

HUMAN ACTIVITY CLASSIFICATION VIA TRAINABLE FRACTIONAL FOURIER TRANSFORM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Özenç Uğur
August 2025

Human Activity Classification via Trainable Fractional Fourier Transform

By Özenç Uğur

August 2025

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Ömer Morgül(Advisor)

Aykut Koç

İsmail Uyanık

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

HUMAN ACTIVITY CLASSIFICATION VIA TRAINABLE FRACTIONAL FOURIER TRANSFORM

Özenç Uğur

M.S. in Electrical and Electronics Engineering

Advisor: Ömer Morgül

August 2025

The human activity classification regarding micro-Doppler characteristics has been a captivating research area since the introduction of the micro-Doppler concept in various fields such as security surveillance, healthcare monitoring/diagnostics, and gait analysis. The extraction of micro-Doppler signatures through radar sensing and subsequent time-frequency analysis of humans or other targets enables precise and reliable characterization of their motion dynamics, allowing for robust classification even in cluttered or low-resolution environments. Compared to conventional vision-based or wearable sensing methods, radar systems are favored due to their inherent ability to preserve target privacy without capturing identifiable visual features.

To extract the meaningful micro-Doppler signatures embedded in the radar data, assorted time-frequency analysis approaches have been employed in the literature, especially the Short-Time Fourier Transform (STFT), due to simplicity and computational efficiency. The Fourier transform-based approach might be broadened by utilizing the Fractional Fourier Transform (FrFT), which generalizes the classical Fourier transform by considering the continuum of infinitely many representations between the time and frequency domains. Thanks to the opportunity to work in intermediate domains provided by the Fractional Fourier transform, more distinguishable representations of the micro-Doppler signatures might be obtained.

Motivated by the limitations of conventional approaches that depend on fixed time-frequency representations, this thesis proposes a deep learning-based classification methodology for the micro-Doppler signature classification of human activities based on the Fractional Fourier transform. Instead of using the singular

usage of the fractional domain representations by searching for optimal transform order empirically, trainable Fractional Fourier transform blocks are engaged in different deep learning architectures. Spectrograms representing the time-frequency characteristics are constructed within the deep-learning architectures, aiming for optimum representations via trainable fractional Fourier transform blocks dynamically.

The proposed approach is evaluated with multiple radar-based datasets, including both simulated and real-world measurements with different radars and configurations. A comprehensive set of experiments is conducted with various deep learning architectures, including single-branch CNN, LSTM, and GRU-based architectures, with the multi-branch configurations including the simultaneous use of different time-frequency representations obtained via fractional Fourier transform and multi-input Siamase-based models. The proposed approach is compared in detail with the conventional Fourier transform approach in distinct experiments, and the behavior of the fractional representations across the models and datasets is examined.

As far as we are aware, this study is the first to combine a trainable Fractional Fourier Transform with micro-Doppler signature-based human activity classification. The experimental results demonstrate that the proposed approach consistently outperforms traditional Fourier Transform-based methods. Moreover, the simultaneous utilization of multiple time-frequency representations with distinct FrFT orders is, to the best of our knowledge, a novel contribution to the literature, yielding considerable performance improvements. It is believed that the findings of this study pave the way for future research on adaptive time-frequency analysis in radar-based sensing applications.

Keywords: Micro-Doppler Signatures, Radar Sensing, Time-Frequency Analysis, Fractional Fourier Transform, Human Activity Recognition, Deep Learning, Trainable Transformations, Spectrogram Classification.

