BASKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONICS ENGINEERING

OBJECT CLASSIFICATION ON NOISE REDUCED AND DATA AUGMENTED MICRO-DOPPLER RADAR SPECTROGRAMS

BY

ALPEREN ERDOĞAN

MASTER OF SCIENCE THESIS

ANKARA - 2021

BASKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONICS ENGINEERING

OBJECT CLASSIFICATION ON NOISE REDUCED AND DATA AUGMENTED MICRO-DOPPLER RADAR SPECTROGRAMS

BY

ALPEREN ERDOĞAN

MASTER OF SCIENCE THESIS

ADVISOR

ASSOC. PROF. SELDA GÜNEY

ANKARA – 2021

BAŞKENT UNIVERSITY

INSTITUTE OF SCIENCE AND ENGINEERING

This study, which was prepared by Alperen Erdoğan, for the program of Electrical and Electronics Engineering, has been approved in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in Electrical and Electronics Engineering Department by the following committee.

Date of Thesis Defense: 12 / 08 / 2021

Thesis Title: Object Classification on Noise Reduced and Data Augmented Micro-Doppler Radar Spectrograms

Examining Committee Members	**Signature**
Chairman: Prof Mustafa DOĞAN	……
Member (Supervisor): Assoc Prof. Selda GÜNEY	……
Member: Prof. Hamit ERDEM	……

APPROVAL

Prof. Ömer Faruk ELALDI
Director, Institute of Science and Engineering
Date: … / … / 2021

BAŞKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ ÇALIŞMASI ORİJİNALLİK RAPORU

Tarih: … /08/2021

Öğrencinin Adı, Soyadı: Alperen ERDOĞAN

Öğrencinin Numarası: 21910475

Anabilim Dalı: Elektrik/Elektronik Mühendisliği Anabilim Dalı

Programı: Elektrik/Elektronik Mühendisliği Tezli Yüksek Lisans Programı

Danışmanın Unvanı/Adı, Soyadı: Dr. Öğr. Üyesi Selda GÜNEY

Tez Başlığı: Gürültüsü Azaltılmış ve Veri Seti Geliştirilmiş micro-Doppler Radar Spectrogramları ile Nesne Sınıflandırılması

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 123 sayfalık kısmına ilişkin, 08/07/2021 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %8'dir. Uygulanan filtrelemeler:

1. Kaynakça hariç

2. Alıntılar hariç

3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

"Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını" inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:

ONAY

Tarih: … / … / ……..

Öğrenci Danışmanı Unvan, Ad, Soyad, İmza:

Dr. Öğr. Üyesi Selda GÜNEY

Dedicated to my lovely cat…

Alperen ERDOĞAN

Ankara – 2021

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my thesis advisor Assoc. Prof. Selda Güney, for always guiding me during my both bachelor and master degree, for her supports and advices. Her valuable guidance helped me during the whole period of my research.

I would like to acknowledge Prof. Hamit Erdem for inspiring my interest in the research area of Neural Networks, thoughtful comments, and recommendations.

From bottom of my heart, I would like to express my deepest appreciation to Assoc. Prof. Radim Burget for helping me by answering whole of my questions and extending my vision.

I wish to acknowledge the help provided by the technical and support staff in the Faculty of Electrical and Communication of the Brno University of Technology.

I also had great pleasure of working with Ph.D student Justyna Skibińska. She illuminated my way and helped me to finalize my thesis.

I gratefully acknowledge to Electrical-Electronics Engineer Yusuf Ergün, Mert Ergün, Melih Aktaş and Software Engineer Burak Özkişi, Roman Suntcov whose help cannot be overestimated.

I would like to thank to Pink Floyd band, my friends are living/had lived in Brno University of Technology A03 dormitory, my childhood friend Kuzey Erdem and Lionel Messi, who uplifted me every time I got stuck during my thesis study and helped me progress in my thesis.

ABSTRACT

ALPEREN ERDOĞAN

OBJECT CLASSIFICATION ON NOISE REDUCED AND DATA AUGMENTED MICRO-DOPPLER RADAR SPECTROGRAMS

Başkent University Institute of Science

Department of Electrical and Electronics Engineering

2021

The classification of targets is one of the most challenging tasks in radar signal processing. Classifying a target can help radar operators figure out the nature of the target, such as its source and activity. However, it is very difficult to find the labeled data necessary to develop radar target classification models. Generating a radar dataset is an expensive and time-consuming process.

To address these issues, we propose a noise reduction method that can be applied to micro-Doppler radar datasets. This method is carried out by averaging the spectrograms of each class in the RadEch micro-Doppler radar datasets and subtracting pixel by pixel from each sample. RadEch dataset has also been augmented with traditional and learning-based data augmentation methods. The learning-based data augmentation method was carried out by using Generative Adversarial Networks.

Raw spectrograms, augmented spectrograms and noise reduced spectrograms have been classified using 5-layer CNN, VGG-16, and VGG-19. Classification results are compared with state-of-art studies. Comparison results shows that classification on noise reduced spectrograms performs better than current state-of-art methods.

KEYWORDS: Micro-Doppler Radar, Radar Signal Classification, Data Augmentation, Noise Reduction, Convolutional Neural Network, Transfer Learning

ÖZET

ALPEREN ERDOĞAN

GÜRÜLTÜSÜ AZALTILMIŞ VE VERİ SETİ ARTTIRILMIŞ MİCRO-DOPPLER RADAR SPEKTROGRAMLARI İLE NESNE SINIFLANDIRILMASI

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik/Elektronik Mühendisliği Anabilim Dalı

2021

Nesnelerin sınıflandırılması, radar sinyal işlemedeki en zorlu görevlerden biridir. Bir hedefi sınıflandırmak, radar operatörlerinin, nesnenin kaynağı ve etkinliği gibi hedefin doğasını anlamalarına yardımcı olabilir. Ancak radar nesne sınıflandırma modelleri geliştirmek için gerekli olan etiketli verileri bulmak çok zordur. Bir radar veri seti oluşturmak pahalı ve zaman alıcı bir süreçtir.

Bu sorunları ele almak için mikro-Doppler radar veri setlerine uygulanabilecek bir gürültü azaltma yöntemi bu tez kapsamında önerilmiştir. Bu yöntem, RadEch micro-Doppler radar veri setlerindeki her sınıfın spektrogramlarının ortalaması alınarak ve bu ortalama spektrogramın, sınıflarda bulunan her örnekten piksel piksel çıkarılması ile gerçekleştirilir. RadEch veri kümesi ayrıca geleneksel ve öğrenme tabanlı veri geliştirme yöntemleriyle de zenginleştirilmiştir. Öğrenmeye dayalı veri artırma yöntemi, Çekişmeli Üretici Ağlar kullanılarak gerçekleştirilmiştir.

Ham spektrogramlar, geliştirilmiş spektrogramlar ve gürültüsü azaltılmış spektrogramlar beş katmanlı evrişimsel sinir ağı, VGG-16 ve VGG-19 kullanılarak sınıflandırılmıştır. Sınıflandırma sonuçları, literatürde yapılmış alan son çalışmalarla karşılaştırılmıştır. Karşılaştırma sonuçları, gürültüsü azaltılmış spektrogramlar üzerinde yapılan sınıflandırma başarısının, mevcut en son yöntemlerden daha iyi bir performansa sahip olduğunu göstermiştir.

ANAHTAR KELİMELER: Mikro-Doppler Radar, Radar Sinyallerinin Sınıflandırılması, Veri Arttırma; Gürültü Azaltma, Evrişimsel Sinir Ağları, Öğrenme Aktarımı

TABLE OF CONTENTS

LIST OF FIGURES

Figure 2. 1 Radar Block Diagram 4
Figure 2. 2 Typical Radar Displays [2] 6
Figure 2. 3 Basic Envelope of Receiving Pulse Radar Signal [2] 8
Figure 2. 4 Pulse Doppler Radar Illustration 11
Figure 2. 5 Principle of Triangular FMCW Ranging [4] 13
Figure 2. 6 Doppler Effect Simulation 15
Figure 2. 7 Micro-Doppler Signature of Simulated ALCM [7] 18
Figure 2. 8 Human Walking Movement in One Cycle [7] 20
Figure 2. 9 Radial Velocity of Human Parts While Walking [7] 21
Figure 2. 10 (a) The orientation of a walking human in relation to a radar pulse, (b) the radar pulse-range profile, (c) and a walking human's micro-Doppler signature. [7] 22
Figure 2. 11 (a) A walking person's micro-Doppler signature, (b) A running person micro-Doppler Signature, and (c) signature of micro-Doppler [7] 23
Figure 2. 12 Machine Learning Approach 24
Figure 2. 13 CaffeNet Architecture 27
Figure 2. 14 Basic CNN Architecture [76] 30
Figure 2. 15 Generative Adversarial Network Architecture [78] 34
Figure 2. 16 GAN, Variational Autoencoder (VAE), Deep Convolutional GAN (DCGAN) and Wasserstein GAN (WGAN) Results on MNIST Dataset. 35

LIST OF TABLES

LIST OF ABBREVIATIONS

ALCM	Air-launched cruise missile
ANN	Artificial Neural Network
BM	Bayesian Models
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRT	Cathode-ray Tube
CW	Continuos Wave
DCGAN	Deep Convolutional GAN
DGM	Deep Generative Models
DL	Deep Learning
DNN	Deep Neural Network
DT	Decision Tree
EM	Electromagnetic
FMCW	Frequency Modulated Continuous Wave
FN	False Negative
FOPEN	Foliage penetration
FP	False Positive
GAN	Generative Adversarial Network
GHz	Gigahertz
GPU	Graphics Processing Unit
HF	High Frequency
IF	Intermediate Frequency
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
KHz	Kilohertz
LNA	Low Noise Amplifier
LO	Local Oscillator
MHz	Megahertz
ML	Machine Learning
PCA	Principle Component Analysis
RADAR	Radio Detecting and Ranging
RCS	Radar Cross Section
RF	Radio Frequency
SVM	Support Vector Machine
TFD-SVD-FT	Time-frequency distribution/singular values decomposition/Fourier transform
TN	True Negative
TP	True Positive
UHF	Ultra High Frequency
VAE	Variational Autoencoder
VHF	Very High Frequency
WGAN	Wasserstein GAN

1. INTRODUCTION

The classification of targets is one of the most challenging tasks in radar signal processing. Classifying a target can help radar operators figure out the nature of the target, such as its source and activity. The original classification of targets was done by listening to the audio signals that were received by the radar. A moving target produces unique sounds caused by the Doppler effect, which trained operators can recognize. Nonetheless, it introduces an inconsistency and a slow classification, making it unsuited to real-time classification. For the military, robust, reliable, and modular targeting algorithms are required to increase efficiency and performance. With increasing use of automatic target recognition algorithms, humans will serve less and less in tasks such as target recognition. Automatic target recognition algorithms not only provide reliable and objective learnings but also serve as a cost-effective alternative to human participation. Due to radar's ability to suppress stationary targets and minimize clutter, Doppler radar is widely used to detect, recognize, and classify moving targets. A moving target introduces a time-varying delay between the radar and the target which causes frequency shifts. A Doppler shift is the major change in frequency caused by a subject's main movement. By adjusting these shifts, micro-Doppler signatures can be created that describe the nature and activity of the target.

1.1. Objectives

In this thesis, RadEch dataset is used for classification. The micro-Doppler radar signals in the RadEch dataset were first converted to spectrograms. Spectrograms have been adapted to the Convolutional Neural Network to be used for classification. Then, the noise reduction technique proposed in this study was applied to the spectrograms. Then, traditional and learning-based data augmentation techniques were applied to Raw spectrograms in order to increase number of the dataset samples. After all these pre-processing, the obtained pre-processed data were classified using three different Convolutional Neural Networks with and without transfer learning.

1.2. Contributions

The main contribution of this thesis is that it has higher classification accuracy with lower demand of the amount of training data than state-of-art studies in the literature. Additionally, a noise reduction method for micro-Doppler radar signals is proposed. It has been observed that the method exceeds the classification accuracy obtained in other studies in the literature. The

performance of VGG-16 and VGG-19 architectures on micro-Doppler radar spectrograms has been tested and compared with and without transfer learning. Traditional data augmentation methods and Learning-Based data augmentation methods using Generative Adversarial Network were compared.

1.3. Thesis Structure

This section describes the rest of thesis structure:

a. Chapter 2 (Literature Review): This chapter defines the necessary concepts that is needed in this thesis and gives a survey of other works which is related to the filed
b. Chapter 3 (Methodology): This chapter clarifies and describes what is contributed to micro-Doppler radar signal classification.
c. Chapter 4 (Results): This chapter shows and describes the outcomes and results that is acquired from the model that is developed.
d. Chapter 5 (Discussion): This chapter summarizes the work and the results of the project. This chapter also compares all results in each other and other study. Future works are also mentioned here.
e. Chapter 6 (Conclusion): This chapter explained briefly what have been done in this thesis and show results in a nutshell.

2. LITERATURE REVIEW

2.1. Radar Systems

Radar is a type of electromagnetic device used for detecting and locating objects. Initially, radar systems were limited to detecting targets and determining their distance. The word radar is derived from the phrase **RA**dio **D**etection **A**nd **R**anging. An electromagnetic energy is transmitted by radar - such as pulsed sine waves - and echoes are collected to provide range, velocity, position, and other target information such as reflectivity. In this chapter, radar systems are defined detailed. Basic radar block diagram is illustrated and every component in the block diagram are explained briefly. Radar equation, radar range ambiguity, radar range resolution, velocity measurement is expressed briefly.

Figure 2.1 illustrates an elementary example of a typical radar system, which omits many details. This system uses an oscillator to generate electromagnetic energy, which is sent to a microwave antenna via a duplexer, a device that simultaneously transmits and receives the signals. As a transducer, the antenna couples electromagnetic energy into free space, where it propagates at the speed of light (approximately 3×10^8 m/s). Radar antennas recover some of the backscattered energy reflected by objects. According to Equation 2.1, the time delay between the transmission by the radar and the reception of the returned echo at a range R is the round-trip time.:

$$t_d = \frac{2R}{c} \tag{2.1}$$

Where speed light c. The R which represents range is multiplied with 2 due to two-way propagation of radar signal.

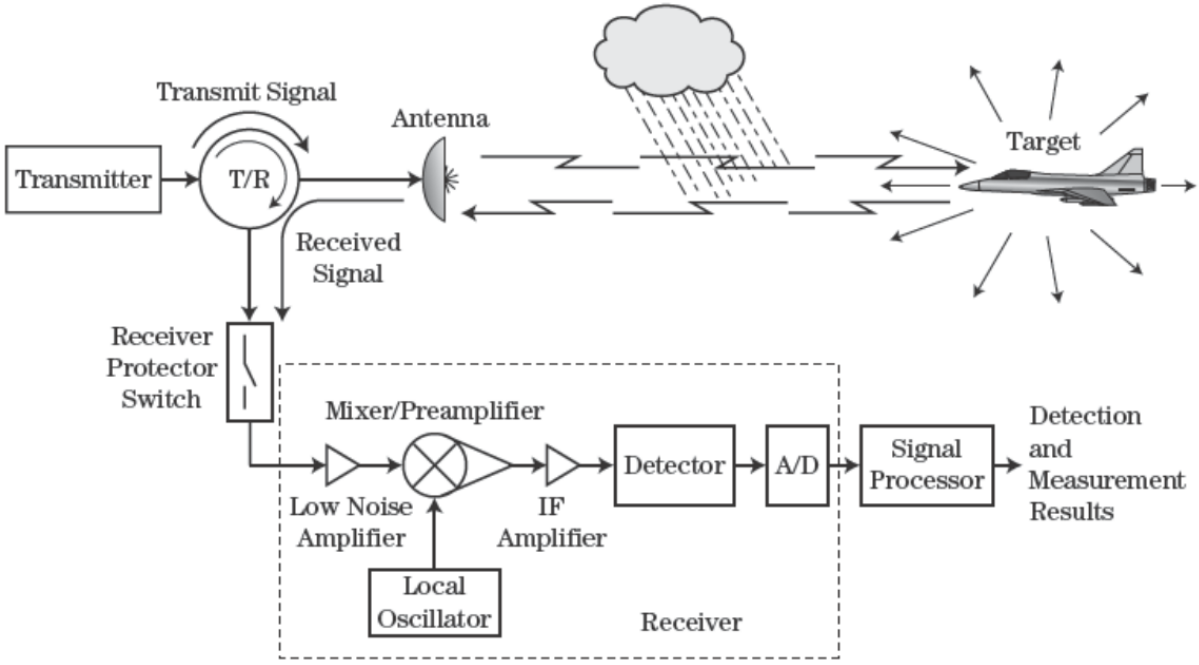

Figure 2. 1 Radar Block Diagram

Receivers are typically equipped with Low Noise Amplifiers (LNA) for amplifying weak reflected radar signals and reducing noise. The receiver also contains a filter-to-filter unnecessary signal in receiving band and noise. Incoming-signals are filtered by the filter and passed to the low noise amplifier so that they can pass the desired bandwidth. Down converters that inside of the transmitter, make radar signals more suitable for signal processing. After required signal processing, indicator shows meaningful information to radar operators. A shift in the carrier frequency of the reflected wave, also known as Doppler shift, can be used to calculate relative velocity if there is relative movement between the radar and the target. Doppler shift and Doppler radar are explained in section 2.1.3 detailly.

There are five basic elements of basic radar systems: a duplexer, a transmitter, an antenna, a receiver, and an indicator. The functions of basic elements are explained in below.

Transmitter: Radar systems have transmitters as a major component, and it produces the radio frequency (RF) power signal to propagate the target. A continuous wave (CW) signal or a pulse signal may be generated in the RF transmitter. An effective radar system will use it with an amplitude and frequency that fit its specific requirements. RF power can be generated in two ways: first method is in the power oscillator approach, where the signal is generated at a sufficiently high level to be applied directly to the antenna; and second method in the master oscillator, where RF signals at a low level are amplified to a higher level [2].

Duplexer: Duplexers allow monostatic radars to use the transmission and reception on the same antenna. Depending on the radar system, two antennas could be used: one for transmitting and one for receiving. This kind of radar system does not required duplexer. Transmission and reception of RF signals via duplexers are usually accomplished by switching between them [2].

Receiver: Basically, the receiver consists of low-noise radio frequency amplifier, intermediate frequency amplifier (IF) and display unit. RF amplifiers on the front end are usually parametric amplifiers or low noise transistors. An IF signal of centre frequency 30 or 60 MHz and bandwidth of about 1 MHz is produced by converting the RF signal to an IF signal using a mixer and local oscillator (LO). Second detector extracts pulse modulations at intermediate frequency, which are matched by the intermediate frequency amplifier. After the signal has been demodulated, it is then amplified by the video amplifier so that it can be displayed on an indicator, usually a cathode-ray tube (CRT) [2].

Antenna: Antennas used in the radar system are mostly directional. Due to the fact that directional antennas propagate signals out to areas of interest. The signal propagates in both vertical and horizontal directions with an isotropic antenna. As a result, antenna gains are weaker and meaningful information is more difficult to obtain. Parabolic dish antennas are fed from a feeding antenna at their focus, which is a common form of radar antenna. By mechanically pointing the antenna, the beam can be scanned in space. Phased-array antennas have also been used for radar. The antenna beam is scanned electronically using phase shifters connected to the elements in a phased-array antenna. By concentrating the signal, the antenna can narrow the beam to a particular preferred direction, and intercept the target echo signal from the same direction. Amplification of the weak energy is then accomplished by using a low noise receiver stage [2].

Indicator: A radar indicator or display visually presents the information contained in the radar echo signal in a way that can be interpreted and acted upon by the operator. You can either directly connect the display to the video output of the radar receiver or you can indirectly connect the display to the radar receiver. The radar receiver output of radars that provide more information per second than the operator can process is normally interpreted and compressed by automatic data processors. Radar displays are typically two-dimensional displays that show the target's location in relation to some reference point. Nonetheless, it can also take the form of a light which indicates a particular status, or a meter showing some value, such as a target's distance or antenna angle. A variety of some display formats are shown in Figure 2.2. For

surveillance radars and tracking radars, target information is displayed on CRT displays. Some of the commonly used radar indicators defined below:

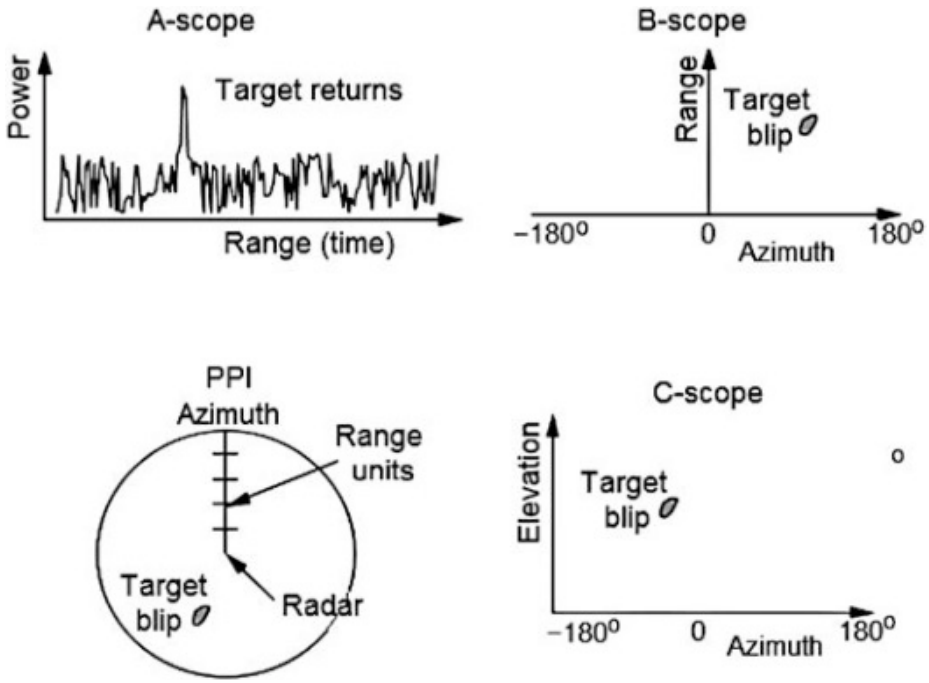

Figure 2. 2 Typical Radar Displays [2]

a. ***A-Scope*****:** It provides target range and signal power data. Deflector vertically deflects according to the target echoes' power, and horizontally in proportion to the range. Radars used to measure or collect data with the A-scope display do not employ the antenna to scan. It is also known as the "R-Scope indicator".

b. ***B-Scope*****:** Basically, it has a rectangular display with degrees indicated by horizontal coordinates and ranges indicated by vertical coordinates. Displays that display B-scope images are used extensively in ground and airborne radar systems, as well as short-range ground surveillance radars.

c. ***C-Scope:*** In this rectangular display, vertical coordinates represent elevation angles, while horizontal coordinates represent azimuth angles.

d. ***E-Scope:*** The rectangular display is either horizontally aligned with the range or vertically aligned with the elevation angle.

e. ***Plan-position-indicator (PPI):*** A circular or polar display shows reflections from reflected targets shown by vector coordinates, and fields and angles indicated by polar

coordinates. It appears that maps are like these displays. Plan-position indicators are used in surveillance and weather radar applications to display 360-degree coverage.

Generally, radars are classified as to the types of waveforms they send and, when they are pulsed, by the rate of transmission. Radiation from continuous wave radars is transmitted and received continuously using separate transmit and receive antennas in general. continuous wave is a high-energy waveform and could be used for long-range applications. Radial velocity and angular position may be accurately measured by unmodulated continuous wave radars using the Doppler shift of the reflection signal. Doppler shift is explained in section 2.1.3 detailed. The range information can be acquired by using time-variant modulation, such as saw-tooth modulation or triangular modulation.

The types of radar systems may also be classified according to specific missions and applications. A variety of missions can be carried out by radars, such as air traffic control, radar for aircraft, radar for satellites, remote sensing, and radar for law enforcement for traffic control.

There can also be a difference in radar classification based on the frequency of the radar. International institutions and organizations have named the frequency ranges of radio frequencies. This naming process has made things much easier in many projects and systems. The band names, it's frequency ranges and general usage are illustrated in Table 2.1.

Radar signals that are reflected by targets are commonly corrupted by unwanted external interference such as thermal noise, electromagnetic interference, atmospheric noise, etc. Whenever the desired signal exceeds the interfering signals, it is possible to acquire information about the target. An echo signal's delay contains information about the range of the returned signal.

Table 2.1 Radar Frequency Bands and Usages [2]

Band	**Frequency (GHz)**	**Usage**
HF	0.003-0.03	Over-the-horizon surveillance
VHF	0.03-0.3	Long range surveillance
UHF	0.3 -1.0	Very long-range surveillance
L	1.0-2.0	Long range military and air traffic control
S	2.0-4.0	Ground-based shipboard search
C	4.0-8.0	Fire control and weather detection radars
X	8.0-12.5	Short range tracking, marine radars

K_u	12.5-18.0	High resolution mapping, satellite altimetry
K	18.0-26.5	Police speed measuring and airport surface detection
K_a	26.5-40.0	Very high-resolution mapping and airport surveillance
MM-Wave	40.0-300.0	Optical targeting system and Laser range finders

By varying the frequency shift, also known as the Doppler frequency, the target can be identified by its radial velocity. Azimuth and elevation of the radar target can be inferred from the direction in which the antenna is pointing for maximum reflected signal strength.

Detection: The main function of radar is to detect whether the target exist or not. The radar cannot detect the weak signal due to the noise energy. Basically, interference and thermal noise corrupt the receiver signal. The majority of noise occurs in regions with low signal levels, such as weak echo signals that corrupt the desired signal in receivers. "The minimum detectable signal" is the weakest signal the radar receiver can detect. In threshold detection, the assumption is made that an object is present when a signal rises above a particular level to ensure that the envelope does not exceed the threshold if only noise is present, the threshold level should be set at an appropriate level.

Range: Range is one of the most important parameters of radar systems. In most cases, the radar's chief feature is its ability to measure the reflection time of its signal in order to determine its range to a target. Around feasible range, the precision of range could be approximately a few centimetres. Timing marks can either be short pulse signals (amplitude modulated signals) or frequency/phase modulated continuous waves. Range measurement's accuracy depends on bandwidth of the radar signal. For more accurate results, bandwidth needs to be wider. This means that the radar signal bandwidth is one of the most important parameters for range accuracy [3]. Pulse radar will be investigated in Section 2.1.1, however, there must be some basic explanation of it. Figure 2.3 shows basic envelope of receiving signal of pulse radar.

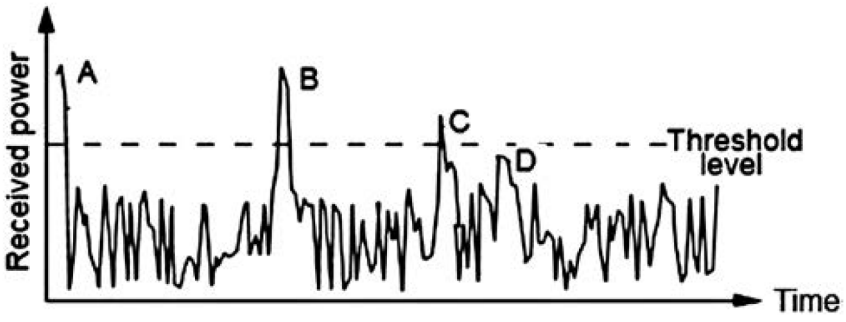

Figure 2. 3 Basic Envelope of Receiving Pulse Radar Signal [2]

Target A has been detected, and it's not a big deal because the receiver's output power is above threshold at targets A and B. It is apparent that the noise voltage accompanying the signal at C is large enough to comfortably exceed the threshold, but target D is not actually detected as a target due to the level of noise [2].

Radar Cross Section: An area equivalent to what is seen by a radar is called a radar cross section. In other words, it is the symbolic area intercepting that amount of power distributed evenly throughout all directions, which produces the same echo at the radar as the target does [4]. Mathematically, it is written as Equation 2.2:

$$\sigma = \lim_{R\to\infty} 4\pi R^2 \frac{|E_S|^2}{|E_i|^2} \tag{2.2}$$

Where:

R: target and radar distance

E_S : radar scattering strength

E_i : strength of incident field at target

Range Equation: Wave propagation from transmitter to receiver is represented by radial equations as a result of transmission power. The radar equation can be used to calculate the power returning to the receiving antenna given the transmit power, the slant range, and the reflecting characteristics of the aim. By knowing the radar receiver's sensitivity, the radar equation can determine the theoretical maximum range that can be achieved by a given radar. As well, the radar equation can be used to estimate a radar system's performance [8]. Radar equations assume electromagnetic waves propagate under ideal circumstances, using an isotropic antenna, and can be modified to fit specific conditions. In the scope of this thesis, the most general version of the radar equation used in distance calculations will be given. Other specific equations, equations involving the shape of the earth and the effects of the atmosphere will not be given. The radar equation is given in Equation 2.3 below.

$$R = \sqrt[4]{\frac{P_T G_T G_R \lambda^2 \sigma}{(4\pi)^3 S_{min}}} \tag{2.3}$$

R: range

P_T : transmitted power

G_T: transmit antenna gain

G_R: received antenna gain

λ : radar signal wavelength

σ : radar cross-section

S_{min} : minimum detectable signal

Range Resolution: Radar's target resolution capability, it is capable of discriminating targets that are very close to it in either range or bearing. In weapons-control radars, for instance, which require high precision, targets can be distinguished by only yards or meters. Radar search is usually less precise and having ability to distinguish between targets only a few meters or even kilometres apart.

Radar systems are capable of distinguishing between two or more targets at different ranges on the same bearing. There are various factors that contribute to the degree of range resolution, including pulse width, type and size of target, and receiver and indicator efficiency. Range resolution is primarily determined by pulse width. Radar systems that have been well-designed are capable of separating targets separated by one half the pulse width duration τ. Based on Equation 2.4, we can determine the theoretical range resolution cell of radar system:

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

where,

ΔR: range resolution in meter

c: propagation velocity of electromagnetic wave in m/s

τ: bandwidth of pulse signal

In this section, the definition of radar systems is defined and all components in a simple radar structure are briefly explained. Classification parameters of radar systems, the frequencies used in radar systems and their intended use were shown. Radar equation was given, and radar range resolution was explained. In the next section, pulse doppler radars will be briefly explained. Since Doppler radar is used in the dataset used within the scope of this thesis, the doppler radar is going to be explained in detail in Section 2.1.3.

2.1.1. Pulse Doppler Radar

An intense and short pulse is propagated by pulse radar. Receives the echo signals during the silent period. Transmitter power is shut off as soon as a measurement is completed in

continuous wave radar, unlike continuous wave radar. The radar pulse modulation used in this method is characterized by very short transmission pulses. Generally, transmit pulse is durations of $\tau \approx 0.1, \ldots, 1\ \mu s$. Between the transmit pulses are very large pulse pauses $T >> \tau$, which are referred to as the receiving time and typically $T \approx 1$ millisecond as shown in Figure 2.4.

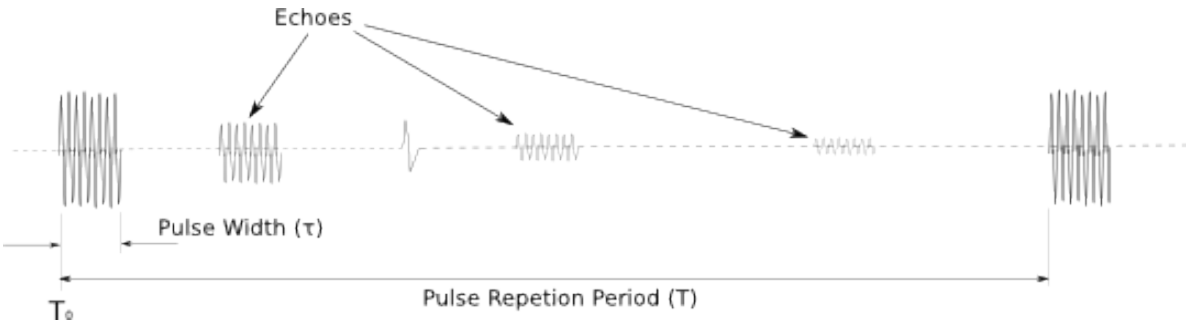

Figure 2. 4 Pulse Doppler Radar Illustration

Reflecting objects are measured at runtime in order to determine how far they are from one another. The pulse radar transmits relatively high pulse power and is primarily used for long-range measurements. Pulse radars are similar to bats' techniques for detecting obstacles. bats send a sound signal to find obstacles in front of them. Based on the duration of the incoming sound signal, they can determine the distance to the target based on the reflected sound signals; pulse radars also work with this principle.

The pulse radar differs from other radar methods because all processes within it must be controlled in time. Time is determined by the leading edge of the transmitted pulse during runtime measurement. When the echo signal's rising edge transitions to the pulse top, the pulse top ends. Calculating the distance requires correcting signal processing delays. Pulse radars are influenced by random deviations.

2.1.2. Continuous Wave Radar

Radar's first and modern versions are based on transmitting a continuous wave of electromagnetic energy and then receiving that reflected back by a moving target. Radar signals are shifted by an amount called Doppler shift if the target is changing spatial location relative to the radar, which is the basis for continuous wave radar. Besides indicating that a target is present, the returned echo also provides a measure of its distance, since the time elapsed between the transmission of the signal and receipt of the echo is measured. Depending on the

characteristics of the transmitted wave, it is possible to extract the amount of target information. The transmission of a signal can be modulated to detect targets based on its amplitude, frequency, or phase [2].

It was recognized early in the development of radar that continuous waves would be useful in measuring Doppler effects. The continuous wave radar's greatest advantage is that it is simple and capable of handling targets at all ranges and speeds without velocity ambiguity. As a rule of thumb, continuous wave radar has all the advantages without corresponding disadvantages. In continuous wave radar, there is a major problem concerning direct leakage from the transmitter to the receiver. The use of pulsed radars or continuous wave radar depends on the application. There are several advantages of continuous wave radars over pulsed radars. It consists of simpler, lighter, and more compact hardware. A continuous wave radar transmits less power than a pulsed radar. In contrast to pulsed radars, continuous wave radar detects targets at shorter ranges. Due to the nature of continuous wave radars, an unmodulated continuous wave radar cannot measure target distance. When the signal is reflected back from the target; continuous wave signal cannot be used to obtain distance. Time stamps must accompany continuous wave signals. Thus, continuous wave radars can be classified into two types: modulated and unmodulated.

2.1.2.1. Frequency Modulated Continuous Wave (FMCW) Radar

In continuous wave radar measurements, the Doppler frequency of the target can be measured, but not its range. Radar that uses frequency modulated continuous wave (FMCW) measurements can determine both the target's range and its Doppler frequency. The FMCW radar transmitted its frequency according to a known time-dependent pattern. The figure 2.5(A) illustrates the principles of triangular FMCW ranging with no Doppler shift for a single target. Range can be determined by comparing the signal echo with the radar's current transmitting frequency [4]. In the absence of a Doppler frequency, the difference frequency's value is given by Equation 2.5:

$$R = \frac{cTf_r}{2B} \tag{2.4}$$

where,

f_r: frequency difference between the signal echoes and the current transmission

R: range between transmitter and target

B: transmitted signal bandwidth

c: speed of light

T: period of modulation wave

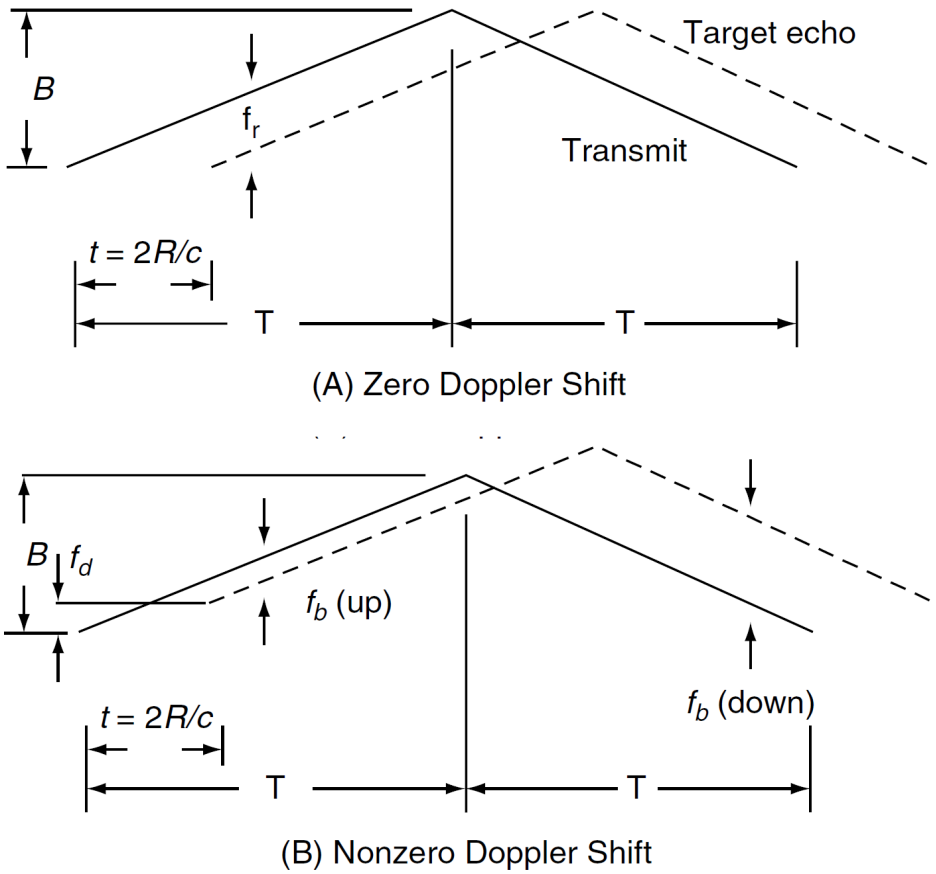

Figure 2. 5 Principle of Triangular FMCW Ranging [4]

Figure 2.5(B) shows a frequency–time relationship if a Doppler shift is present. Differences in frequency exist on both sides: the upper frequency, f_b(up), and the down frequency, f_b(down). The frequency indicates range f_r and the Doppler frequency f_d can be finding out by Equation 2.5:

$$f_r = \frac{1}{2}[f_b(up) + f_b(down)]$$
$$f_d = \frac{1}{2}[f_b(down) + f_b(up)] \quad (2.4)$$

Radars based on FMCW technology are applicable to airborne applications. For instance, for measuring the height above Earth's surface, the aircraft can be equipped with an FMCW altimeter. FMCW radars are used in the system which measure relatively short distance predominantly. Self-driving cars is good example for application of the FMCW radar. FMCW radar is used for obstacle detection in self-driving cars instead of sonar radars and cameras, as it is not affected by weather conditions, darkness and many other factors [5].

2.1.3. Doppler Radar

2.1.3.1. Doppler Effect

Christian Doppler, an Austrian mathematician and physicist, described the coloured light effect of stars in 1842 [6]. Light sources appear different colours when they move; the colour of the light would appear bluer while moving away with respect to an observer, the light would appear redder. This phenomenon was first discovered as the Doppler effect [7]. Light sources are seen as having a frequency that depends on the motion of the source in relation to the observer. As a result of the motion of the source, waves at the front and the back of the source are compressed and stretched, respectively.

Imagine an ambulance or police car with open sirens approaching you from the side. The siren sound will be different when the ambulance is approaching you versus when it is moving away from you. This example is valuable for the doppler effect in everyday life. Again, the doppler effect accounts for this difference. Doppler effect is illustrated in Figure 2.6. This electromagnetic wave source is motioning through to the observer, as can be seen in Figure 2.6. The wavelength of an EM wave, on the other hand, is larger than its exact wavelength when measured with respect to observer and smaller than its exact wavelength when measured with respect to source motion. Any wave propagating source would do, such as sound waves, electromagnetic waves, and so on.

