number of $25.6 \cdot 10^6$ parameters. No layers were frozen when training `ResNet50`, meaning all layers had updating weights when training with novel data (see section 3.5.1). The training parameters are presented in table 4.3.

Opt. Algorithm	ADAM
Train/Validation	60%/40%
Epochs	15
Minibatch size	64
Learning rate	0.0001
L2-reg. constant	0.001

Table 4.3: Training options for `ResNet50`. "Opt. Algorithm": Optimization Algorithm.

The validation results are presented in figure 4.4. The final validation accuracy for the completely trained network is presented in table 4.4, for Range-Time, Range-Doppler and WFT data respectively. Confusion matrices for the three data representations can be seen in appendix (figure A.2).

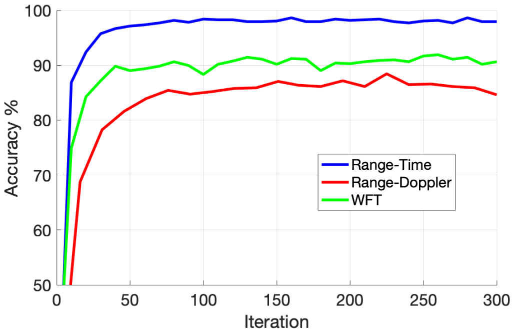


Figure 4.4: Validation accuracy for `ResNet50`. Results are from a single training session.

	Range-Time	Range-Doppler	WFT
Val. Acc (%)	97.9	84.6	90.6

Table 4.4: Validation accuracy of different data types on `ResNet50`.

In figure 4.4, it is seen that Range-Time data performs best, followed by the WFT data. The worst performance is seen with Range-Doppler data. This is reasonable, as no time-information is present here (section 2.3.2) thus removing a dimension of interest to the classifier. The WFT data has both range, velocity (Doppler) and time as dimensions. A representation of time in frequency-domain is thus evidently important for radar hand gesture classification using CNN. Although, more than three frames might yield even better classification results.

4.3.2 Choice of `SimpleNet` Architecture

The data presented in table 4.5 are the results from the network design experiments detailed in section 3.5.2.

Network	Validation Accuracy (%)	Time (min)	Total nr of params
24 x 24			
Net 1	91.8	9.2	394'098
Net 2	89.7	5.4	378'248
Net 3	90.8	9.3	125'662
Net 4	90.9	5.3	105'552
48 x 48			
Net 1	92.7	40.5	1'568'798
Net 2	89.9	9.3	1'492'808
Net 3	92.1	40.2	407'182
Net 4	89.9	8.9	331'192

Table 4.5: Performance of different networks for different image input sizes.

The models that utilize a single fully connected (FC) layer has a higher total amount of parameters compared to the ones using multiple FC layers. This is because the FC and Conv layers are of a smaller size when there are multiples of them compared to the networks only utilizing one, as can be seen in table 3.4. Net 1 is the best performing network regarding validation accuracy, the 48 x 48 input is better still but more time consuming. Due to limited time and the need to test many different permutations of the data the computational time is of importance when choosing the network. Having a simple fast network that is complemented by ResNet50 that is a more complex and slow network makes Net 1 with input size 24 x 24 a suitable choice.

To increase performance in the network, hyperparameter optimization with Bayesian optimization (section 3.5.3) was performed to maximize validation accuracy. The results can be seen in table 4.6:

Parameter	max(Accuracy)
Learning Rate	0.0046595
L2 reg rate	0.005242
Mini Batch Size	246
Dropout Rate	0.87

Table 4.6: Result of hyperparameter Bayesian optimization, maximizing validation accuracy.

The yielded learning rate in table 4.6 was after the optimization process lowered to 0.0005 (an approximate factor of 10). This was to decrease the relatively aggressive weight updates after each iteration. This manifested as large dips in accuracy at some places, even though the accuracy at the end of training was of a good degree. A decrease in learning rate to 0.0005 did not yield any worse accuracy results, but resulted in a ”smoother” training process. Thus, the final and applied learning rate is documented in table 4.7.

4.3.3 Classification Using `SimpleNet`

The `SimpleNet` network was trained using the hyper-parameter configuration seen in table 4.7, the network was trained with Range-Time data, Range-Doppler data, and three-frame WFT data. The image input sizes in pixels are 24x24x1 for Range-Time and Range-Doppler (1 channel: gray scale) and 24x24x3 for WFT (three channels, RGB).

As validation accuracy varies between training sessions, probably due to a random division of training and validation data as well as random initial weight values causing the network to find different local minima (as described in section 2.4.2 and shown in figure 2.11), training of the model is performed 25 times with averaged validation accuracy. This is seen in figure 4.5 as the mean validation accuracy together with a 95% confidence interval. Table 4.8 shows the mean validation accuracies together with the peak validation accuracy, defined as the maximum measured validation accuracy at the end of a training session.

Optimization Algorithm	ADAM
Train/Validation	60%/40%
Epochs	200
Minibatch size	246
Learning rate	0.0005
L2-reg. constant	0.005242
Dropout rate	0.87

Table 4.7: Training options for `SimpleNet`.

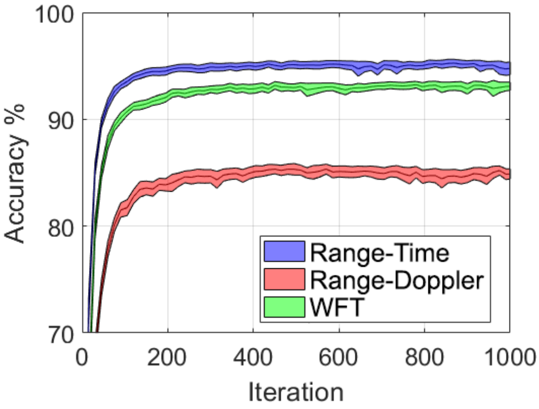


Figure 4.5: Mean value with a 95% confidence interval for validation accuracy during training of Range-Time data, Range-Doppler data, and WFT data.

	Range-Time	Range-Doppler	WFT
Mean Val Acc (%)	95.0	85.0	93.0
Peak Val Acc (%)	97.2	86.6	94.7

Table 4.8: Validation accuracy of different data types on `SimpleNet`.