ÖZET

ÖĞRENİLEBİLİR KESİRLİ FOURİER DÖNÜŞÜMÜ İLE İNSAN AKTİVİTESİ SINIFLANDIRMASI

Özenç Uğur

Elektrik ve Elektronik Mühendisliği, Yüksek Lisans

Tez Danışmanı: Ömer Morgül

Ağustos 2025

Mikro-Doppler karakteristiklerine dayalı insan aktivitesi sınıflandırması, mikro-Doppler kavramının güvenlik gözetimi, sağlık izleme/teşhisi ve yürüme analizi gibi çeşitli alanlarda kullanılmaya başlanmasından bu yana ilgi çekici bir araştırma alanı haline gelmiştir. Mikro-Doppler imzalarının radar algılaması yoluyla elde edilmesi ve ardından insanlara veya diğer hedeflere ait zaman-frekans analizlerinin gerçekleştirilmesi, hareket dinamiklerinin hassas ve güvenilir bir şekilde karakterize edilmesini sağlamak ve bu sayede karmaşık ya da düşük çözünürlüklü ortamlarda dahi sağlam bir sınıflandırma yapılmasına olanak tanımaktadır. Geleneksel görsel tabanlı veya giyilebilir algılayıcılara kıyasla, radar sistemleri yalnızca hareket bilgisi toplaması ve kimlik belirleyici görsel özellikleri içermemesi nedeniyle gizlilik açısından daha avantajlı bulunmakta ve bu nedenle sıklıkla tercih edilmektedir.

Radar verisine gömülü anlamlı mikro-Doppler imzalarını ortaya çıkarmak amacıyla, literatürde çeşitli zaman-frekans analiz yöntemleri kullanılmıştır. Bu yöntemler arasında özellikle Kısa Süreli Fourier Dönüşümü (STFT), sade yapısı ve hesaplama verimliliği nedeniyle yaygın olarak tercih edilmektedir. Bununla birlikte, klasik Fourier dönüşümüne dayalı yaklaşımlar, zaman ve frekans alanları arasında sonsuz sayıda ara gösterimi mümkün kılan Kesirli Fourier Dönüşümü (FrFT) ile genişletilebilir. FrFT'nin sunduğu bu ara alanlarda çalışabilme yetisi sayesinde, mikro-Doppler imzalarının daha ayırt edici ve bilgilendirici temsilleri elde edilebilmektedir.

Sabit zaman-frekans çözünürlüğüne bağlı geleneksel yaklaşımların kısıtlarından yola çıkarak, bu tez, Kesirli Fourier Dönüşümüne dayalı yeni bir derin öğrenme tabanlı insan aktivitesi sınıflandırma yöntemi önermektedir. FrFT temsillerinin yalnızca tek bir dönüşüm mertebesiyle ve ampirik aramalarla kullanılması yerine,

öğrenilebilir FrFT blokları farklı derin öğrenme mimarilerine entegre edilmiştir. Bu mimarilerde zaman-frekans özelliklerini temsil eden spektrogramlar, öğrenme sürecinin bir parçası olarak dinamik biçimde üretilmekte ve dönüşüm mertebeleri ağ tarafından optimize edilmektedir.

Önerilen yaklaşım, farklı radar yapılandırmalarıyla toplanmış hem simülasyon hem de gerçek dünya verilerini içeren çeşitli radar tabanlı veri kümeleri üzerinde değerlendirilmiştir. Tek dallı CNN, LSTM ve GRU mimarilerinin yanı sıra, farklı FrFT temsillerinin eşzamanlı kullanıldığı çok dallı (multi-branch) yapılar ve çok girişli Siamese tabanlı modellerle kapsamlı deneyler gerçekleştirilmiştir. Önerilen yöntem, klasik Fourier dönüşümüne dayalı yaklaşımlarla detaylı olarak karşılaştırılmış ve FrFT temsillerinin modeller ve veri kümeleri üzerindeki davranışları analiz edilmiştir.

Bildiğimiz kadarıyla, bu çalışma mikro-Doppler imzalarına dayalı insan aktivitesi sınıflandırma problemini öğrenilebilir Kesirli Fourier Dönüşümüyle birleştiren ilk çalışmadır. Deneysel sonuçlar, önerilen yaklaşımın geleneksel Fourier tabanlı yöntemleri tutarlı şekilde geride bıraktığını göstermektedir. Ayrıca, farklı FrFT mertebeleriyle elde edilen çoklu zaman-frekans temsillerinin eşzamanlı kullanımı da literatürde ilk kez önerilmekte olup, performans açısından kayda değer iyileşmeler sağlamaktadır. Bu çalışmanın bulgularının, radar tabanlı algılama sistemlerinde uyarlanabilir zaman-frekans analizine yönelik gelecekteki araştırmalara yol göstereceğine inanılmaktadır.