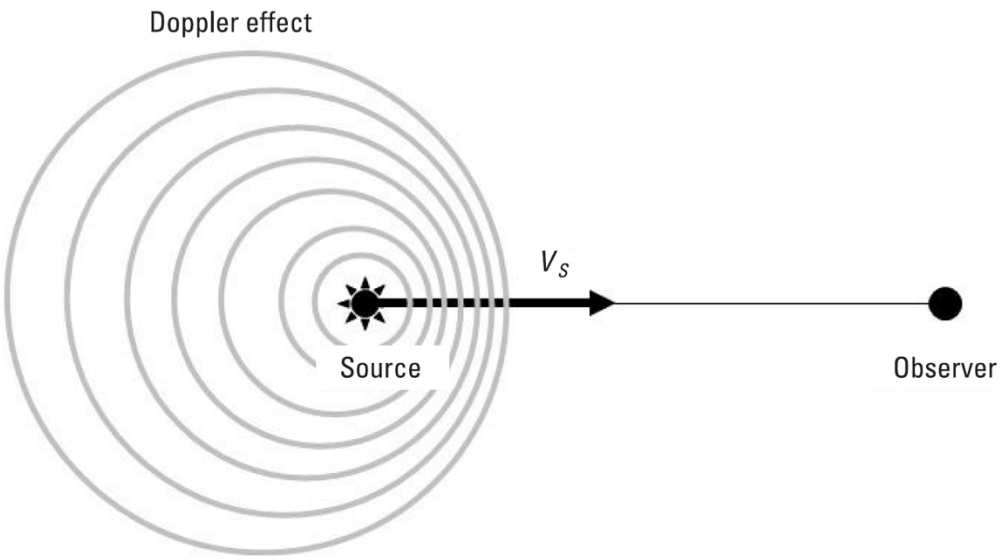

Figure 2. 6 Doppler Effect Simulation

Equation 2.5 shows the relationship between observed frequency and emitted frequency for a scatterer in the radar field of view moving with a velocity component.

$$f = f_0(\frac{c \pm v_r}{c \pm v_s}) \tag{2.5}$$

where,

c: speed of light

f_0: emitted frequency

F : observation frequency

v_r: speed of receiver

v_s: speed of source

2.1.3.2. Doppler Effect in Radar

In Doppler radar, electromagnetic waves with a specific frequency are sent up to the moving object. It is possible to use Doppler radar with a stationary target, but it is less useful if the target moves. Upon hitting the moving object, the electromagnetic radiation wave "bounces" back to the source, which contains both the original transmitter and a receiver. The relativistic Doppler effect, however, states that the wave is shifted whenever it reflects off of a moving object. Radar guns treat the wave that bounces back toward them as if it were an entirely new wave that came from the target that reflected it. The new wave basically originates from the target. Waves sent toward the target have a certain frequency, but when they are received at the gun, it has another frequency. Due to the fact that electromagnetic radiation has a exact

frequency when it is transmitted and changes to a frequency when it returns, the velocity of the target can be calculated [9].

2.1.3.3. Micro-Doppler Effect

In coherent laser radar systems, micro-Doppler was first introduced in [10]. Modulating a laser beam's amplitude, frequency, phase, and even polarization when transmitting and receiving electromagnetic waves, laser detection and ranging systems determine an object's range, velocity, and other properties by receiving the reflected or backscattered light waves. In high-frequency systems even with a low vibration rate or low vibration amplitude, phase shifts caused by Doppler shifts are readily detectable.

Micromotion refers to oscillating activity occurring in an target or body element of the object. A micro motion defined here includes a broader definition of the "micro," and includes motions of objects other than bulk motion, such as oscillations within the stuff or structural aspects of the stuff. Another example of a micro motion source includes a rotating propeller or rotor blade, bird wings flapping, or a person walking and swinging arms and legs. In micro-Doppler studies, human motion plays a significant role. Human bodies are capable of performing articulated motions by moving individual parts. The high articulation and flexibility result in a complex micromotion. Humans can articulate their motion through walking. During the transmitting of radar signals, micro activity induces frequency modulations alongside the carrier frequency. Periodically shifting the carrier frequency does not produce side-band Doppler motion but only a consummate periodical motion. According to the carrier frequency, shaking rate, and angle between the shaking direction and wave incidence, harmonic frequencies are included in the modulation. By modulating the frequency, it can be determined the kinematic properties of any object that interested in. There have been proposals to use propeller or rotor blade modulation to identify targets since a very long time. The research in [7] covers how to extract the kinematic information as well as the ways of representing the frequency modulation as a target signature.

2.1.3.4. Micro-Doppler Analysis

There is a correlation between the frequency band of the signal and the micro-Doppler effect. Micro-Doppler radars operate at high frequency bands using microwaves effect is detectable only if the target's oscillation rate and displacement are both high enough. For an X band radar, vibration at the frequency of 15 Hz coupled with a displacement of 0.3 cm can cause a maximum micro-Doppler shift of 18.8 Hz. The displacement requirement may be too large even with a 10-cm-wavelength radar operating in the L-band for the same micro-Doppler shift

of 18.8 Hz and vibration rate of 15 Hz, which implies the required displacement must be 1 cm. Consequently, in radar systems operating at lower frequencies, vibrations may not produce micro-Doppler shifts which can be detected. Causing by their longer rotary motion arms and quicker tip speeds, micro-Doppler shifts from rotating blades, such as those in rotor blades, may be detectable. Radar in the UHF band operates at frequencies ranging from 300 to 1,000 MHz, which is commonly used for foliage penetration detection (FOPEN). FOPEN radars are unable to detect the micro-Doppler shift when a target vibrates. Nevertheless, rotating blades or propellers can still be detected by micro-Doppler shifts. Radar in the UHF band can detect the micro-Doppler shifts of helicopter rotor blades at 666 Hz if they rotate at 200 m/s and their frequency can be measured with a 0.6-m wavelength [7].

The signature of the object can be called a characteristic expression of the object or process. Micro-Doppler characteristics in an object provide evidence of its identity as a moving object when the Doppler phenomenon is observed. An object's movement can be distinguished from other movements by means of its micro-Doppler signature. An intricate frequency modulation that takes place almost simultaneously in same the time and the frequency of Doppler domains makes an object identifiable. As shown in Figure 2.7, a rotating air-launched cruise missile (ALCM) exhibits a micro-Doppler signature in simulation software [11, 12]. There are 6.4m of length and 3.4m of wingspan for the cruise missile. An X-band pulse radar burst of 1-s chirp simulation is presumed to be based on pulses the electromagnetic backscattering field. A total of eight thousand one hundred and twenty-two pulses are transmitted from the radar once over a period of 0.55 second to cover the full vision of the target of 360°.

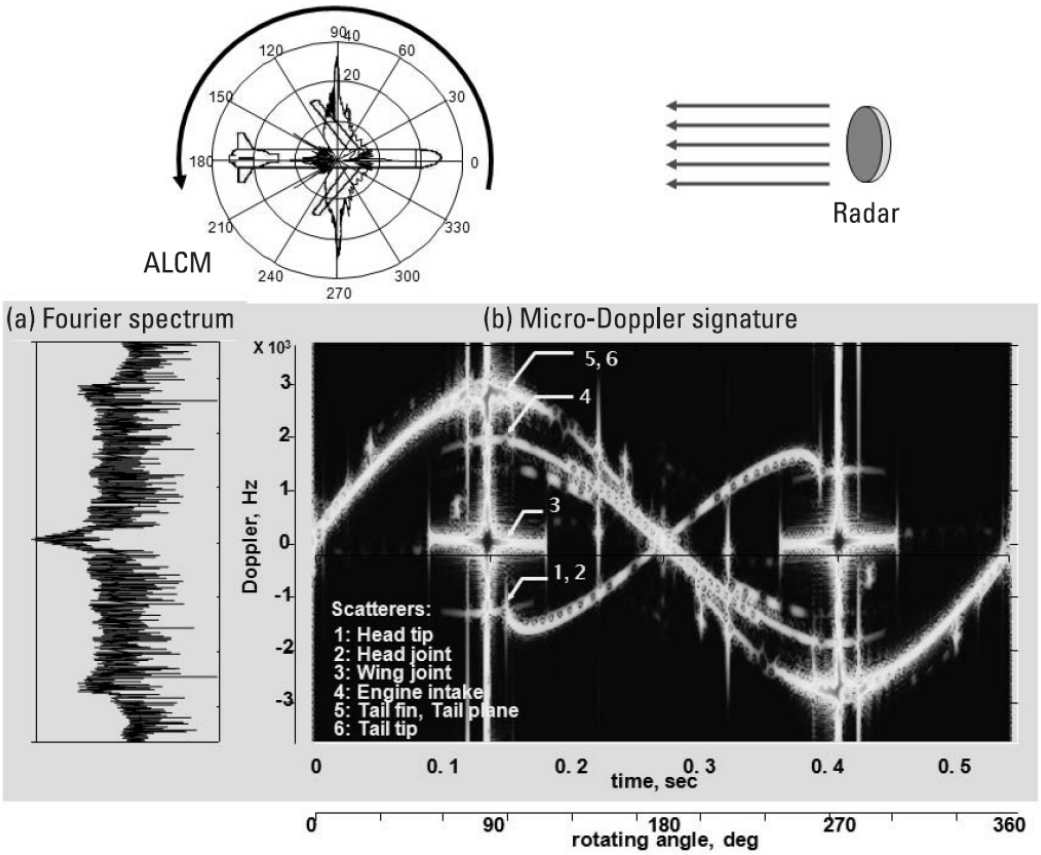

Figure 2. 7 Micro-Doppler Signature of Simulated ALCM [7]

It is possible to observe micro-Doppler phenomena in time-frequency domain, also in the domain of frequency of rotating ALCMs. As it can be seen in Figure 2.7(b), the rotating ALCM produces a time-frequency micro-Doppler signature. Figure 2.7(a) shows the conventional Fourier spectrum as a comparison. A rotation of 360° takes 0.55 seconds, so the missile rotates at a rate of 1.8 cycles per second. For the ALCM model, (1) the missile head tip, (2) head joint, (3 wing joint, (4) turbine engine intake, (5) tail fin and tail plane, and (6) tail tip and engine exhaust are located at about −2.5m, −1.8m, 0.2m, 2.5m, 3.5m, and 4.2m, respectively, off the pivot point at 0 [7].

2.1.3.4.1. Doppler Effect Due to Vibration

The target's micro-Doppler signature can also be acquired if it is composed of multiple parts, which might rotate or move independently from the bulk. When a vehicle is moving, the rotating elements can be the wheels; when a person is moving, the elements can be their arms or legs [13]. In the presence of these movements, Micro-Doppler effect can be utilized to extract a target's signature during measurement [14,15]. Radar signals may be extended to detect small

vibrations by extracting the micro-Doppler information contained in them [16,17]. A target's vibrations generate a phase modulation in the radar signal, and specific algorithms enable the phase information to be extracted. It can be referred to the discussion and the radar model in [20-22] to explain how micro-Doppler appears in the signal generated by vibrating targets.

2.1.3.4.2. Doppler Effect Due to Human Motion

In the late 1990s, researchers began studying the molecular signatures of human gaiting through radar [23,24]. Nonrigid bodies can be deformed, that is, the distance between two points in the body can change while the body moves, and as a result, the body shape can be altered. Nonrigid body motion and radar scattering, however, can be treated as the motion of multiple rigid bodies since the body can be modelled as joints or segments.

The radar has demonstrated its ability to detect targets with small cross sections, including humans and animals. However, the methods for analysing dynamic characteristics in humans and animals and identifying movement patterns from radar returns are challenging. Hence, many radars range-Doppler images have been observed to demonstrate Doppler modulation associated with a target's rotation, vibration, or motion In the range cells corresponding to the motion sources, these appear as characteristic Doppler frequency distributions. For example, rotating ship antennae, helicopter rotor blades, motioning arm and leg, or other rotationary motion characteristics may be detected. Radar imagery of a moving target is produced by removing target translational motion and oscillatory motion components with motion compensation and image autofocusing algorithms for minimizing the induced Doppler distributions in the radar imagery.

A human body's movement is articulated. A regular pattern of periodic movement can be observed in the human body's limbs. Humans walk with a highly coordinated pattern of muscles, joints, and nerves. Gait cycles are an example of an articulated human motion, and walking can be dissected into periodic motions. The human walking period is divided into two cases: swing and standing. Stepping is performed with the heel striking the ground and the toe off the ground. During the swing case, the foot is raised off the floor, either accelerating or decelerating. For gait analysis, methods include visual assessment, sensor measurement, and kinematic analysis that traces movement patterns and measures accelerations, displacements, and orientations of body segments and joints. Whenever a person walks, runs, jumps, or changes directions, their body movements change. Radar micro-Doppler signatures, since they do not suffer from distances, visible-light circumstances, and background divergency, can be used to guess the impact phase sequences, gait phases, and swing phases. [7].

Walking is characterized by its periodicity. Walking in a cycle is illustrated in Figure 2.8. Standing comprises about half of the cycle, while swinging occupies the remaining half. Standing on one foot, swinging the leg, and preparing the next stride occurs in the swing phase. This cyclic movement is repeated over and over again.

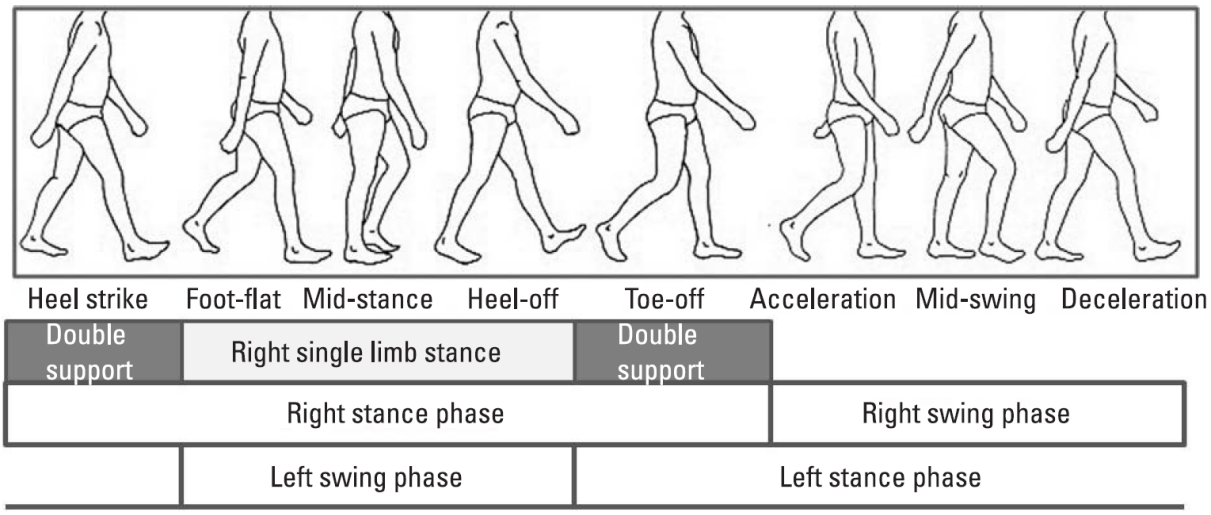

Figure 2. 8 Human Walking Movement in One Cycle [7]

One-loop human gait is shown in Figure 2.8. In this walk, all parts of the human body move periodically. The speed of each moving part increases and decreases in a specific period, relative to a reference point. The graph showing the radial velocities of all parts is shown in Figure 2.9. Every part moving in the body has a Doppler signature due to their speed. The micro-doppler signature created by all parts come together to form the integral micro-doppler sign of that movement.

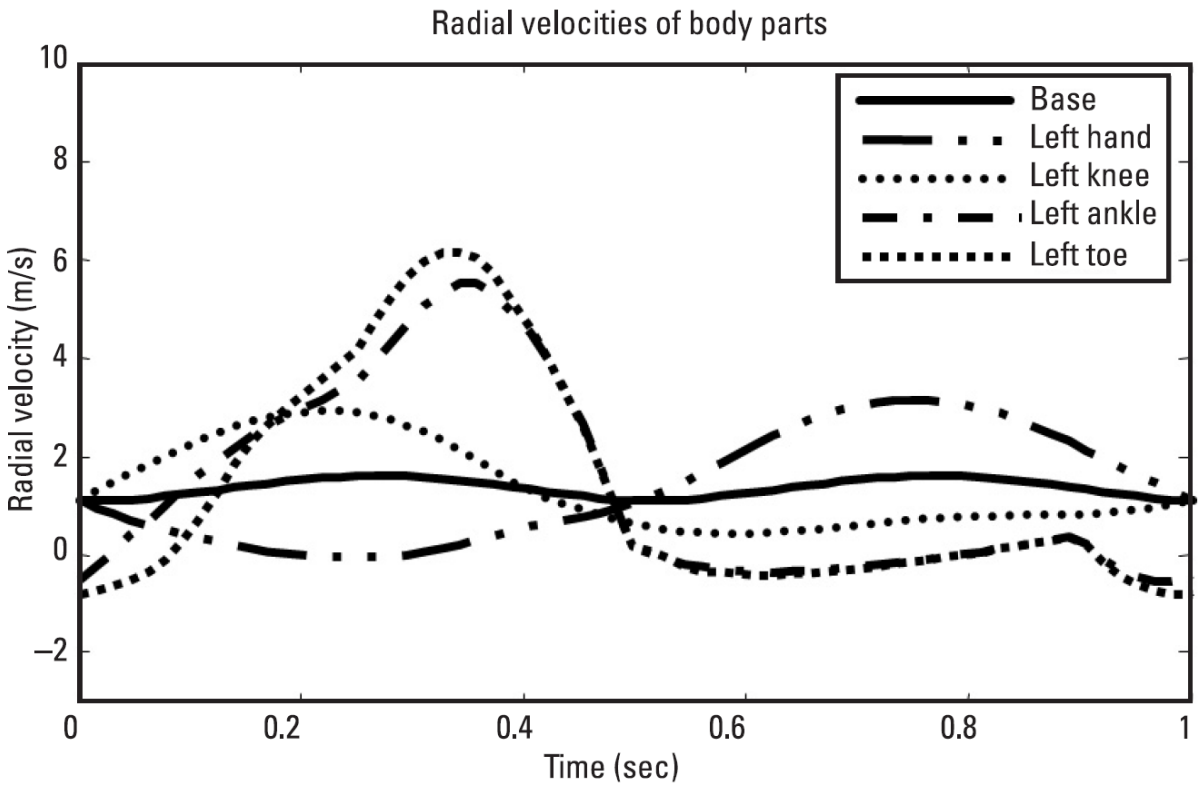

Figure 2. 9 Radial Velocity of Human Parts While Walking [7]

The time domain view of micro-doppler radar signals reflected from the human body will be rather confusing. Although frequency domain views, that is, Fourier transform graphs, create a more meaningful graph compared to the time domain, it will be very difficult for radar operators to interpret. Micro-Doppler signals reflected from the human body go through many processes while processing and extracting meaningful data. Extracting meaningful information mostly depends on the competence of the operator but can contain a lot of error. Such situations can be improved and optimized using artificial neural networks. Within the scope of this thesis, it is discussed how to process the radar signal reflected off the body and from vehicles, using artificial neural networks to extract meaningful information from the processed data. In terms of the radar signal as being input to the ANN, it must first be processed. We can think of the micro-doppler signal as a time series signal. This signal includes both the time component and the frequency component. If these signals are transformed into a spectrogram with the correct parameters, we obtain an image in which the pixels in an axis representing the frequency and the time indicate the strength of the signal.

Using the time-frequency transform (spectrogram) of the radar imaging range profile, a walking human's micro-Doppler signature is derived in Figure 2.10. As the Gait is completed

by swinging both legs simultaneously, they appear as peaks in the micro-Doppler signature. The motion underneath the legs tend to have a slight saw-tooth shape because of the way the body speeds up and slows down while it swings.

In figure 2.10 (a), the human is walking at a constant speed towards the fixed radar. In Figure 2.10 (b) its distance to the radar is given as pulse range profiles. Figure 2.10 (c), micro-doppler spectrogram of radar signal is given. There is a periodic pattern on the spectrogram as expected. It is the movement of a body part relative to the radar receiver that creates every fluctuation in the pattern. Since the movements of the legs are greater during walking than other piece of human body, the micro-Doppler signal created by the legs is expressed in a wider frequency range (in other words, the speed relative to the radar changes more). As the movements of the tibia and the clavicle will be less compared to the leg and more than the torso, an expected situation is observed again. Finally, since the torso moves at a more constant speed than other body parts during walking, it is the part with the least frequency change in the pattern in the spectrogram.

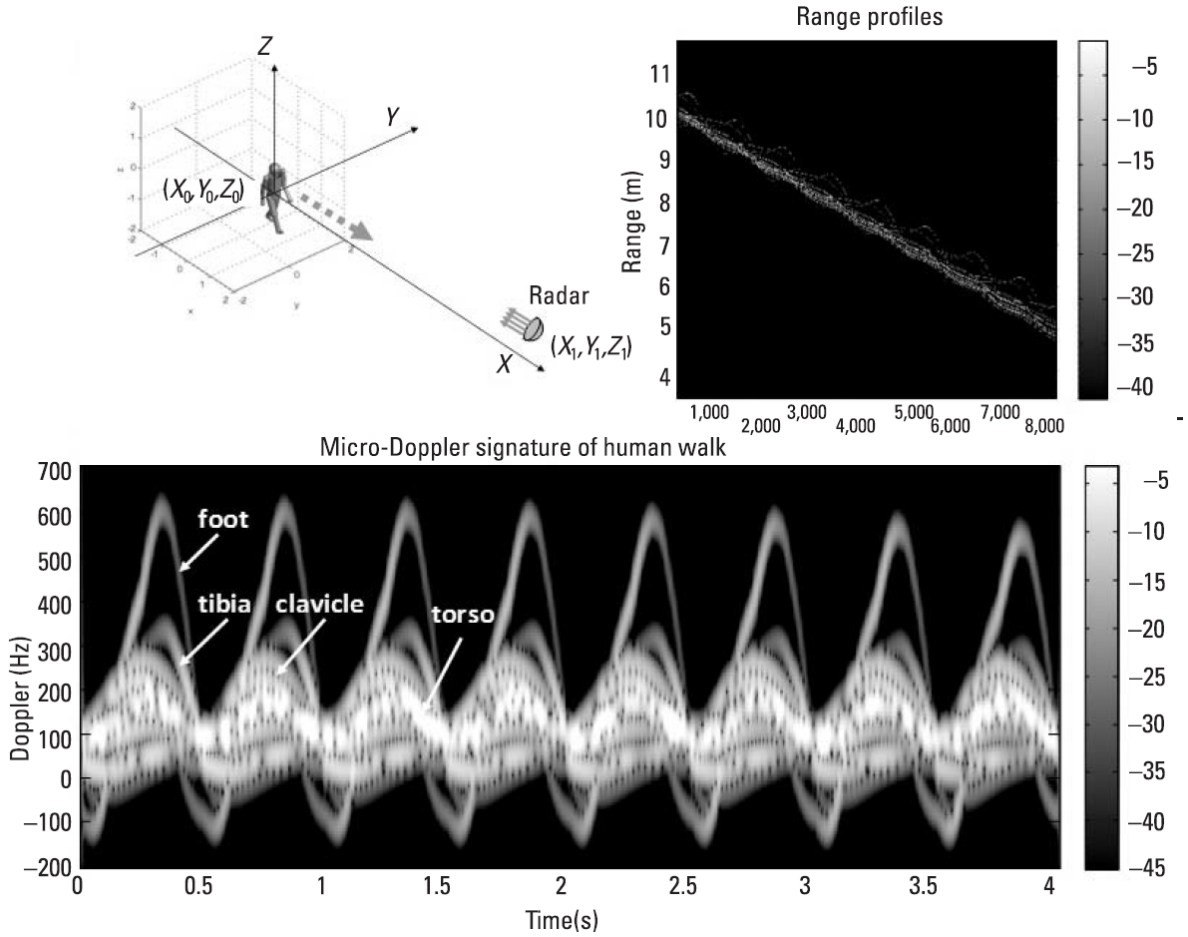

Figure 2. 10 (a) The orientation of a walking human in relation to a radar pulse, (b) the radar pulse-range profile, (c) and a walking human's micro-Doppler signature. [7]

Figure 2.11 illustrates examples of radar micro-doppler signatures detected by collected X-band radar data of humans running, crawling, and walking. In contrast to the micro-Doppler expression of a walking human in Figure 2.11 (a), the expression of a running human in Figure 2.11 (b) has a higher Doppler frequency shift and a shorter gait cycle. Figure 2.11 (c) shows that the Doppler frequency shift of a crawling person is much lower, as well as the maximum value of the Doppler shift.

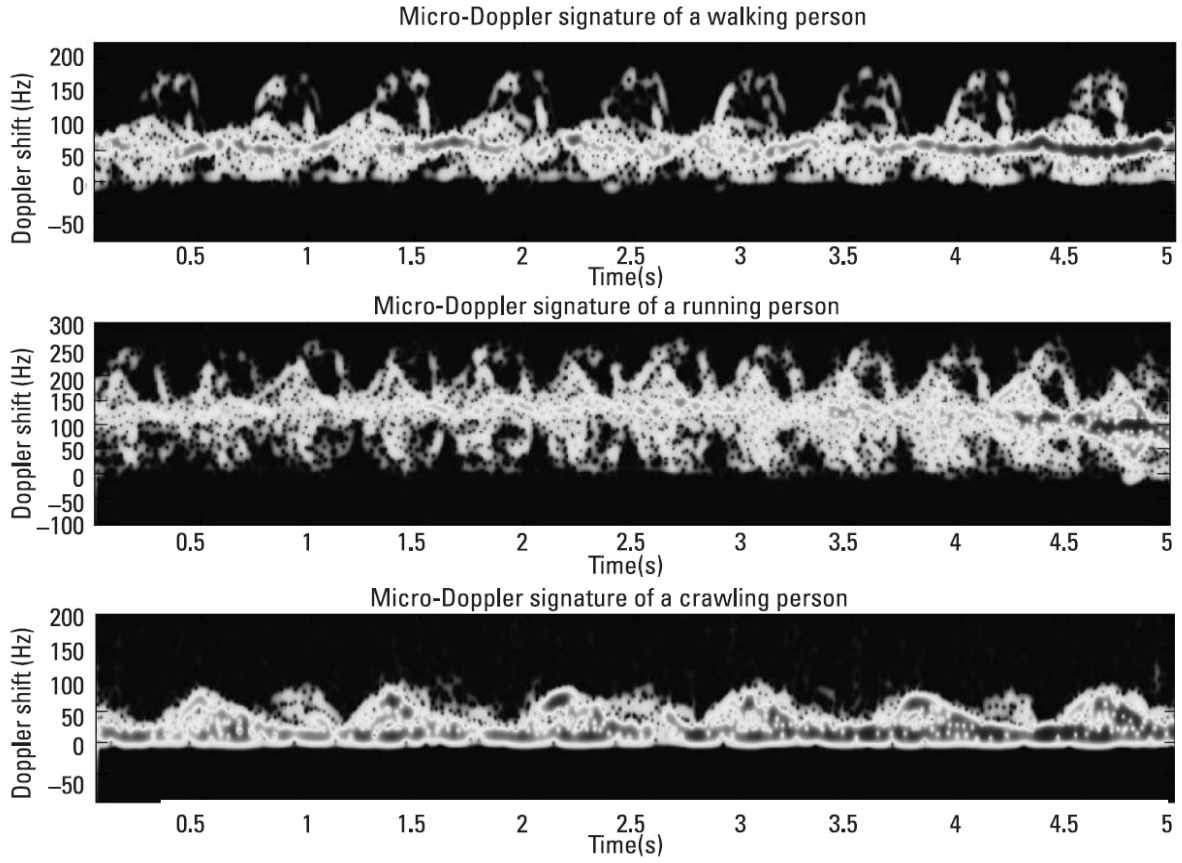

Figure 2. 11 (a) A walking person's micro-Doppler signature, (b) A running person micro-Doppler Signature, and (c) signature of micro-Doppler [7]

2.2. Machine Learning

In computer science, machine learning (ML) is a subfield which focuses on creating algorithms based on examples of some phenomenon in order to be useful. An algorithm or a method can provide examples from nature, or by hand creating them. Another way to define machine learning is as a process of solving a practical problem by 1) collecting data, and 2) Using that dataset, build a statistical model algorithmically. We assume that to solve the practical problem, that statistical model will be used somehow.

Typically, as part of machine learning process, training data is used to learn how to perform a particular task, where the objective is to achieve that task using the acquired experience. Data in machine learning is made up of samples. Attributes, features, variables, and characteristics are usually used to describe individual samples. There are several different types of features: nominal (list), binary, ordinal, and numeric (integer, real, etc.). Performance metrics are generally used for measuring machine learning model's performance in a particular task, which improves with experience over time. For machine learning models and algorithms, a variety of mathematical and statistical models are employed. Training then results in the trained model predicting, classifying, or clustering new examples using the knowledge it acquired during the training process. Figure 2.12 shows that machine learning approach.

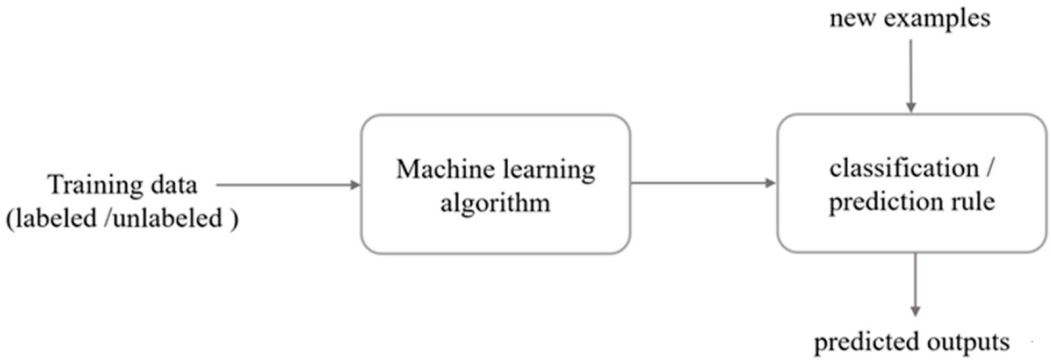

Figure 2. 12 Machine Learning Approach

Machine-learning tasks can be categorized broadly according to their learning methodology (supervised/unsupervised), their learning models (regression, clustering, classification) or the particular machines used for implementation of the task. There are two major categories of machine learning tasks: unsupervised and supervised learning, according to the learning signals of the learning systems. In terms of supervised learning, the training data is paired with the instance inputs and outputs, with aim of developing a universal rule mapping inputs to the outputs. There are often incomplete or only partial inputs and outputs in dynamic environments, with some missing or being only provided in a feedback loop (such as reinforcement-learning). A supervised approach employs expert knowledge to predict missing results for test data. Although supervised learning distinguishes between training and test sets, unsupervised learning does not. During processing, the learner seeks out hidden patterns [25]. The goal of dimension reduction is to enhance the quality of a dataset while protecting as much of the original data as possible as a result of merging supervised and unsupervised learning

types. A dimension reduction algorithm is usually used before classification-model or regression-model is applied to stay away the effects of dimensionality [25]. There are a number of DR algorithms that are commonly used:

- Principal-component analysis (PCA) [26]
- Partial-least-squares regression [27]
- linear-discriminant analysis [28]

It has been limited to presenting the models of machine learning that have been used in the thesis.

2.2.1. Regression

Regression model is one of the supervised learning models that desire to predict an output variable based on known input variables. There are several popular algorithms for regression, including linear regression, logistic regression, and stepwise regression [29, 30]. Additionally, they have developed their own regression spline to make regression calculations more complicated [31], multivariate-adaptive-regression splines [32], multiple linear regression, cubist [33], and locally estimated scatterplot smoothing [34].

2.2.2. Decision Trees

A decision tree (briefly DT) could be considered as a classification or regression-based model on a tree structure [35]. DT organizes the dataset into increasingly smaller and homogeneous subpopulations, while simultaneously generating a tree graph. Throughout the tree structure, the inside node represents the result of different pairwise comparisons of a selected feature, while each branch tells us how the comparisons were made. The leaf points are the results of mapping the entire system (expressed in terms of a classification rule). Classification and regression trees are two of the most common learning algorithms in this category [36].

2.2.3. Clustering

Clustering is widely used applications of unsupervised learning, namely, finding natural groupings of data [37]. Clustering techniques such as k-means [38], hierarchical clustering [39], and expectation maximization [40] have been used by researchers for decades.

2.2.4. Bayesian Models

Bayesian models (BMs) are statistical-based probabilistic graphic models within which the analyse to be performed using a Bayesian framework. The supervised learning category includes models

such as this one, which can be applied to supervised learning problems. Naive bayes [41], gaussian naive bayes, multinomial naive bayes, Bayesian network [42], mixture of gaussians [43], and Bayesian belief network [44] are illustradted most famous algorithms have been developed for different applications in literature.

2.2.5. Support Vector Machines (SVM)

By definition, support vector machines (short for SVMs) construct a linear hyperplane against which data examples are classified based on their characteristics. In traditional SVMs, the classification abilities can be significantly improved by Utilizing the kernel trick to create a higher-dimensional feature set from the original one. SVMs have been used for classification, regression, and clustering in literature. Developed on the basis of global optimization in high-dimensional spaces, SVMs help overcome overfitting problems, making them applicable in a wide range of applications [45,46]. Some of the most popular SVM algorithms are, the support vector regression method [47] and least squares support vector machine [48].

2.3. Deep Learning

Traditional Machine Learning (ML) is enhanced by deep learning (DL), which introduces additional complexity (depth) to the model and alters data using various functions that represent the input hierarchy. [49]. One of the most attractive aspects of deep learning is its ability to automatically extract features from raw data, where higher-level features are derived from lower-level ones [50]. In particular, deep learning solves complex problems well and quickly because the models used are more complex, which allows for massive parallelization. A deep learning model utilizes complex models to reduce errors in regression problems and enhance classification accuracy when enough data is available to represent the problem. Based on the network architecture are using, including such Unsupervised Pre-trained Networks, Convolutional Neural Networks, Recurrent Neural Networks, and Recursive Neural Networks, deep learning includes a variety separate aspect also including convolutions, pooling layers, fully connected layers, gates, memory cells, activation functions, encode/decode schemes, and so on. [51].

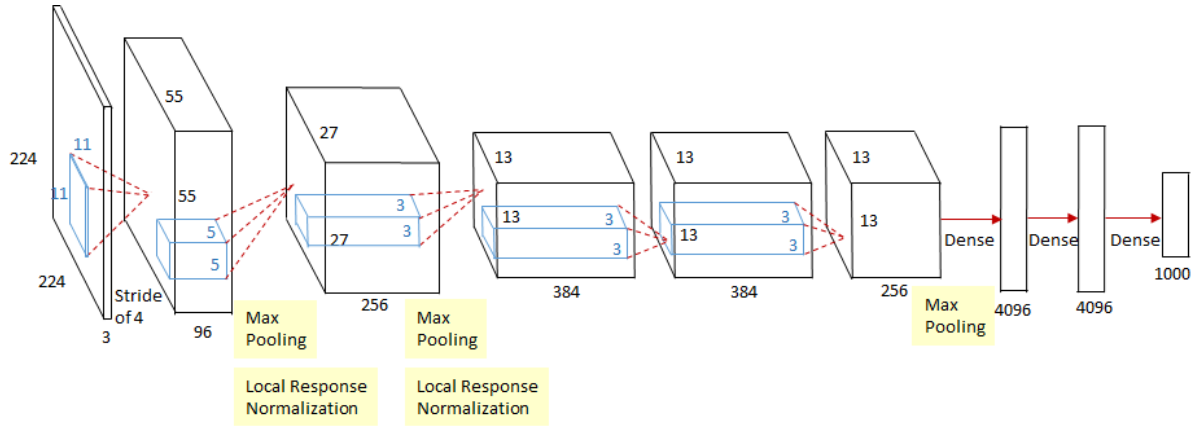

Figure 2. 13 CaffeNet Architecture

Video and image contour data have gained popularity in an array of applications using deep learning. It can process audio as well as voice and natural language, as well as abstract data like weather [52], soil chemistry [53], and population information [54]. Figure 2.3 illustrates an instance of DL architecture, CaffeNet [55], which is comprised by convolutional-layers and fully connected layers.

Deep learning reduces the need for feature engineering in image processing, which is one of the most significant advantages of this method. Standard techniques used to classify images were affected by hand-engineered features, which affected the overall results. It is a time-consuming and complex process to implement features; it must be adjusted every time the dataset or the problem changes. The time-consuming process of feature engineering, therefore, requires local knowledge and is unlikely to generalize beyond a narrow area [56]. In contrast, deep learning does not require the creation of features, but instead figures out which features are important through training. The training time for deep learning is usually longer than for other methods based on machine learning, but it is usually faster to test using machine learning than using other methods [57]. The models' complexity can also create optimization issues, as well as hardware limitations. Pre-trained models can also pose risks when used on small datasets or datasets with significant differences.

2.3.1. Artificial Neural Networks

Artificial neural networks (ANN) are divided two parts into: “Traditional ANNs” and “Deep ANNs”. An artificial neural network emulates the complex functions of the brain, such as pattern recognition, realization, and learning, and makes some decisions based on that information [58]. The brain is composed of billions of neurons that interact with one another

and process any information provided. A biochemical neural network, modelled loosely on the structure of an ANN, consists of interconnected units similar to ANN. Numerous nodes are present in the network arranged in multiple layers, such as:

- Input layer, where data is used as input to the system,
- It consists of one or more hidden layers where learning takes place.
- Providing the decision/prediction in an output layer.

In artificial neural networks, the learning process is supervised, and they are used to predict and classify data. They are also used for regression task. ANNs use various learning algorithms, including radial-basis-function networks [59], perceptron [60], back-propagation [61], and also resilient-back-propagation [62]. There have also been several ANN-based learning-algorithms proposed, including counter-propagation-algorithms [63] and neural fuzzy inference-systems [64], autoencoder and supervised-Kohonen-networks [65], as well as Hopfield-networks [66], multilayer-perceptron [67], self-organising-maps [68], extreme-learning-machines [69], regression neural network [70], ensemble-neural-networks and self-adaptive-learning-machines [71]. The majority of people refer to deep ANNs as deep neural networks (DNNs) [72]. Machine learning analysis is a relatively new field, where sanctionative models combining multiple layers of computation find out high abstraction levels for complex information representations. In some cases, AI-based deep learning performs the step of feature extraction by itself. This is one of the main advantages of deep learning. Many different sectors and industries, including agriculture, have greatly benefited from deep learning models. There are several different topologies of DNNs including supervised, semi supervised, or even unsupervised layers. Convolutional neural networks, in which feature maps are exploited for generating maps in the image domain, are a good example of a standard deep learning model. Comprehensive information is available on CNNs in the literature [73]. Other general deep learning architectures are Boltzmann-machines, deep-belief-networks [74], and auto encoders [75].

2.3.2. Convolutional Neural Networks

Convolutional Neural Networks (briefly CNNs) are one of the famous subsection of artificial neural-networks that have shown tremendous performance in areas such as image-classification. The CNNs are able to distinguish faces, objects, and traffic signs, in addition to supplying the power to autonomous vehicles and robots. Researchers have revealed that CNNs are effective in many Natural Language Processing tasks, including the classification of

sentences, the detection of emotions in text among others. The primary purpose of this thesis is not to develop an understanding of how Convolutional Neural Networks work. Accordingly, CNNs architecture, how it works and what is the parameters of CNNs are briefly explained in this section.

Neurons in ANNs tend to self-optimize through learning, just as traditional ANNs do, CNNs are composed of neurons. Each nerve cell can still receive associate degree input associate degreed perform an operation the idea of in numerous ANNs. Although each component of the network will have its own perceptual score function, whole perception scores will continue to come from a single weighting function. Loss functions associated with classes will be found in the last layer. It is only noteworthy that, unlike regular ANNs, CNNs are primarily used to recognize patterns within images. The design may include options specific to specific image types, hence making the network more appropriate for image-based computations, whereas reducing the parameters necessary for lining up the model.

In CNNs, the input consists primarily of bunch-of-images. Consequently, the architecture should be set up to accommodate any particular data types that might be encountered. A major difference is that there are three dimensions to the CNN layers. These layers have neurons organized into three dimensions, and also, they have spatial spatiality (height, width, as well as depth) to the input. An activation volume is a volume within the ANN that is not related to the number of layers in the model. A layer of the ANN does not connect with a large area of the layer preceding it, unlike traditional ANNs.

In follow this way this means that for the instance has given earlier, the input will have a dimensionality of 224 × 224 × 3 (height, width, and depth), leading to a final output layer comprised of a dimensionality of 1 × 1 × k. Here in the dimensionality, “k” represents that the quantify of class. This will result in a smaller set of classes across depth depending on how the input dimensionality is condensed.

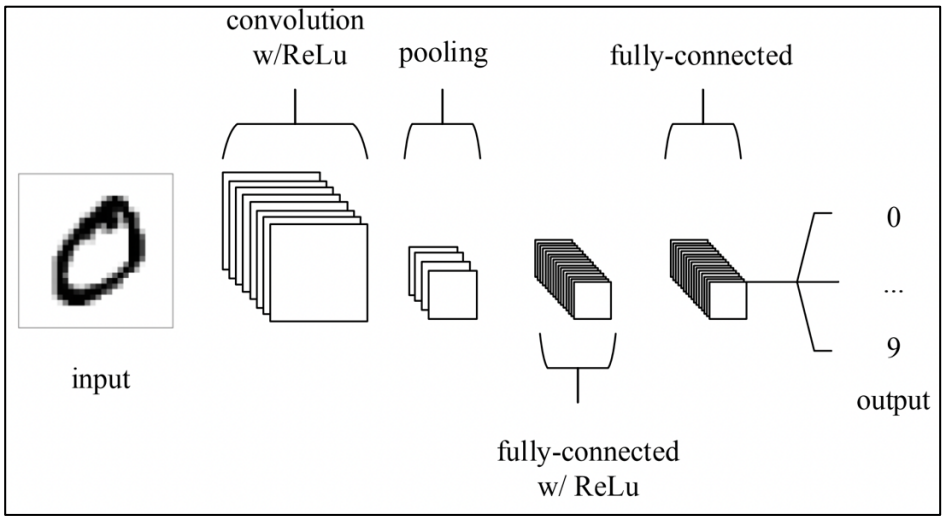

Figure 2. 14 Basic CNN Architecture [76]

Convolutional Neural Networks are constructed of three kinds of layers. These are convolutional, pooling, and dense layers. In order to construct a convolutional neural network, these layers are stacked together. Using a convolutional neural network for classification on MNIST is depicted in Figure 2.14. A CNN with the basic functionality described above is divided into four sections:

1. An ANN's input layer normally consists of the layer that holds the pixel values.
2. In the convolutional layer, output of neurons associated with local areas of input is determined by the scalar product between their weights and input volume. As the output of an earlier layer is activated by a sigmoid activation function, a rectified linear unit commonly called a ReLu are used as an activation function.
3. Pooling layer further reduces the number of parameters in the activation by sampling along the spatial dimensions of input.
4. Once the fully connected layers have been trained, they can then act constantly as artificial neural networks, retrieving score categories from activations that are used for classification. According to the authors, ReLU can be used between the fully connected layers, thereby improving performance.