`SimpleNet` performed best on the Range-Time data, closely followed by the WFT data. The Range-Doppler data was noticeably the worst. These results agree with the results from `ResNet50`.

From a trained network, a confusion matrix can be produced with the validation data. The matrix presents true classes vs predictions, illustrating the classification performance for different classes (gestures). For the results of multiple `SimpleNet` training sessions (figure 4.5), confusion matrices was calculated at each training session, then summed up and averaged. The matrices are shown in figure 4.6 top left, top right, bottom, for Range-Time, Range-Doppler and WFT data respectively.

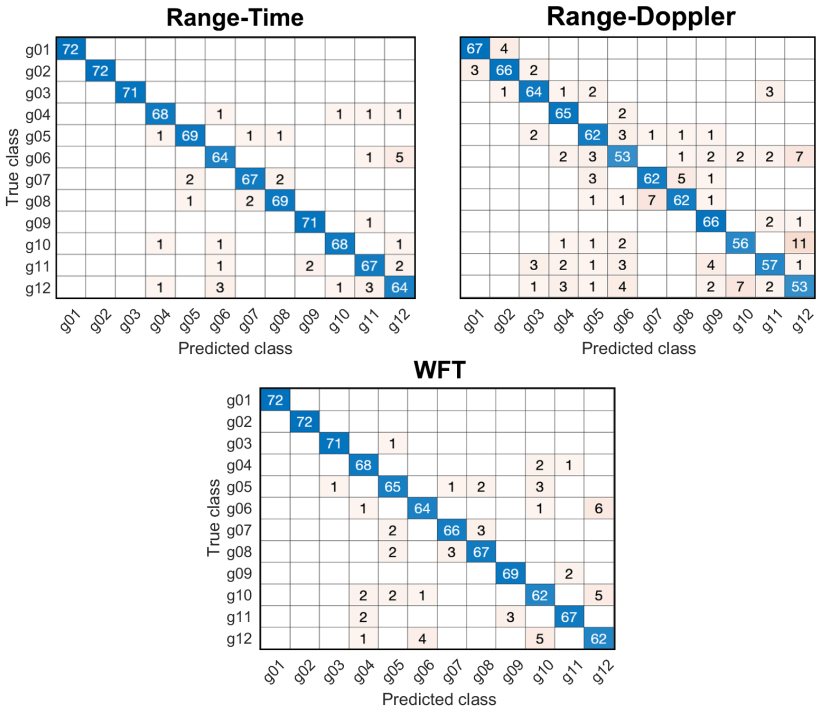


Figure 4.6: Confusion Matrices for `SimpleNet` classification. Range-Time (top left), Range-Doppler (top right), WFT (bottom). Results are averaged for 25 training sessions and rounded to closest integer.

The results in figure 4.6 mirror those from figure 4.5: Range-Time and WFT data performs best for `SimpleNet` classification. For all three data representations, the errors exist mainly for gestures G04-G12 (see figure 3.4 and 4.3 for information about the gestures). This is reasonable as G01 (hand forth), G02 (hand back) and G03 (waving hand) can be seen as the gestures with least complexity and clearest "intent" among the 12 gestures. For these three gestures, the whole hand is moving as a flat surface, without any separate finger movement.

One relatively prominent trend for all three data representations is that of G06 (fold two fingers) being classified as G12 (slide thumb). Studying the Range-Time response in figure 4.3, there are similarities in the range measurements, as folding two fingers first approaches then moves away from the radar similarly to the thumb sliding along the index finger. One would expect a reciprocal error, where G12 is also classified as G06. This is observed as well, but not quite as prominent. It is possible that measurements from G12 thus is more uniform than those from G06.

For the frequency-domain data, it is observed that G10 (click fingers) and G12 is classified as each other to some extent, as well as G07 (hand swipe right) and G08 (hand swipe left). As above, G10 and G12 have similar measured spectras in figure 4.3. Both gestures are based on stationary hands with one or two fingers moving. This is not observed in Range-Time to the same extent. For G07 and G08, they can ideally be seen as mirror images to each other. This is grounds for possible difficulties in separating them with the Range-Doppler data, as both G07 and G08 has a stationary part either before or after the movement, thus the information of whether it is located before or after the motion is lost in the Range-Doppler data. The WFT classification is observed to decrease the amount of errors for faulty G10-G12 and G07-G08 classification. This might be due to the reintroduction of a time-dependence, yielding additional information to the classifying network. The loss of the time-aspect in measurements is evidently a large factor in Range-Doppler classifications under-performance.

Overall, the results shows that both distinct and less distinct gestures can be classified to a relatively high degree for Range-Time and WFT data, but with lower accuracy for the Range-Doppler data. The classification results are also noteworthy when taking the range resolution (see section 4.1) of the setup into account. Even though some gestures utilize sub-or on-the-border-resolution movement with different scatter points on the hand, for example G10, G11, G12, a >95% classification result can be presented. Additionally, the unambiguous velocity limit might be another reason for the sub-par performance of the Range-Doppler data for cases where this limit might have been surpassed. This is expected to affect both the Range-Doppler and WFT data classification performance, but the WFT representation might counter this with the time-dependent frame order which might help differentiate the gestures.

Additional experiments for data with a thirteenth gesture G13 which consisted only of noise (no gesture taking place under measurement) were performed to investigate if any of the original 12 gestures were classified because of a lower signal strength and thus recognized because of its higher level of noise. The `SimpleNet` network was thus trained using Range-Time data while including a 13th gesture that consisted of the noise data, the results showed a 100% validation accuracy for G13. Further, no other gestures were classified as G13, which indicates that none of the 12 original gestures relies on only noise for classification.

4.3.4 Comparison of SimpleNet and ResNet50

The classification of Range-Time data proved to be effective with both ResNet50 and SimpleNet. While ResNet50 had a higher accuracy for Range-Time data, it also presents a large model complexity ($25.6 \cdot 10^6$ parameters) and is more computationally heavy compared to SimpleNet which presents a much lower complexity ($394 \cdot 10^3$ parameters). Additionally, the image input size for ResNet50 is approximately a factor of 10 larger than for SimpleNet.