Anahtar sözcükler: Mikro-Doppler İmzaları, Radar Algılama, Zaman-Frekans Analizi, Kesirli Fourier Dönüşümü, İnsan Aktivitesi Sınıflandırması, Derin Öğrenme, Öğrenilebilir Dönüşümler, Spektrogram Tabanlı Sınıflandırma.

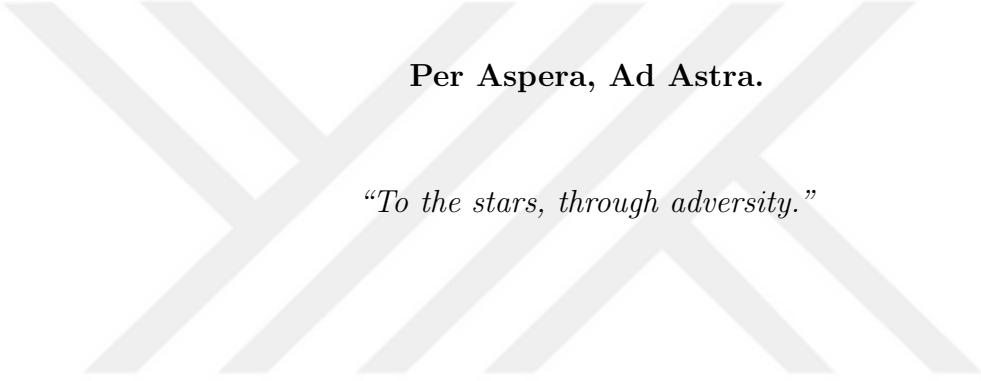
Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Ömer Morgül, for his patience, insightful guidance, valuable suggestions, and continuous support throughout my master's studies and research. In light of his guidance, I was able to complete the work presented in this thesis. This achievement would not have been possible without his mentorship and encouragement. I would also like to extend my thanks to the jury members, Asst. Prof. Aykut Koç and Assoc. Prof. İsmail Uyanık for their time and valuable comments on this thesis.

I also wish to express my sincere appreciation to ASELSAN for providing me with the opportunity and support to pursue my master's studies. I am thankful to my colleagues for their understanding and cooperation throughout this journey. I especially want to thank my manager, Ali Demirci, for his support and encouragement. I am particularly grateful to Dr. Caner Odabaş and Dr. Muharrem Arik, whose generous support, insightful discussions, and willingness to answer my endless questions guided me to clarify my ideas, overcome challenges, and shape the presented work in this thesis. I am also deeply thankful to Emre Aydın for always being there when I needed help. His constant support, kindness, and generosity, whether through sharing his time, offering encouragement, or even lending his computer when I needed it, truly made a difference in this journey.

I would also like to thank my long-time friends Atakan Yiğit and Erkan Bayram for their years of friendship, unwavering support, and thoughtful ideas. Our enduring bond and their constant encouragement have always been a source of strength and motivation for me, both during this thesis and beyond.

Last but not least, I would like to express my deepest gratitude to my family and relatives for their unconditional love, endless patience, and constant support. Their belief in me, even during the most challenging times, gave me the strength and determination to keep going.



Per Aspera, Ad Astra.

“To the stars, through adversity.”