In this simple way, Convolutional neural networks are able to encrypt the initial computer files layer by layer, using techniques such as convolution and down-sampling to provide category scores.

2.3.3. Deep Generative Models

Deep generative models (DGMs) are one of the classes of neural networks that contains many hidden layers trained to mimic probability distributions of quite large number of samples in the certain dataset. When successfully trained, it can be used to estimate the possibility of each observation and to create new examples from the underlying distribution. Developing DGMs has emerged one of the most popular research topics in machine learning recently. DGMs are growing rapidly in the literature. Some DGMs have already become too interesting AI models. For example, the recent achieving is so fascinating in the field of generating realistic voices, images, or videos that called deep fakes. Notwithstanding these achieving, a lot of mathematical and practical issues limit the use of DGMs. The DGM models must be designed for specific dataset. It does not show same results on different dataset. It is even more challenging to understand that why there is not certain model and hyper-parameter for all datasets.

Applications of deep generative models (DGM), such as creating fake videos from famous images, have recently become popular. Although these fake images may already appear to pose ethical problems it also promises new useful technologies [77]. For example, in physics and computational chemistry, DGMs used as new scientific applications. In the scope of this thesis, DGMs will be used to create new samples for radar dataset and will be evaluated its performance.

Deep generative models are neural organizations with many secret layers prepared to rough confounded, high-dimensional likelihood distributions. In brief, the driven objective in DGM preparing is to memorize an obscure or recalcitrant likelihood dissemination from a ordinarily little number of autonomous and indistinguishably dispersed samples. When trained successfully, that be able lie used the DGM in conformity with score the probability concerning a partial sample then to create recent samples so are similar after samples from the uncouth distribution. These troubles bear been at the bottom of probability and statistics for decades however continue to be computationally difficult to solve, particularly into excessive dimension. DGMs can be used to augment small datasets. Generative Adversarial Networks (GAN), a type of DGM, are used in this thesis to augment relatively small dataset. Therefore, it will be more valuable to discuss about Generative Adversarial Networks at the rest of this section.

2.3.3.1. Generative Adversarial Networks

Generative adversarial network (GAN) is an algorithm that uses two neural networks, which are combined in an adversarial race, New synthetic data should be created so that they can play the role of real data. In a study that published by Ian Goodfellow et al. at the University of Montreal, GANs were introduced as a tool for synthetic data generator such as image generation. Since GANs can learn to mimic any distribution of data, they have the potential to create worlds that are remarkably similar to be in any domain: images, any signals, songs and speeches. In a sense, they are robotic artists, with stunningly good output. But they could be utilized to create fake digital content as well, and they are behind of "Deepfakes". For a better understanding of GANs, a comparison with discriminative algorithms is useful to understand how generative algorithms work. A discriminative algorithm tries to classify data; that is, given a set of features about the data, it identifies the category or label to which the data belongs Using a discriminative algorithm, a data instance of images of bikes and motorbikes could predict whether photos were "bikes" or "motorbikes.". An image of a bicycle is included as one of the labels, while the bag of images is part of the dataset. This problem can be expressed mathematically as y is the label, and x is the feature. The equation $p(y|x)$ means "the probability of y given x", that in this case would understand as "the probability that an image has bike inside of it."

As a result, Labels are assigned to features by discriminative algorithms. Only this correlation is of concern to them. A generative algorithm does the opposite of what a deterministic algorithm does. The process of predicting labels is done by looking at specific labels instead of certain characteristics. In generative algorithms, the question is: How likely are these features to appear in a bike image? In contrast to discriminative models, generative models are concerned with "how you get x?". In addition to being able to capture $p(y|x)$, the probability of x given y, generative algorithms could be used as classifiers as well. Interestingly, they have ability to do more than categorize input data. In mathematical representation of GAN, here are two-player minimax games with the value function where Discriminator and Generator participate in Equation 2.6:

$$min_G\ max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z))] \quad (2.6)$$

A neural network, called the generator, is responsible for generating new data instances, while another, the discriminator, is responsible for evaluating if those examples belong to the underlying training dataset or not. At the same time, the generator creates new artificial images that it passes on to the discriminator. Even though they are fake, it tries to pass them off as real, hoping that they will be considered authentic as well. The generator's aim is to generate handwriting: “to lie without being caught”. A discriminator's aim is to recognize fake images generated from the generator. In terms of mathematical representation, D(x) need to be maximized and D(G(z)) need to be minimized in purpose for Discriminator network; D(G(z)) need to be maximized in purpose for Generator network. It is going to be clearer to explain working steps of GAN:

1. It generates a fake input-label pair for the fake label by feeding noise from a random distribution to the Generator G.
2. 2. This false pair is fed alternately to Discriminator D along with the real pair (label y = 1).
3. In the discriminator D, that uses a binary-classification neural network, these x values are combined into a loss known as the D loss as it calculates both fake and real values.
4. It also calculates the noise level from the G generator as the G loss since each objective function is different.
5. These two losses are sent back to their respective networks, which adjust their parameters based on the loss.
6. The algorithm used can be Adam, Gradient Descent etc. and repeating this process for a desired number of epochs.

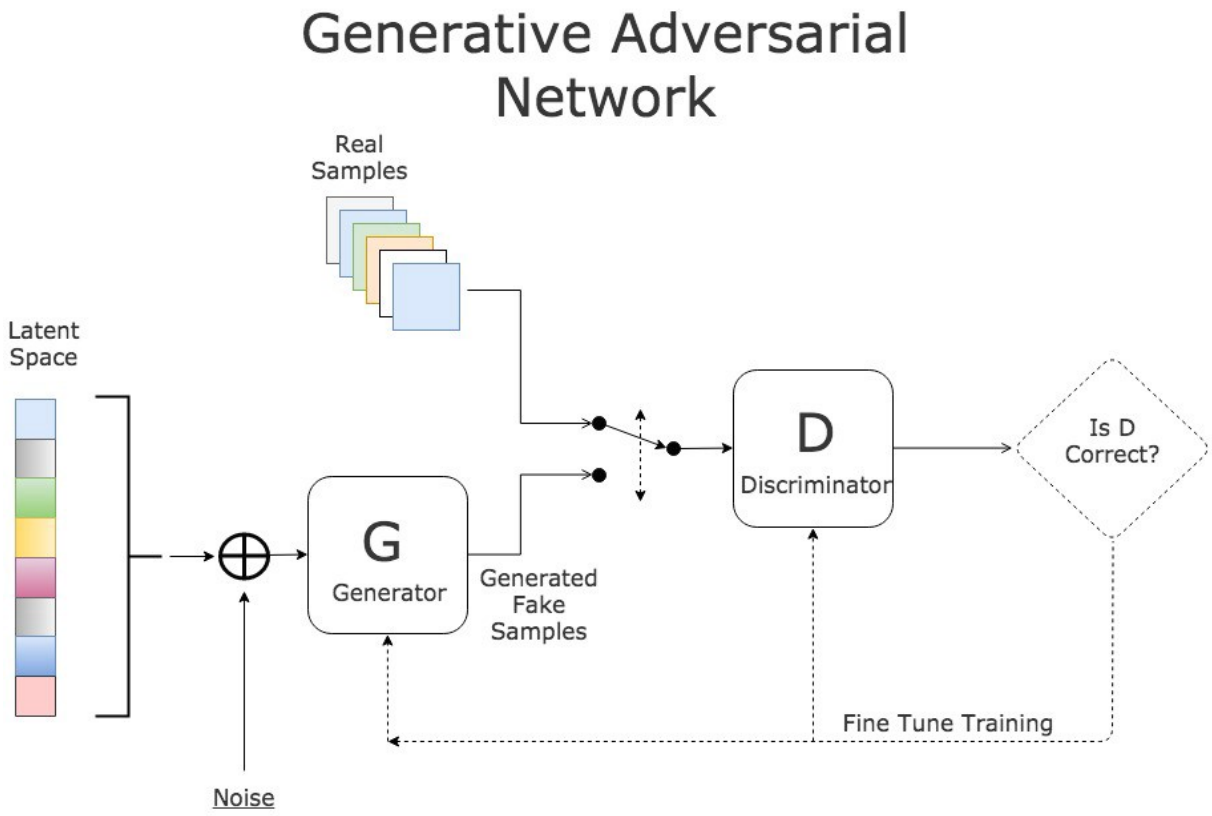

Figure 2. 15 Generative Adversarial Network Architecture [78]

The discriminator network for the MNIST dataset [79] is a standard convolutional network that divides images into two groups, real or fake. Generators are inverse convolutional networks, basically: Rather than taking an image and down sampling it, these models using stochastic models, the noise vector is up sampled and used as an image. Using techniques such as maximum pooling, the first net discards data and the second one generates it. A zero-sum game is a situation in which each network tries to optimize a different objective function, or loss function. As the discriminator changes itself in terms of the discriminator loss, also the generator changes. The losses of each force-against each other. Very first architecture of GAN is shown in Figure 2.15. How GANs give results on MNIST data is shown in Figure 2.16.

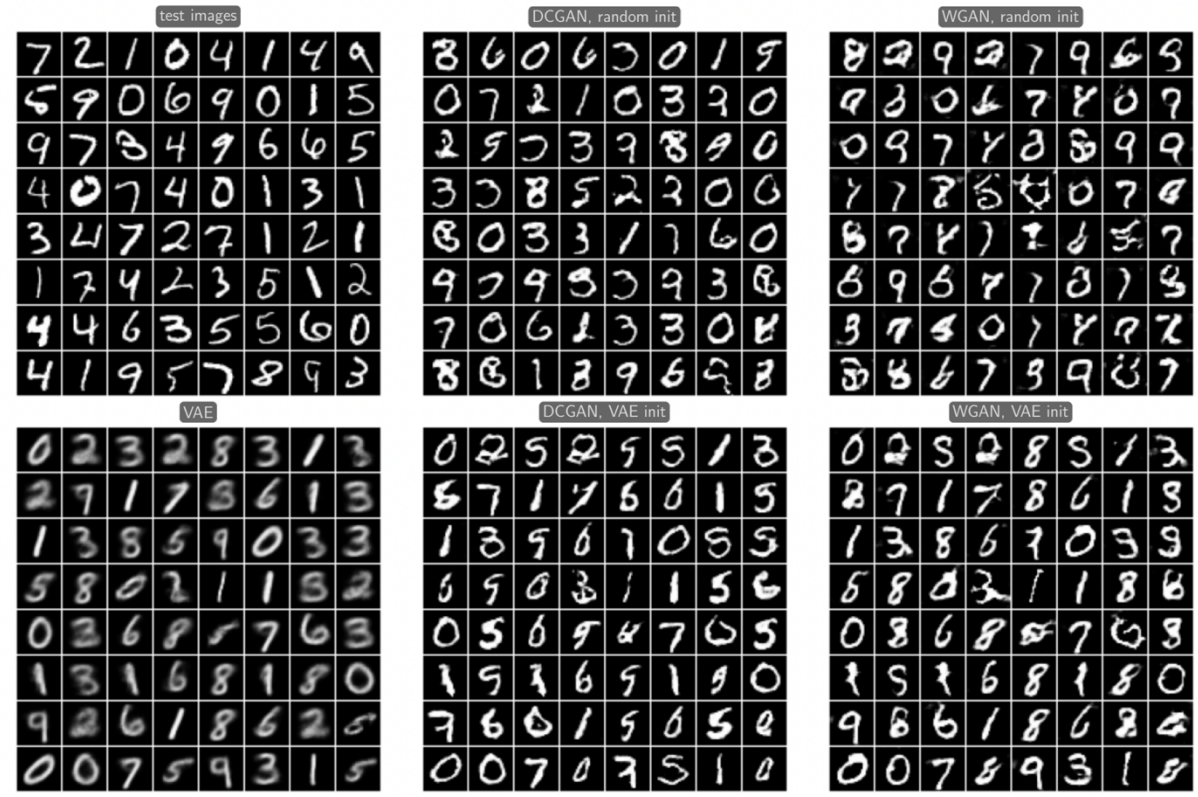

Figure 2. 16 GAN, Variational Autoencoder (VAE), Deep Convolutional GAN (DCGAN) and Wasserstein GAN (WGAN) Results on MNIST Dataset.

The ability of GANs to mimic on datasets is quite high. GAN structures have been used to reproduce data sets in many studies up to now. GAN structures can be trained not only on image data but also on audio, text, numeric, time series data. In the scope of this thesis, GAN structures will be used to reproduce the radar dataset and the results will be analysed.

2.4. Related Works

Radar is currently being used to detect and track people in highly dispersed environments in response to the growing demand for security and surveillance. With electromagnetic-based sensors, people can be located through walls and their activity can be recorded in real-time. There has been a noteworthy number of research to develop human activity detection and tracking. Using wide bandwidth for imaging radar signals at close range is arguably one of the most important approaches. In various studies, such as those mentioned in [81] and [82], building interior images have demonstrated the capability of detecting and tracking people indoors. Use of a Doppler radar is another way to achieve it. It is easy to detect movement using a Doppler-based radar during a background repression of clutters. The radar indicator screen contains "micro-Doppler" features when Doppler returns are being sent from humans. Using a

micro-Doppler radar, humans generate reflections from their non-rigid moving bodies. Information about human movements can be found in it. In the years since micro-Dopplers for human activity classification first appeared in [83], new research has taken place exploring their potential. For instance, Otero et al. [85] created a basic classifier to distinguish human gait using a spectral analysis. A study by Van Dorp et al. [84] used FMCW radar data to estimate the parameters of human gait. Humans, animals, and vehicles were distinguished by the use of Micro-Doppler features in [89] According to [86]– [88], several time-frequency analyses were used to extract micro-Doppler features from targets.

P. van Dorp et al. [99] utilized conventional methods to classify radar signals. They did not utilize machine learning methods. Similarly, S. Groot et al. [100] did not use machine learning approach to classify. They applied particle filter to micro-Doppler radar signal's spectrum. In [101], Authors applied Principal Component Analysis methods to classify human motions on micro-Doppler radar signal spectrograms.

Youngwook et al. [90] used Support Vector Machine to classify doppler radar signal which measured 8 human activities. They extracted feature by using spectrograms of radar signals. They acquired %90 of classification accuracy. By using mel-cepstrum coefficients, Van Eden et al. [91] have been able to differentiate between animals and humans with a Gaussian mixture model and hidden Markov model. They compared their methods with current micro-Doppler classification techniques. They illustrated that their technique has classification accuracy between %75 and %90. Micro-Doppler signatures can also be measured from the gyroidal blades on helicopters, the gyroidal antennas on ships, and the vibrational engines in vehicles [10], [21].

Tivive et al. [92] investigated image-based approach by using time-frequency domain signals. They consider that signal as image and applied Gabor-filters to extract feature such as edge, from signals. In [93] authors used high order spectral processing to extract meaningful features from reflected Doppler radar signals. These all method could be called as "conventional feature extraction method". Conventional feature extraction methods of radar signal have many drawbacks. It requires meaningful features to classify radar signals. Additionally, adaptation of these methods to different domain is relatively difficult. To get rid of these difficulties, Convolutional Neural Networks get on the stage. The Convolutional Neural Network is one of the most widely used artificial neural networks for tasks in computer vision, such as, image recognition, image classification, object detection, etc. [94]. Classifying images involves taking images as inputs, analysing them for their probability of matching a particular

class, and then synthesizing the data based on the results, afterwards show in the classification probability as output. CNNs require much fewer image pre-processing compared to other classification algorithms. It generally involves convolutional layers to extract features, pooling layers to reduce size of images and fully connected layers to classify images.

By combining different levels of abstracting, CNNs are able to extract and learn features automatically. It has been demonstrated that CNN-based approaches outperform conventional approaches in the classification of radar signals [90], [95], [96].

The study by Y. Kim et al. [95] classified radar signals of humans with 97.6% accuracy using a pre-trained CNN method. Cars, dogs, and horses were the targets. Additionally, 90.9% of the predicted activities by their methods have been accurate, such as crawling, running and walking.

In [102], R.P. Trommel et al. utilized DCNN to classify human gait by using micro-Doppler radar-spectrograms. They acquire an overall 86.9% classification accuracy by using DCNN and significantly outperforming the 3-NN with 68.3% classification accuracy and SVM with 60.3% classification accuracy.

B. K. Kim et al. [97] utilized the GoogleNet architecture which is pre-trained model and they obtained 100% accuracy. They classified two different kind of drones with using radar operates in Ku-band. They converted radar signals into spectrogram images and fed CNN with that spectrogram images for training CNN Model

In [98], The doppler radar echoes of human activities were classified by J. Park et al. with a finetuned CNN model which is pre-trained. They attained 80.3% classification accuracy for five human activities. Additionally, t A feature-based approach generates 45% classification accuracy. A pre-trained CNN generates 66.7% accuracy with the radar dataset.

According to papers that published in the literature, using CNNs are quite better for feature extraction of micro-Doppler radar signal spectrogram. E. Alhadhrami et al. [103] used CNN for classifying micro-Doppler radar signal spectrograms. Using Micro-Doppler radar signals spectrograms three CNNs were trained to extract feature representations which are AlexNet, VGG-16, and VGG-19 networks in a transfer learning mode. They utilized public RadEch dataset [104] to classify. Their method is superior to other methods which used same dataset for classification, with %99.95 accuracy. This study could consider as the state-of-art. However, they used conventional data augmentation techniques before training with

unbalanced dataset. They did not apply noise reduction methods to improve classification accuracy. Their data augmentation process did not make unbalanced RadEch dataset to be balanced. On the contrary, it made dataset more unbalanced. Unbalanced datasets are not well suited for deep learning models. When deep learning models are trained with unbalanced datasets, their generalization ability is low and could be misleading.

In the scope of this thesis, RadEch dataset [104] is utilized to classify. Contrary to what E. Alhadhrami et al. [103] had done; this thesis shows the effect of transfer learning techniques on radar data sets in detail. Furthermore, in this thesis, both time series radar signal and spectrograms of radar signal are used to classify. Their results are compared and analysed in detail. Conventional data augmentation techniques have been applied to make the RadEch dataset more balanced. Not only conventional data augmentation techniques have been applied, but also learning based data augmentation techniques with using Generative Adversarial Networks (GANs) has been applied and results are analysed. Additionally, unlike what E. Alhadhrami et al. [103] did, noise reduction technique has been applied to dataset. The results illustrated that without any data augmentation, same CNN models which trained with noise reduced dataset, have same accuracy or superior to state-of-art studies.

3. METHODOLOGY

Analysing various human body movements have attracted much attention in recent years. The most commonly used method for human movement analysis uses visual image sequences. However, moving parts of the human body are perceived differently depending on distance, variations in lighting, clothing conditions, and occlusion from distractions. Radar has been widely used for detecting, tracking, and imaging targets of interest due to its long-range capability, wonderful morning and evening performance, and ability to penetrate wall and ground. Thus, radar has become a tool for detecting and tracking humans and animals. Beside human body motion, animal motion is also an important nonrigid body motion. Compared to human bipedal motion, the four-legged animal's motion has more choices for its feet striking the ground.

In this section, the study of classification of human movements and some vehicles will be explained. Five different human movements, two different vehicle movements and one clutter will be classified. The classification process will be done over the reflections of the micro-Doppler radar signal. Within the scope of this thesis, Radech Database [104] was used for the classification process. To increase the classification accuracy, dataset's noise will be reduced with using proposed method in this thesis. After the classification process, the results will be compared with the dataset without noise reduction. Conventional data augmentation methods will be utilized to enhance the classification success of the noise-reduced dataset. In addition to traditional data augmentation methods, learning-based data augmentation methods will also be applied by using Generative Adversarial Network. All results will be compared among themselves and with state-of-art publications.

For classification all class in dataset, Convolutional Neural Networks will be used. If it is first considered the micro-Doppler radar data as time-series signals, it could be seen that these signals are suitable for one-dimensional CNN architecture. One-dimensional CNN architectures are used to classify one-dimensional data. It performs feature extraction with one-dimensional convolutional filter on data set. In this study, the RadEch dataset was first classified with one-dimensional CNN and the classification success was recorded. The dataset was then converted into two-dimensional spectrograms and made suitable for two-dimensional CNN structures. Different convolutional neural networks were used to observe the classification successes on the same dataset. One of these CNNs is a simple multi-layer and two-dimensional CNN created specifically for this dataset. Hyper-parameters are tuned in accordance with the

dataset and the highest classification accuracy is tried to be achieved. Apart from the CNN specially designed for this dataset, it has been trained for classification in different architectures to compare the success of the designed architecture. One of the trained architectures is the VGG-16 and VGG-19 CNN architectures published by Simonyan et al. [105], which is successful in classifying relatively small-sized images as in the dataset used in this thesis. The difference between VGG-16 and VGG-19 is all about the depths of the architectures. VGG-19 is a deeper CNN architecture than VGG-16. It shows higher classification success in relatively larger datasets. In Addition, within the scope of this thesis, transfer learning process was carried out using VGG-16 and VGG-19 architectures. Classification successes were compared with architectures without transfer learning.

Generative Adversarial Networks [78] architecture's been used in many areas in recently. It is one of the most used architectures in areas such as virtual image generation, virtual video generation, and natural language processing. In addition to such areas, generative adversarial networks are also useful in data set augmentation by imitating samples in data sets and producing new samples. This feature of architecture is not a frequently encountered field compared to other fields used in the literature. This situation can be explained by many reasons. most importantly, the training process of generative adversarial networks is quite challenging. They can be easily overfit, as well as it requires hyper-parameter tuning, which can even differentiate from class to class in the dataset. Finally, they need a large data set because they will learn the distribution in the dataset they are in and imitate it. Considering that artificial learning-based augmentation of an already large data set is also highly desirable, generative adversarial networks are not used much in the literature for data augmentation. Within the scope of this thesis, the success of the generative adversarial network in data set augmentation will be compared with the conventional data augmentation techniques, and then applied to the dataset augmented with traditional methods to reach the largest and most balanced dataset. RadEch dataset is going to be explained in detail in the next section 3.1.

3.1. Dataset

RadEch, a database developed by the Military Academy in Belgrade, Serbia, is available online [104]. For radars operating in the Ku-band, the Doppler frequencies on this carrier frequency are within the audio band, which means that, through headphones, the radar operator can hear the Doppler frequencies via an audio tone. When a ground moving target is heard for the first time, the sound is instantly recognizable. Data is collected using a pulse-Doppler ground-surveillance radar operates at 16.8 GHz. According to the radar's parameters, it has high

average power and pulse width, a range resolution of 150 meters, an elevation resolution 7.5 degrees, and an azimuth resolution of 5 degrees. By using the radar for target detection and tracking, the target echo could be recorded continuously. The radar and target were set at a short distance (between 100 and 1000m). The moving targets were sighted in line-of-sight despite low vegetation and interference from the sky. All target movements were controlled. In each scenario, one target was recorded at a time. A sampling frequency of 4 kHz was used for the raw radar data with an amplitude of ± 1V. There are .MAT files recorded in the database containing radar echoes from various targets. MAT-files are the data file format of MATLAB software.

The large database of raw audio Doppler signals was created from more than 80 different scenarios. Each scenario was recorded for at least 20 seconds. Approximately 453 real record 4-second interval were collected for each of the five target classes in the database. Data distributions and sample counts of the datasets are shown in Table 3.1. The flow diagram of the system designed within the scope of this thesis is shown in Figure 3.1.

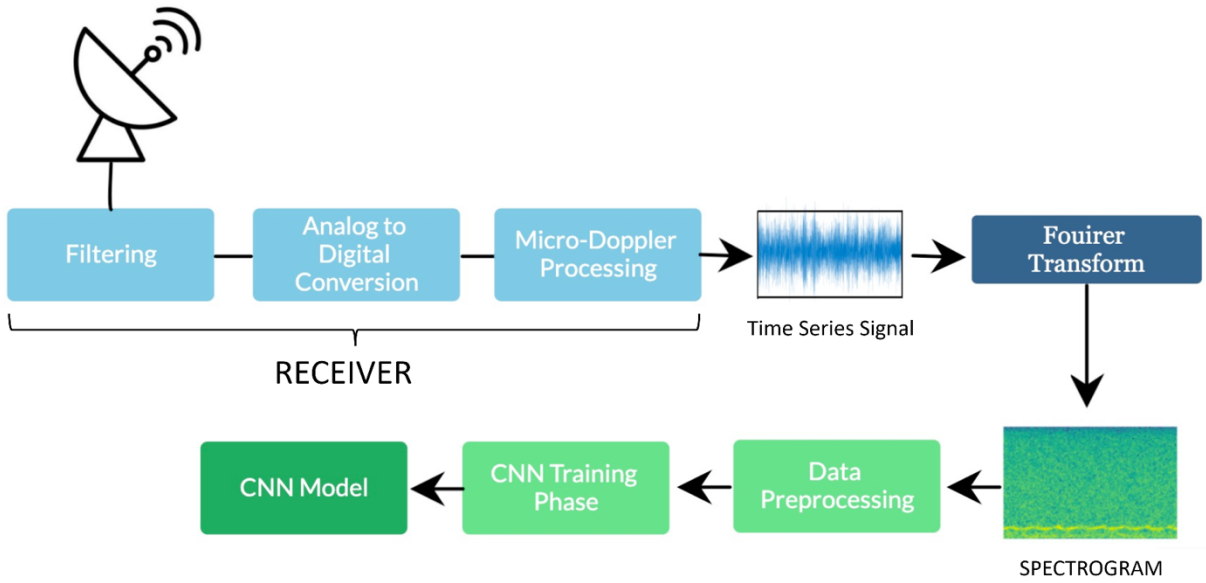

Figure 3. 1 Flow Diagram

Table 3. 1: Dataset Distribution

Class	One person walking	One person running	One person crawling	Group walking	Group running	Wheeled	Truck	Clutter
Samples	99	71	18	124	50	26	47	17

As it can be seen from Table 3.1, the sample numbers of the classes in the data set are quite unbalanced. For example, while the "Group Walking" class contains 124 samples; The "Clutter" class contains only 17 samples, 7 times less. RadEch dataset, which contains 452 samples in total, can be considered relatively small for deep learning architectures. It is important to augment the sample distribution in the dataset so that it is balanced.

In Figure 3.2 the dataset is shown with a simple tree diagram. It can be easily seen through this diagram which movements or vehicles are in which class. There are no subsections in “Clutter” class due to it contains only plants and trees reflections.

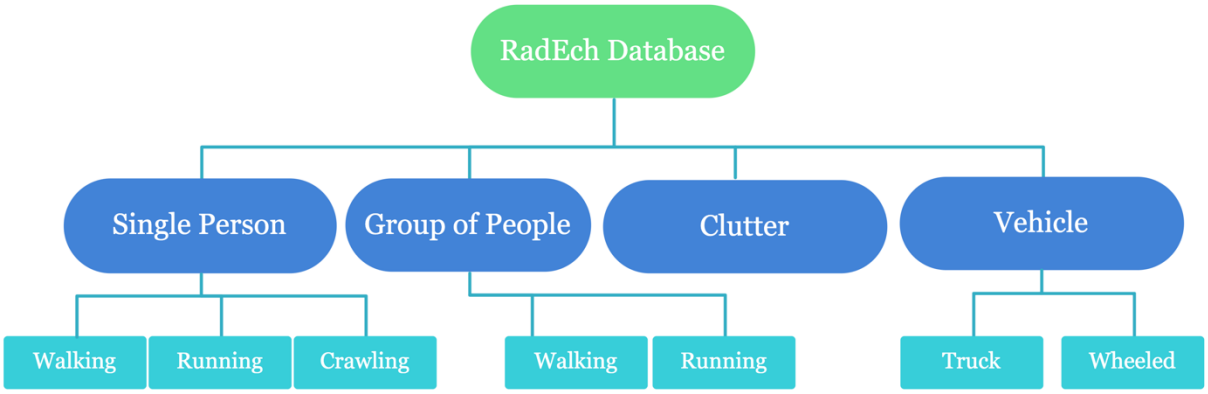

Figure 3. 2 RadEch Database Tree Diagram

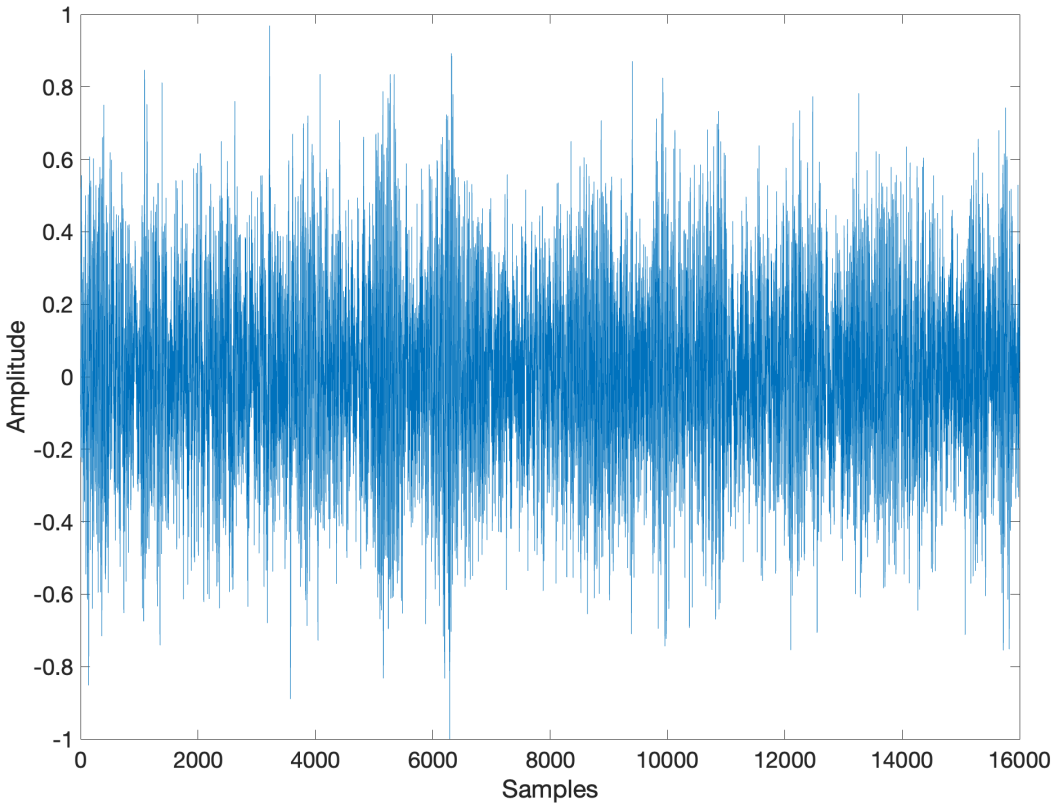

Figure 3. 3 Example of "One Person Walking" Class

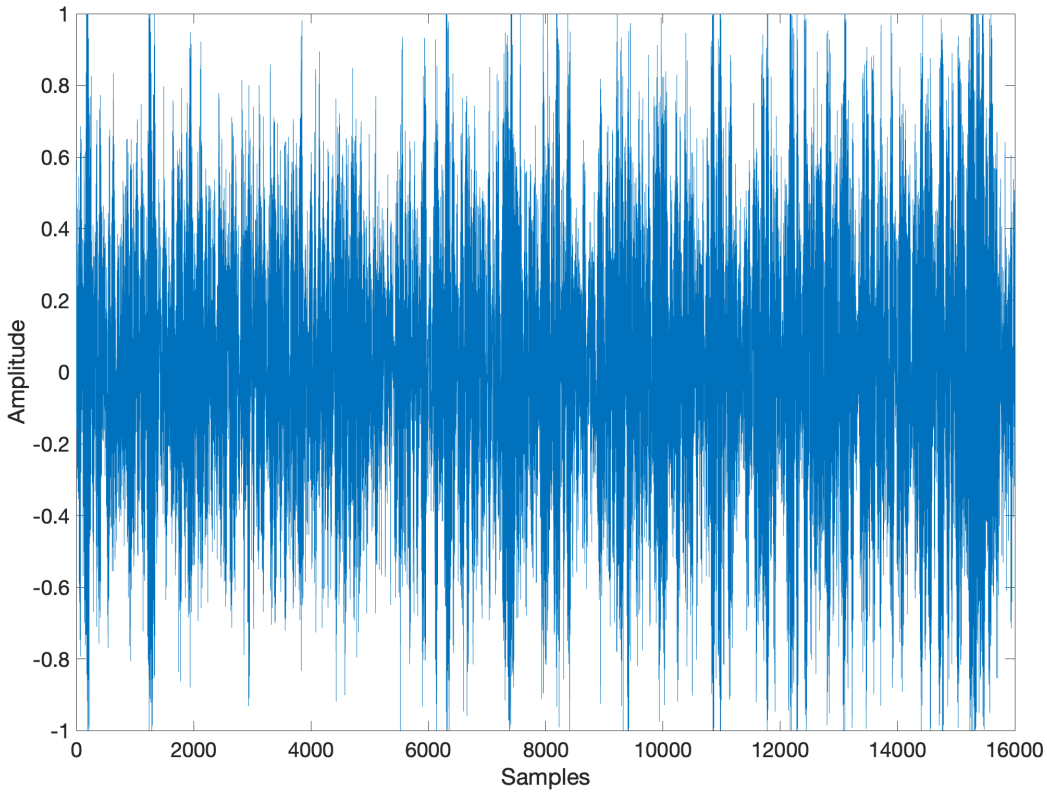

Figure 3. 4 Example of "One Person Running" Class

An example of the "One Person Walking" class is shown in Figure 3.3 Each sample in the dataset contains 16000 subsamples. Considering the sampling rate of the dataset is 4 KHz, these 16000 samples were recorded in approximately 4 seconds. Likewise, an example from the "One Person Running" class is shown in Figure 3.4. If examined carefully, it can be seen that the sample of the "One Person Running" class is more frequent between samples. This explains that the frequency of the signal is higher. A higher micro-Doppler frequency indicates that the reflected signal has a higher velocity of the reflected object. This is an expected situation, because a running person has a higher speed than a walking person.

Section 3.2 will provide detailed information about the spectrograms presented in the dataset. It is relatively difficult to analyse or extract information from a time series signal over a time graph. Therefore, it is more appropriate to use spectrograms.

3.2. Micro-Doppler Signals to Spectrograms Conversion

In this section, micro-Doppler reflection signals will be converted to spectrograms using the short-time Fast Fourier Transform. First, all 452 samples were converted to .wav files. The conversion process was performed with Matlab software. 4 KHz was chosen as the sampling frequency, and 16 was selected as the bit resolution.

As a result of the process, each sample has been converted into 4 second .wav files. Samples converted to .wav file are now ready to be converted to time-frequency domain spectrograms. Unlike .mat file files, .wav file audio files can be easily converted to spectrograms on many platforms. The .wav files were converted to spectrograms via “scipy” [106], an open-source python programming language library. It is also graphically displayed and recorded via “matplotlib”, which is an open-source python library as well. While creating the spectrograms, the non-uniform fast fourier transform (NFFT) value of 200 and the dots per inch (dpi) value of 100 were selected from the selected parameters. Choosing a higher dpi value will increase the image quality of the spectrogram, but the dpi value has been chosen at the optimum level since each higher resolution image will increase the training time. All spectrograms are saved as .png image format.

Table 3.2 shows the process time spent for spectrogram conversion of each class. All operations are carried out on Apple M1 processor with 8 Gb ram and Apple M1 graphic card.

Table 3. 2: Spectrogram Conversion Processing Time

Class	One person walking	One person running	One person crawling	Group walking	Group running	Wheeled	Truck	Clutter
Processing time (s)	14,20	10,31	2,56	17,86	7,19	3,76	6,79	2,50

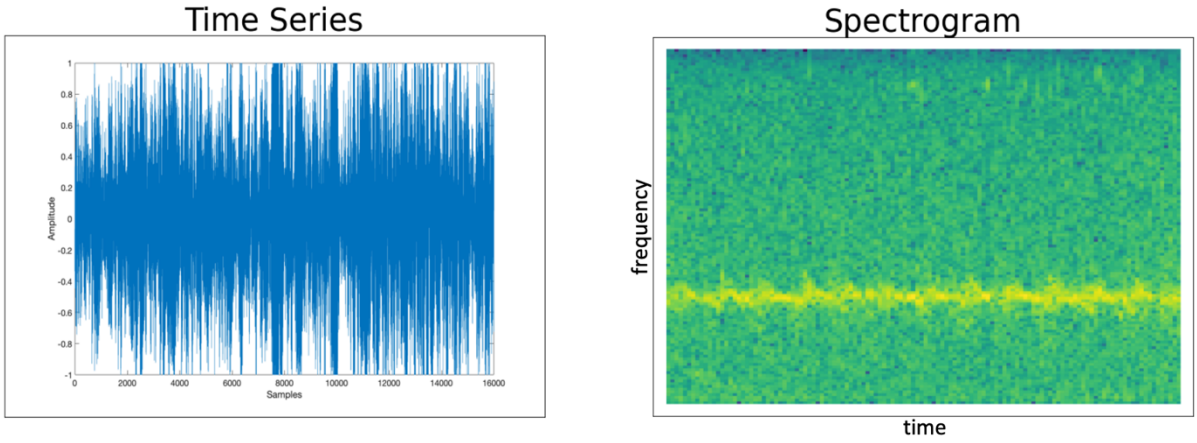

Figure 3. 5 Time Series Signal to Spectrogram

In Figure 3.5, the time series graph and spectrogram of an example from the "One Person, Running" class are shown. Time is plotted on the x-axis, and micro-Doppler frequency is shown on the y-axis. The highest micro-Doppler frequency measured in the RadEch dataset was specified as 2 KHz [104]. The colours in the spectrogram indicate the strength of the signal. Yellow colour means it has higher power than dark blue colour. The yellow colour oscillating in a certain narrow frequency range for 4 seconds in the spectrogram shows the frequency of the micro-Doppler signal reflected from the running person. Within the oscillation frequency, there are all micro-Doppler signals reflected from the body, arms and legs of the human body. Therefore, it looks more like an oscillating line rather than a straight line. All in all, the speed of a running person's arms and legs are constantly changing at the time of running relative to a reference point.

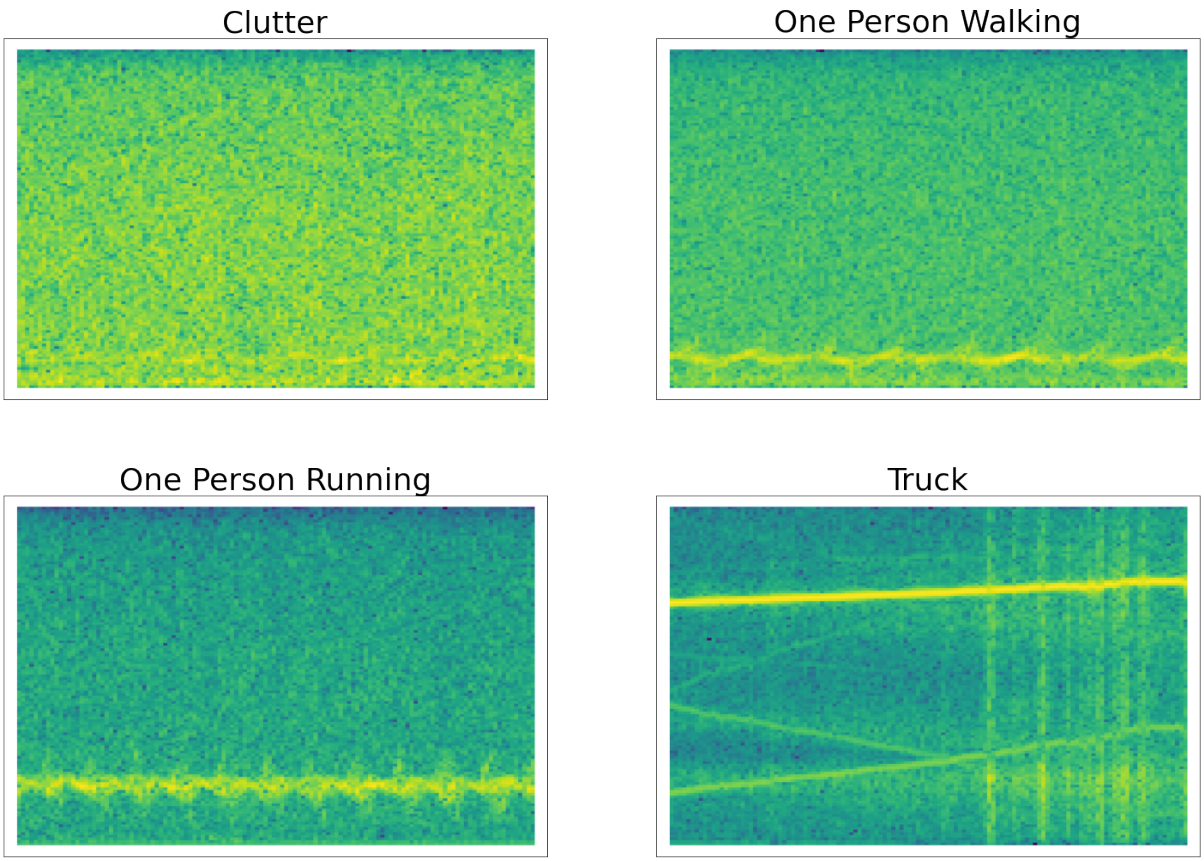

Figure 3. 6 Spectrogram Examples

Figure 3.6 shows examples of spectrograms for four different classes. The Clutter spectrogram was obtained as a result of measurements made in the empty field. It shows micro-Doppler signals from various shrubs and trees. As expected, a high frequency micro-Doppler signal is not seen. There are micro-Doppler signs, which are reflected from the leaves and branches of the plants that move only from the wind and for various reasons, and which are seen intermittently in the spectrogram.