For the WFT data, the SimpleNet outperforms ResNet50 with 2.4 %. One reason for the worse performance on the WFT data with ResNet50 could be due to the type of data used. ResNet50 is pre-trained on large amounts of images that has a correlation between the different color chanels in the spatial domain, the RGB channels previously had no time-dependence between them and thus the model might be unsuitable for classification of frequency-domain data with a time-dependence between its color channels. Another reason might be that the model needs more regularization to increase its ability to generalize, although L2 regularization was in fact used for training ResNet50. A Bayesian hyperparameter optimization was performed on SimpleNet but not on ResNet50, which might be a reason for the slight under-performance of WFT data with ResNet50. The reason for no Bayesian hyperparameter optimization for ResNet50 was the very large time required for network training (section 3.5.2 and 3.5.3).

The Range-Doppler data was the worst performing data representation for both networks. As discussed, a time-dependency in the data seems important.

To note is that the results from ResNet50 comes from one training session, and are not averaged as for SimpleNet.

The performance of ResNet50 compared to SimpleNet illustrates that the gap of model complexity does not necessarily mean a gap of classification performance. It is possibly so that ResNet50 is unnecessarily complex as classifying the radar data used in this thesis arguably is not as difficult as classifying 100'000s or 1'000'000s of widely varying images in 1000's of classes. Still, controlling devices with hand gestures and radar data will have high requirements on speed and it is a positive outlook that the classification methods can go down in complexity and data input size, and still achieve acceptable performance.

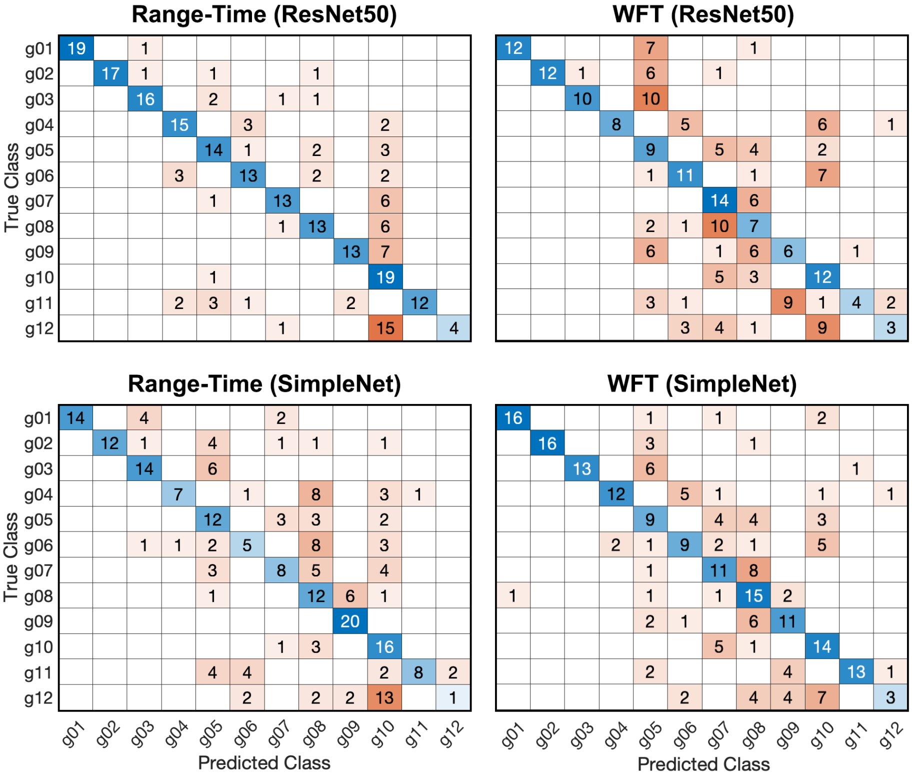
4.3.5 Classifying Data from Test Group

Gesture measurements from the test group (section 3.2.3) were used as validation data on both the SimpleNet and ResNet50 networks, after training using the authors data set. The data from the test group thus had no influence on the training and were only used for testing the model classification capabilities. Observe that for SimpleNet, the network was that of one single training session, without any type of averaging.

Range-Time and three-frame WFT data was classified. The results of the validation is presented in table 4.9, with confusion matrices in figure 4.7.

	SimpleNet	ResNet50
All gestures		
Range-Time	53.75%	70.0%
WFT	59.17%	45.0%

Table 4.9: Validation accuracy for data from test group.



Range-Time (ResNet50)

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12
g01	19		1									
g02		17	1		1			1				
g03			16		2		1	1				
g04				15		3				2		
g05					14	1		2		3		
g06				3		13		2		2		
g07					1		13			6		
g08							1	13		6		
g09									13	7		
g10					1					19		
g11				2	3	1			2		12	
g12							1			15		4

WFT (ResNet50)

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12
g01	12				7			1				
g02		12	1		6		1					
g03			10		10							
g04				8		5				6		1
g05					9		5	4		2		
g06					1	11		1		7		
g07							14	6				
g08					2	1	10	7				
g09					6		1	6	6		1	
g10							5	3		12		
g11					3	1			9	1	4	2
g12						3	4	1		9		3

Range-Time (SimpleNet)

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12
g01	14		4				2					
g02		12	1		4		1	1		1		
g03			14		6							
g04				7		1		8		3	1	
g05					12		3	3		2		
g06			1	1	2	5		8		3		
g07					3		8	5		4		
g08					1			12	6	1		
g09									20			
g10							1	3		16		
g11					4	4				2	8	2
g12						2		2	2	13		1

WFT (SimpleNet)

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12
g01	16				1		1			2		
g02		16			3			1				
g03			13		6						1	
g04				12		5	1			1		1
g05					9		4	4		3		
g06				2	1	9	2	1		5		
g07					1		11	8				
g08	1				1		1	15	2			
g09					2	1		6	11			
g10							5	1		14		
g11					2				4		13	1
g12						2		4	4	7		3

Figure 4.7: Confusion matrices for data from test group. `ResNet50` Range-Time (top left) and WFT (top right), `SimpleNet` Range-Time (bottom left) and WFT (bottom right).