One person walking and one person running spectrograms show micro-Doppler reflections when a person is walking and running. In the spectrogram of a running person, the micro-Doppler frequency is higher than a walking person because the person moves faster. Similarly, fluctuation is greater in the one person running class because the runner's arms and legs will move more frequently. The less fluctuation in the one person walking spectrogram can be explained by the fact that the arms and legs move less when walking compared to running.

The truck spectrogram shows a four-second micro-Doppler reflection of a truck. The reason for having a higher micro-Doppler frequency, unlike other spectrograms, is that the truck

has a higher speed than the human walking and running. The spectrogram here shows a truck with a regularly increasing frequency for 4 seconds. one might interpret this truck as having increased acceleration in four seconds. As a result of truck moving, their signature has a predominant spectral component (Doppler frequency). Wheeled vehicles are compact targets, without sub reflectors moving, so they have a small spectral band around central Doppler frequency. Considering that the Doppler frequency for the target is about 1400 Hz, the car would be moving at about 12.5 m/s (45 km/h).

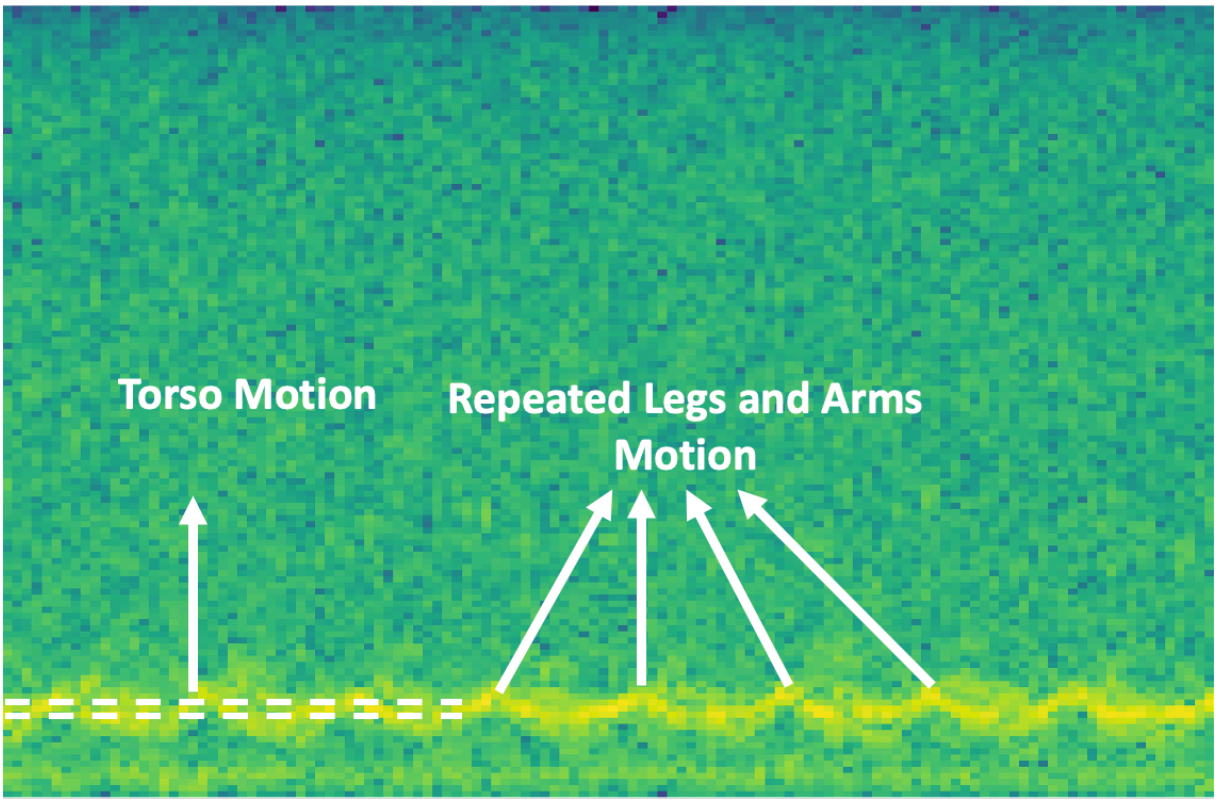

Figure 3. 7 One Person Walking Spectrogram

As mentioned earlier in this section, interpretations of human body movements can be extracted from spectrograms. In Figure 3.7, a spectrogram of the "One Person, Walking" class is examined. Again, as mentioned before, the yellow pixels on the spectrogram are micro-Doppler radar signals reflected from the walking human body. These yellow pixels can be seen to oscillate periodically. The reason for this oscillation is the movement of the arms and legs of the human body while walking. If we imagine a walking person, the body moves at a more constant speed when looking at the overall movement, no matter how oscillating the arms and legs move. This is the explanation why the yellow pixels oscillating in spectrograms oscillate around a thicker line rather than a sinusoidal curve, that is, the movement of the human torso.

If the spectrograms given so far are carefully examined, it can be seen that there are reflections from sources other than the original source of the signals reflected to the receiver. These reflections can be seen as noise in the spectrograms. There may be various reasons for these reflections. During the measurement, there may be reflections from trees, moving leaves of plants, surrounding structures (which may cause a phase change in the signal), and other creatures such as cats and dogs, apart from the region of interest. This is generally undesirable. If we train the spectrograms as such, with deep learning architectures, the architecture will also learn the noise in the spectrogram. This may reduce the generalization ability of the architecture, as well as reduce the classification accuracy during testing. To avoid this situation, various noise reduction techniques can be applied to spectrograms. The noise reduction technique applied in this study is explained in detail in Section 3.3.

3.3. Noise Reduction

Noise can be seen as a big problem in image datasets. While noise in images is sometimes undesirable, sometimes it is artificially added to the image dataset so that the deep learning architecture does not overfit. Traditional noise reduction techniques are not suitable for every dataset. For example, image noise can be reduced by using Auto-Encoder. This may be a viable technique for images containing frames from everyday life, but it is not very suitable for datasets that require sensitive pre-processing such as micro-Doppler radar datasets, as it will cause information loss.

Blurring images is another noise reduction technique. Although image blurring seems to be more suitable for micro-Doppler radar dataset than noise reduction with Auto-Encoder, its applicability to RadEch dataset is debatable. Considering that the spectrograms seen in Figure 3.6 are blurred, the data in the spectrogram carries will change. It will make the noise in the spectrograms more invisible by mapping them to neighbouring pixels, but the yellow pixels that carry the actual information will be mapped to the neighbouring pixels and their area will increase. This may cause the "One Person Walking" class to be confused with the "Group Walking" class. Because the "Group Walking" class contains micro-Doppler marks reflected from the bodies of multiple people when they move together, the yellow pixels in their spectrograms cover a larger area.

The noise reduction technique proposed for the RadEch dataset in this study is averaging all class in the dataset. In other words, finding the average spectrogram of each class in the dataset and finding the difference between this average spectrogram image and each sample in the dataset. Due to the lack of much difference between spectrograms of the same class, besides

the information sections, this procedure is applied. Therefore, if all spectrograms in a class are averaged, the noise zones in the spectrograms will generally remain the same. Figure 3.8 shows where the noises are on the spectrograms. In these parts, the desired power is zero or close to zero. Adapting this to the spectrogram is that the noise zones are dark blue.

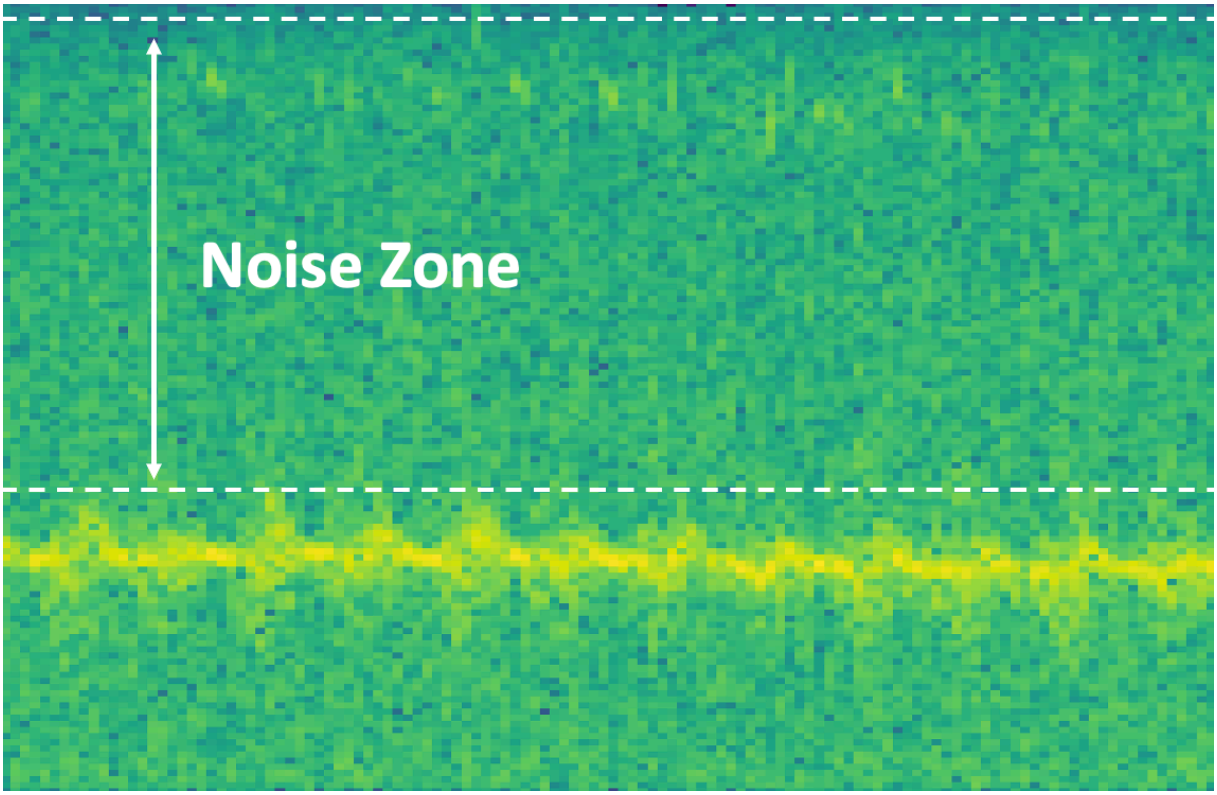

Figure 3. 8 Noise Zone in the Spectrogram

The average image of 71 spectrograms for the "One Person, Running" class was created and shown in Figure 3.9.

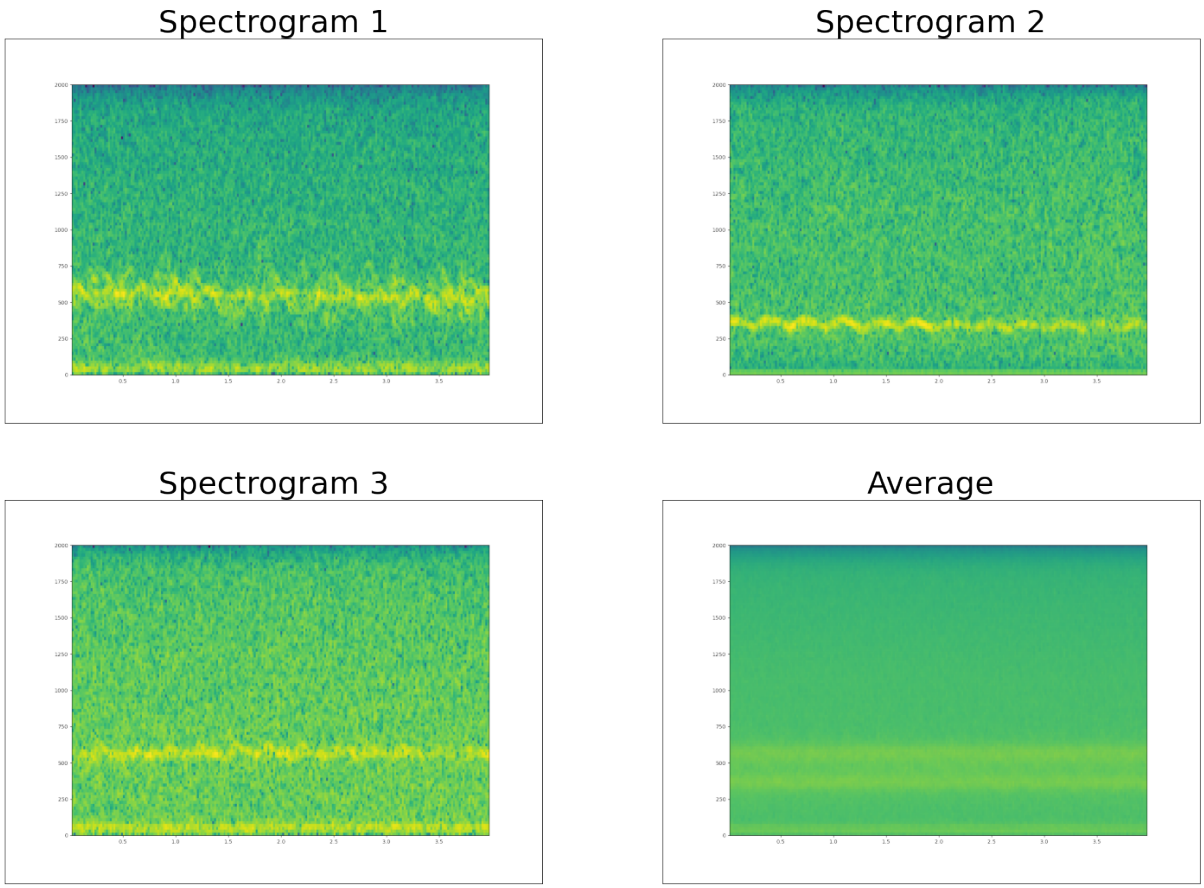

Figure 3. 9 Average Spectrograms of "One Person, Running" Class

The average image of all spectrograms in the dataset was created. PIL [108] and NumPy [109] libraries, which are open-source python libraries, are used to get the average of the images. In the next step, the difference of each sample in all classes with the average image was taken from the actual spectrogram. The differences between the spectrograms and average spectrogram given in Figure 3.9 and the average spectrogram are shown in Figure 3.10.

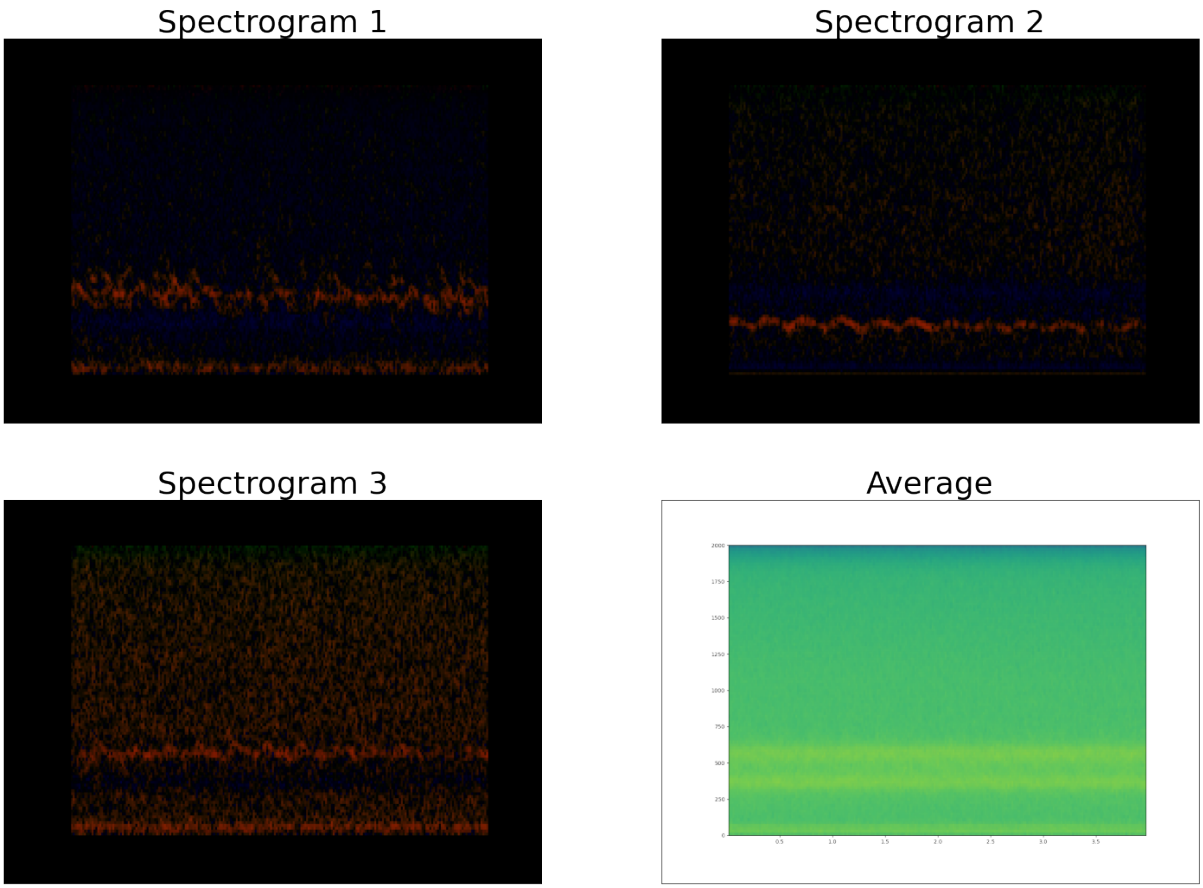

Figure 3. 10 Spectrograms Subtraction with Average Spectrogram

Belonging to the "One Person, Running" class given in Figure 3.9 and 3.10; The pixel-by-pixel subtractions of Spectrogram 1, Spectrogram 2 and Spectrogram 3 are shown in Figure 3.11. Thus, the noise reduced version of the dataset is obtained for each class. In Spectrogram 2 and Spectrogram 3, it can be seen that the colour difference in the noise zones decreased and turned into a more stable green colour. In Spectrogram 1, this situation was less common than the others. The reason is that the noise scale of Spectrogram 1 is wider and more. Therefore, it can be deduced that the noise cannot be reduced much in the noise reduction process of the samples with large and high noise scale. Fortunately, not every sample in the dataset has such high noise as in Spectrogram 1. The few high noise samples in the dataset are negligible compared to the benefit of noise reduction. Noise reduction process is carried out for whole 452 samples in the dataset. All noise reduced samples are saved as a new dataset and stored for the classification process in next sections.

In Section 3.4, dataset augmentation process will be explained. Apart from the noise reduction operations made in this section, traditional and learning-based data augmentation techniques will be applied to the raw dataset.

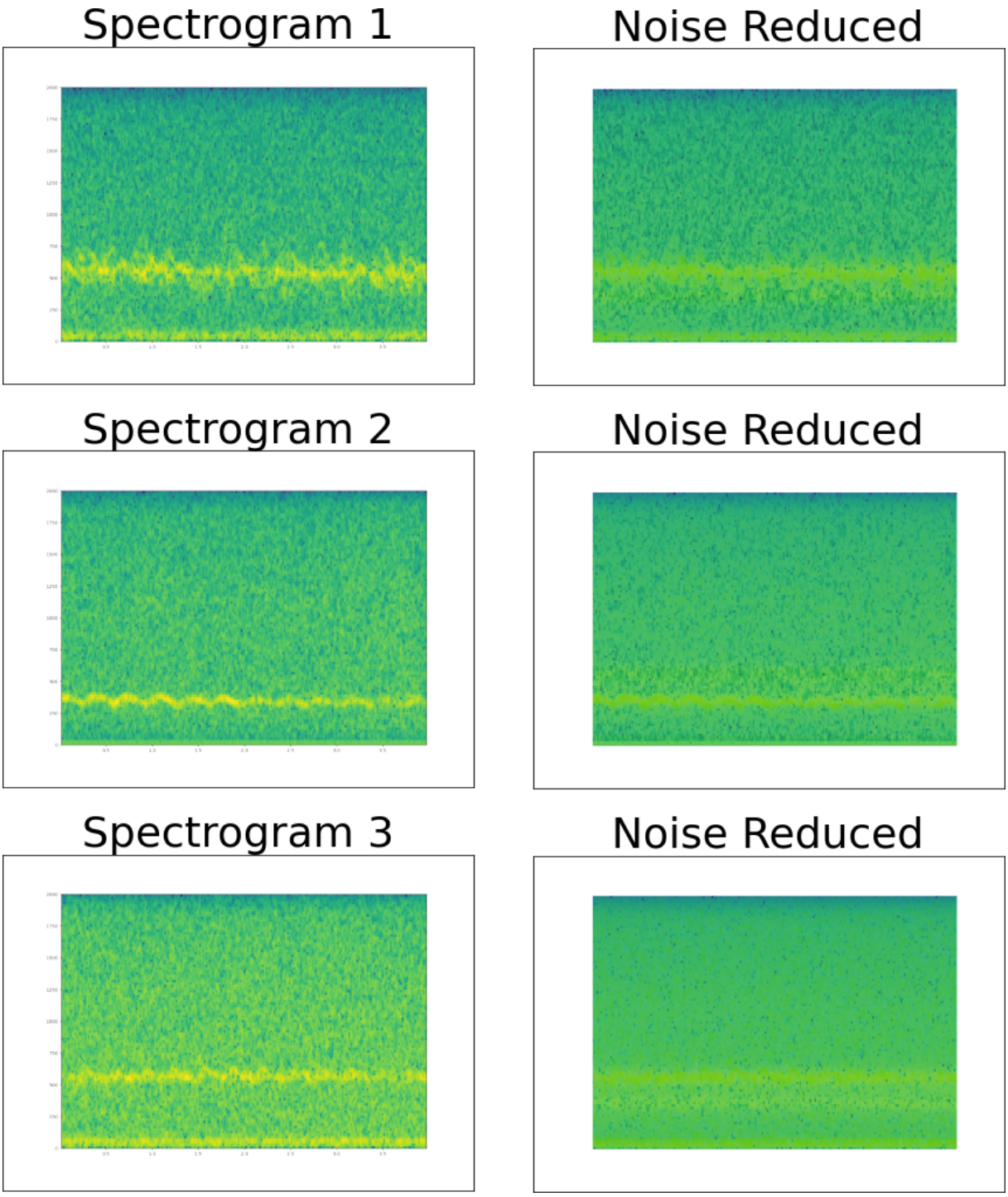

Figure 3. 11 Regular and Noise Reduced Spectrograms

3.4. Data Augmentation

Deep learning tasks have attracted great interest since they were first used in the literature. Researchers have published many studies for years in order to develop deep learning models, and they have tried to find solutions about the problems encountered while training these models. One of the most important of these problems is the overfitting problem. Overfitting briefly means that the model loses its generalization ability and memorizes the dataset; While it has good classification accuracy on the dataset, it cannot show the same success on different datasets. Researchers have proposed many methods for preventing this situation. One of them is to finish the model training early; Another is to reduce the model depth-size or to reduce the number of neurons in the model. These recommendations are important for limited datasets. But the best way to avoid overfitting is to feed the model with more and more representative data. Finding data with this specification is quite time consuming and requires huge cost. At this point, researchers suggested data augmentation techniques.

Data augmentation can reduce overfitting issue in training phase of deep learning models. There are many data augmentation techniques. These techniques can be basically divided into two: traditional data augmentation techniques and learning-based data augmentation techniques. The images in the dataset can be rotated at different angles, flipped vertically and horizontally, scaled, added noise, and manipulated in colour as well as other traditional techniques. Some of these techniques are suitable not only for image datasets, but also for time-series datasets and numeric datasets. Another data augmentation technique used in the literature is learning-based data augmentation. In this technique, the dataset is trained with Generative deep learning models and is expected to generate new data that mimics the dataset. These new data generated are added to the dataset and the dataset is augmented. Examples of deep learning models used in these techniques are Generative Adversarial Network and Auto-encoders which of these techniques can be applied to dataset are depending entirely on the dataset. While flipping and rotation are not suitable for some datasets, learning-based data augmentation techniques may be suitable for some datasets. Too much data augmentation can also cause overfitting problem. This trade-off should be fine-tuned.

3.4.1. Time Series Data Augmentation

Within the scope of this study, traditional data augmentation techniques suitable for RadEch dataset were applied. When the dataset is analysed as time-series, it is seen that it is a time-series dataset with 452 samples consisting of 4-second micro-Doppler recordings. Thus, flipping and shifting techniques seem appropriate for most of the classes in the dataset. Shifting

technique cannot be applied to "Truck" and "Wheeled" classes since they have samples with a speed that changes within four seconds, unlike other classes. The problems that will arise if these techniques are applied to these classes will be shared later in this section.

Unlike Esra Alhadhrami et al. [103] study, data augmentation was used in this study by trying to equalize the number of samples contained in each class in the dataset. The imbalance in the dataset was tried to be balanced by doing data augmentation. Esra Alhadhrami et al. [103] shifted each class in the dataset at 15 different rates and flipped it once. This means that the number of samples contained in each class in the dataset has increased by 17 times. Although it may seem like a good situation at first, this will not handle the imbalance in the dataset, and that much a data augmentation may cause overfitting again. Shifting factors that used in study is shown in Table 3.3.

Table 3. 3: Shifting Factors

Class	**Shifting Factors**	**Number of Samples**	**Number of Samples after Augmentation**
Clutter	50, 100, 250, 400, 550, 700, 850, 1000, 1150, 1300	17	187
Group, Running	250, 550, 850, 1150	50	250
Group, Walking	400, 1000	124	372
One Person, Crawling	50, 100, 250, 400, 550, 700, 850, 1000, 1150, 1300	18	198
One Person, Running	400, 850, 1300	71	284
One Person, Walking	400, 1000	99	297

In Figure 3.12, a circular shifted time-series representation of 400, 850 and 1300 samples of a sample belonging to the "One Person, Running" class is given, respectively. In Figure 3.13, the spectrograms of the signals shown as these time-series are shown in the same order.

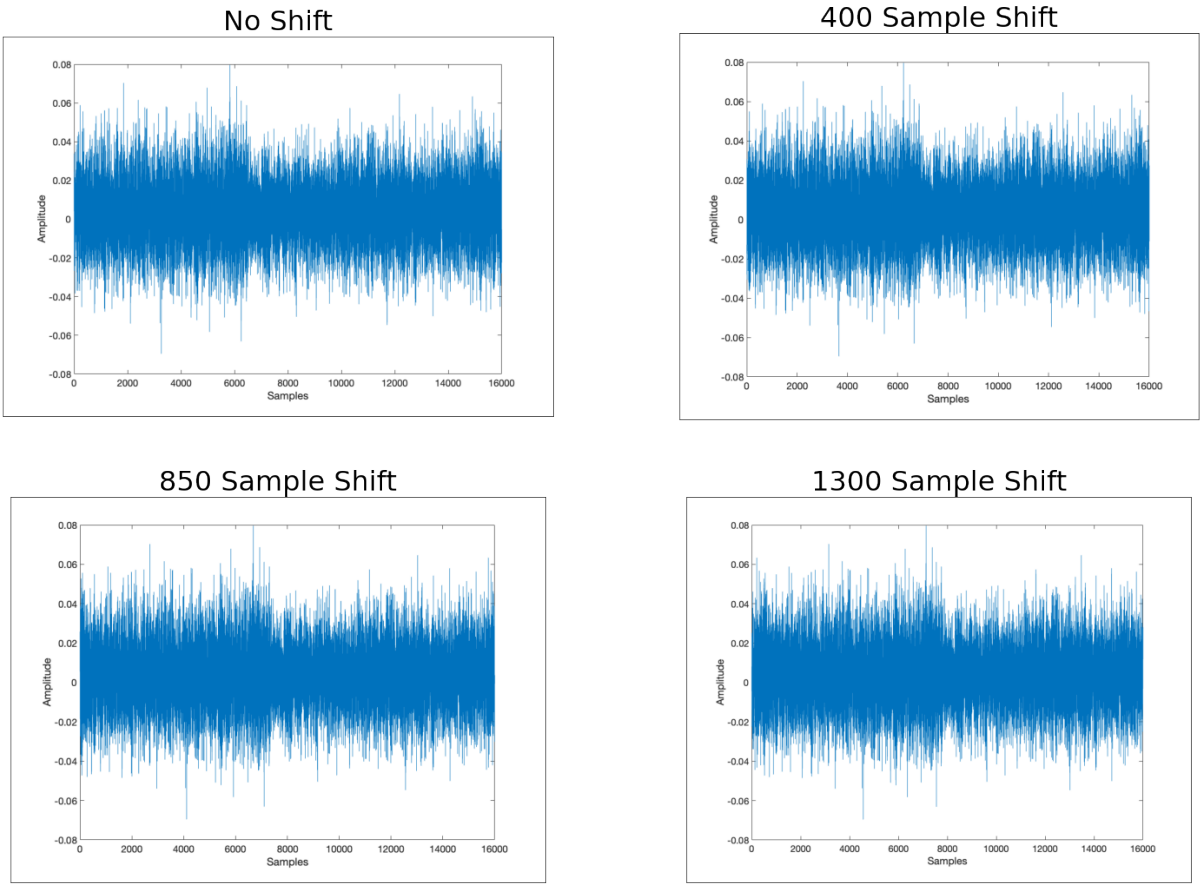

Figure 3. 12 Sample Shifting

If you look carefully at Figure 3.12 and Figure 3.13, the shifting process in the time domain can be seen. What should be considered here is the contribution of the shifting process to the model. The spectrograms in Figure 3.13 are actually micro-Doppler reflections of a single person running. As long as, we do not exaggerate the number of shifting operations, we can actually use it as if it belongs to four different people. But if we increase the number of shifting processes too much for the dataset to get bigger, we get a lot of shifted spectrograms of the same person at the same time. The point to be noted here is that although each person's running spectrogram has the same distribution to a certain extent, it also has a different pattern person by person. Shifting the spectrograms means shifting the noise in the spectrogram. In over augmented datasets, the model also learns these noises and may misinterpret spectrograms without noise with this pattern. As mentioned earlier in this section, the "Truck" and "Wheeled" classes are not shifted. The reason for this is that Truck and Wheeled vehicles have a much more compact structure compared to the human body; speed can change more according to

human movements. This velocity change corresponds to an unsteady yellow line in the spectrograms.

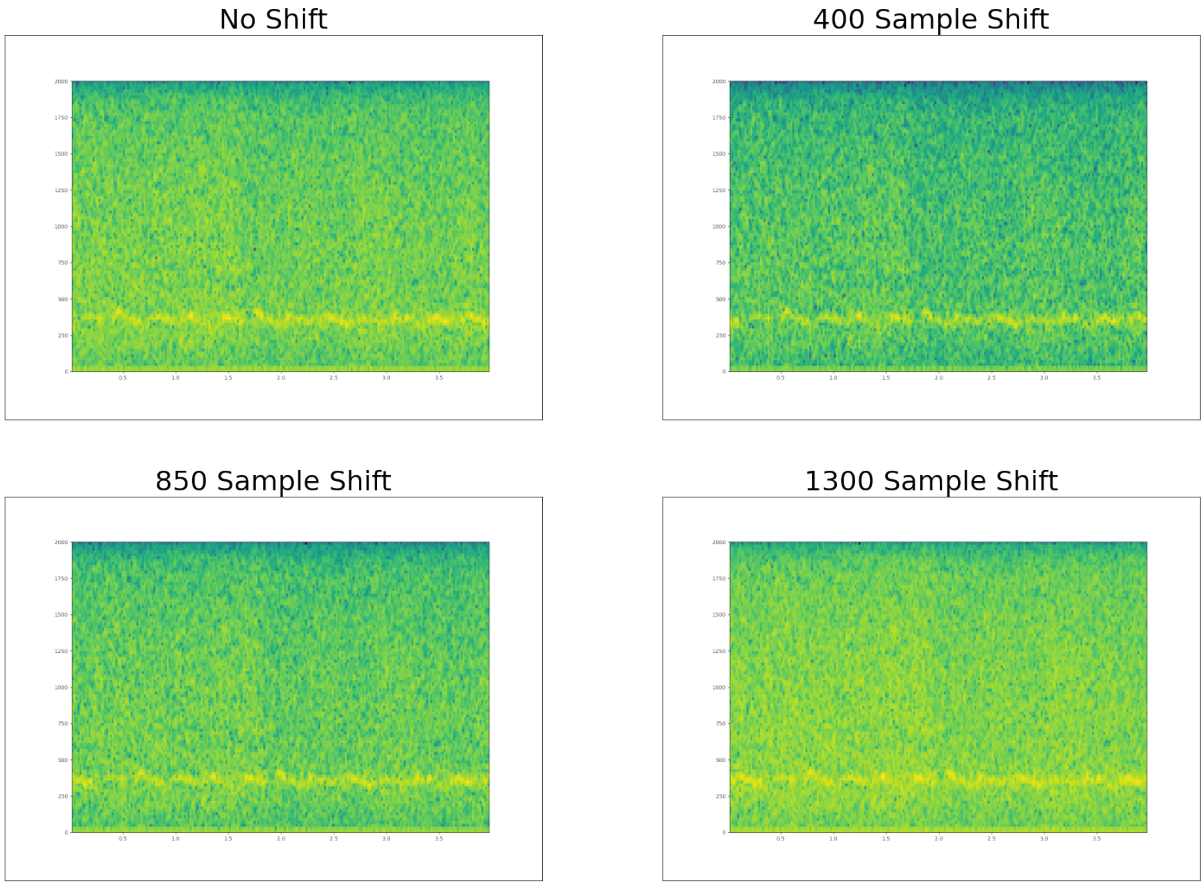

Figure 3. 13 Sample Shifting Spectrogram

If circular shifting is applied to such spectrograms, we get samples with a sharp jump from the micro-Doppler frequency in the time axis of the spectrogram. In reality, it means the speed of a Truck moving at a constant speed or with a certain acceleration decreases or increases by a certain amount in zero seconds. The spectrogram representation of this situation is shown in Figure 3.14. The first purpose of the data augmentation process was to increase the dataset with samples similar to the samples in the dataset and avoid overfitting. The first requirement of data augmentation is that all artificially generated samples are as realistic as possible. If samples unrelated to the dataset are generated and added to the dataset, only noise is added to the dataset and reduces the classification success. At this point, a situation like Figure 3.14 is not actually possible. Therefore, circular shift is not suitable for classes like "Truck" and "Wheeled". A different method was used to augment the "Truck" and "Wheeled" classes. When

the samples contained in these two classes are examined; It can be seen that it would be more appropriate to shift the spectrograms in the frequency axis rather than in the time axis. Samples within the "Truck" and "Wheeled" classes are generally those that do not have a very broad frequency spectrum, such as a tone signal. These tone signals represent micro-Doppler reflection signals. These signals contain information about the object's speed. If the spectrograms are shifted up in the direction of the frequency axis, a faster object can be obtained, and if they are shifted down, a slower object can be obtained. In terms of its real-life reflection, this situation is quite realizable. Figure 3.15 shows a sample of the "Truck" class, shifted up and down.

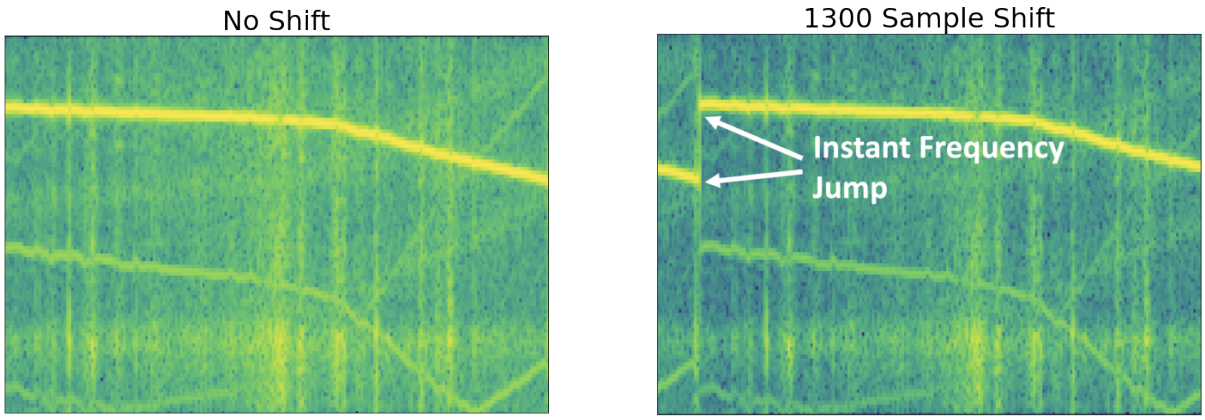

Figure 3. 14 Instantaneous Frequency Jump on Spectrogram

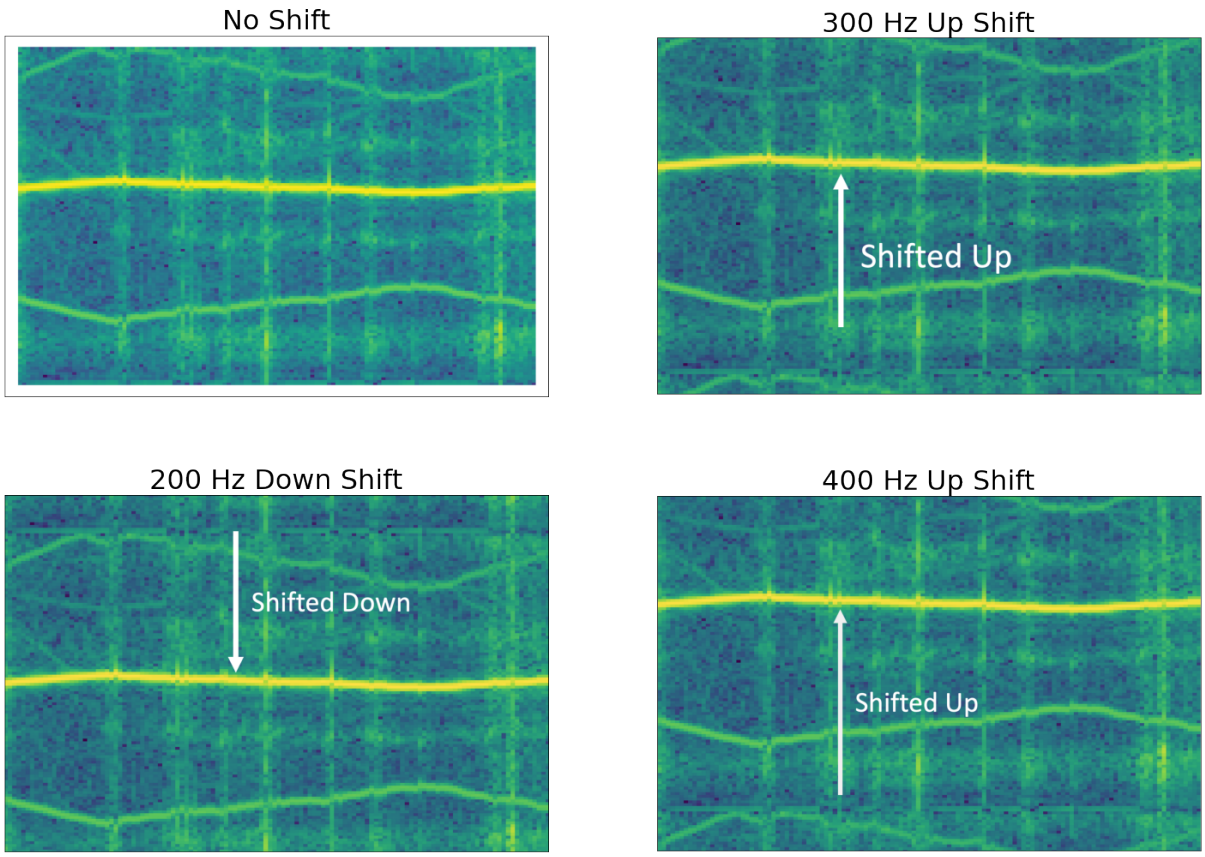

Figure 3. 15 Spectrogram Up and Down Shift

Figure 3.15 shows a sample of the "Truck" class, shifted up and down. The first spectrogram in Figure is the normal state of the sample. The second sample was shifted up to 300 Hz. This means that the speed of the object is increased by approximately 9.6 km/h. The third spectrogram was shifted down by 200 Hz and finally the fourth spectrogram was shifted up by 400 Hz. As can be understood from these examples, samples belonging to the "Truck" and "Wheeled" class can be augmented with this technique. The final version of the augmented RadEch data is shown in Table 3.4. Finally, total number of samples reached 1953. All these augmented spectrograms were saved for classification in later sections.