The test-only data shows relatively poor results when validated on the networks trained on data from the authors. The classification for the Range-Time data with `ResNet50` performs the best with 70% accuracy, which is 27.9% lower than in table 4.4. The discrepancy between `ResNet50` and `SimpleNet` results for Range-Time data is here larger than previous tests (section 4.3.4) and `ResNet50` performs 16.3% better than `SimpleNet`. On the other hand `SimpleNet` outperforms `ResNet50` on the WFT data with 14.2%. Again, as in section 4.3.1, a weaker performance on WFT data is observet for `ResNet50`.

While the Range-Time previously showed the best results for `SimpleNet`, WFT data here provides the highest classification results with an accuracy of 59.2%. This could mean that `SimpleNet` can generalize the model for WFT data better than for Range-Time data. Combining this with the good results from Range-Time training in table 4.8, one conclusion could be that the model is more accurate for Range-Time data but more general for WFT data.

According to figure 4.7, G12 generally has the worst performance for both of the networks. It is also noticeable that G12 to a high degree is classified as G10 and many of the other gestures in the Range-Time data also has the same tendency. When analyzing the training data from the authors it was observed that the average image of G10 has a relatively high level of noise, meaning that the signal level for the gesture is generally quite weak. Thus this might indicate that some gestures is partially categorized due to its weak signal instead of the measured radar pattern. This trend mostly seems to be prevalent in the Range-Time data and not as obvious in the WFT, thus the signal processing steps performed in the WFT seems to alleviate this issue. For the classification of Range-Time data using `SimpleNet`, a trend in error where G04-G07 are classified as G08 is also observed.

The results from the test group would point towards the need of personalized training of the model, alternatively a larger number of persons contributing to measurements. To note is the relatively small size of the test-only data set: it is hard to draw conclusions, as more data would be needed for more confidence in the result. Additionally, classification results from training on data from one author and validation on the other are presented in appendix A.2.

Chapter 5

Conclusion

Initially, two different radar setups were tested and evaluated, one using equivalent time (ET) sampling and the other using real time (RT) sampling and down-mixing of the signal. Due to a higher acquisition rate (effectively pulse repetition frequency PRF) and more convenient implementations of data recording the RT setup was found to be the most suitable setup. A pulse length of 600 ps and a PRF of 140 Hz was used.

A data set consisting of 12 different gestures was recorded, each gesture had 180 measurements per gesture, 90 from each of the two authors, and an additional 20 measurements from a test group with two volunteers for a total of 200 measurements per gesture.

The gestures were selected with different degrees of complexity. They were chosen continuously up until the time for big scale measurements. They were performed under the same time-span and in the same distance interval to the radar.

The measured gesture data was classified with convolutional neural networks (CNNs). The data was presented in three different image representations: Range-Time data (no processing), Range-Doppler data (Fourier transformation (FT) on data), and WFT data (Windowed FT). The different data representations were presented to the CNN as spectrum images in time and frequency (Doppler) domain respectively. WFT processing produced a Range-Doppler representation with three frames, represented as the R, G, and B-channel in an image. In order to verify the signal processing, radar simulations in MATLAB was performed and analyzed.

Two different networks were used to analyze the data, one pre-defined and pre-trained network called `ResNet50` used for transfer learning, and a network of lower complexity designed for this thesis called `SimpleNet`. During the design of `SimpleNet` multiple permutations of different layer structures were tested and evaluated, resulting in the choice of the network with second-best validation accuracy but with fastest training time. The accuracy difference between the best and second-best was $\approx 1\%$. Bayesian optimization was used to optimize `SimpleNet` hyper-parameters. The hyperparameters of `ResNet50` was not optimized with an algorithm, but manually to due to the significantly larger time consumption for network training.

The results from `SimpleNet` and `ResNet50` training-validation is summarized in table 5.1. The `SimpleNet` results are averaged for 25 number of trainings.

Range-Doppler data was the worst performing for the networks, probably due to no time-dependency to be found in the data.

Some difference was seen between SimpleNet and ResNet50 for Range-Time data, which was the best data representation result-wise. WFT data classification performed worse for ResNet50 than for SimpleNet, possibly owing to the fact that ResNet50 is pre-trained with data without any time-dependence between the R, G, and B-channels. The relatively similar results contrast the relatively large difference in complexity between the networks. Thus, relatively low-complex CNNs can be used for radar gesture classification using the presented methods.

The results from validating trained models with test-only data (Range-Time and WFT) from the test group is also summarized in table 5.1. ResNet50 outperformed SimpleNet with Range-Time data, while SimpleNet outperformed ResNet50 with WFT data. The generalization abilities of the networks is from these results shown to be sub-par for accurate classification of gesture data from different individuals than those who provided data for model training. Important to consider is that the test-only set is relatively small, and thus it is hard to draw any conclusions with high confidence. These results would then be considered as an indication towards what a result with more data possibly would look like.

	SimpleNet	ResNet50
Data from author (averaged)		
Range-Time	95.0	97.9
Range-Doppler	85.0	84.6
WFT	93.0	90.6
Test-only data		
Range-Time	53.8	70
WFT	59.2	45.0

Table 5.1: Summary of classification results, as validation accuracy (%).

The goal of this thesis, to investigate the possibilities of hand gesture sensing and classification using an in-house pulsed radar setup and convolutional neural networks was thus accomplished, with validation accuracies in the mid-and upper-90% range for two of the three signal processing methods under test. Both spatial and reciprocal data were tested. The setup and classification method show great potential for pulsed millimeter-wave radar hand gesture recognition using a low-powered pulsed resonant-tunneling diode (RTD) wavelet generator, with future possibilities to refine results by for example modifying the setup or performing more extensive measurements.

Chapter 6
Outlook

While the setup used in this thesis is not considered low-powered, except for the wavelet generator, it serves as a proof-of concept for said generator. It would be of further value to investigate complete low-powered setups, possibly also implementing on-chip detection. On-chip detection of 60 GHz ISM band signals would be a field of large possibilities, where the limit of small implementation size as well as power consumption can be explored.

Regarding the radar operations, a higher acqusition rate or recording frame rate (PRF), would be attractive because of the increase if unambiguous velocity limit (see section 2.2.5). As the PRF increases, the unambiguous range decreases, but it would be of no concern regarding short-range detection due to the linear decrease and excessively long limit in the case of this thesis (≈ 1000 km for PRF=140 Hz, see section 4.1). Additionally, a shorter pulse length would yield a higher range resolution, which together with a higher possible measured velocity could prove more robust for hand gesture radar detection.