Table 3. 4: Number of Samples

Class	**Shifting Factors**	**Number of Samples**	**Number of Samples after Augmentation**
Clutter	50, 100, 250, 400, 550, 700, 850, 1000, 1150, 1300	17	187
Group, Running	250, 550, 850, 1150	50	250

Group, Walking	400, 1000	124	372
One Person, Crawling	50, 100, 250, 400, 550, 700, 850, 1000, 1150, 1300	18	198
One Person, Running	400, 850, 1300	71	284
One Person, Walking	400, 1000	99	297
Wheeled	200, 300, 400 Hz Up; 200 Hz Down	26	130
Truck	200, 300, 400 Hz Up; 200 Hz Down	47	235

3.4.2. Learning-Based Data Augmentation

After time-series data augmentation was done in the previous section, data augmentation will be done with the generative deep learning model in this section. Generative Adversarial Networks is used in this section for data augmentation. Due to the relatively small size of the Raw dataset, it is actually not a very suitable dataset for GAN. However, GAN model will be trained experimentally, and the data obtained will be recorded for classification. The GAN architecture used in this thesis is shown in Figure 3.16 and Figure 3.17 respectively. The training of the GAN model will be done on a class basis. In other words, each class will be trained as if it is a single dataset, and new samples will be produced with the Generator at the end of the training.

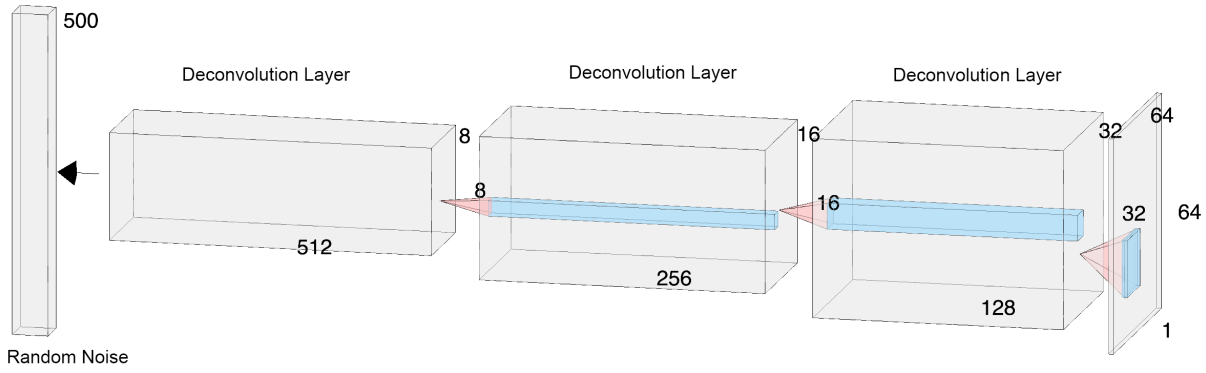

Figure 3. 16 Generator Architecture

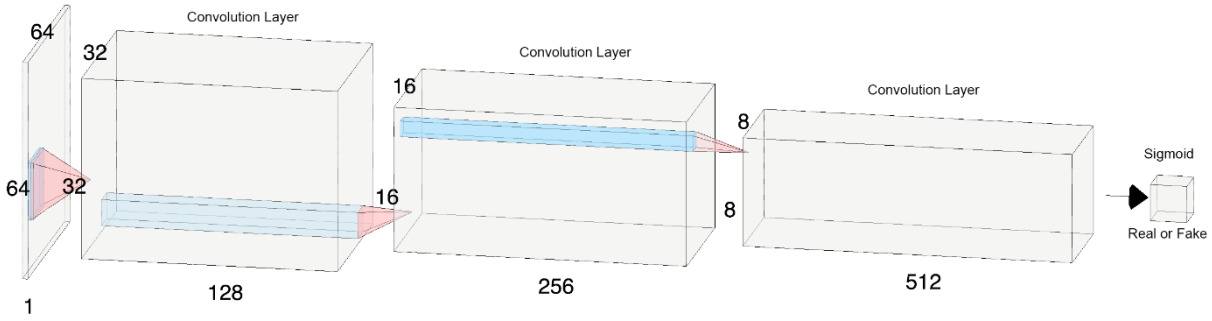

Figure 3. 17 Discriminator Architecture

The Generator architecture seen in Figure 3.16 starts production by inputting a random noise vector with a length of 500. The random noise vector passes through deconvolution layers of different sizes, respectively, and finally turns into a 64x64 image with 3 channels. This image enters the Discriminator network as an input, as seen in Figure 3.17, and is subjected to feature extraction by passing through the convolution layers. Finally, it is decided whether it is real or fake by coming to the sigmoid layer. A more detailed code-script summary of the Generator and Discriminator networks is given below:

```
Generator (
  (main): Sequential(
    (0): ConvTranspose2d(500, 512, kernel_size=(4, 4), stride=(1, 1), bias=False)
    (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): ConvTranspose2d(512, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (4): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): ConvTranspose2d(256, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (7): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (8): ReLU(inplace=True)
    (9): ConvTranspose2d(128, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (10): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (11): ReLU(inplace=True)
    (12): ConvTranspose2d(64, 3, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (13): Tanh()
  ))

Discriminator (
  (main): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (1): LeakyReLU(negative_slope=0.2, inplace=True)
    (2): Dropout(p=0.4, inplace=False)
    (3): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
```

```
  (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (5): LeakyReLU(negative_slope=0.2, inplace=True)
  (6): Dropout(p=0.4, inplace=False)
  (7): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
  (8): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (9): LeakyReLU(negative_slope=0.2, inplace=True)
  (10): Dropout(p=0.4, inplace=False)
  (11): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
  (12): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (13): LeakyReLU(negative_slope=0.2, inplace=True)
  (14): Dropout(p=0.4, inplace=False)
  (15): Conv2d(256, 1, kernel_size=(4, 4), stride=(1, 1), bias=False)
  (16): Sigmoid()
))
```

The GAN architecture is built and trained using a PyTorch [110] a python library. The parameters that used during training of the architecture are given in Table 3.5. However, the parameters given in this table were not used in this way in the training of every class. There are parameters that differ greatly from class to class. Ultimately, training the GAN model is quite difficult as it is easily overfitted. Therefore, the parameters had to be fine-tune even when training different classes within the same dataset. Table 3.5 shows the hyper-parameters of the "One Person, Crawling" class.

Table 3. 5: Hyper-parameters of GAN

Quantity of workers for dataloader	2
Batch size	4
Spatial size of image	64
Quantity of channels of image	3
Size of latent vector	500
Size of feature maps in Generator	64
Size of feature maps in Discriminator	32
Quantity of training epoch	150
Learning rate for optimizer	0,0001
Beta1 hyper-parameter for Adam optimizer	0,5

After training the GAN model for each class, new samples were generated. The numbers of samples generated are shown in Table 3.6.

Table 3. 6: Number of Generated Samples

Class	**Number of Samples**	**Number of Generated Samples**
Clutter	17	160
Group, Running	50	200
Group, Walking	124	450
One Person, Crawling	18	160
One Person, Running	71	250
One Person, Walking	99	350
Wheeled	26	200
Truck	47	200

After the new samples were produced and imported into the class they belonged to, the dataset increased to a total of 2422 samples. A spectrogram sample of the "Truck" class is shown in Figure 3.18. In order to compare the similarity, 16 spectrograms produced for the "Truck" class are shown in Figure 3.19.

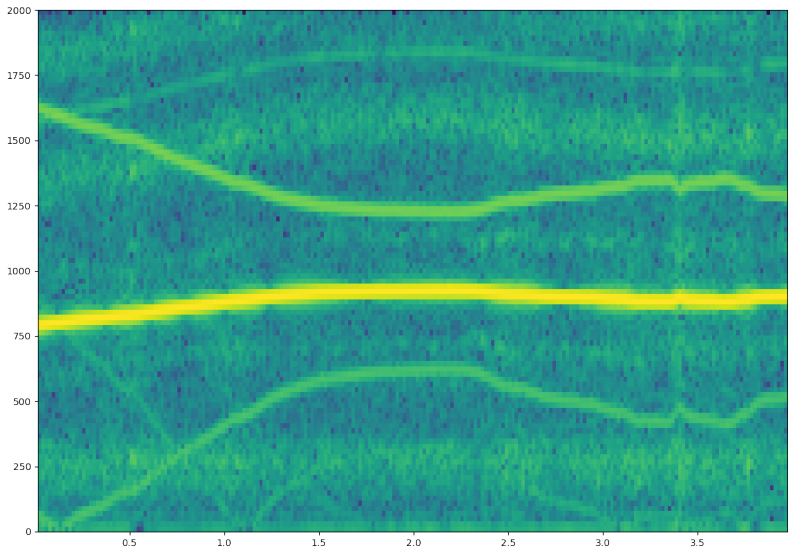

Figure 3. 18 Spectrogram Example for Truck

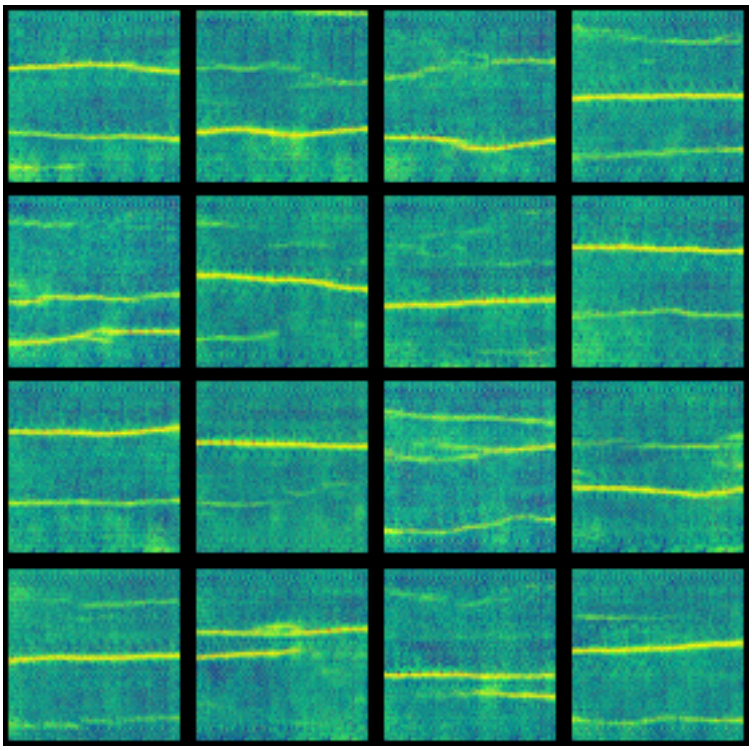

Figure 3. 19 Generated Spectrograms for Class of Truck

The GAN model used can produce 64x64 images. This size can be enlarged. But creating a higher size image will require more computational resources. Besides, the training time will increase considerably. Since the resolution of the spectrograms created in RadEch dataset is not very high and their size will be reduced by resizing during the classification process; 64x64 image size is considered sufficient. As can be seen from the spectrograms produced in Figure 3.19, the GAN model learned the distribution of the "Truck" class well and generated very similar but different samples. The samples are very similar to the samples contained in the raw RadEch dataset. The effect of the generated samples on classification accuracy will be explained in the following sections. In Figure 3.20, the change of the Loss parameter according to iteration during the training of the "Truck" class is shown for both the Generator and the Discriminator Network.

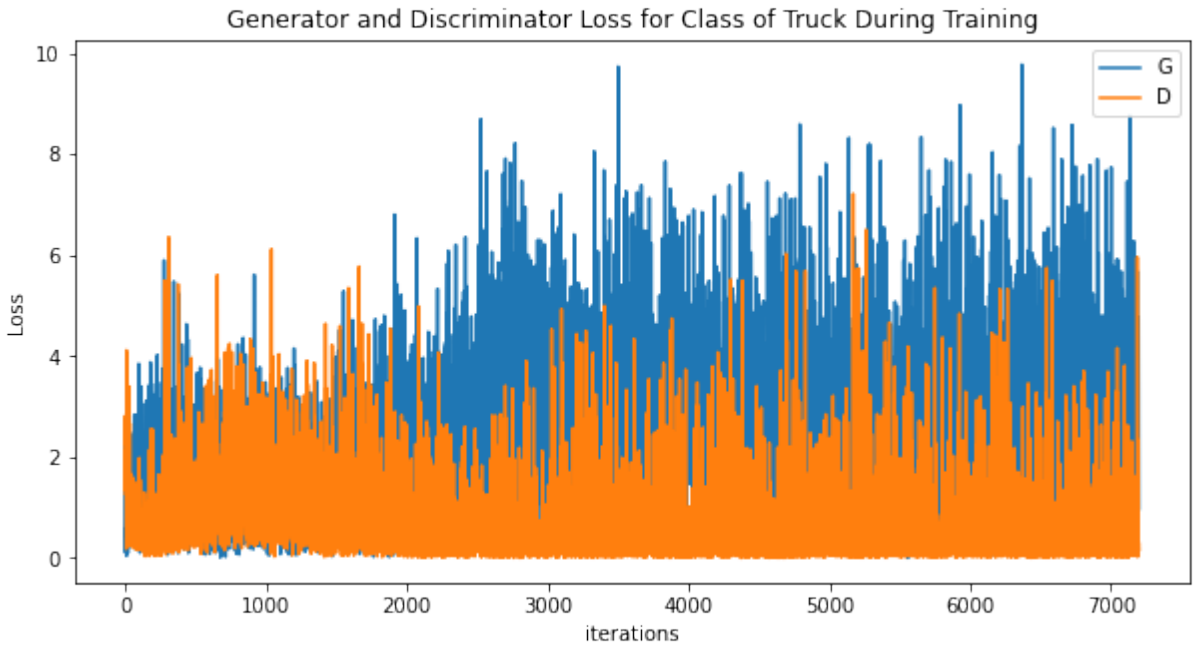

Figure 3. 20 Loss over Iteration Graph

3.5. Network Architecture

In this part of the thesis, deep learning models that will classify the pre-processed dataset will be explained in detail. Due to the relatively small size of the dataset, a very deep model was not needed. Training small datasets with large deep learning models takes time and requires computational cost. Also, small datasets trained with deep models are likely to be overfitting. Three different CNNs were used to classify the pre-processed dataset. One of them is the CNN network, which has five layers of Convolution and three fully connected layers, created for this dataset. The other two models are VGG-16 and VGG-19 [105] architectures. The VGG-16 and VGG-19 architectures are pre-trained models. Both these architectures are trained with ImageNet [111] database. The weights obtained after this training have been saved and are publicly accessible. In this thesis, VGG-16 and VGG-19 will be trained with both pre-trained weights and random weights. Thus, when the models, which are trained on ImageNet, are trained on the RadEch dataset with transfer learning, performance of transfer learning will be evaluated and compared. The CNN architecture used in this thesis is shown in Figure 3.21.

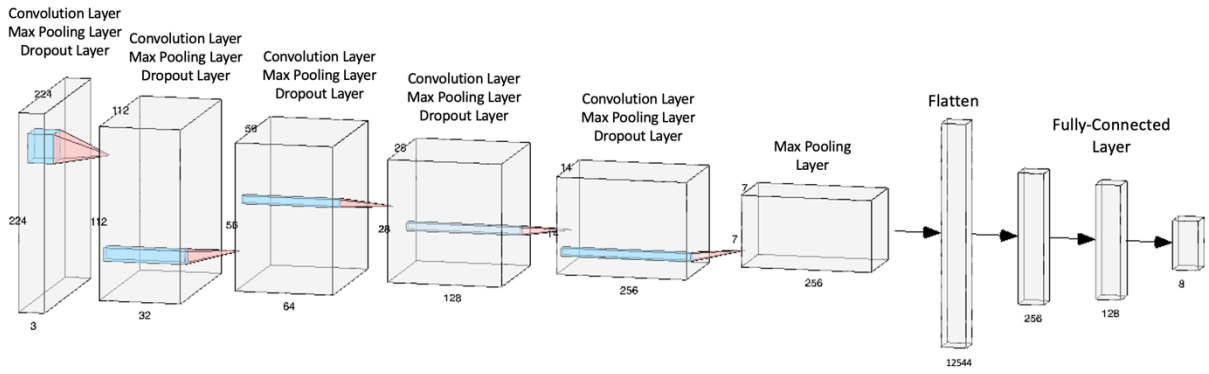

Figure 3. 21 Five Layer CNN Architecture

Table 3.7 illustrates specifications of the Model shown in Figure 3.21. All spectrograms with and without pre-processing in the dataset have large image sizes. Training CNN with high size images increases computational cost. Feature extraction can be done by reducing the image sizes. For this reason, all spectrograms with different image sizes are rescaled before being given to the model for training. All spectrograms are rescaled to 224 x 224 x 3 size before feeding to the CNN model shown in Figure 3.21. In this metric, "224" represents the pixel height and width of the image, and "3" represents the image channel. By using rescaling, the need for lower computational cost is met and the features are extracted from the spectrograms. Rescaled spectrograms will be passing through a 5-layer convolution network for feature extraction. These 5 layers all contain convolution layers and max pooling layers. In convolution layers, convolutions of spectrograms are taken through convolution filters. Each convolution filter extracts an image from the result of the convolution process. This means that for each spectrogram, there are as many images as the number of filters in each layer. While larger features of the spectrograms are extracted in the first layers of the model, smaller features are extracted towards the last layers.

Table 3. 7: CNN Architecture Specifications

Layer	**Output Shape**	**Number of Parameters**
Rescaling_1	224 x 224 x 3	0
Conv2d_1	224 x 224 x 16	448
Max Pooling_1	112 x 112 x 16	0
Dropout_1	112 x 112 x 16	0

Conv2d_2	112 x 112 x 32	4640
Max Pooling_2	56 x 56 x 32	0
Dropout_2	56 x 56 x 32	0
Conv2d_3	56 x 56 x 64	18496
Max Pooling_3	28 x 28 x 64	0
Dropout_3	28 x 28 x 64	0
Conv2d_4	28 x 28 x 128	73856
Max Pooling_4	14 x 14 x 128	0
Conv2d_5	14 x 14 x 256	295168
Max Pooling_5	7 x 7 x 256	0
Flatten_1	12544	0
Dense_1	256	3211520
Dense_2	128	32896
Dense_3	8	1032

"Max pooling" layers reduce the size of images by a process called "pooling" in response to the increasing number of images in layers of model. The pooling process in this model can be thought of as taking one of the maximum pixel values of each neighbouring 2x2 pixel in the image and reducing it to a single pixel. An image with a size of 224 x 224 will turn into 112 x 112 after passing through the max pooling layer in this model. The dropout layer in the first three layers is there to prevent the model from being overfitted. After each back-propagation operation, the model cuts the connections between different layers; and actually, prevents overfitting. After passing through the last max pooling layer, the spectrograms are combined into a single vector in the flatten layer. As such, each sample is ready to be trained in a fully connected layer. In the "Dense_3" layer, which is the last layer of the model, classification is done for 8 classes.

Other architectures used to classify RadEch dataset are VGG-16 and VGG-19. VGG16 is a convolutional neural network model proposed by K. Simonyan et al. [112]. In an ImageNet dataset consisting of 14 million images belonging to 1000 classes, the model achieves 92.7% accuracy in its top-5 tests. It is one of the very famous models presented at ILSVRC-2014. It is also based on CNNs. When compared with AlexNet [113], major improvements of VGG include the use of either a large kernel-size filter (size 11 in the case of VGG) or multiple (3*3) kernel-size filters one after another.

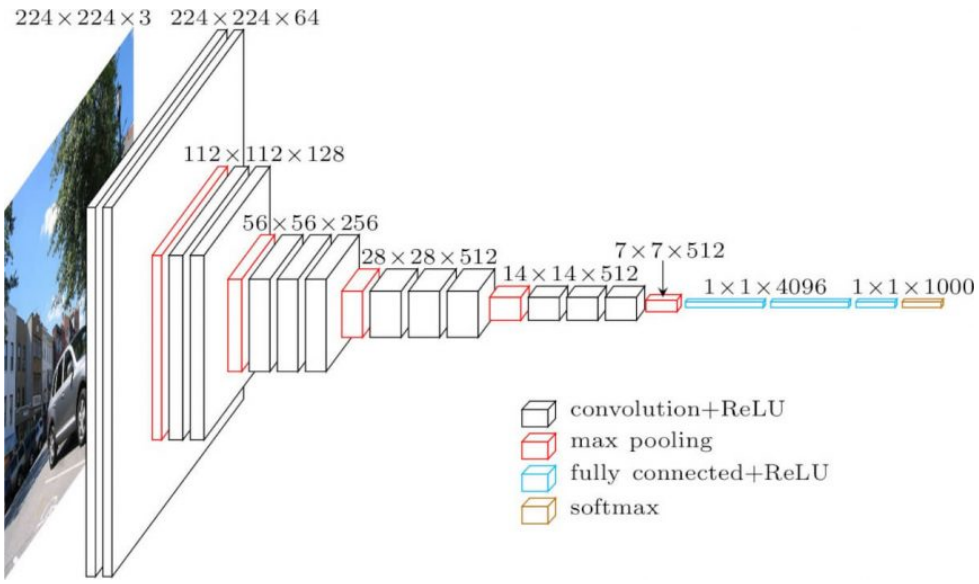

Figure 3. 22 VGG-16 Network Architecture [112]

As an architecture, the dimensions are set by the image size, (224 × 244). As part of pre-processing, each pixel in an image is subtracted from its mean RGB value. After completing the pre-processing step, a stack of convolutional layers with small receptive-field filters (3x3) are used. A filter of (1x1) size is used occasionally, which can be described as a linear transform of the input channels (followed by nonlinearity). During a convolution operation, the stride is fixed to 1. Several convolutional layers follow the five pooling layers to perform the spatial pooling. In max-pooling, the length of each stride is fixed to 2. The window size for the pooling is (2 * 2) pixels. All fully connected layers are configured the same way; the first and second layers each consist of 4096 channels, the third layer performs a 1000-way ILSVRC classification, therefore, contains 1000 channels, one for each class). The final layer is the softmax layer. A ReLu activation function follows all the hidden layers for the VGG network. Figure 3.23 shows all VGG configurations published in the study of Simonyan et al. [112]. The size of the architectures increases from (A) to (E). Although all configurations are successful, the most popular are D and E configurations with 16 and 19 layers. They called VGG-16 and VGG-19.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3. 23 VGG Configurations [112]

There are no architectural differences between the configurations. From 11 weight layers in network A that contains 8 convolutional and 3 fully connected layer, to 19 weight layers in network E that contains 16 convolutional and 3 fully connected layer. Each convolutional layer starts with 64 channels, and after every max-pooling layer, the number doubles, increasing by two more to 512; the convolutional layers have relatively few channels. Gradient descent coupled with backpropagation optimized the objective function of a multinomial logistic regression model, completing the entire training procedure. The batch size and the momentum are set to 256 and 0.9, respectively. There has been added a dropout ratio

of 0.5 to the first two fully connected layers in order to smooth out the dropouts. The learning rate was initially set at 0.001, but when the validation set accuracy stopped improving, it dropped by a factor of 10.

As in ILSVRC-2012 [114] as well as ILSVRC-2013 [114], In comparison to the previous generation of models, VGG16 significantly outperformed. When VGG network is used, there are two important drawbacks to consider. In the first place, the training process takes a long time and requires a lot of computing power. Secondly, the weights associated with network architectures are quite large. VGG16 is over 500MB in size since the model is so deep and contains so many fully connected nodes.

As can be seen from Figure 3.23, there is no architectural difference between VGG-16 and VGG-19. They differ only in the number of parameters they contain due to their depth. Within the scope of this thesis, VGG-16 and VGG-19 models will be trained with RadEch dataset in order to see the effects of the size of deep learning models on classification accuracy regardless of the dataset.

3.6. Training Stage

All dataset pre-processing described in Sections 3.2, 3.3 and 3.4 will be classified in Section 4. Specifications of training process are explained in this section. Three different CNN models described in Section 3.5 will be used for the classification process. These three models will be trained with all pre-processed datasets. At the end of the training, parameters such as classification accuracy and training duration will be taken into consideration and a comparison will be made. Considering the models to be used for training and the size of the dataset, it would make sense to do all the training on the GPU (Graphics Processing Unit). Since GPU technologies can perform parallel processing, they can complete model training in a shorter time than CPUs (Central Processing Unit). The Google Colaboratory platform was used to train the models on the GPU. Colaboratory, as known as “Colab”, is a product of Google Research. It is especially well suited for machine learning, data analysis and educational applications. Anyone can write and run arbitrary python code through the browser with Colab. A more detailly definition of Colab is a hosted Jupyter notebook service that offers free access to computing resources, such as GPUs, without the need for any installation. The Google Colab platform is open to everyone and free of charge. However, the training time of the models is limited to 12 hours. Google Colab offers users 12 GB NVIDIA Tesla K80 GPU in its free version. In this thesis, Google Colab Pro version, which is the paid version of Google Colab, was used. Google Colab Pro version offers users faster GPU, longer runtime, and more RAM

memory. The hardware specifications that Google Colab provided, which the three CNN models to be trained are given in Table 3.8.

Table 3. 8: Hardware Specifications

GPU	NVIDIA Tesla P100
GPU Memory	16,2 Gb
CUDA Architecture	Applicable
RAM Memory	27,3 Gb

Raw version, noise reduced version, traditional augmented version, and learning-based augmented versions of RadEch dataset will be classified with three different CNN models. The results will be compared both within themselves and with state-of-art studies in the literature. The specifications of the datasets are given in Table 3.9.

Table 3. 9: Dataset Specifications

Dataset	**Type**	**Number of Samples**
RadEch Raw Dataset	.mat files	452
RadEch Dataset Raw Spectrogram	.png files	452
RadEch Dataset Noise Reduced Spectrogram	.png files	452
RadEch Dataset Traditional Augmented	.png files	1953
RadEch Dataset Learning-Based Augmented	.png files	2422

3.7. Evaluation Metrics

As it examined the CNN results, there is a set of well-known measurements that can be utilized as benchmarks to evaluate how efficient the model is. These measures included the recall, precision, accuracy, and F1-score it is used to evaluate the proposed classification model.

A famous measure used to solve classification problems is the confusion matrix. Both binary classification and multiclass classification issues can be addressed by this technique. The confusion matrix given in Table 3.10 illustrates binary classification by confusion.

Table 3. 10: Confusion Matrix for Binary Classification

		Predicted	
		Negative	**Positive**
Real	**Negative**	TN	FP
	Positive	FN	TP

A confusion matrix is a count of predicted values versus real values. The output "TN" means “True Negative”, which indicates how many examples were correctly classified as negative. As well, "TP" means “True Positive”, which refers to the number of correctly classified positive examples. The term "FP" represents False Positives, which are actual negative examples classified as positives. "FN" stands for False Negatives, which are actual positive examples classified as negatives. While performing classification, accuracy is commonly used as a metric. A model's accuracy is determined by calculating its confusion matrix with using the Equation 3.1.

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{3.1}$$

Accuracy is the number of correctly classified objects that determines the accuracy of the method. With imbalanced datasets, accuracy can be misleading, and In addition to confusion matrix metrics, there are other performance measures that can be used. Within the scope of this thesis, all evaluations were made with the "Accuracy" metric. "Accuracy" was used as the evaluation metric in related works of this thesis.

The "precision" is measured by the ratio between true positives and false positives. In the Precision analysis, it is looked at how many false positives in the results. In the absence of

False Positives, the model had 100% precision. A model's precision is determined by calculating its confusion matrix with using the Equation 3.2.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

A recall rate is also called a sensitivity rate or a true positive rate. The definition is as follows in Equation 3.3:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

In ideally good classification models, recall should be 1 (high). Numerator and denominator must be equal for recall to be 1, i.e., TP = TP + FN, which means that FN is 0. In a decreasing FN, the denominator value becomes larger than the numerator and the recall value decreases (which is not desirable). The ideal precision and recall in a classifier are both 1, meaning FP and FN are both zero. F1-score is a metric that takes into account both precision and recall and is defined as follows in Equation 3.4:

$$F1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \tag{3.4}$$

Precision and recall must both be 1 for F1 Score to be 1. When precision and recall are both high, the F1 score becomes high. F1 score is the arithmetic mean of precision and recall, which is a superior measure to accuracy.

In addition to all these evaluation metrics, there is also the stage of deciding how much of the data set will be allocated for training and how much for testing. The process of separating this data set, which is one of the most important stages of any deep learning model, can be done in many ways. The higher the training set is set, the better the learning process will be because the model will be fed with more data. On the other hand, keeping the training set high will mean reducing the validation set. Lower amounts of validation sets, no matter how well the model is

trained, will render the model's evaluation results inaccurate as its ability to represent the whole data is reduced. Studies published in the literature are generally divided into 15% validation set and 85% training set.

In this thesis, a data separation method called cross-validation was used. A cross-validation method is a way of estimating how well machine learning models perform (or how accurate they are). Especially when limited data is available, it protects against overfitting a predictive model. Cross-validation involves folding a fixed number of data sets, analysing each fold, and averaging the error estimate on the average. One of the cross-validation methods widely used in the literature is the "Holdout" method. It is a rather basic and simple approach where we separate our dataset into two parts, the training data and the testing data. Training is carried out on training data, and the results are then evaluated on the testing set. Typically, training data is bigger than testing data, so the data is split in the ratio 85:15 or 80:20.

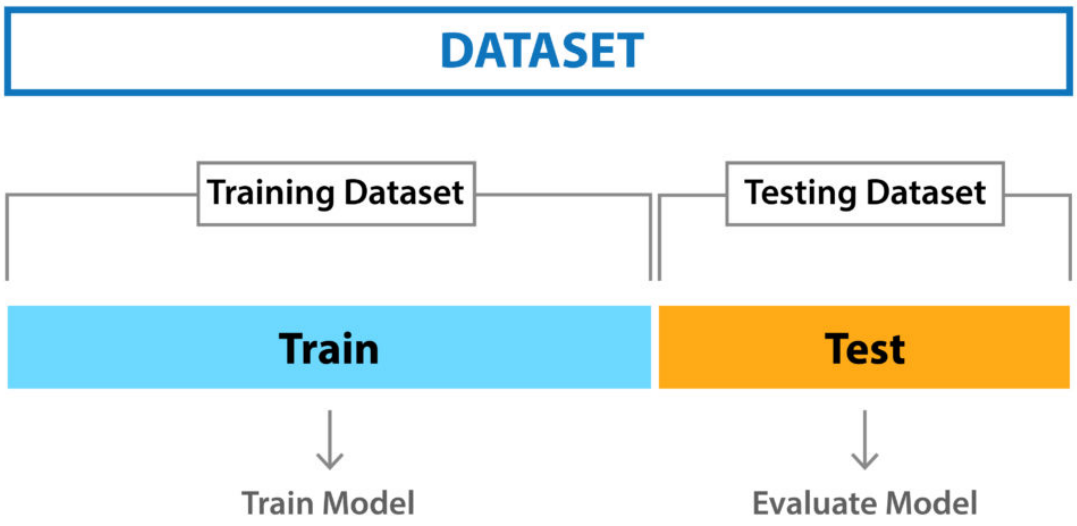

Figure 3. 24 Holdout Method of Validation [115]

The holdout method is shown in Figure 3.24. The data is initially shuffled randomly before it is split according to this method. Every time we train the model, it can give different results because it's training on a different combination of data, and this can cause instability. Additionally, it is not possible to ensure that the train set selected represents the whole dataset. Also, when a dataset is not very large, there is a chance that some important information

contained in the testing data might be lost since the model is not trained on the testing set. The hold-out method is good to use when using a very large dataset.

Holdout method can be improved by using K-fold cross validation. By using this method, it is ensured that the score of the model is not affected by how it is chosen the training and testing sets. In a nutshell, k number of subsets are created, and then k number of holdout methods are applied. Here is step of this method below:

1. Dividing your dataset into k number of folds (subsets) at random
2. Create a model based on k - 1 folded folds of your dataset per fold. Then, test the model to determine its effectiveness for kth fold
3. Continue to do this until all the k-folds have been tested
4. In cross-validation, the average accuracy of your obtained k samples will be used as a performance metric.

The training and test sets can have any observation from the original dataset since it ensures that every observation is included. The model produced by this method is generally less biased than those produced by other methods. Using this method has the disadvantage that the training algorithm must be the same each time, so that it requires k times as much computing power to evaluate the algorithm. The k-fold cross-validation method is shown in Figure 3.25

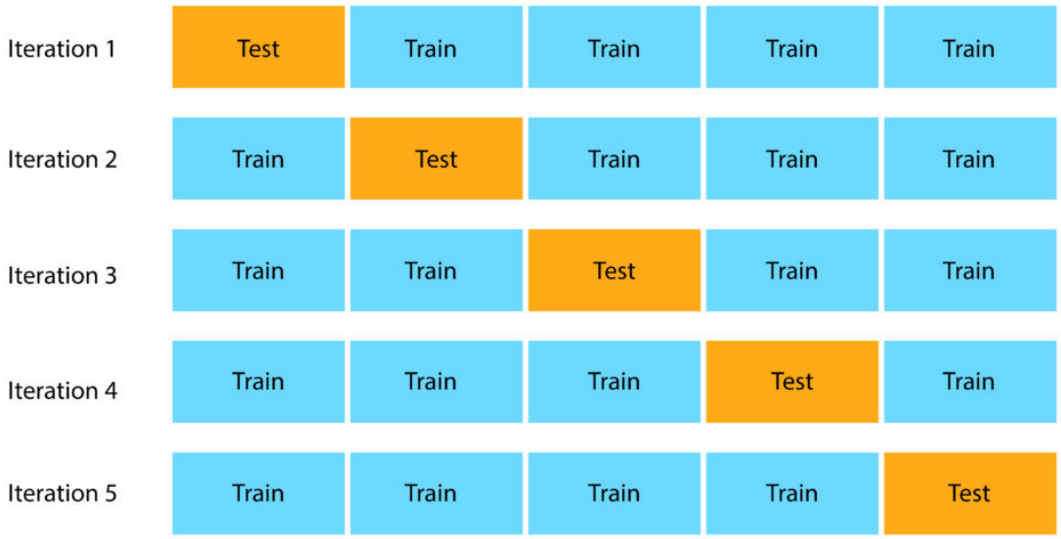

Figure 3. 25 5-Fold Cross-validation

4. RESULTS

Training and classification of the augmented datasets obtained in the previous sections with the deep learning models described in the previous sections will be explained in this section. The training process will be explained in three different titles according to the three deep learning models used in the thesis. In each topic, that is, each deep learning model will be trained with 5 different datasets. The results obtained will be displayed in the section of each deep learning model with graphs and tables.

4.1. CNN with 5-Layer

Raw RadEch spectrograms and datasets that augmented in previous sections will be trained with 5-layer CNN. This CNN architecture is shown in Figure 3.21. The specifications of training process are shown in table 4.1.

Table 4. 1: Training Specifications

Batch Size	8
Cross Validation	5-Fold Cross Validation
Image Size	224 x 224 x 3
Epoch	50
Model Optimizer	Adam optimizer
Learning Rate	0,001
Loss Function	Categorical Cross-entropy
Evaluation Metric	Accuracy
Activation Function of Classification Layer	Sigmoid Function

4.1.1. Raw RadEch Spectrograms

Raw RadEch Spectrograms contains 452 samples. The dataset is trained with 50 Epochs. The results obtained as a result of the training are shown in Table 4.2. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.1 and 4.2 respectively.

Table 4. 2: Training Results

Training Accuracy	99,45%
Validation Accuracy	91,11%
Training Loss	0,022
Validation Loss	0,197
Training Time	269,53 seconds

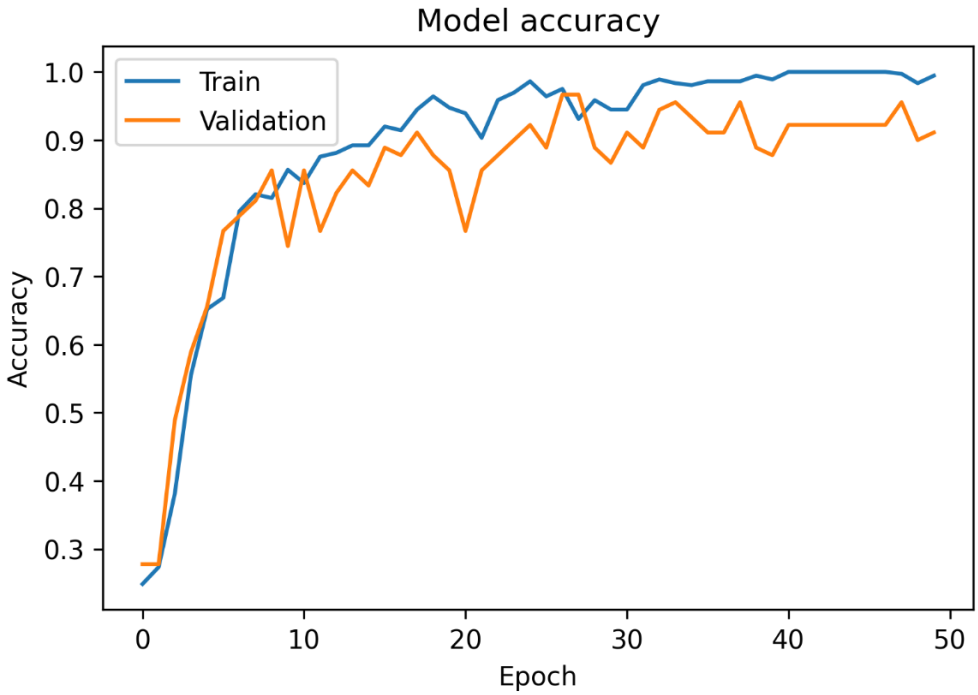

Figure 4. 1 Accuracy over Epoch

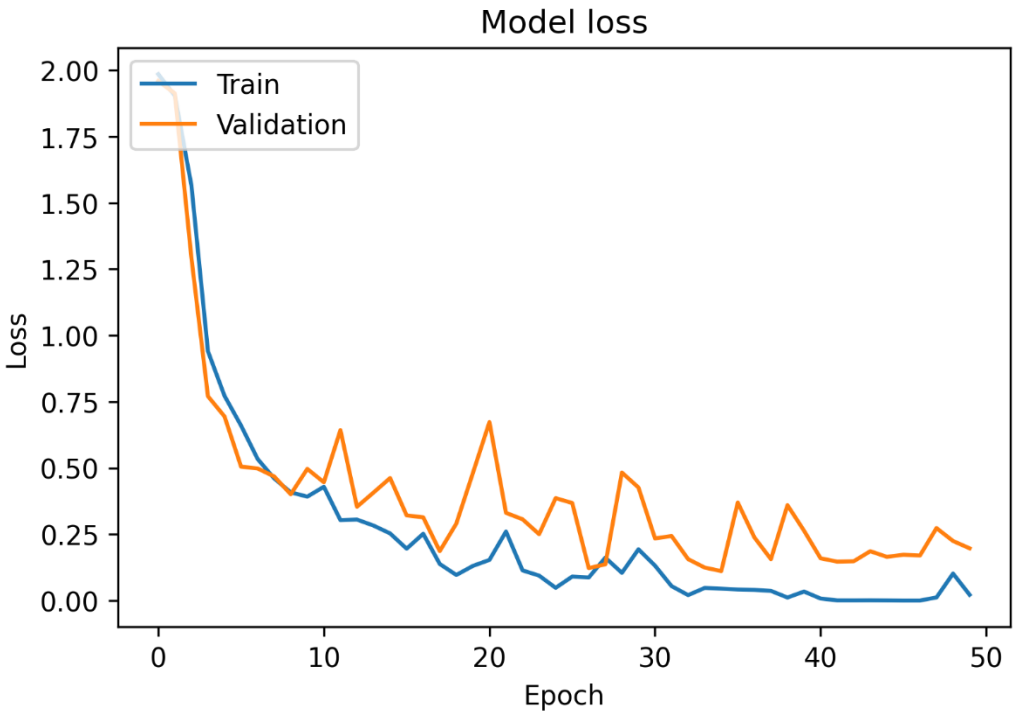

Figure 4. 2 Loss over Epoch

4.1.2. Noise Reduced RadEch Spectrograms

Noise Reduced RadEch Spectrograms contains 452 samples. Training epochs are selected as 50. The results obtained as a result of the training are shown in Table 4.3. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.3 and 4.4 respectively.

Table 4. 3: Training Results

Training Accuracy	99,19%
Validation Accuracy	99,12%
Training Loss	0,005
Validation Loss	0,003
Training Time	161,56 seconds

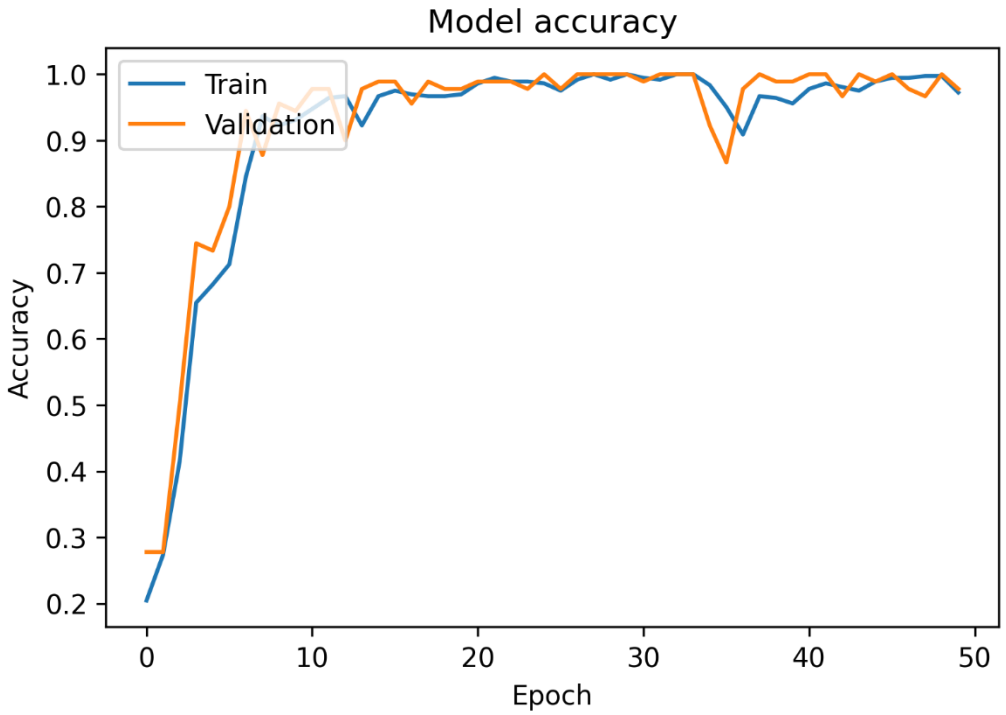

Figure 4. 3 Accuracy over Epoch

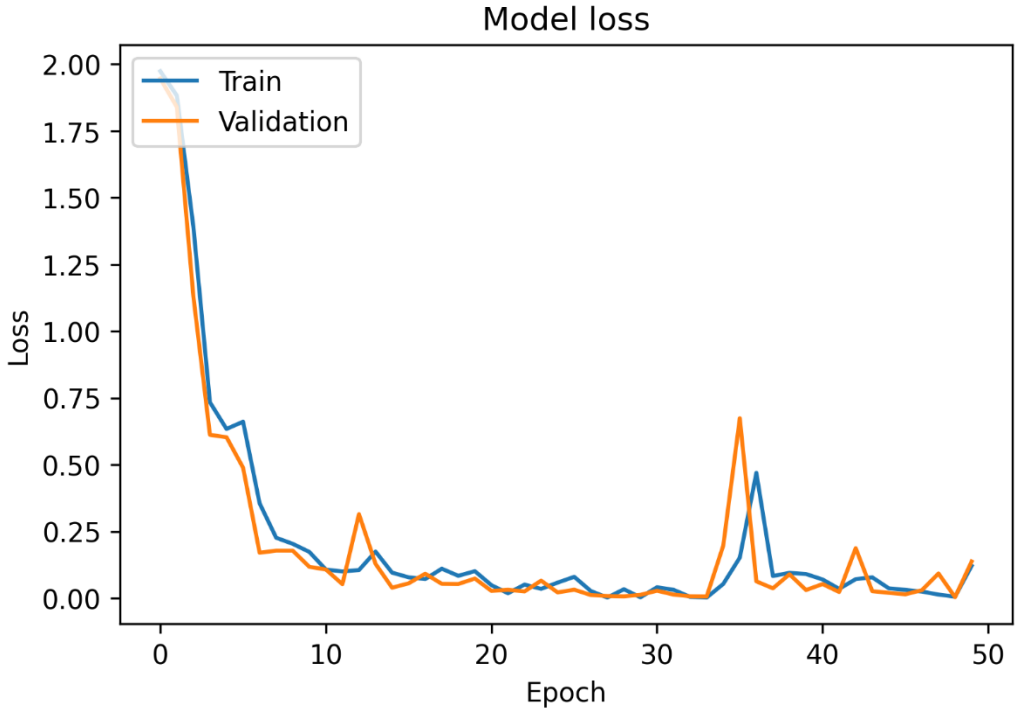

Figure 4. 4 Loss over Epoch

4.1.3. Traditional Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.4. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.5 and 4.6 respectively.

Table 4. 4: Training Results

Training Accuracy	99,88%
Validation Accuracy	99,76%
Training Loss	0,039
Validation Loss	0,021
Training Time	1204,09 seconds

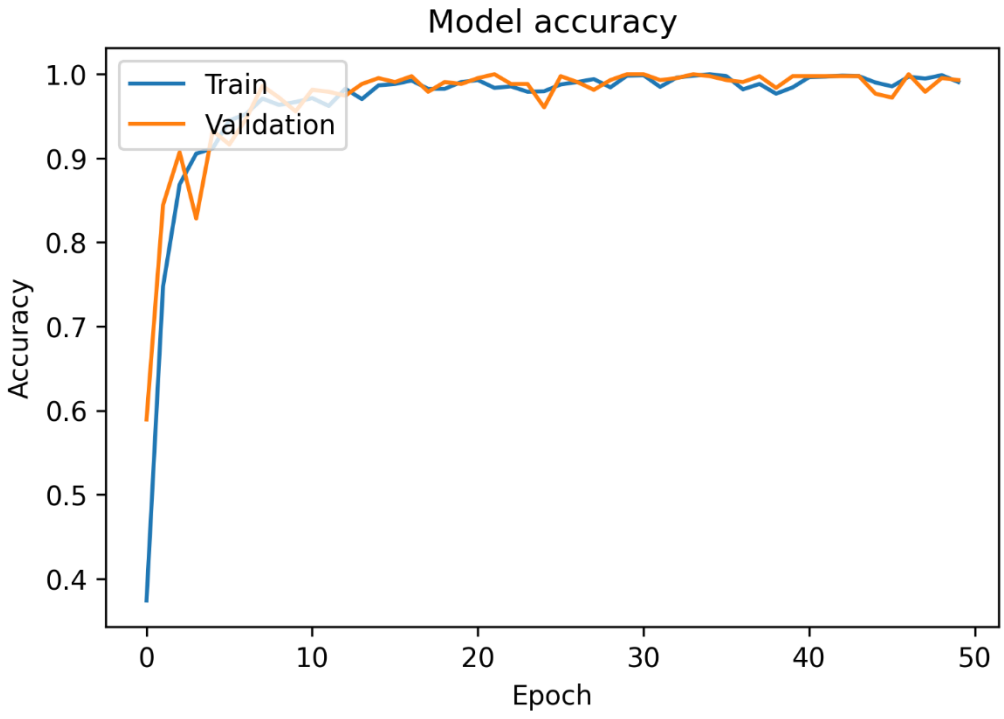

Figure 4. 5 Accuracy over Epoch

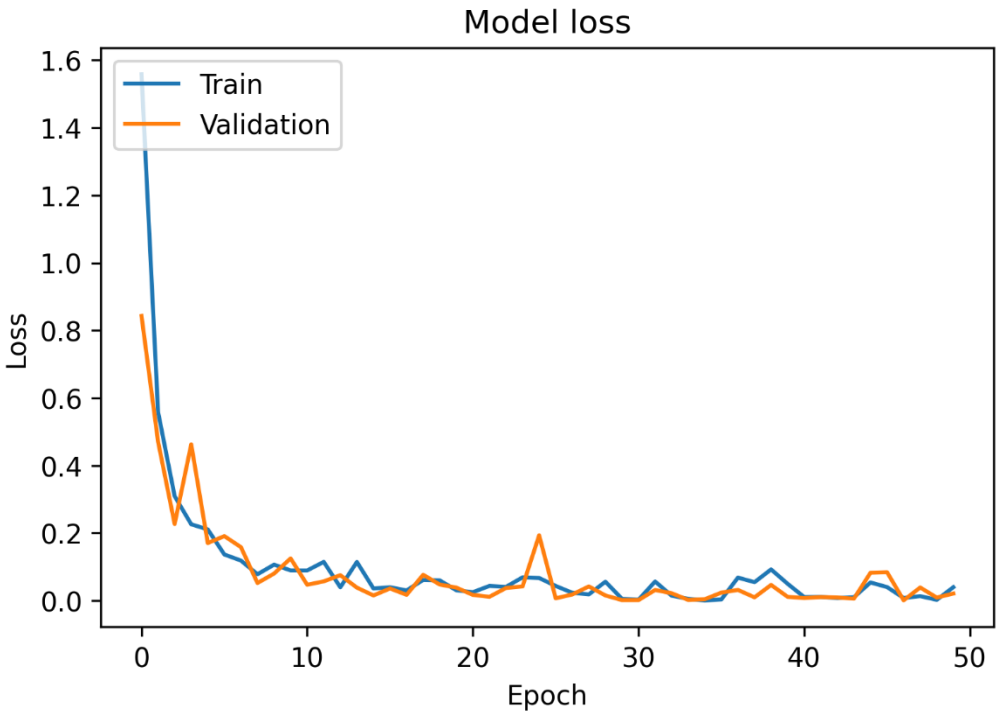

Figure 4. 6 Loss over Epoch

4.1.4. Learning-Based Augmented RadEch Dataset

Learning-Based Augmented RadEch Dataset contains 2422 samples. The samples are generated by GAN. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.5. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.7 and 4.8 respectively.

Table 4. 5: Training Results

Training Accuracy	98,99%
Validation Accuracy	94,07%
Training Loss	0,011
Validation Loss	0,156
Training Time	379,43 seconds

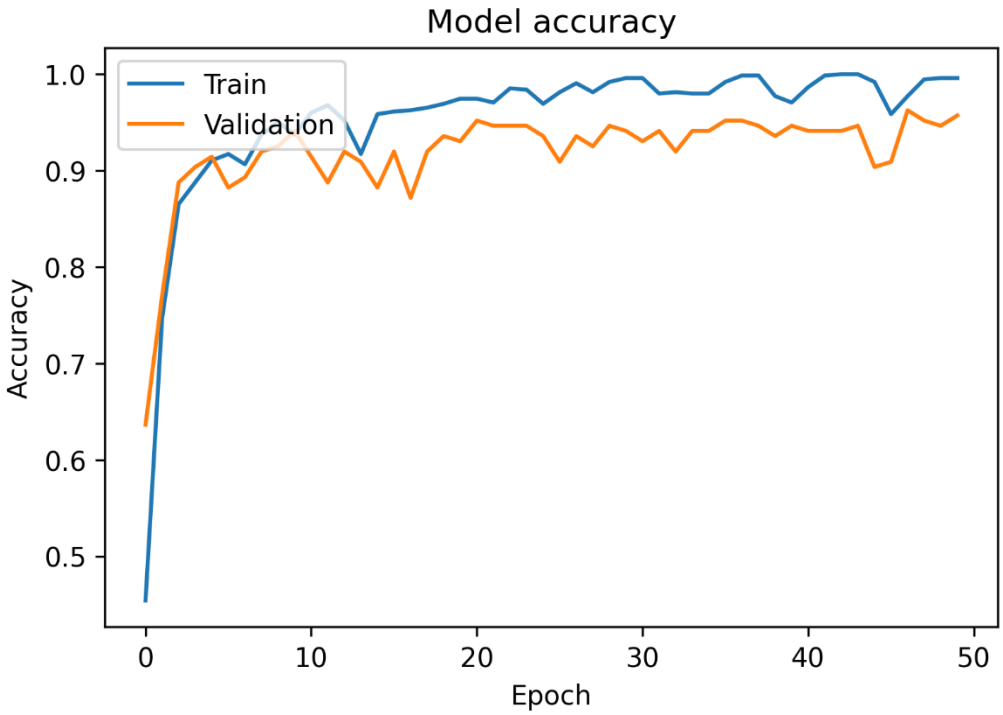

Figure 4. 7 Accuracy over Epoch

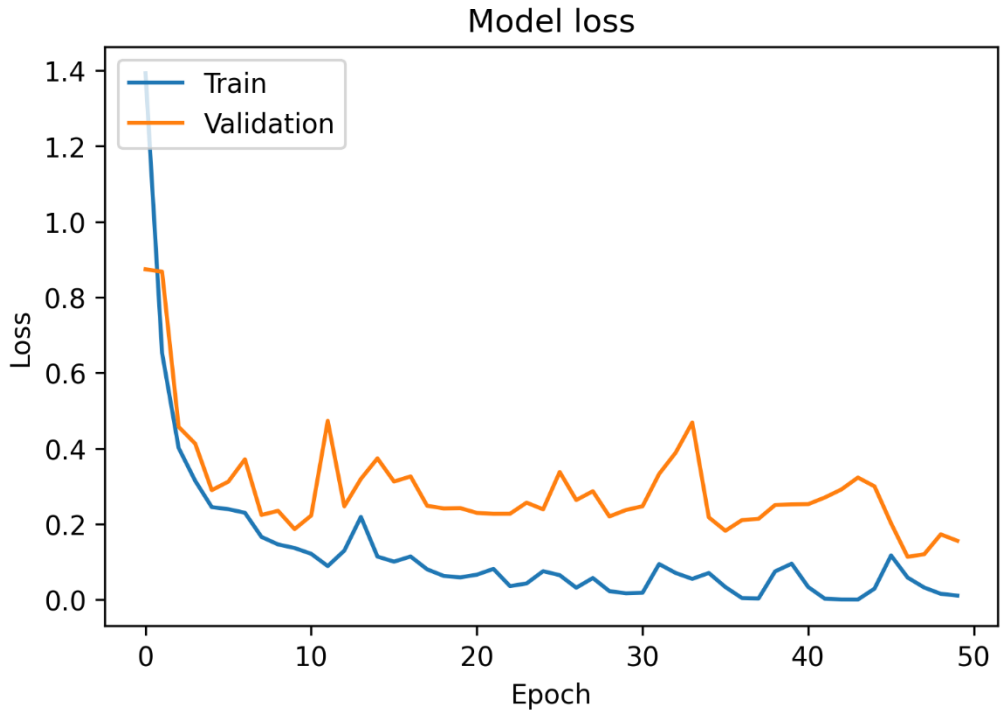

Figure 4. 8 Loss over Epoch

In this section, trained results of all datasets created in this thesis with 5-Layer CNN are shown in tables and graphs. In the next section, the same datasets will be trained with the VGG-16 model.

4.2. VGG-16 without Transfer Learning

Raw RadEch spectrograms and datasets that augmented in previous sections will be trained with VGG-16 network. This CNN architecture is shown in Figure 3.22. The specifications of training process are shown in Table 4.6. In fact, the VGG-16 network has been trained and weighted with a dataset of more than 15 million images called ImageNet that explained in Section 3.5. Datasets will be trained without using these weights, that is, without transfer learning.

Table 4. 6: Training Specifications

Batch Size	8
Cross Validation	5-Fold Cross Validation
Image Size	224 x 224 x 3
Epoch	50
Model Optimizer	Adam optimizer
Learning Rate	0,001
Loss Function	Sparse Categorical Cross-entropy
Evaluation Metric	Accuracy
Activation Function of Classification Layer	Softmax Function

4.2.1. Raw RadEch Spectrograms

Raw RadEch Spectrograms contains 452 samples. The dataset is trained with 50 Epochs. The results obtained as a result of the training are shown in Table 4.7. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.9 and 4.10 respectively.

Table 4. 7: Training Results

Training Accuracy	99,99%

Validation Accuracy	94,44%
Training Loss	2,96e-5
Validation Loss	0,279
Training Time	409,71 seconds

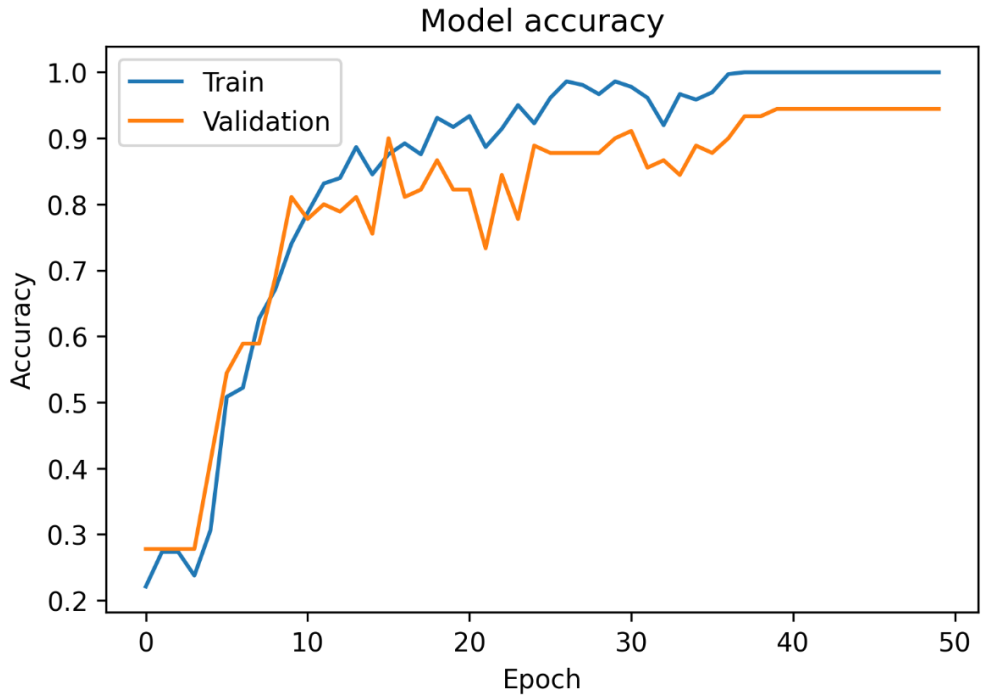

Figure 4. 9 Accuracy over Epoch

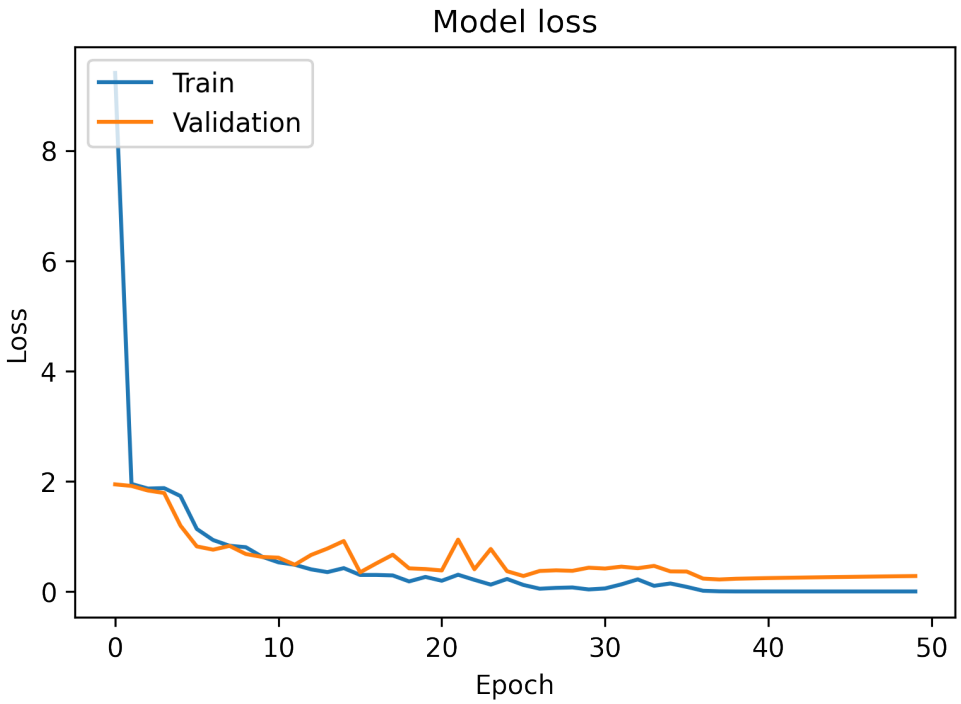

Figure 4. 10 Loss over Epoch

4.2.2. Noise Reduced RadEch Spectrograms

Noise Reduced RadEch Spectrograms contains 452 samples. Training epochs are selected as 50. The results obtained as a result of the training are shown in Table 4.8. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.11 and 4.12 respectively.

Table 4. 8: Training Results

Training Accuracy	99,99%
Validation Accuracy	98,89%
Training Loss	1,926e-5
Validation Loss	0,058
Training Time	343,63 seconds

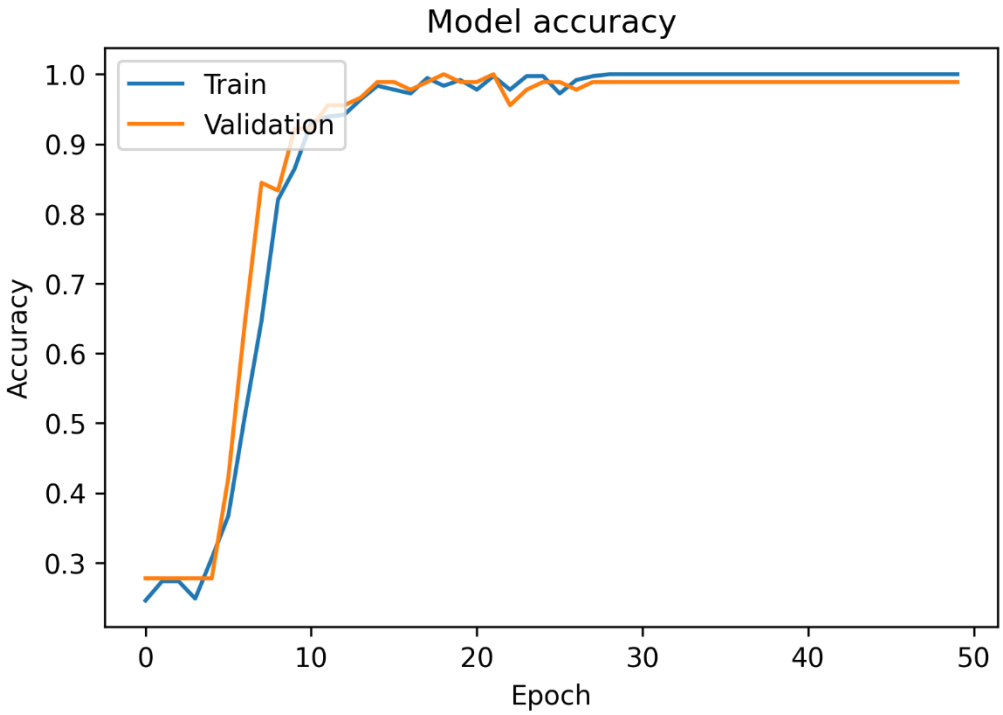

Figure 4. 11 Accuracy over Epoch

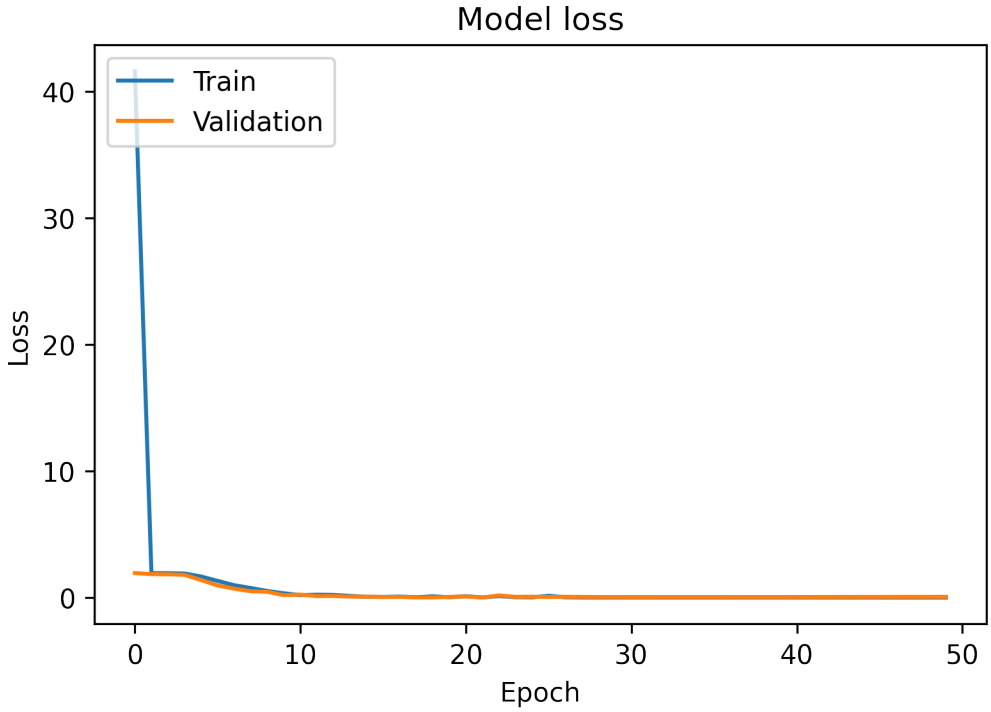

Figure 4. 12 Loss over Epoch

4.2.3. Traditional Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.9. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.13 and 4.14 respectively.

Table 4. 9: Training Results

Training Accuracy	100,00%
Validation Accuracy	99,77%
Training Loss	2,62e-9
Validation Loss	0,011
Training Time	1857,00 seconds

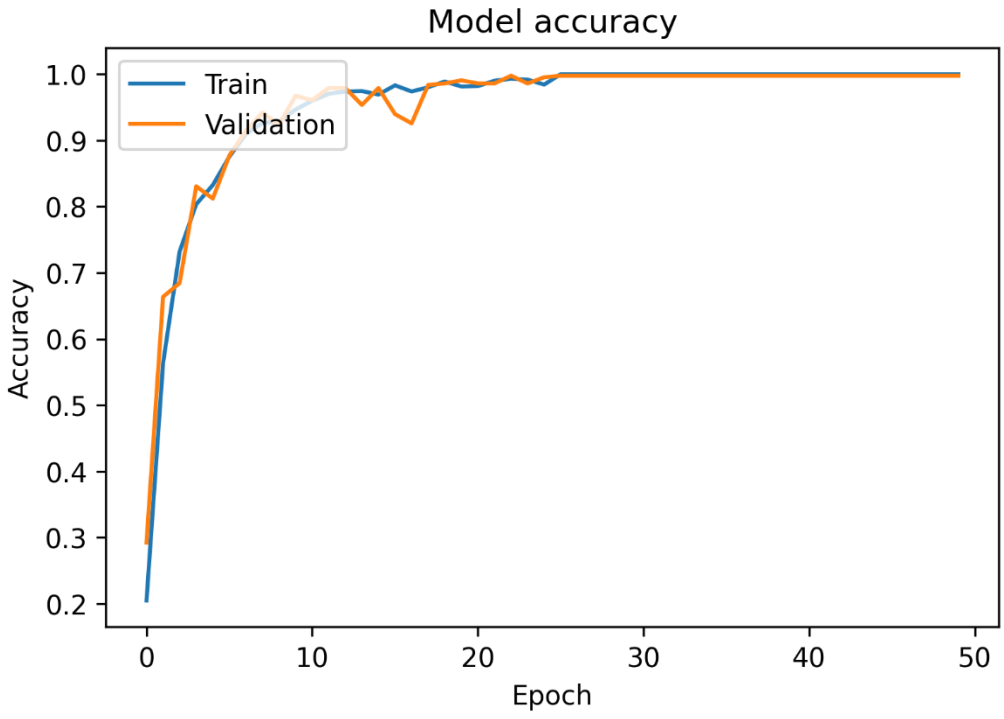

Figure 4. 13 Accuracy over Epoch

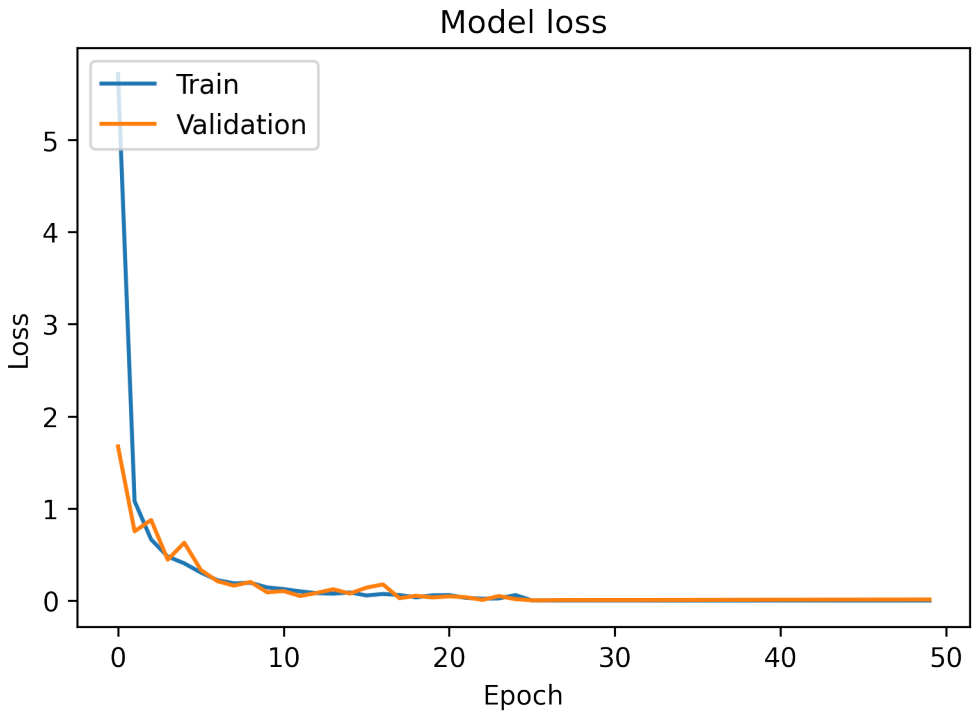

Figure 4. 14 Loss over Epoch

4.2.4. Learning-Based Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.10. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.15 and 4.16 respectively.

Table 4. 10: Training Results

Training Accuracy	99,60%
Validation Accuracy	94,65%
Training Loss	1,28e-4
Validation Loss	0,200
Training Time	559,54 seconds

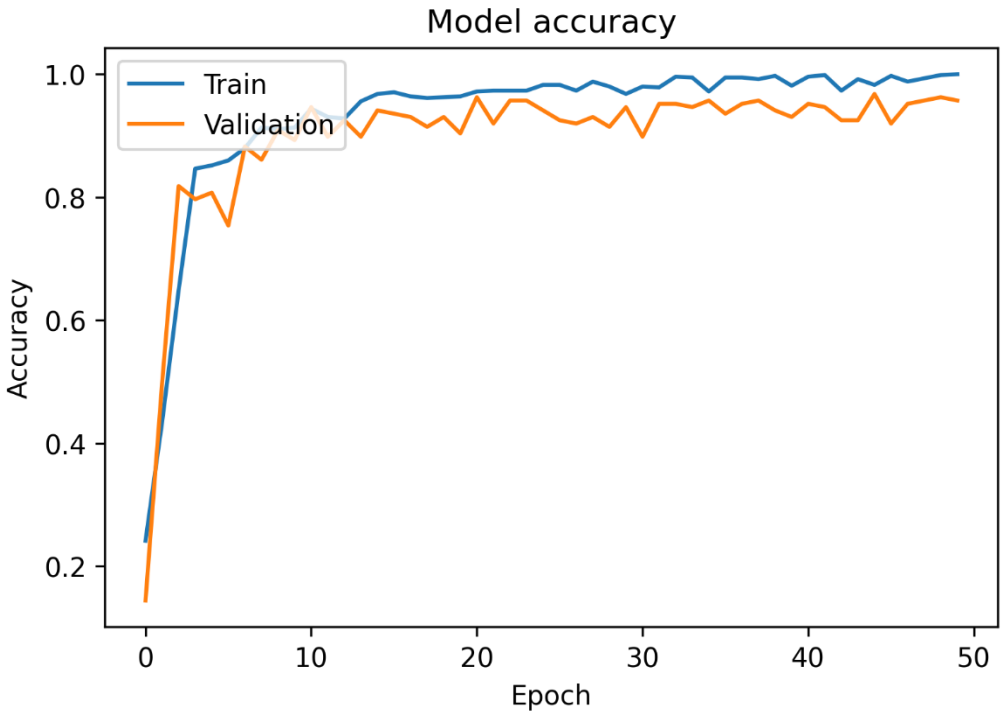

Figure 4. 15 Accuracy over Epoch

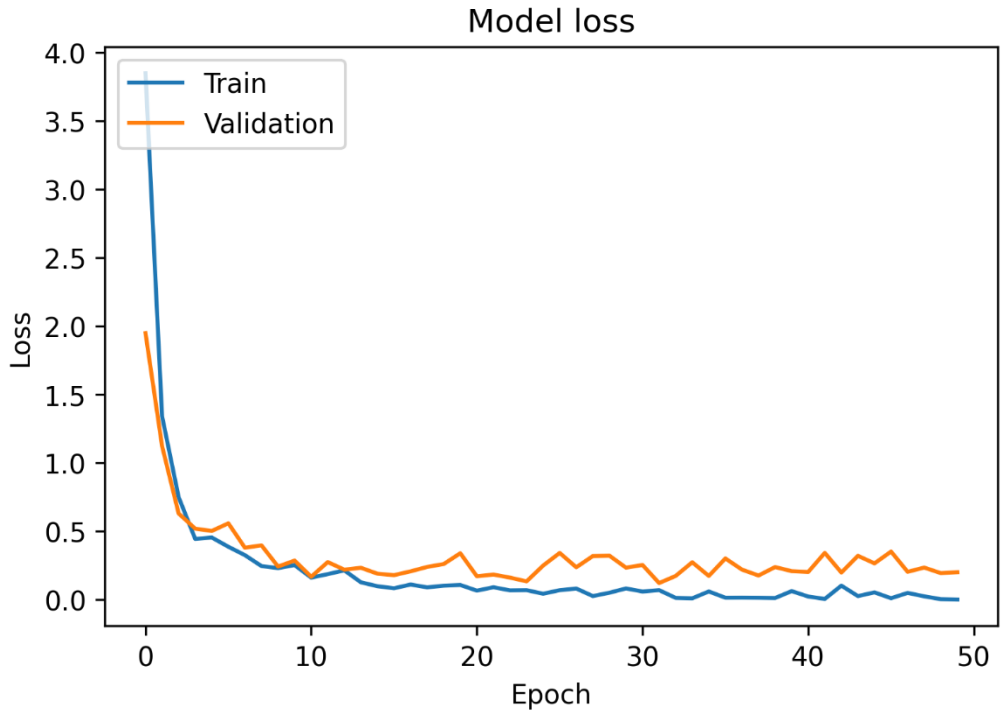

Figure 4. 16 Loss over Epoch

4.3. VGG-16 with Transfer Learning

Raw RadEch spectrograms and datasets that augmented in previous sections will be trained with VGG-16 network. This CNN architecture is shown in Figure 3.22. The specifications of training process are shown in Table 4.11. In fact, the VGG-16 network has been trained and weighted with a dataset of more than 15 million images called ImageNet that explained in Section 3.5. Datasets will be trained with using these weights that is, with transfer learning. So that the results of transfer learning on micro-Doppler radar data can also be seen.

Table 4. 11: Training Specifications

Batch Size	8
Cross Validation	5-Fold Cross Validation
Image Size	224 x 224 x 3
Epoch	50
Model Optimizer	Adam optimizer
Learning Rate	0,001
Loss Function	Sparse Categorical Cross-entropy
Evaluation Metric	Accuracy
Activation Function of Classification Layer	Softmax Function

4.3.1. Raw RadEch Spectrograms

Raw RadEch Spectrograms contains 452 samples. The dataset is trained with 50 Epochs. The results obtained as a result of the training are shown in Table 4.12. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.17 and 4.18 respectively.

Table 4. 12: Training Results

Training Accuracy	71,12%
Validation Accuracy	65,22%
Training Loss	0,582
Validation Loss	0,775
Training Time	270,72 seconds

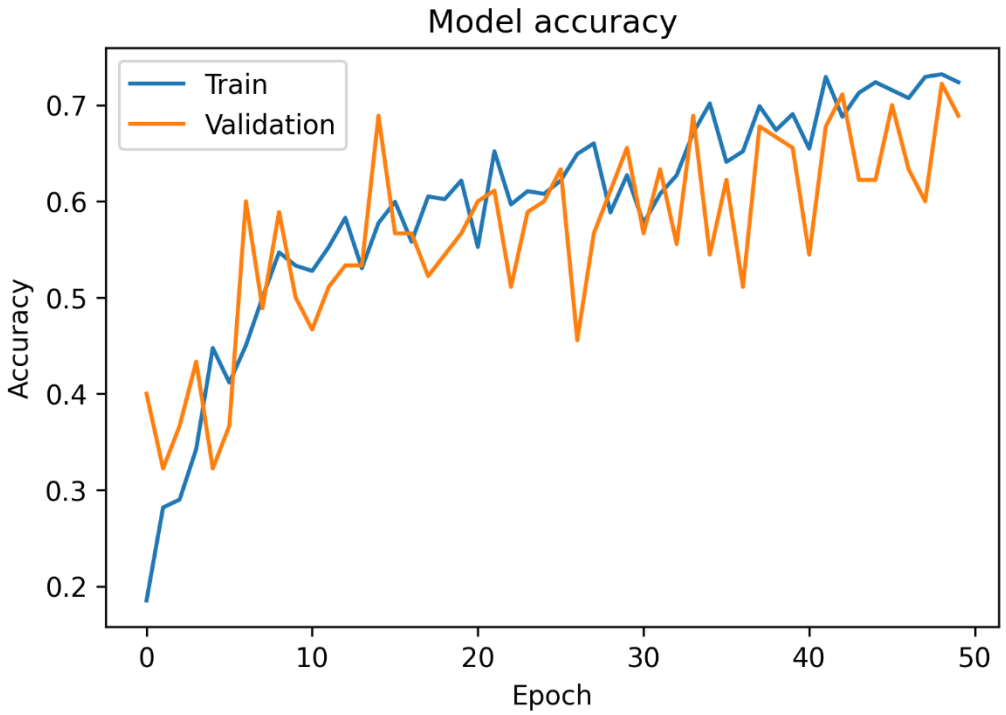

Figure 4. 17 Accuracy over Epoch

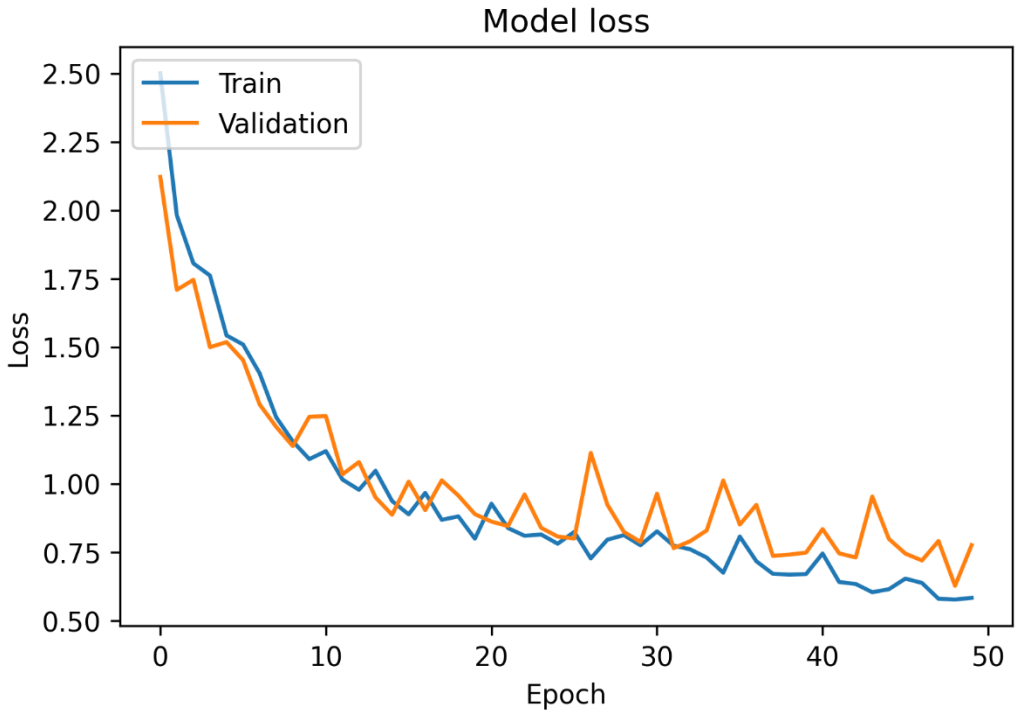

Figure 4. 18 Loss over Epoch

4.3.2. Noise Reduced RadEch Spectrograms

Noise Reduced RadEch Spectrograms contains 452 samples. Training epochs are selected as 50. The results obtained as a result of the training are shown in Table 4.13. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.19 and 4.20 respectively.