To further investigate the process of classifying radar gesture data, a larger data set would be attractive. Mainly, as mentioned in section 4.3.5, a larger number of participating individuals would be desirable due to the probable increase in model generalization ability. Additionally, tests could be performed in how much personal data a new user needs to provide to a pre-trained (on gesture data) model until classification yields acceptable results. Alternatively, investigations could be made into how much variation is needed for acceptable results on novel data, something achieved with training data augmentation for example.

Like discussed in section 4.3.4 it is of importance that the network complexity can be relatively low while still maintaining good classification results. Alternatives to CNN classification could yield less complex while still accurate classification. This could be processes where the data is analyzed as discrete signals, and not as spectrum images. For example, feeding pulsed signals to a RNN (Recurrent Neural Network) could yield good results due to the networks possession of a long-short-term-memory and ability and analyze time-dependent data [42]. Also, when performing transfer learning on `ResNet50` one could reduce the effective complexity by ”freezing” layers that does not need to train specifically on the provided data (for example the top CNN layers that detect overall and general patterns) while perform transfer learning on the layers that would need to be more unique for the provided data (the last layers in the network which performs very fine image analysis together with the fully connected layers). Additionally, as was observed in figure 4.4, some generalization issues exists for the WFT data in `ResNet50`, even though L2-regularization is implemented. This could be countered by adding drop-out or augmenting the images, for example by stretching along the x and y-axis and shearing of the image. This is an effective method against under-generalization [43].

When it comes to signal processing and data representation, an interesting analysis would be to classify WFT with more than three frames. This would introduce a more accurate time representation as well as more discrete velocities (see section 2.3.2 and 3.3). This would create the demand on > 3 frame input in CNNs, possibly utlizing CNNs with more than three channel (R, G, B) inputs. Additional investigation could also be made into combining WFT and Range-Time data in a complementary way for classification.

While this thesis deals with the technical aspects of radar gesture sensing, it would be interesting to evaluate the method with an industry perspective. As power, latency, device size and cost needs to be taken into account, it is important to look at all these factors and analyze the viability of this method of touchless interaction. With the performance derived in this thesis being as good as it is in some aspects, it would be feasible to continue researching the subject and other aspects needed for implementing a system of this kind.

References

[1] T. K. Sarkar and M. Salazar Palma. A history of the evolution of radar. In *2014 44th European Microwave Conference*, pages 734–737, Oct 2014.

[2] M. Skolnik. Role of radar in microwaves. *IEEE Transactions on Microwave Theory and Techniques*, 50(3):625–632, March 2002.

[3] Niraj Bhatta and M Geethapriya. Radar and its applications. *International Science Pres*, pages 1–9, 07 2016.

[4] Michael Inggs, R T. Lord, and WG VII. Current applications of imaging radar. *International Conference on Advanced Remote Sensing for Earth Observation Systems, Techniques and Applications*, 05 2019.

[5] Sergios Theodoridis Rama Chellappa. *Academic Press Library in Signal Processing, Volume 7.* Academic Press, 2018.

[6] Y. Kim and B. Toomajian. Hand gesture recognition using micro-doppler signatures with convolutional neural network. *IEEE Access*, 4:7125–7130, 2016.

[7] Heinz Willebrand. *Advantages of the 60 GHz frequency band and new 60 GHz backhaul radios.* Lightpointe, 2015.

[8] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Trans. Graph.*, 35(4):142:1–142:19, July 2016.

[9] P. Hügler, M. Geiger, and C. Waldschmidt. Rcs measurements of a human hand for radar-based gesture recognition at e-band. In *2016 German Microwave Conference (GeMiC)*, pages 259–262, March 2016.

[10] Md. Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Mahmudul Hasan, Brian C. Van Esesn, Abdul A. S. Awwal, and Vijayan K. Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *CoRR*, abs/1803.01164, 2018.

[11] B. Dekker, S. Jacobs, A. S. Kossen, M. C. Kruithof, A. G. Huizing, and M. Geurts. Gesture recognition with a low power fmcw radar and a deep convolutional neural network. In *2017 European Radar Conference (EURAD)*, pages 163–166, Oct 2017.

[12] Zhi Zhou, Zongjie Cao, and Yiming Pi. Dynamic gesture recognition with a terahertz radar based on range profile sequences and doppler signatures. *Sensors*, 18(1), 2018.

[13] L. E. Wernersson L. Ohlsson, P. Fay. Picosecond dynamics in a millimetre-wave rtd–mosfet wavelet generator. *Electronics Letters*, Vol. 51 No. 21 pp. 1671–1673, 2015.

[14] William A. Holm Mark A. Richards, James A. Scheer. *Perinciples of Modern Radars: Basic Principle*. SciTech Publishing, 2010.

[15] Mark A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005.

[16] Christopher Marki Ferenc Marki. *Mixer Basics Primer - A Tutorial for RF & Microwave Mixers*. Marki Microwave, 2010.

[17] C. Chen, S. Wu, S. Meng, J. Chen, G. Fang, and H. Yin. Application of equivalent-time sampling combined with real-time sampling in uwb through-wall imaging radar. In *2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pages 721–724, Oct 2011.

[18] Real-time versus equivalent-time sampling. `https://www.tek.com/document/application-note/real-time-versus-equivalent-time-sampling`. Accessed: 2019-05-07.

[19] Armin W Doerry. Balancing i/q data in radar range-doppler images. In *Radar Sensor Technology XIX; and Active and Passive Signatures VI*, volume 9461, page 94611Y. International Society for Optics and Photonics, 2015.

[20] R.N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill Kogakusha, Ltd., Tokyo, second edition, 1978.

[21] Gerald Kaiser. *A Friendly Guide to Wavelets*. Birkhäuser, 1994.

[22] Douglas A Lyon. The discrete fourier transform, part 4: spectral leakage. *Journal of object technology*, 8(7), 2009.

[23] Juergen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61, 04 2014.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, page 108. MIT Press, 2016. `http://www.deeplearningbook.org`.