Table 4. 13: Training Results

Training Accuracy	99,17%
Validation Accuracy	97,78%
Training Loss	0,035
Validation Loss	0,066
Training Time	281,50 seconds

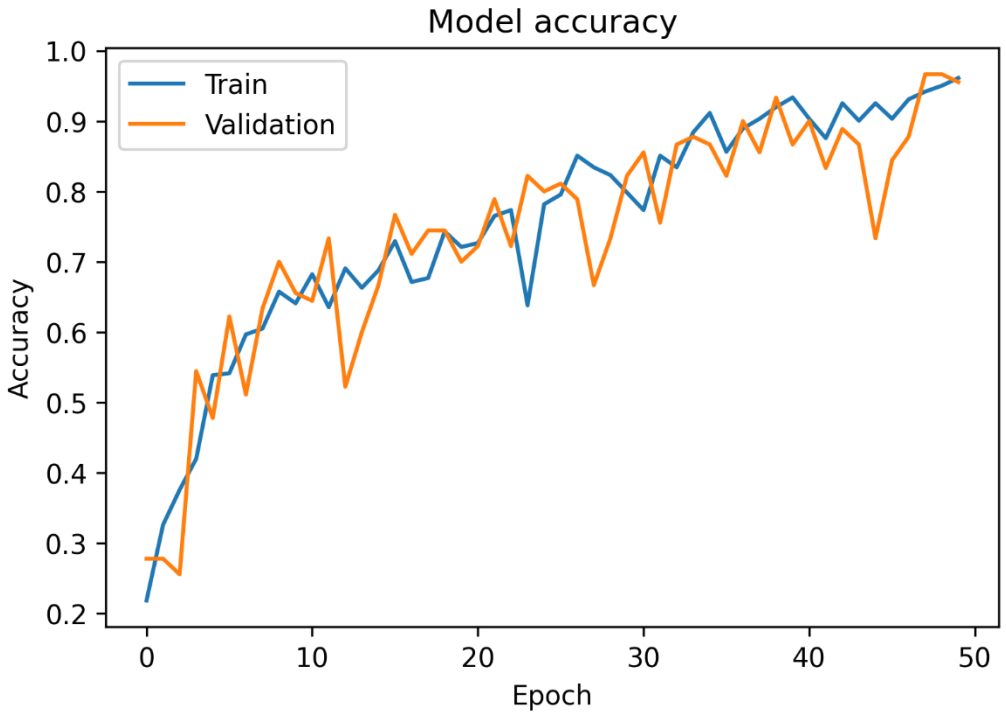

Figure 4. 19 Accuracy over Epoch

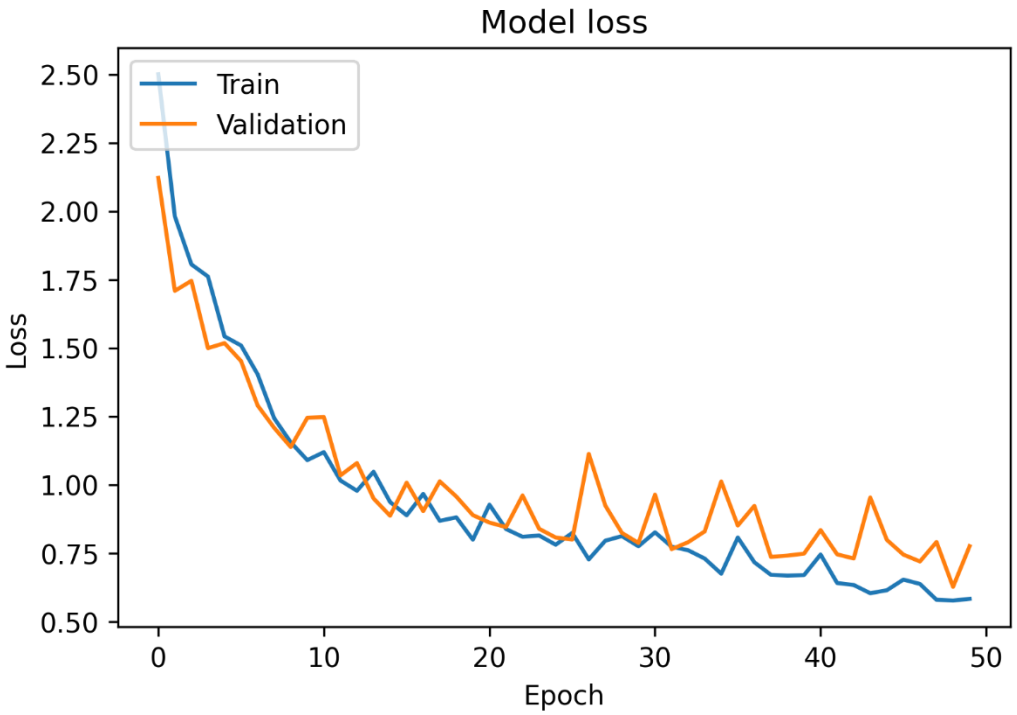

Figure 4. 20 Loss over Epoch

4.3.3. Traditional Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.14. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.21 and 4.22 respectively.

Table 4. 14: Training Results

Training Accuracy	100,00%
Validation Accuracy	99,77%
Training Loss	6,993e-8
Validation Loss	0,022
Training Time	1902,37 seconds

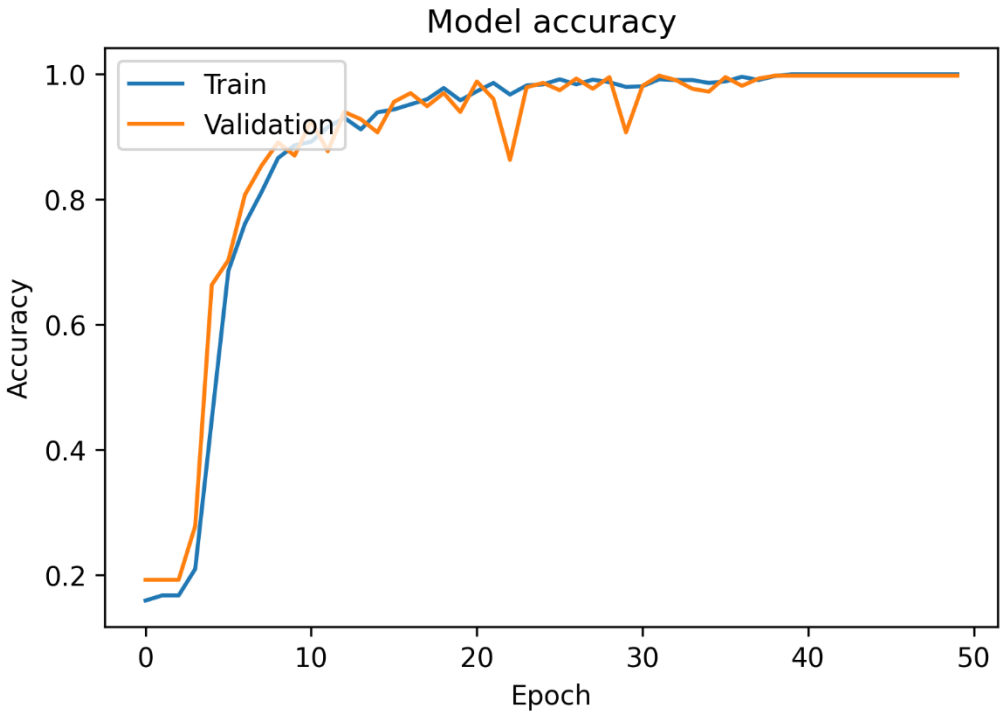

Figure 4. 21 Accuracy over Epoch

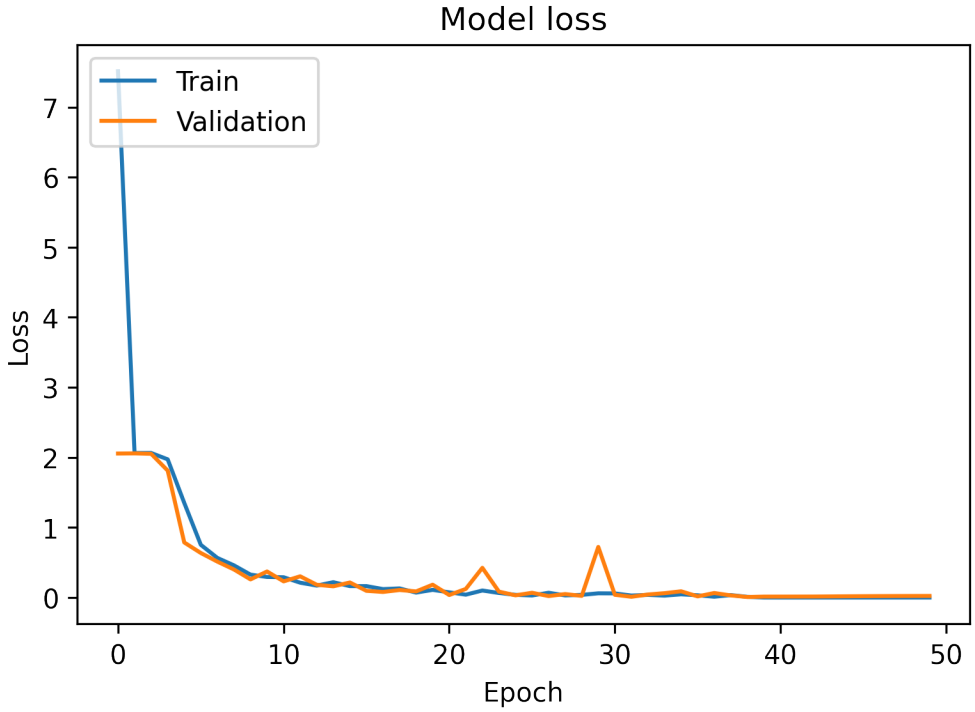

Figure 4. 22 Loss over Epoch

4.3.4. Learning-Based Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.15. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.23 and 4.24 respectively.

Table 4. 15: Training Results

Training Accuracy	99,05%
Validation Accuracy	95,72%
Training Loss	6,263e-5
Validation Loss	0,131
Training Time	632,96 seconds

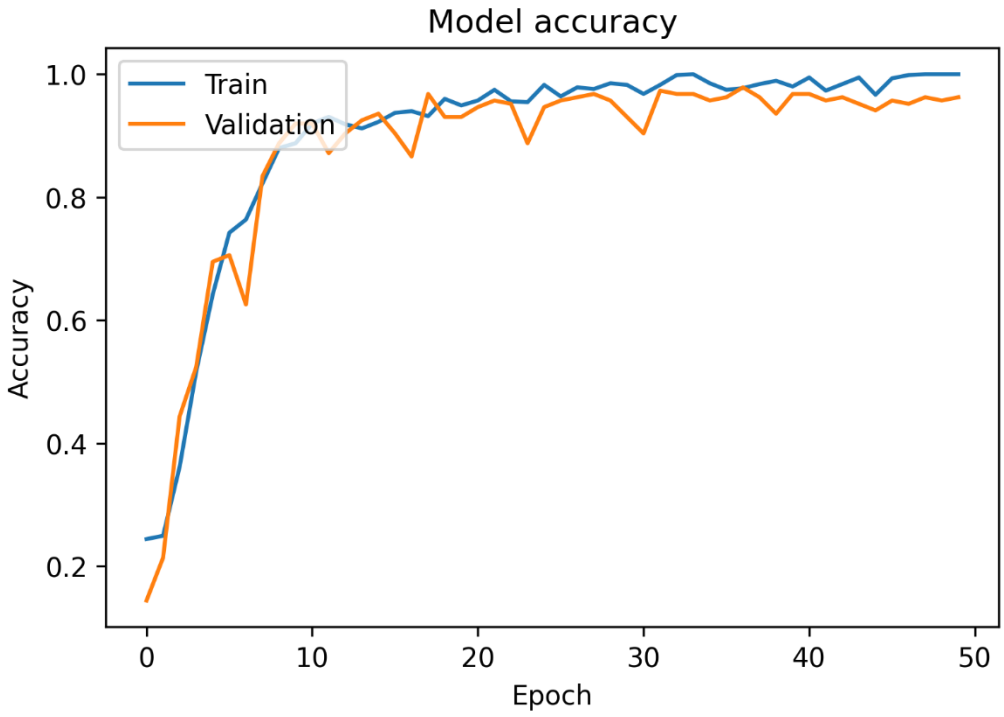

Figure 4. 23 Accuracy over Epoch

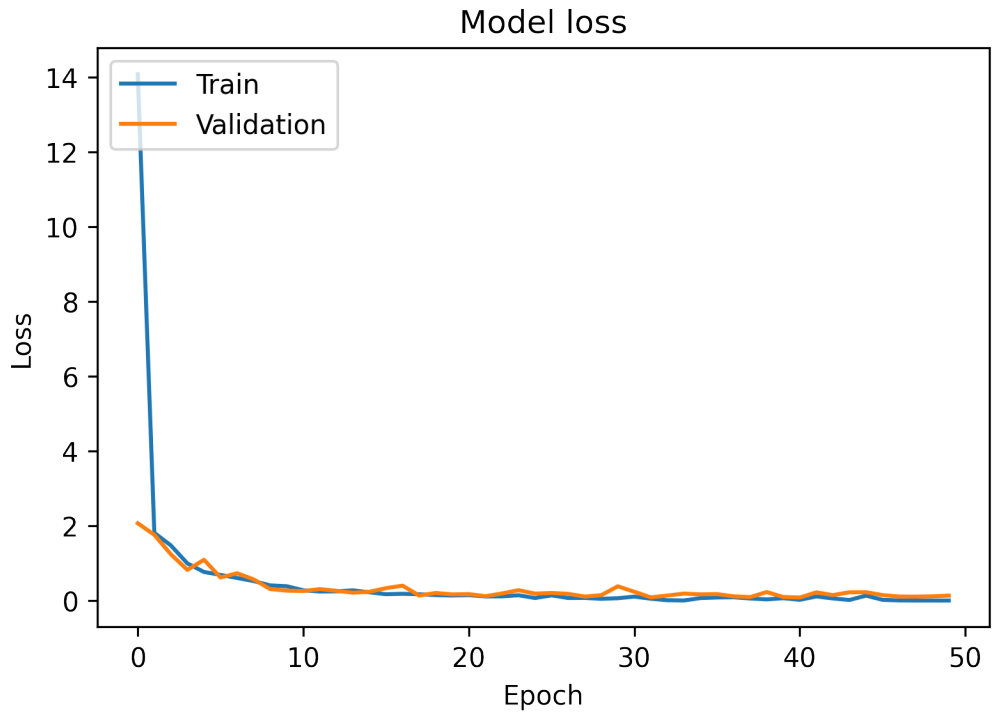

Figure 4. 24 Loss over Epoch

4.4. VGG-19 without Transfer Learning

Raw RadEch spectrograms and datasets that augmented in previous sections will be trained with VGG-16 network. This CNN configuration is shown in Figure 3.23. The specifications of training process are shown in Table 4.16. In fact, the VGG-19 network has been trained and weighted with a dataset of more than 15 million images called ImageNet that explained in Section 3.5. Datasets will be trained without using these weights, that is, without transfer learning.

Table 4. 16: Training Specifications

Batch Size	8
Cross Validation	5-Fold Cross Validation
Image Size	224 x 224 x 3
Epoch	50
Model Optimizer	Adam optimizer
Learning Rate	0,001
Loss Function	Sparse Categorical Cross-entropy
Evaluation Metric	Accuracy

Activation Function of Classification Layer	Softmax Function

4.4.1. Raw RadEch Spectrograms

Raw RadEch Spectrograms contains 452 samples. The dataset is trained with 50 Epochs. The results obtained as a result of the training are shown in Table 4.17. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.25 and 4.26 respectively.

Table 4. 17: Training Results

Training Accuracy	99,99%
Validation Accuracy	91,11%
Training Loss	1,059e-5
Validation Loss	0,330
Training Time	402,60 seconds

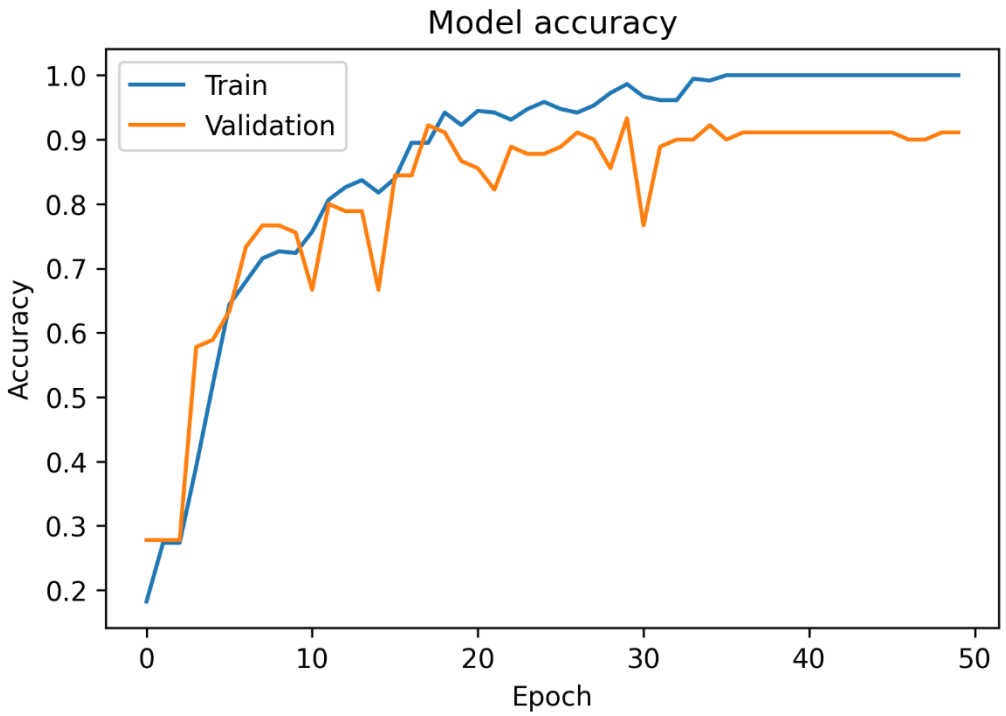

Figure 4. 25 Accuracy over Epoch

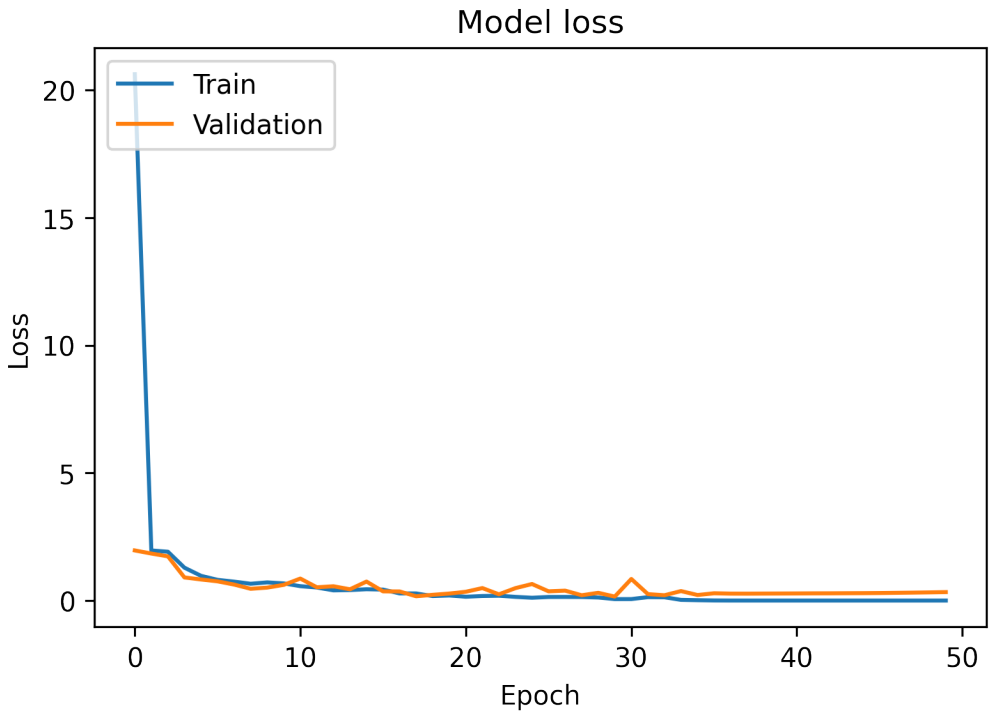

Figure 4. 26 Loss over Epoch

4.4.2. Noise Reduced RadEch Spectrograms

Noise Reduced RadEch Spectrograms contains 452 samples. Training epochs are selected as 50. The results obtained as a result of the training are shown in Table 4.18. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.27 and 4.28 respectively

Table 4. 18: Training Results

Training Accuracy	100,00 %
Validation Accuracy	100,00 %
Training Loss	1,135e-6
Validation Loss	2,315e-4
Training Time	339,72 seconds

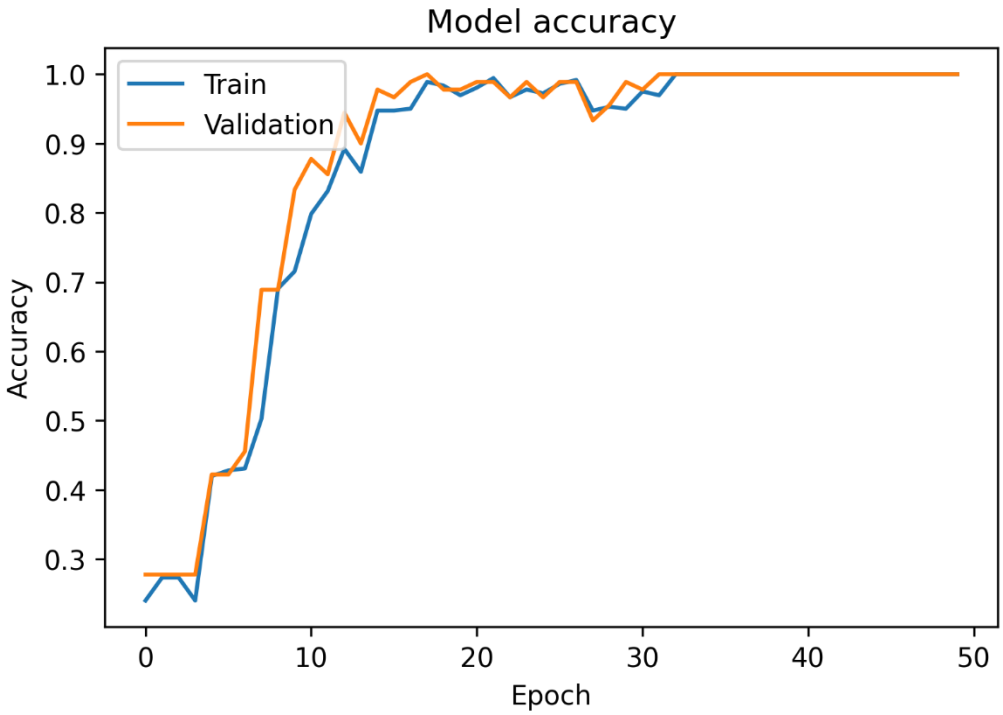

Figure 4. 27 Accuracy over Epoch

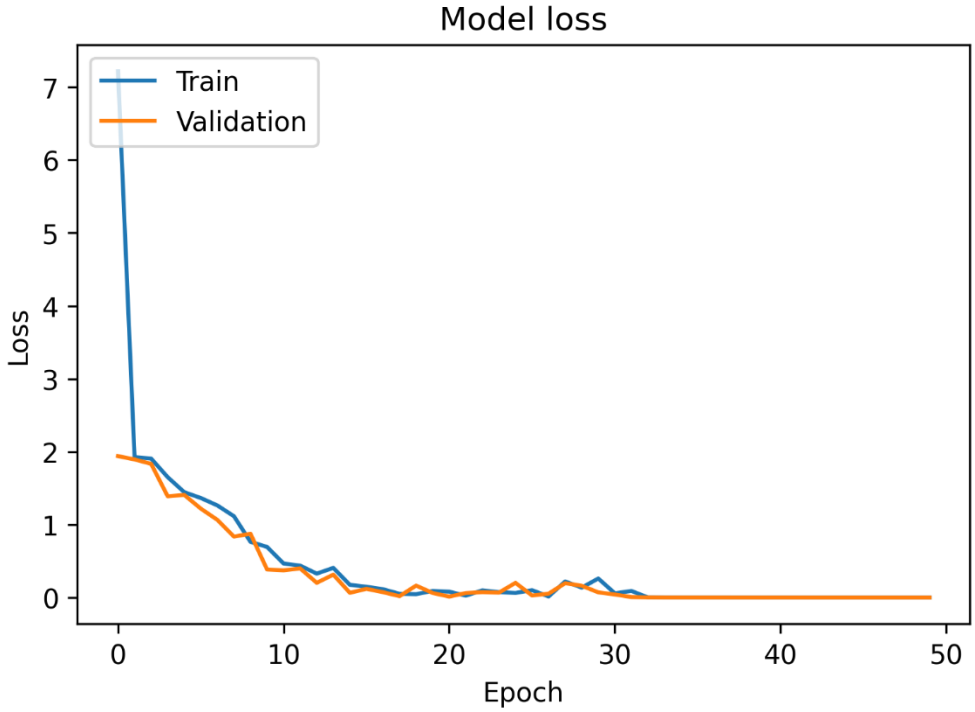

Figure 4. 28 Loss over Epoch

4.4.3. Traditional Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.19. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.29 and 4.30 respectively.

Table 4. 19: Training Results

Training Accuracy	100,00%
Validation Accuracy	99,77%
Training Loss	2,62e-9
Validation Loss	0,011
Training Time	1857,00 seconds

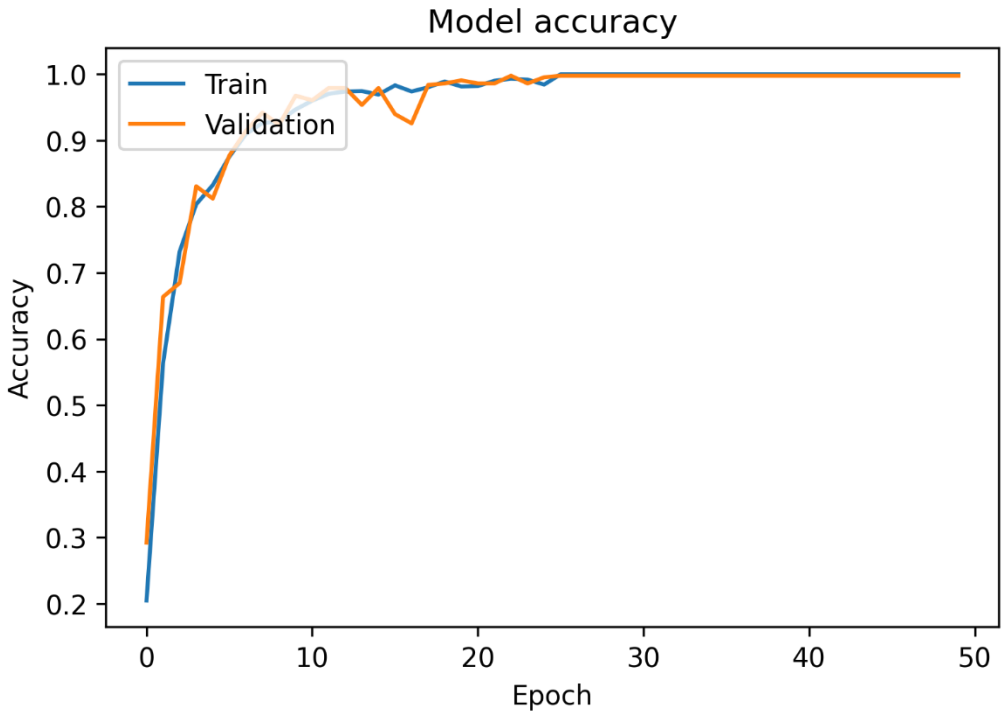

Figure 4. 29 Accuracy over Epoch

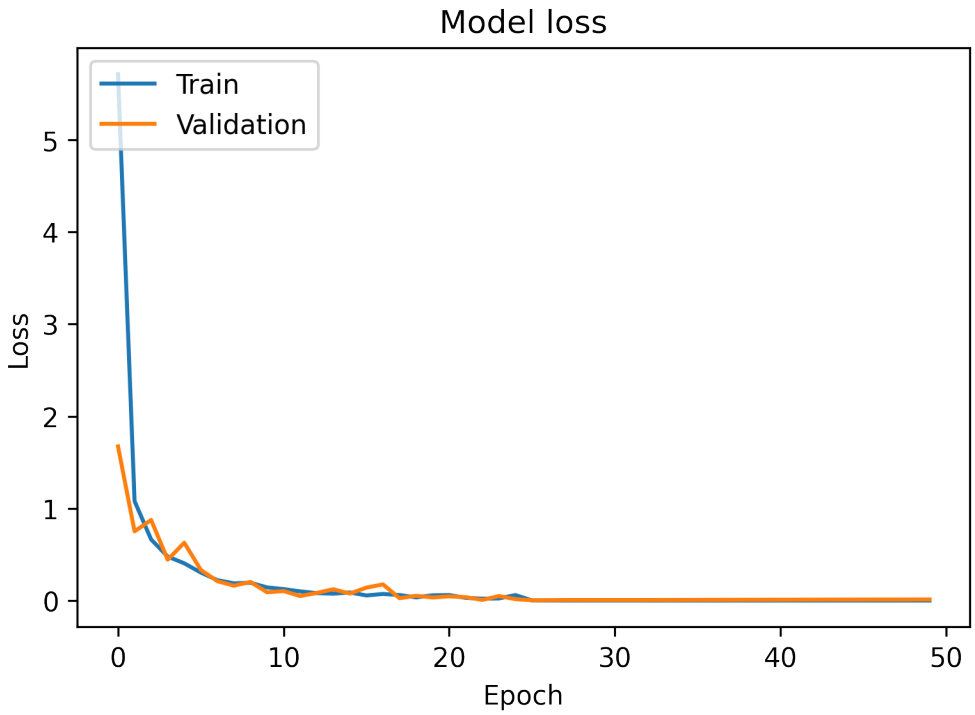

Figure 4. 30 Loss over Epoch

4.4.4. Learning-Based Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.20. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.31 and 4.32 respectively.

Table 4. 20: Training Results

Training Accuracy	99,60%
Validation Accuracy	94,65%
Training Loss	1,28e-4
Validation Loss	0,200
Training Time	559,54 seconds

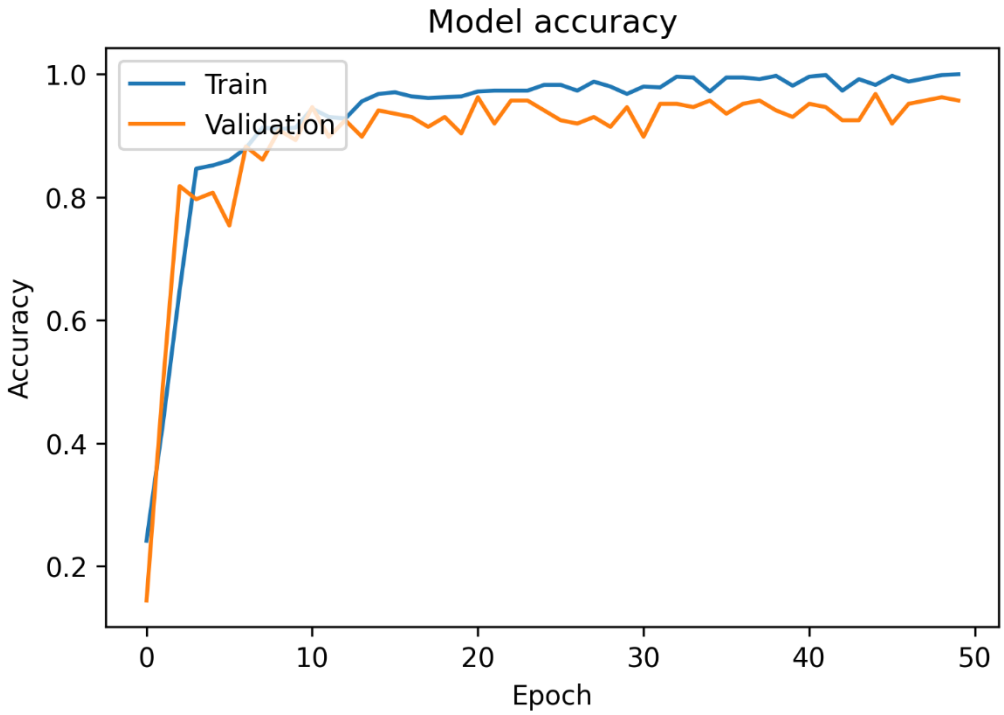

Figure 4. 31 Accuracy over Epoch

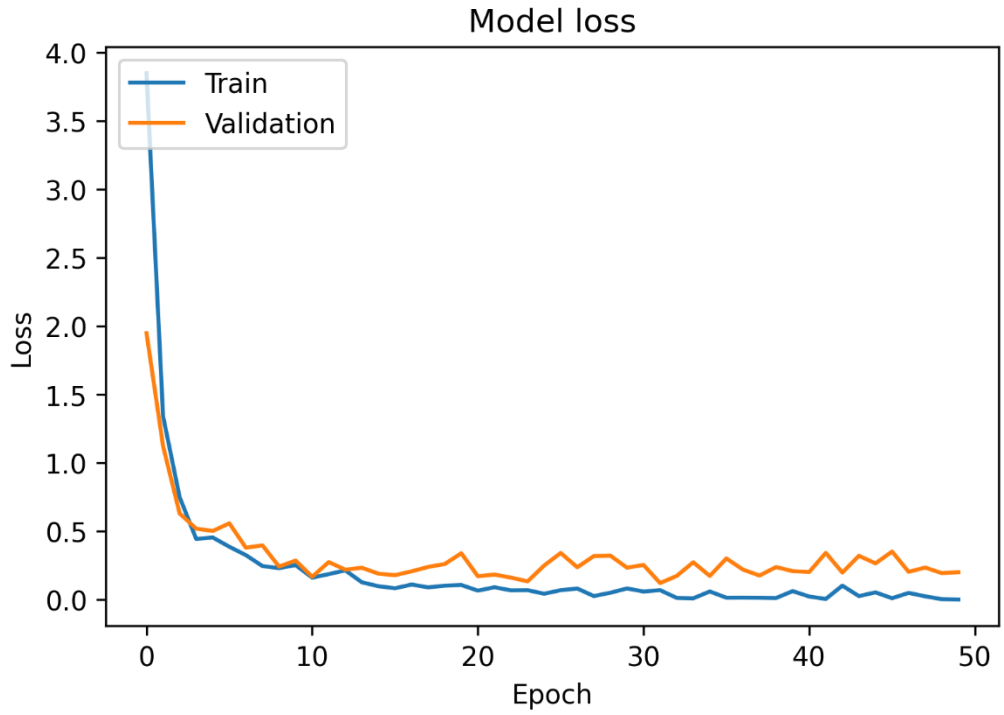

Figure 4. 32 Loss over Epoch

4.5. VGG-19 with Transfer Learning

Raw RadEch spectrograms and datasets that augmented in previous sections will be trained with VGG-19 network. This CNN configuration is shown in Figure 3.23. The

specifications of training process are shown in Table 4.21. In fact, the VGG-16 network has been trained and weighted with a dataset of more than 15 million images called ImageNet that explained in Section 3.5. Datasets will be trained with using these weights that is, with transfer learning. So that the results of transfer learning on micro-Doppler radar data can also be seen.

Table 4. 21: Training Specifications

Batch Size	8
Cross Validation	5-Fold Cross Validation
Image Size	224 x 224 x 3
Epoch	50
Model Optimizer	Adam optimizer
Learning Rate	0,001
Loss Function	Sparse Categorical Cross-entropy
Evaluation Metric	Accuracy
Activation Function of Classification Layer	Softmax Function

4.5.1. Raw RadEch Spectrograms

Raw RadEch Spectrograms contains 452 samples. The dataset is trained with 50 Epochs. The results obtained as a result of the training are shown in Table 4.22. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.33 and 4.34 respectively.

Table 4. 22: Training Results

Training Accuracy	81,57%
Validation Accuracy	74,77%
Training Loss	0,355
Validation Loss	0,532
Training Time	269,73 seconds

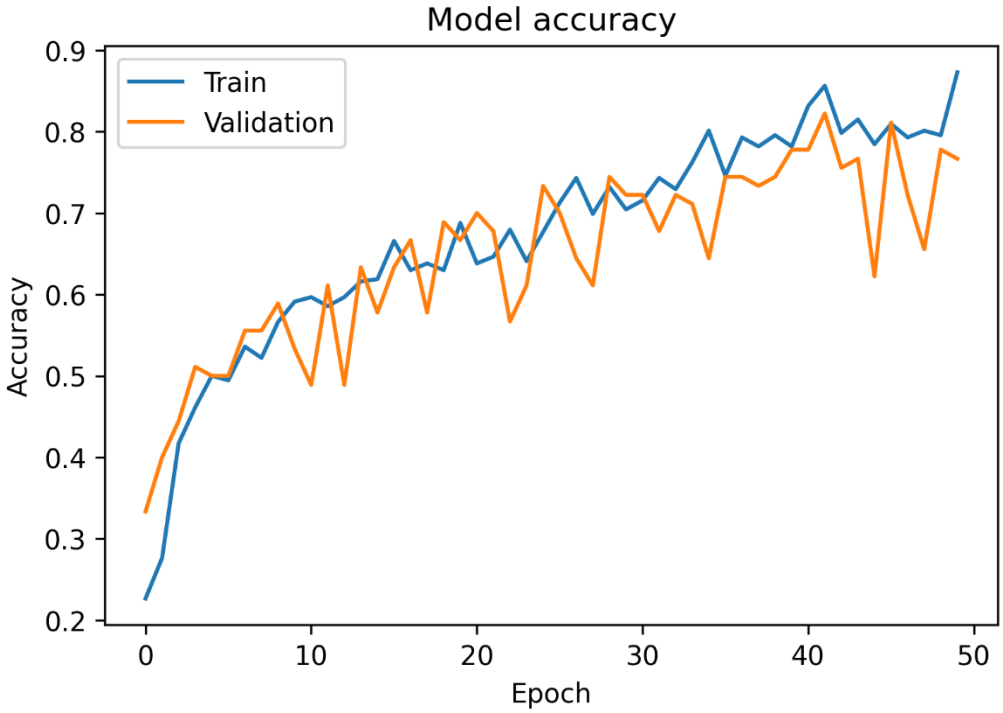

Figure 4. 33 Accuracy over Epoch

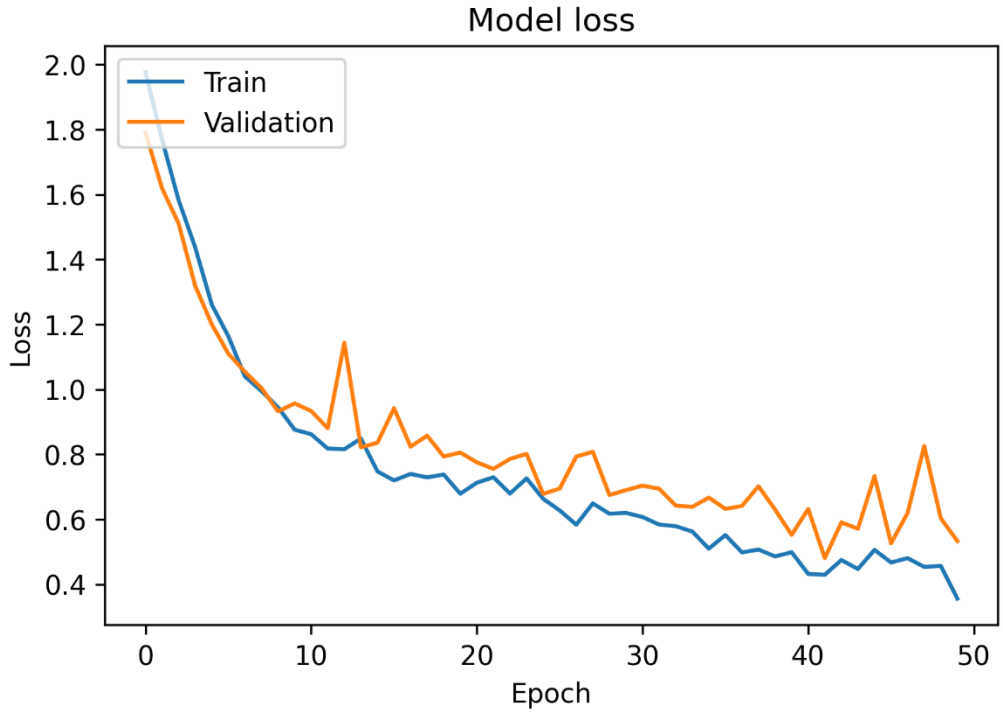

Figure 4. 34 Loss over Epoch

4.5.2. Noise Reduced RadEch Spectrograms

Noise Reduced RadEch Spectrograms contains 452 samples. Training epochs are selected as 50. The results obtained as a result of the training are shown in Table 4.23. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.35 and 4.36 respectively.