[25] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[26] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[27] A. J. Thomas, M. Petridis, S. D. Walters, S. M. Gheytassi, and R. E. Morgan. On predicting the optimal number of hidden nodes. In *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 565–570, Dec 2015.

[28] Terrence Hinton, Jeffrey; Sejnowski. *Unsupervised Learning: Foundations of Neural Computation.* MIT Press, 1999.

[29] Martin A. Riedmiller. Advanced supervised learning in multi-layer perceptrons — from backpropagation to adaptive learning algorithms. *Computer Standards and Interfaces*, 16:265–278, 1994.

[30] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5:185–196, 06 1993.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[32] Andrew Y. Ng. Preventing "overfitting" of cross-validation data. In *In Proceedings of the Fourteenth International Conference on Machine Learning*, pages 245–253. Morgan Kaufmann, 1997.

[33] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[34] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML '04*, 2004.

[35] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[36] S. Heunisch, L. Ohlsson, and L. Wernersson. Reflection of coherent millimeter-wave wavelets on dispersive materials: A study on porcine skin. *IEEE Transactions on Microwave Theory and Techniques*, 66(4):2047–2054, April 2018.

[37] Flann 25240 standard gain horn antenna data sheet. `http://flann.com/wp-content/uploads/2016/01/Series-240.pdf`. Accessed: 2019-04-23.

[38] Hxi hlnav-383 data sheet. `http://www.hxi.com/Datasheets/HLNAV-383%20Data%20Sheet.pdf`. Accessed: 2019-04-23.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[40] An overview of resnet and its variants. `https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035`. Accessed: 2019-05-14.

[41] Peter I. Frazier. A tutorial on bayesian optimization, 2018.

[42] Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018.

[43] Data augmentation | how to use deep learning when you have limited data — part 2. `https://medium.com (link)`. Accessed: 2019-05-27.

Appendix A

Appendix

A.1 Results of Alternative Classification Methods

Various other methods of data classification are here superficially tested for comparison. The methods used was those present in the Matlab "Classification Learner" application[1].

The input data is the large data set from the authors and is in all cases the Range-Time spectrum, downsized to a 25x25x1 pixel image and flattened to a 25x25x1 long vector. 75%-25% training-validation is used. The validation accuracy of the three best performing classification methods are presented in table A.1.

Method	Validation Acc. (%)
Medium Gaussian SVM	67.0
Linear SVM	64.1
Quadratic SVM	63.5

Table A.1: Validation accuracy for alternative classification methods. "SVM": Support Vector Machine.

[1] `https://au.mathworks.com/help/stats/classificationlearner-app.html`

A.2 Classification of Data Divided Between Authors

`SimpleNet` is here trained on data from one author and validated on data from the other. There are 90 measurements per gesture per author. The settings are used as in section 4.3.3, but here with 50%-50% training-validation (one author for training, one for validation). The validation results for classification on Range-Time and WFT data are presented in figure A.1 table A.2. The results are averaged over eight training sessions.

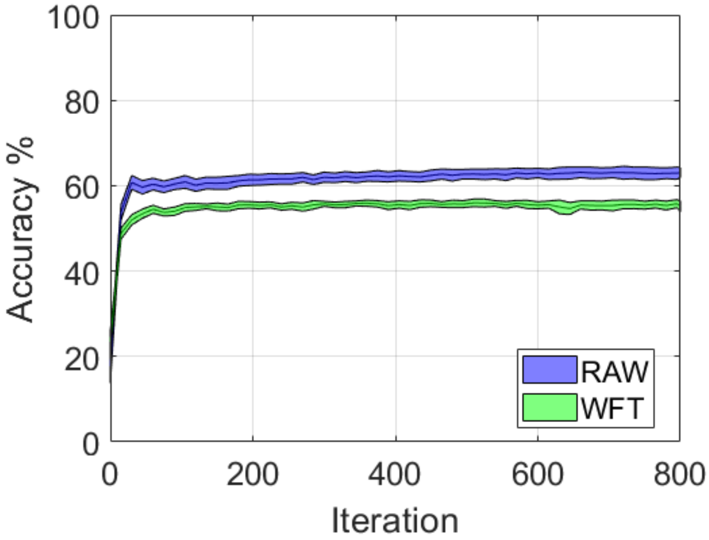
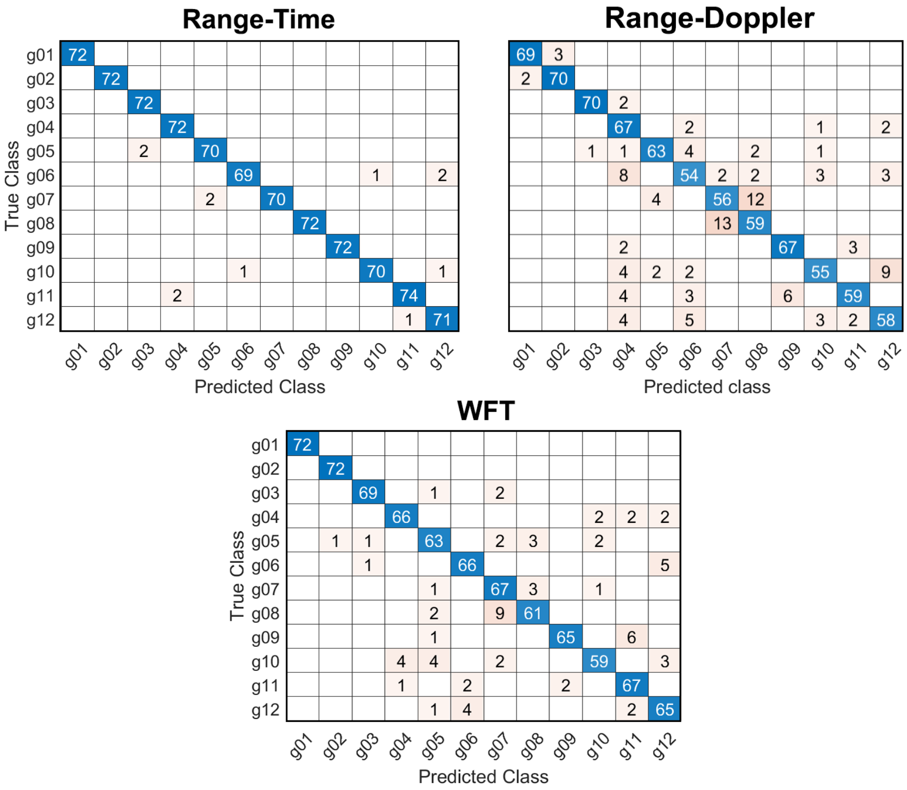


Figure A.1: Validation accuracy (%) for `SimpleNet` (author vs author, averaged). The results are averaged for author 1 vs author 2, and author 2 vs author 1.