Table 4. 23: Training Results

Training Accuracy	93,81%
Validation Accuracy	90,44%
Training Loss	0,128
Validation Loss	0,149
Training Time	214,72 seconds

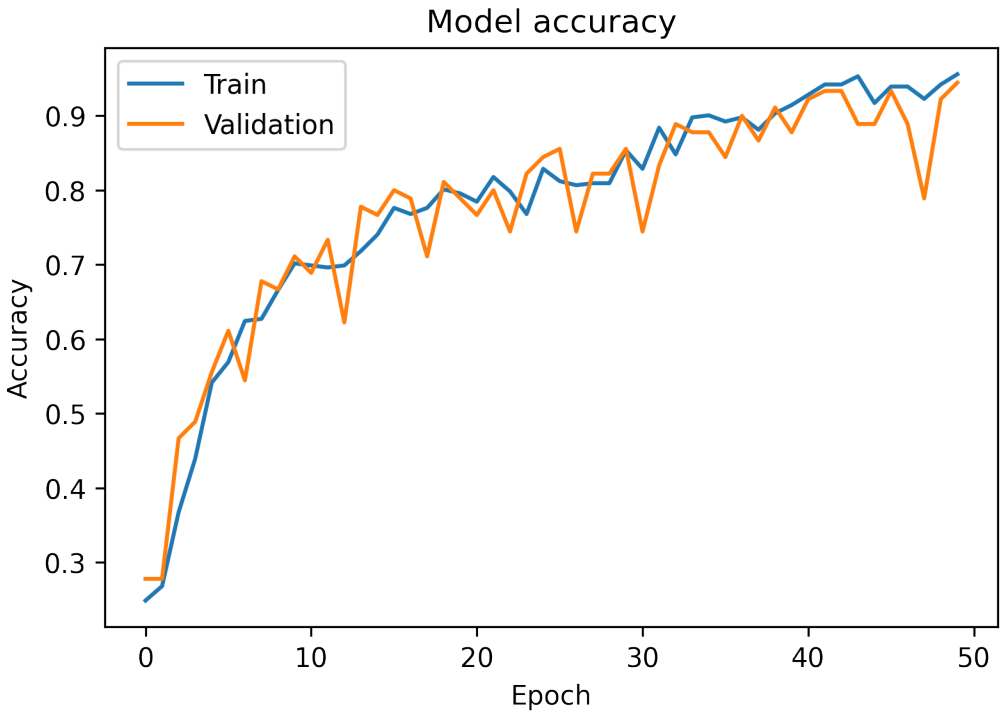

Figure 4. 35 Accuracy over Epoch

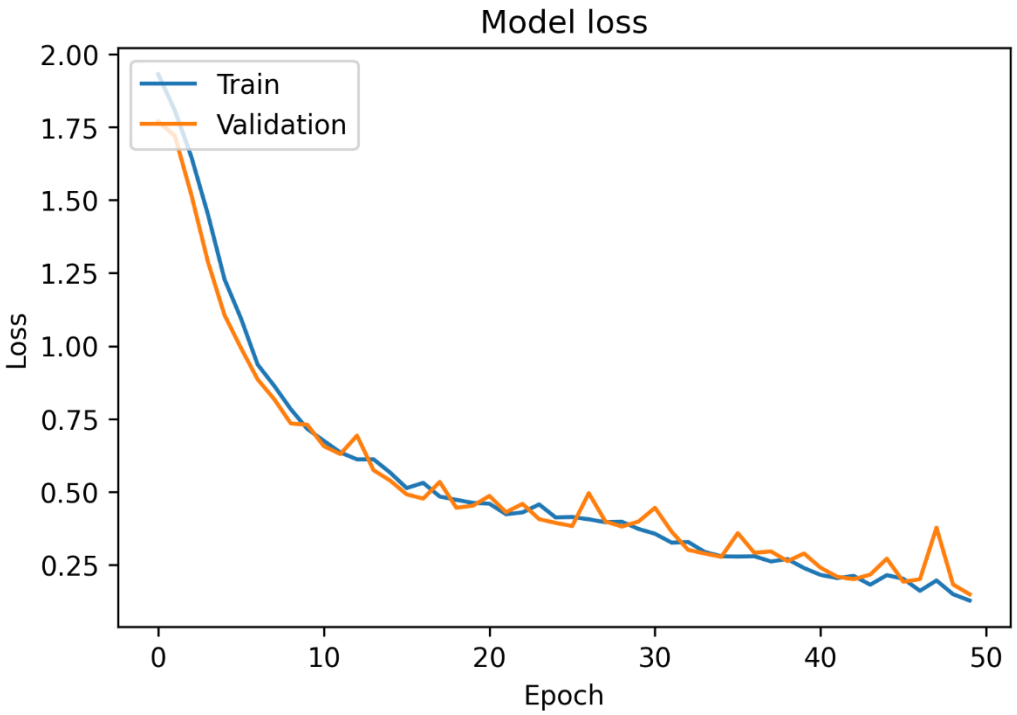

Figure 4. 36 Loss over Epoch

4.5.3. Traditional Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.24. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.37 and 4.38 respectively.

Table 4. 24: Training Results

Training Accuracy	90,49%
Validation Accuracy	83,17%
Training Loss	0,230
Validation Loss	0,274
Training Time	1311,30 seconds

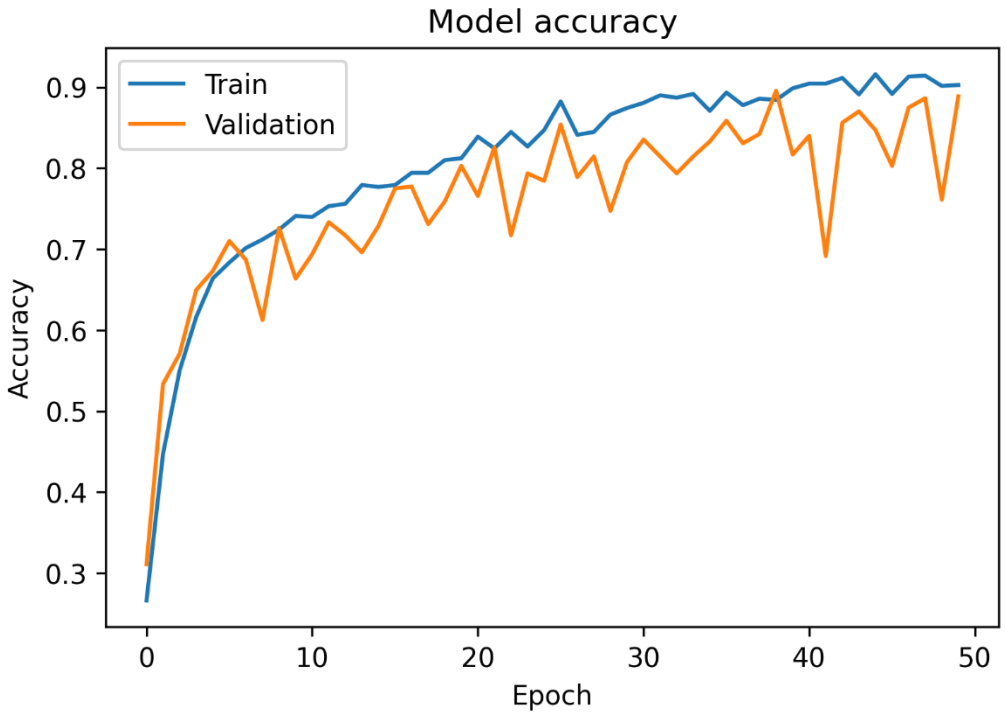

Figure 4. 37 Accuracy over Epoch

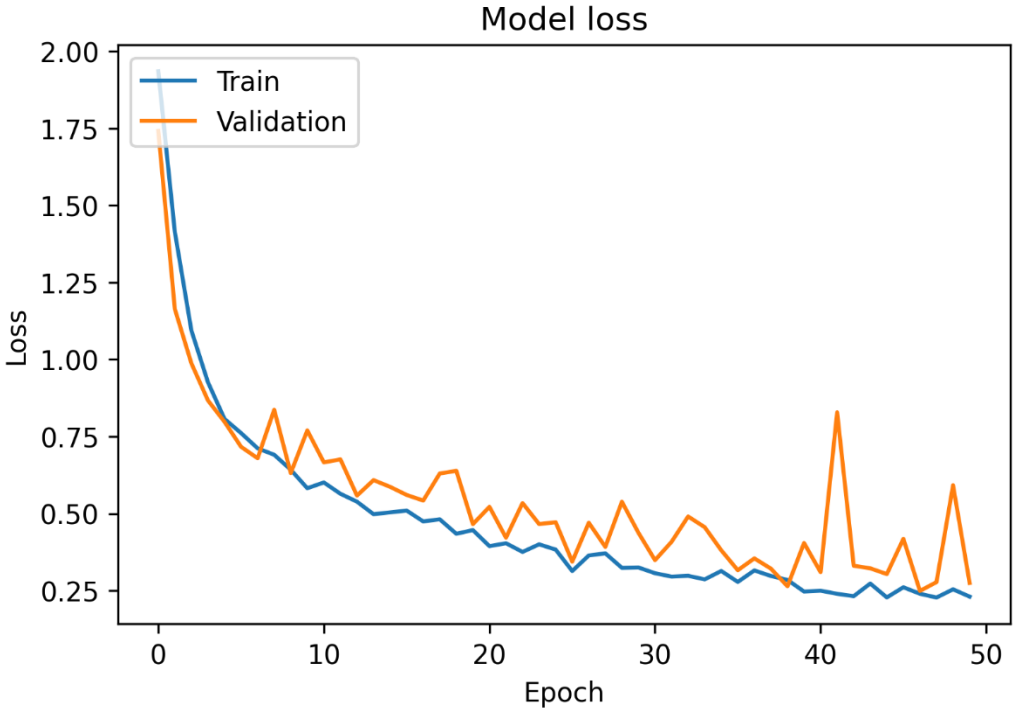

Figure 4. 38 Loss over Epoch

4.5.4. Learning-Based Augmented RadEch Dataset

Traditional Augmented RadEch Dataset contains 1953 samples. Training epochs are selected as 50. It is important to note that no augmented data were used in the testing process. It has only been validated with real spectrograms. The results obtained as a result of the training are shown in Table 4.25. Additionally, Training and validation accuracy over epoch graphs and training and validation loss over epoch graph are shown in Figure 4.39 and 4.40 respectively.

Table 4. 25: Training Results

Training Accuracy	93,56%
Validation Accuracy	90,42%
Training Loss	0,141
Validation Loss	0,153
Training Time	379,83 seconds

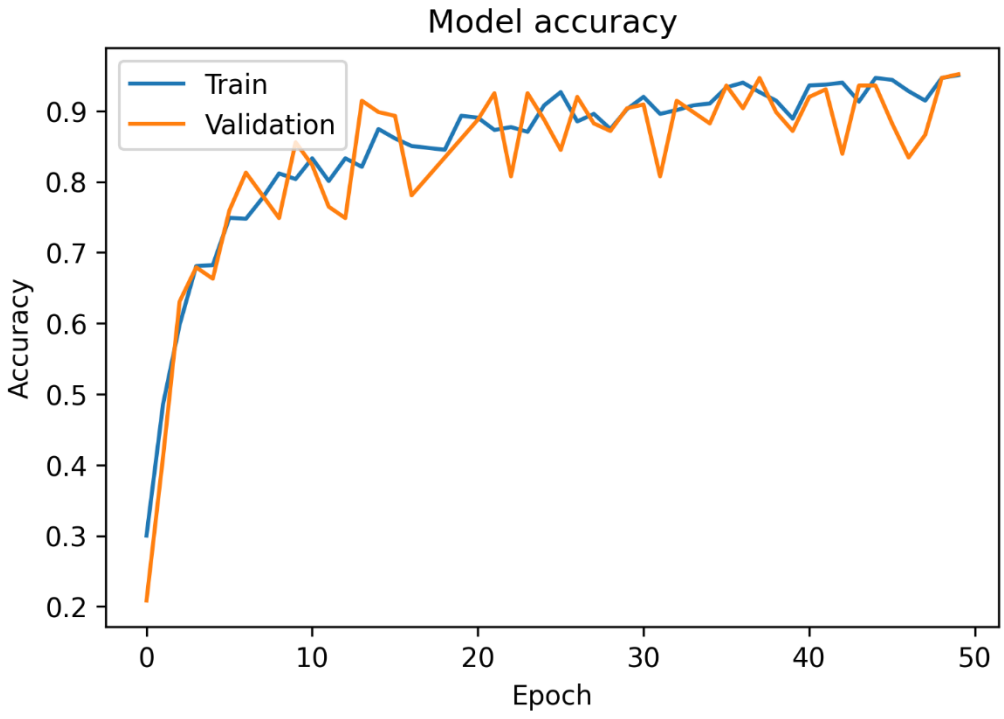

Figure 4. 39 Accuracy over Epoch

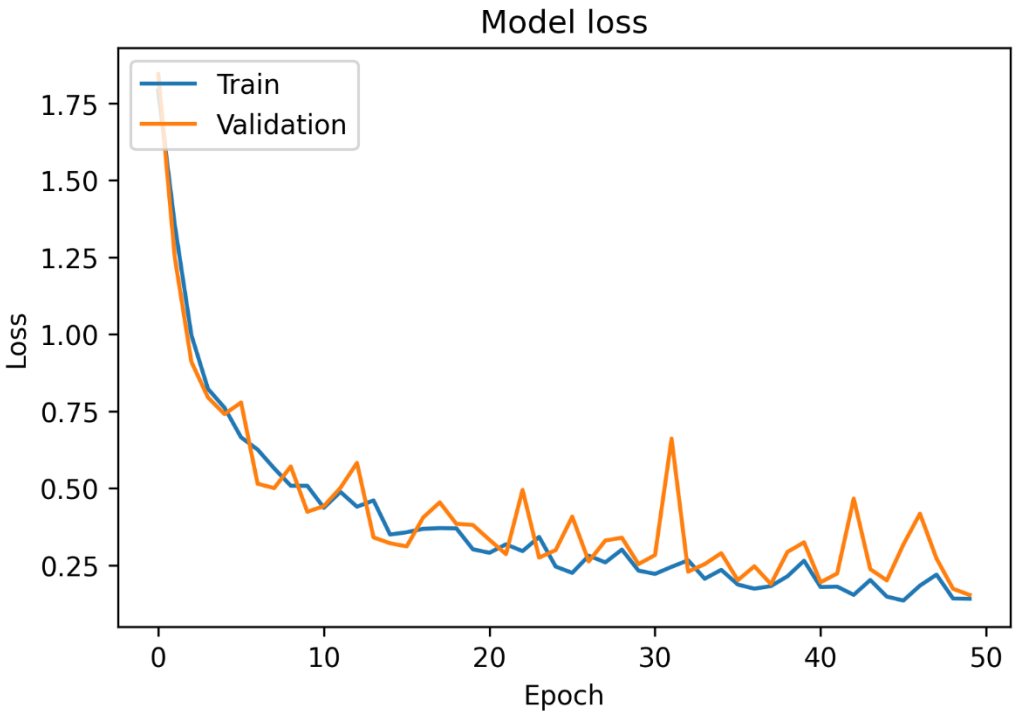

Figure 4. 40 Loss over Epoch

5. DISCUSSION

In Section 4, Raw RadEch dataset and pre-processed datasets are trained with three different deep learning models. Their results are shown with tables and graphs. In this section these results, and graphs are going to be evaluated. Table 5.1 shows 5-Layer CNN results, Table 5.2 shows VGG-16 without Transfer learning results, Table 5.3 shows VGG-16 with transfer learning results, Table 5.4 shows VGG-19 without transfer learning results and Table 5.5 shows VGG-19 with transfer learning results.

Table 5. 1: 5-Layer CNN Results

	Raw RadEch Spectrograms	**Noise Reduced Spectrograms**	**Traditional Augmented Spectrograms**	**Learning-Based Augmented Spectrograms**
Training Accuracy	99,45%	99,19%	99,88%	98,99%
Validation Accuracy	91,11%	99,12%	**99,76%**	94,07%
Training Loss	0,022	0,005	0,039	0,011
Validation Loss	0,197	0,003	0,021	0,156
Training Time	269,53 seconds	161,56 seconds	1204,09 seconds	379,43 seconds

Table 5. 2: VGG-16 without Transfer Learning Results

	Raw RadEch Spectrograms	**Noise Reduced Spectrograms**	**Traditional Augmented Spectrograms**	**Learning-Based Augmented Spectrograms**
Training Accuracy	99,99%	99,99%	100,00%	99,60%
Validation Accuracy	94,44%	98,89%	**99,77%**	94,65%
Training Loss	2,96e-5	1,926e-5	2,62e-9	1,28e-4
Validation Loss	0,279	0,058	0,011	0,200

Training Time	409,71 seconds	343,63 seconds	1857,00 seconds	559,54 seconds

Table 5. 3: VGG-16 with Transfer Learning Results

	Raw RadEch Spectrograms	**Noise Reduced Spectrograms**	**Traditional Augmented Spectrograms**	**Learning-Based Augmented Spectrograms**
Training Accuracy	71,12%	99,17%	100,00%	99,05%
Validation Accuracy	65,22%	97,78%	**99,77%**	95,72%
Training Loss	0,582	0,035	6,993e-8	6,263e-5
Validation Loss	0,775	0,066	0,022	0,131
Training Time	270,72 seconds	281,50 seconds	1902,37 seconds	632,96 seconds

Table 5. 4: VGG-19 without Transfer Learning Results

	Raw RadEch Spectrograms	**Noise Reduced Spectrograms**	**Traditional Augmented Spectrograms**	**Learning-Based Augmented Spectrograms**
Training Accuracy	99,99%	100,00 %	100,00%	99,60%
Validation Accuracy	91,11%	**100,00 %**	99,77%	94,65%
Training Loss	1,059e-5	1,135e-6	2,62e-9	1,28e-4
Validation Loss	0,330	2,315e-4	0,011	0,200
Training Time	402,60 seconds	339,72 seconds	1857,00 seconds	559,54 seconds

Table 5. 5: VGG-19 with Transfer Learning Results

	Raw RadEch Spectrograms	**Noise Reduced Spectrograms**	**Traditional Augmented Spectrograms**	**Learning-Based Augmented Spectrograms**
Training Accuracy	81,57%	93,81%	90,49%	93,56%
Validation Accuracy	74,77%	**90,44%**	83,17%	90,42%
Training Loss	0,355	0,128	0,230	0,141
Validation Loss	0,532	0,149	0,274	0,153
Training Time	269,73 seconds	214,72 seconds	1311,30 seconds	379,83 seconds

The first comment that can be made according to the results is that the transfer learning process using the VGG-16 and VGG-19 architectures did not give very successful results with the RadEch dataset. This situation can be explained by domain adaptation. VGG-16 and VGG-19 architectures are trained with a dataset called ImageNet [111]. This dataset contains many images from daily life. Trees, cars, airplanes, balloons and strawberries are examples of these images. Since these images in ImageNet do not resemble micro-Doppler radar spectrograms, it is quite normal that the weights of a model trained with this dataset will not be very successful in spectrogram classification. First of all, they have different domains. Therefore, using a deep learning architecture trained with ImageNet to classify micro-Doppler radar spectrograms would not be correct.

Another interpretation to be drawn from the results is that the classification success of Raw RadEch spectrograms is lower than noise reduced and augmented spectrograms. Noise reduced spectrograms are relatively noise-reduced spectrograms, so deep learning models do not have to learn noises. Based on this, the classification success of noise reduced spectrograms is expected to be higher than Raw Spectrograms. Augmented spectrograms, on the other hand, are expected to have higher classification success since they have more samples. Augmented spectrograms also protect deep learning models from overfitting.

When comparing traditional augmented and Learning-based augmented spectrograms, it can be seen that traditional augmented spectrograms have higher classification success. At this point, it should be considered that Generative Adversarial Networks used for learning-based augmentation need a lot of data in their training, otherwise they can easily be overfitted. Considering that the RadEch dataset is a relatively small dataset, it is quite normal that these GAN augmented spectrograms do not mimic the dataset well. For this reason, learning-based augmentation has had lower classification success than traditional data augmentation. It can be seen from the tables that the learning-based augmentation class is trained in shorter times compared to the traditional augmentation class. This can be explained by the fact that the dimensions of the spectrograms produced with GAN are 3 x 64 x 64. Although it is increased to 3 x 224 x 224 during classification, it will take much less time to learn this information as it contains less information.

Considering all the results, the highest classification success is 100% in noise reduced spectrograms trained with VGG-19 architecture, without transfer learning. Moreover, this result was obtained using only 452 samples. This result surpasses the results of E. Alhadhrami et al [103], a state-of-art study using the RadEch dataset.

E. Alhadhrami et al. In their early work [116], they achieved 99.9% classification success by increasing 452 samples to a total of 7684 samples using traditional augmentation. in their next study [103], they achieved 98.51% classification success with 452 samples.

Table 5. 6 Results Comparison

Study	**Number of Sample for RadEch Dataset**	**Feature Extraction Method**	**Classifier**	**Classification Accuracy**
[93]	452	TFD SVD-FT	SVM	93,00 %
[103]	452	VGG-19	Softmax	98,51 %
[116]	7681	AlexNet	Softmax	99,90 %
This Study	**452**	**VGG-19**	**Softmax**	**100,00 %**

5.1. Future Works

Since we achieved the main goal of this solution, there is some parts of the solution needs further work and development. In order to increase the generalization ability of the models, it is necessary to collect more micro-Doppler radar data at first. After more data is

obtained, depending on the number of samples, training with GAN and augmenting the data with better quality results and retraining with this augmented data. After acquiring more data and augmenting it with GAN, applying noise reduction to the entire dataset and observing the results. In order to increase the generalization ability rather than model performance, this dataset can be trained using the model previously trained with the micro-Doppler radar dataset. In this case, the transfer learning success of two different datasets with compatible domains on the same model can be compared.

6. CONCLUSION

The aim of this study is to classify micro-Doppler radar data, pre-process it before classification, augment the dataset due to its small size, and compare all these results first within themselves and then with state-of-art studies.

In this context, using various deep learning architectures, first the raw dataset, then the noise reduced dataset and then the augmented dataset are classified. As a result of the classification, the classification success of 100.00% was obtained as the highest success in the training of the noise reduced dataset, which has 452 samples, with the VGG-16 architecture. While training the dataset, 5-fold Cross-validation technique was applied. The results showed that traditional data augmentation methods produce these and better results than learning-based data augmentation methods. In order for the learning-based data augmentation method to be better, the dataset must be very large. Otherwise, the GAN will not be able to learn the correct distribution.

The VGG-16 and VGG-19 models have a domain incompatible with the RadEch dataset, as they are trained with a huge dataset called ImageNet, which contains many examples from everyday life. Since there is a domain adaptation problem, the models trained with transfer learning showed lower classification success in this dataset than the models trained without transfer learning.

7. REFERENCES

[1] Mark A. Richards, James A. Scheer, William A. Holm, “Principles of Modern Radar Basic Principles”, 2010

[2] Habibur Rahman, “Fundamental Principles of Radar”, 2019, ISBN 9781138387799

[3] Merrill Skolnik, “Radar Handbook”, Third Edition, McGraw-Hill Education, 2008

[4] Wai Kai Chen, “The electrical engineering handbook”,

[5] K. V. Mishra, Z. Slavik and O. Bringmann, "ReMCW: Reduced Bandwidth FMCW Radar for Autonomous Driving," *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, 2019, pp. 1427-1431, doi: 10.1109/IEEECONF44664.2019.9048773.

[6] Johansson, G., “Visual Perception of Biological Motion and a Model for Its Analysis,” Perception & Psychophysics, Vol. 14, 1973, pp. 201–211.

[7] Victor Chen, The Micro-Doppler Effect in Radar, Artech, 2011.

[8] “The Radar Equation” [Online] Access: 26.02.2021, Available: radartutorial.eu

[9] “ThoughtCo” [Online] Access: 01.03.2021, Available: thought.co

[10] Chen, V. C., et al., “Micro-Doppler Effect in Radar: Phenomenon, Model, and Simulation Study,” IEEE Transactions on Aerospace and Electronics Systems, Vol. 42, No. 1, 2006, pp. 2–21.

[11] Shirman, Y. D., “Computer Simulation of Aerial Target Radar Scattering, Recognition, Detection, and Tracking”, Norwood, MA: Artech House, 2002.

[12] Gorshkov, S. A., et al., “Radar Target Backscattering Simulation: Software and User’s Manual”, Norwood, MA: Artech House, 2002.

[13] Senigagliesi, L.; Ciattaglia, G.; De Santis, A.; Gambi, E. “People Walking Classification Using Automotive Radar”. Electronics 2020, 9, 588.

[14] Parker, K.J.; Robert, M.L.; Huang, S.-R. “Method and Apparatus for Using Doppler Modulation Parameters for Estimation of Vibration Amplitude.” U.S. Patent No. 5,086,775, 11 February 1992.

[15] Chen, V.C.; Li, F.; Ho, S.S.; Wechsler, H. “Micro-Doppler effect in radar: Phenomenon, model, and simulation study.” IEEE Trans. Aerosp. Electron. Syst. 2006, 42, 2–21

[16] Chen, V.C.; Li, F.; Ho, S.S.; Wechsler, H. “Analysis of micro-Doppler signatures.” IEEE Proc. Radar Sonar Navig. 2003, 150, 271–276

[17] Zeintl, C.; Eibensteiner, F.; Langer, J. “Evaluation of FMCW Radar for Vibration Sensing in Industrial Environments.” In Proceedings of the 29th IEEE International Conference Radioelektronika (RADIOELEKTRONIKA), Pardubice, Czech Republic, 16–18 April 2019.

[18] Thayananthan, T.; Stankovi´c, L.J.; Djurovi´c, I. “Micro-Doppler-based target detection and feature extraction in indoor and outdoor environments.” J. Frankl. Inst. 2008, 345, 700–722.

[19] Ding, L.; Ali, M.; Patole, S.; Dabak, A. “Vibration parameter estimation using FMCW Radar.” In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 2224–2228.

[20] Clemente, C.; Balleri, A.; Woodbridge, K.; Soraghan, J.J. “Developments in target micro-Doppler signatures analysis: Radar imaging, ultrasound and through-the-wall radar.” Eurasip J. Adv. Signal Process. 2013, 2013, 47

[21] Chen, V.C.; Tahmoush, D.; William J.M. “Radar Micro-Doppler Signatures: Processing and Applications”; Institution of Engineering and Technology: London, UK, 2014.

[22] Gianluca Ciattaglia, “Performance Evaluation of Vibrational Measurements through mmWave Automotive Radars”, Remote Sens. 2021, 13, 98.

[23] Chen, V. C., “Analysis of Radar Micro-Doppler Signature with Time-Frequency Transform,” Proc. of the IEEE Workshop on Statistical Signal and Array Processing (SSAP), Pocono, PA, 2000, pp. 463–466.

[24] Baker, C. J., and B. D. Trimmer, “Short-Range Surveillance Radar Systems,” Electronics & Communication Engineering Journal, August 2000, pp. 181–191.

[25] Konstantinos G. Liakos, “Machine Learning in Agriculture: A Review”

[26] Pearson, K. “On lines and planes of closest fit to systems of points in space”. Lond. Edinb. Dublin Philos. Mag. J. Sci. 1901, 2, 559–572.

[27] Wold, H. Partial Least Squares. In Encyclopedia of Statistical Sciences; John Wiley & Sons: Chichester, NY, USA, 1985; Volume 6, pp. 581–591, ISBN 9788578110796.

[28] Fisher, R.A. “The use of multiple measures in taxonomic problems”. Ann. Eugen 1936, 7, 179–188.

[29] Cox, D.R. “The Regression Analysis of Binary Sequences.” J. R. Stat. Soc. Ser. B 1958, 20, 215-242

[30] Efroymson, M.A. “Multiple regression analysis. Math. Methods Digit. Comput.” 1960, 1, 191–203

[31] Craven, B.D.; Islam, S.M.N. “Ordinary least-squares regression.” SAGE Dict. Quant. Manag. Res. 2011, 224–228.

[32] Friedman, J.H. “Multivariate Adaptive Regression Splines.” Ann. Stat. 1991, 19, 1–67

[33] Quinlan, J.R. “Learning with continuous classes.” Mach. Learn. 1992, 92, 343–348.

[34] Cleveland, W.S. “Robust locally weighted regression and smoothing scatterplots.” J. Am. Stat. Assoc. 1979, 74, 829–836.

[35] Belson, W.A. “Matching and Prediction on the Principle of Biological Classification.” Appl. Stat. 1959, 8, 65–75.

[36] Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. “Classification and Regression Trees”; Routledge: Abingdon, UK, 1984; Volume 19, ISBN 0412048418.

[37] Tryon, R.C. Communality of a variable: “Formulation by cluster analysis.” Psychometrika 1957, 22, 241–260.

[38] Lloyd, S.P. “Least Squares Quantization in PCM.” IEEE Trans. Inf. Theory 1982, 28, 129–137.

[39] Johnson, S.C. “Hierarchical clustering schemes.” Psychometrika 1967, 32, 241–254

[40] Dempster, A.P.; Laird, N.M.; Rubin, D.B. “Maximum likelihood from incomplete data via the EM algorithm”. J. R. Stat. Soc. Ser. B Methodol. 1977, 39, 1–38

[41] Kohonen, T. “Learning vector quantization.” Neural Netw. 1988, 1, 303.

[42] Belson, W.A. “Matching and Prediction on the Principle of Biological Classification.” Appl. Stat. 1959, 8, 65–75.

[43] Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. “Classification and Regression Trees”; Routledge: Abingdon, UK, 1984; Volume 19, ISBN 0412048418.

[44] Kass, G.V. “An Exploratory Technique for Investigating Large Quantities of Categorical Data.” Appl. Stat. 1980, 29, 119.

[45] Suykens, J.A.K.; Vandewalle, J. “Least Squares Support Vector Machine Classifiers.” Neural Process. Lett. 1999, 9, 293–300.

[46] Chang, C.; Lin, C. LIBSVM: “A Library for Support Vector Machines.” ACM Trans. Intell. Syst. Technol. 2013, 2, 1–39.

[47] Smola, A. “Regression Estimation with Support Vector Learning Machines.” Master’s Thesis, The Technical University of Munich, Munich, Germany, 1996; pp. 1–78.

[48] Suykens, J.A.K.; Van Gestel, T.; De Brabanter, J.; De Moor, B.; Vandewalle, J. “Least Squares Support Vector Machines”; World Scientific: Singapore, 2002; ISBN 9812381511.

[49] LeCun, Y., Bengio, Y., 1995. “Convolutional networks for images, speech, and time series.” Handbook Brain Theory Neural Networks 3361.

[50] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

[51] Andreas Kamilaris, Deep learning in agriculture: A survey

[52] Sehgal, G., Gupta, B., Paneri, K., Singh, K., Sharma, G., Shroff, G., 2017. “Crop Planning using Stochastic Visual Optimization.” arXiv: 1710.09077.

[53] Song, X., Zhang, G., Liu, F., Li, D., Zhao, Y., Yang, J., 2016. „Modeling spatio-temporal distribution of soil moisture by deep learning-based cellular automata model.” J. Arid Land 8 (5), 734–748.

[54] Demmers, T.G., Cao, Y., Parsons, D.J., Gauss, S., Wathes, C.M., 2012. “Simultaneous monitoring and control of pig growth and ammonia emissions.” IX International Livestock Environment Symposium (ILES IX). American Society of Agricultural and Biological Engineers, Valencia, Spain.

[55] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T., 2014. “Caffe: Convolutional architecture for fast feature embedding.” In: Proceedings of the 22nd International Conference on Multimedia. ACM, Orlando, FL, USA, pp. 675–678.

[56] Amara, J., Bouaziz, B., Algergawy, A., 2017. „A Deep Learning-Based Approach for Banana Leaf Diseases Classification.” BTW workshop, Stuttgart, pp. 79–88.

[57] Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y., 2014. “Deep learning-based classification of hyperspectral data.” IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (6), 2094–2107.

[58] McCulloch,W.S.; Pitts,W. “A logical calculus of the ideas immanent in nervous activity.” Bull. Math. Biophys. 1943, 5, 115–133

[59] Broomhead, D.S.; Lowe, D. “Multivariable Functional Interpolation and Adaptive Networks.” Complex Syst. 1988, 2, 321–355.

[60] Rosenblatt, F. “The perceptron: A probabilistic model for information storage and organization in the brain.” Psychol. Rev. 1958, 65, 386–408.

[61] Linnainmaa, S. “Taylor expansion of the accumulated rounding error.” BIT 16, 146–160 (1976). https://doi.org/10.1007/BF0193136

[62] Riedmiller, M.; Braun, H. “A direct adaptive method for faster backpropagation learning: The RPROP algorithm.” In Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; pp. 586–591.

[63] Hecht-Nielsen, R. „Counterpropagation networks.“ Appl. Opt. 1987, 26, 4979–4983.

[64] Jang, J.S.R. “ANFIS: Adaptive-Network-Based Fuzzy Inference System.” IEEE Trans. Syst. Man Cybern. 1993, 23, 665–685.

[65] Melssen, W.; Wehrens, R.; Buydens, L. “Supervised Kohonen networks for classification problems.” Chemom. Intell. Lab. Syst. 2006, 83, 99–113.

[66] Hopfield, J.J. “Neural networks and physical systems with emergent collective computational abilities.” Proc. Natl. Acad. Sci. USA 1982, 79, 2554–2558.

[67] Pal, S.K.; Mitra, S. “Multilayer Perceptron, Fuzzy Sets, and Classification.” IEEE Trans. Neural Netw. 1992, 3, 683–697

[68] Kohonen, T. “The Self-Organizing Map.” Proc. IEEE 1990, 78, 1464–1480.

[69] Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. “Extreme learning machine: Theory and applications.” Neurocomputing 2006, 70, 489–501.

[70] Specht, D.F. “A general regression neural network.” IEEE Trans. Neural Netw. 1991, 2, 568–576.

[71] Cao, J.; Lin, Z.; Huang, G. Bin “Self-adaptive evolutionary extreme learning machine. “Neural Process. Lett. 2012, 36, 285–305

[72] LeCun, Y.; Bengio, Y.; Hinton, G. “Deep learning.” Nature 2015, 521, 436–444

[73] Goodfellow, I.; Bengio, Y.; Courville, “A. Deep Learning”; MIT Press: Cambridge, MA, USA, 2016; pp. 216–261.

[74] Salakhutdinov, R.; Hinton, G. “Deep Boltzmann Machines.” Aistats 2009, 1, 448–455. [CrossRef]

[75] Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.-A. “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion” Pierre-Antoine Manzagol. J. Mach. Learn. Res. 2010, 11, 3371–3408.

[76] O'Shea, Keiron & Nash, Ryan. (2015). “An Introduction to Convolutional Neural Networks”. ArXiv e-prints.

[77] D. Chu, I. Demir, K. Eichensehr, J. G. Foster, M. L. Green, K. Lerman, F. Menczer, C. O’Connor, E. Parson, and L. Ruthotto. White paper: “Deep fakery|an action plan”. Technical report, IPAM, 2020.

[78] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). “Generative Adversarial Networks.” *ArXiv, abs/1406.2661*.

[79] LeCun, Yann and Cortes, Corinna. "MNIST handwritten digit database." (2010):

[80] Ruthotto & Haber, “An Introduction to Deep Generative Modeling” (2021)

[81] S. Nag, M. A. Barnes, T. Payment, and G. Holladay, “Ultrawideband through-wall radar for detecting the motion of people in real time,” in Proc. SPIE Radar Sensor Technol. Data Vis., Jul. 2002, vol. 4744, pp. 48–57.

[82] A. R. Hunt, “A wideband imaging radar for through-the-wall surveillance,” in Proc. SPIE Sensors, Command, Control, Commun., Intell. (C3I) Technol., Sep. 2004, vol. 5403, pp. 590–596.

[83] J. L. Geisheimer, E. F. Greneker, and W. S. Marshall, “High-resolution Doppler model of the human gait,” in Proc. SPIE Radar Sensor Technol. Data Vis., Jul. 2002, vol. 4744, pp. 8–18.

[84] P. van Dorp and F. C. A. Groen, “Human walking estimation with radar,” Proc. Inst. Elect. Eng.—Radar, Sonar Navig., vol. 150, no. 5, pp. 356–365, Oct. 2003.

[85] M. Otero, “Application of a continuous wave radar for human gait recognition,” in Proc. SPIE Signal Process., Sensor Fusion, Target Recog., May 2005, vol. 5809, pp. 538–548.

[86] J. Li and H. Ling, “ISAR feature extraction from non-rigid body targets using adaptive chirplet signal representation,” Proc. Inst. Elect. Eng.—Radar, Sonar Navig., vol. 150, pp. 284–291, Aug. 2003.

[87] T. Thayaparan, S. Abrol, E. Riseborough, L. Stankovic, D. Lamothe, and G. Duff, “Analysis of radar micro-Doppler signatures from experimental helicopter and human data,” IET Radar, Sonar Navig., vol. 1, no. 4, pp. 289–299, Aug. 2007.

[88] P. Setlur, M. Amin, and F. Ahmad, “Urban target classifications using time-frequency micro-Doppler signatures,” in Proc. 9th Int. Symp. Signal Process. Appl., Feb. 2007, pp. 1–4.

[89] A. G. Stove and S. R. Sykes, “A Doppler-based automatic target classifier for a battlefield surveillance radar,” in Proc. IEEE Radar Conf., Oct. 2002, pp. 419–423.

[90] Y. Kim and H. Ling, "Human Activity Classification Based on Micro-Doppler Signatures Using a Support Vector Machine," in IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 5, pp. 1328-1337, May 2009, doi: 10.1109/TGRS.2009.2012849.

[91] W.D. van Eeden, J.P. de Villiers, R.J. Berndt, W.A.J. Nel, E. Blasch, Micro-Doppler radar classification of humans and animals in an operational environment, Expert Systems with Applications, Volume 102, 2018, Pages 1-11, ISSN 0957-4174,

[92] F. H. C. Tivive, S. L. Phung, and A. Bouzerdoum, ``An image-based approach for classification of human micro-Doppler radar signatures," Proc. SPIE, vol. 8734, May 2013, Art. no. 873406.

[93] P. Molchanov, J. Astola, K. Egiazarian, and A. Totsky, ``Frequency and phase coupling phenomenon in micro-Doppler radar signature of walking human," in Proc. 13th Int. Radar Symp., May 2012, pp. 49.53.

[94] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, ‘‘Review of deep convolution neural network in image classification,’’ in Proc. Int. Conf. Radar, Antenna Microw. Electron. Telecommun. (ICRAMET), Oct. 2017, pp. 26–31.

[95] Y. Kim and T. Moon, ‘‘Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks,’’ IEEE Geosci. Remote Sens. Lett., vol. 13, no. 1, pp. 8–12, Jan. 2016.

[96] J. Lundén and V. Koivunen, ‘‘Deep learning for HRRP-based target recognition in multistatic radar systems,’’ in Proc. IEEE Radar Conf., May 2016, pp. 1–6.

[97] B. K. Kim, H.-S. Kang, and S.-O. Park, ‘‘Drone classification using convolutional neural networks with merged Doppler images,’’ IEEE Geosci. Remote Sens. Lett., vol. 14, no. 1, pp. 38–42, Jan. 2017.

[98] J. Park, R. J. Javier, T. Moon, and Y. Kim, ‘‘Micro-Doppler based classification of human aquatic activities via transfer learning of convolutional neural networks,’’ Sensors, vol. 16, no. 12, p. 1990, Nov. 2016.

[99] P. van Dorp and F. Groen, “Human walking estimation with radar,” IEEE Proceedings - Radar, Sonar and Navigation, vol. 150, pp. 356–365(9), October 2003

[100] S. Groot et al., “Human motion classification using a particle filter approach: multiple model particle filtering applied to the micro-doppler spectrum,” International Journal of Microwave and Wireless Technologies, vol. 5, pp. 391–399, 6 2013.

[101] J. Li et al., “Automatic classification of human motions using doppler radar,” in Proc. IEEE IJCNN, June 2012, pp. 1–6.

[102] R. P. Trommel, R. I. A. Harmanny, L. Cifola and J. N. Driessen, "Multi-target human gait classification using deep convolutional neural networks on micro-doppler spectrograms," 2016 European Radar Conference (EuRAD), 2016, pp. 81-84.

[103] E. Alhadhrami, M. Al-Mufti, B. Taha and N. Werghi, "Learned Micro-Doppler Representations for Targets Classification Based on Spectrogram Images," in IEEE Access, vol. 7, pp. 139377-139387, 2019, doi: 10.1109/ACCESS.2019.2943567.

[104] M. S. Andrić, B. P. Bondžulić, and B. M. Zrnić, ‘‘The database of radar echoes from various targets with spectral analysis,’’ in Proc. 10th Symp. Neural Netw. Appl. Elect. Eng., Sep. 2010, pp. 187–190.

[105] Simonyan, Karen & Zisserman, Andrew. (2014). “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv 1409.1556.

[106] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” Nature Methods, 17(3), 261-272.

[107] John D. Hunter. “Matplotlib: A 2D Graphics Environment,” Computing in Science & Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55 (publisher link)

[108] Umesh, P., 2012. “Image Processing in Python.” CSI Communications, 23.

[109] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. “Array programming with NumPy.” Nature 585, 357–362 (2020). DOI: 0.1038/s41586-020-2649-2.

[110] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library.” In Advances in Neural Information Processing Systems 32 (pp. 8024–8035).

[111] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). “Imagenet: A large-scale hierarchical image database.” In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).

[112] Simonyan, Karen & Zisserman, Andrew. (2014). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. arXiv 1409.1556.

[113] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). “ImageNet Classification with Deep Convolutional Neural Networks.” Neural Information Processing Systems. 25. 10.1145/3065386.

[114] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge.” IJCV, 2015

[115] “What is Cross Validation in Machine learning? Types of Cross Validation,” GreatLearning Blog. 26-Apr-2021. [Online]. Available: mygreatlearning.com, [Accessed: 05-Jul-2021].

[116] E. A. Hadhrami, M. A. Mufti, B. Taha and N. Werghi, "Transfer learning with convolutional neural networks for moving target classification with micro-Doppler radar spectrograms," 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), 2018, pp. 148-154, doi: 10.1109/ICAIBD.2018.8396184.