	Range-Time	WFT
Author 1 train, Author 2 val.	63.8	54.4
Author 2 train, Author 1 val.	62.3	55.9

Table A.2: Validation accuracy (%) for `SimpleNet` (author vs author, averaged).

A.3 Confusion Matrices from ResNet50



Range-Time

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12
g01	72											
g02		72										
g03			72									
g04				72								
g05			2		70							
g06						69				1		2
g07					2		70					
g08								72				
g09									72			
g10						1				70		1
g11				2							74	
g12											1	71

Range-Doppler

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12
g01	69	3										
g02	2	70										
g03			70	2								
g04				67		2				1		2
g05			1	1	63	4		2		1		
g06				8		54	2	2		3		3
g07					4		56	12				
g08							13	59				
g09				2					67		3	
g10				4	2	2				55		9
g11				4		3			6		59	
g12				4		5				3	2	58

WFT

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12
g01	72											
g02		72										
g03			69		1		2					
g04				66						2	2	2
g05		1	1		63		2	3		2		
g06			1			66						5
g07					1		67	3		1		
g08					2		9	61				
g09					1				65		6	
g10				4	4		2			59		3
g11				1		2			2		67	
g12					1	4					2	65

Figure A.2: Confusion Matrices for ResNet50 classification. Range-Time (top left), Range-Doppler (top right), WFT (bottom).

A.4 Additional Representations of Measurement Data

Figure A.3, A.4 and A.5 shows data representations from the same measurements presented in figure 4.3. Shown here is Range-Time data with red lines indicating result from cropping (left column), WFT data from cropped raw data (middle column), Range-Doppler data (right column). Observe the indication of 0 velocity in the middle and right column.

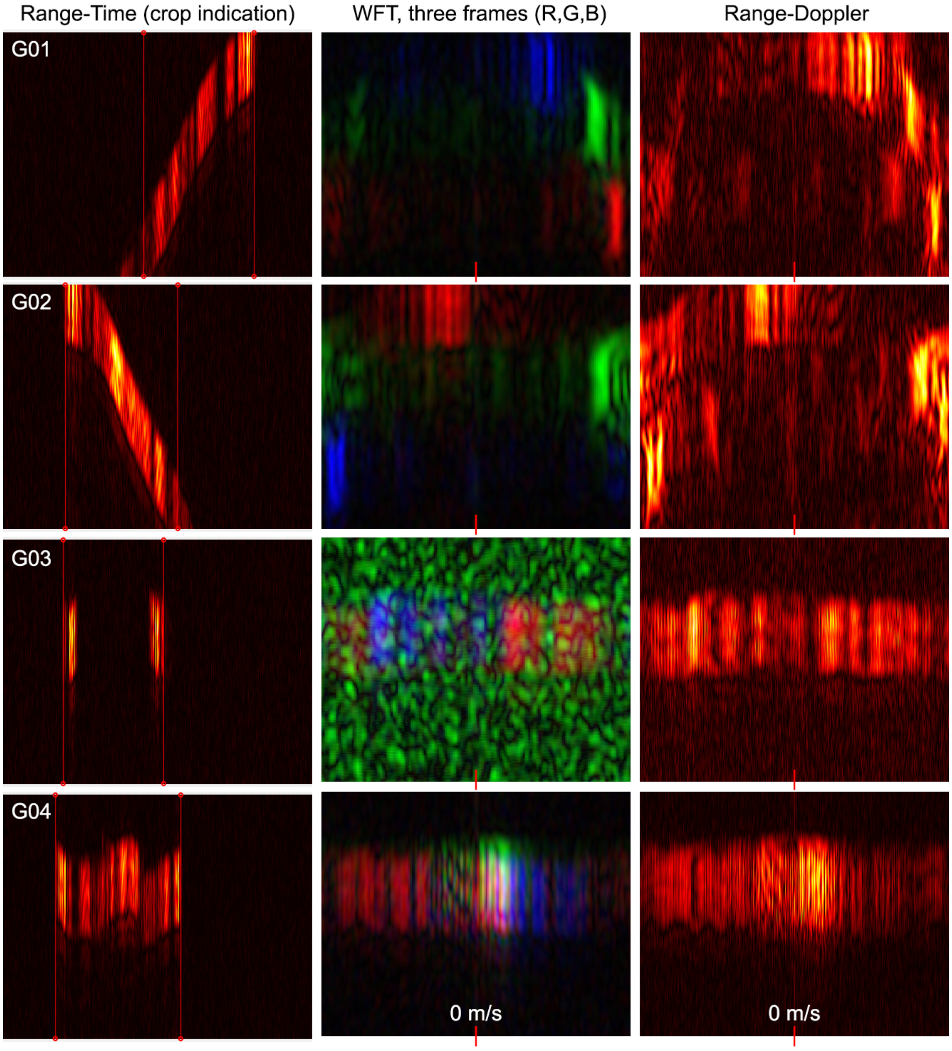


Figure A.3: Range-Time, WFT, Range-Doppler data. G01-G04.

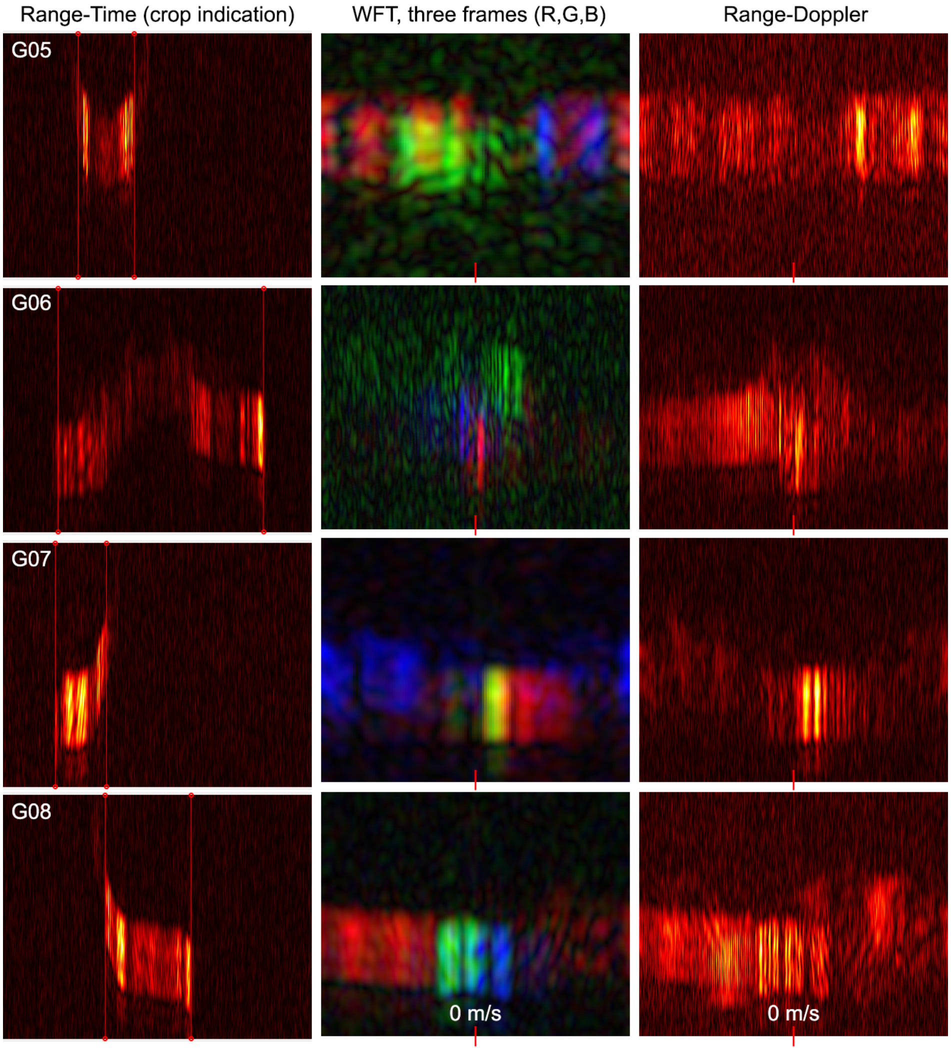


Figure A.4: Range-Time, WFT, Range-Doppler data. G05-G08.

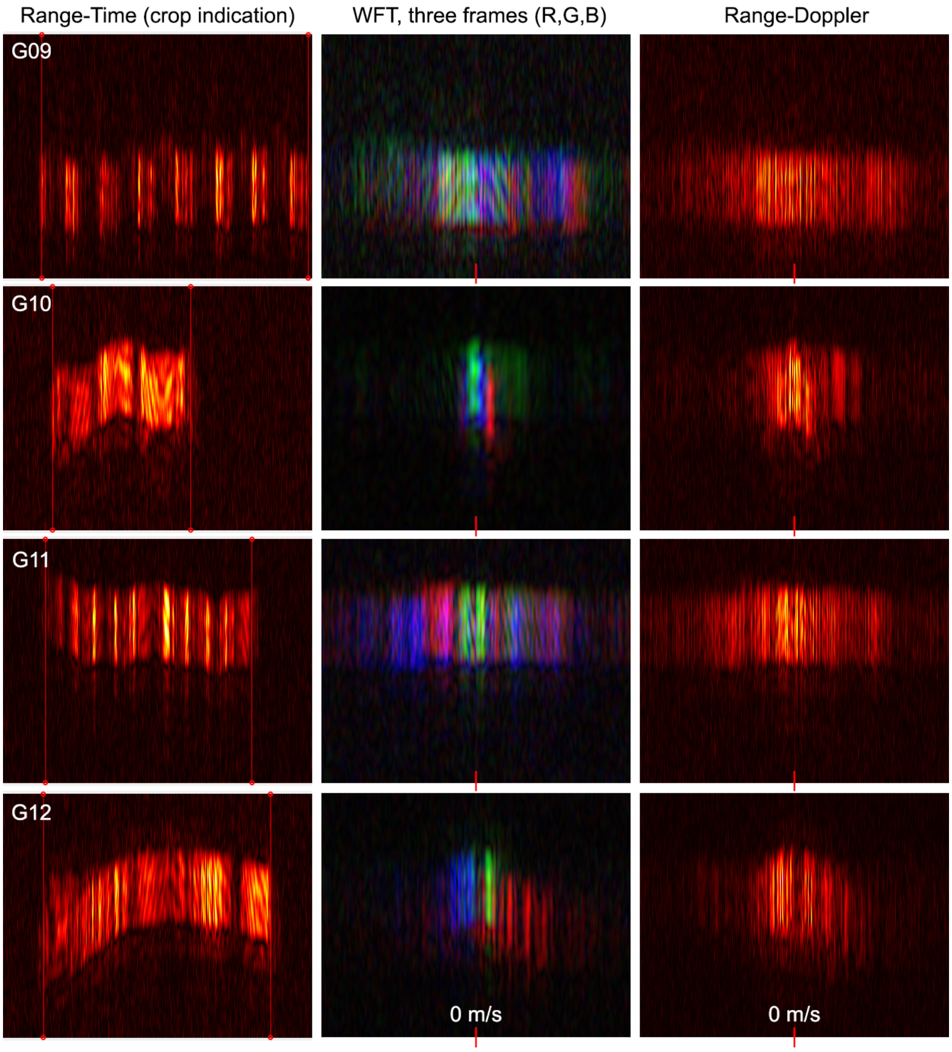


Figure A.5: Range-Time, WFT, Range-Doppler data. G09-G12.

LUND
UNIVERSITY

Series of Master’s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2019-714
http://www.eit.lth.se

Printed by Tryckeriet i E-huset, Lund 2019