Contents

1	Introduction	1
1.1	Problem Overview	1
1.2	Scope of the Thesis	2
1.3	Thesis Outline	5
2	Background and Related Works	9
2.1	Radar Fundamentals	9
2.1.1	FMCW Radar Principles	12
2.1.2	Micro-Doppler Signatures	14
2.2	Time-Frequency Representations	15
2.2.1	Short-Time Fourier Transform	17
2.2.2	Short-Time Fractional Fourier Transform	20
2.3	Deep Learning Architectures	22
2.3.1	Convolutional Neural Networks	24

2.3.2	Recurrent Neural Networks	27
2.4	Augmentation Methods	30
2.4.1	Mix-Up Augmentation	31
2.4.2	Random Stretching	32
2.5	Related Work	34
2.6	Motivation and Contribution of the Thesis	46
3	Datasets	48
3.1	Dataset I, SimHuman Dataset	49
3.2	Dataset II, University of Glasgow Dataset	51
3.3	Dataset III, University of Alabama Dataset	53
3.4	Dataset IV, Drone Classification Dataset	57
4	Model Architectures	63
4.1	Pre-processing Steps	63
4.2	Spectrogram Generation	65
4.3	Single-Branch Models	68
4.3.1	CNN Based Model	70
4.3.2	RNN Based Models	73
4.4	Multi-Branch Models	75

4.4.1	Fusion Models	76
4.4.2	Siamase Models	78
5	Experiments and Results	80
5.1	Experimental Setup	80
5.1.1	Training and Testing Procedure	81
5.1.2	Performance Metrics	83
5.1.3	Hardware and Software Setups	86
5.2	Single Branch Models	87
5.2.1	CNN-Based Model	87
5.2.2	LSTM-Based Model	96
5.2.3	GRU-Based Model	104
5.3	Multi-Branch Models	110
5.3.1	Fusion Based Models	111
5.3.2	Siamase Based Models	115
6	Discussion	118
6.1	Single-Branch Models	118
6.1.1	Performance Comparison	119
6.1.2	Behavior of the Fractional Orders	120

6.1.3	Verification of the Learned Orders	124
6.2	Multi-Branch Models	126
6.3	Configuration and Augmentation Effects	131
6.3.1	Analysis of Configuration Studies	131
6.3.2	Analysis of Augmentation Methods	135
6.4	Drone Target Classification	136
6.5	Additional Experiments with Well-Known Models	137
6.6	Comparison with Work in the Literature	139
6.7	Limitations and Considerations	142
7	Conclusion and Future Work	145
7.1	Conclusion	145
7.2	Future Work	148

List of Figures

2.1	Radar Data Cube	10
2.2	Micro-Doppler signature of a helicopter [1]	15
2.3	Micro-Doppler signature of a walking man [2]	16
2.4	Representation of 2-D convolutional layer	25
2.5	Representation of 2-D max-pooling layer	26
2.6	Representation of dense layer	26
2.7	Representation of an LSTM unit	27
2.8	Representation of a GRU unit	29
2.9	Unfolded LSTM vs Bi-LSTM Structures [3]	30
2.10	Mix-up Augmentation Example with $\lambda = 0.2$	32
2.11	Stretching Augmentation Example with Different Constant α and β Values as Range-Time Representation	33
2.12	Stretching Augmentation Example with Different Constant α and β Values as Micro-Doppler Signatures	34

2.13 Time-Frequency Distribution for Different Dynamics ((A)-Vibration; (B)-Rotation; (C)-Coning; (D)- Tumbling) [4]	36
2.14 Micro-Doppler Signatures of Wheeled Vehicle, Tracked Vehicle, and Walking Human) [5]	37
2.15 Feature Extraction from Micro-Doppler Signature [6]	37
2.16 Simplified Representation of the First DCNN Architecture in the Literature [7]	39
2.17 LSTM vs Bi-LSTM Experiments Results in [8]	40
2.18 Proposed Autoencoder Structure in [9]	41
2.19 Sea Clutter Removal with Frequency-Domain Filtering in [10]	42
2.20 Sea Clutter Removal with FrFT in [10]	43
2.21 Sea Spike Clutter Removal Algorithm in [11]	43
2.22 Feature Extraction Algorithm Proposed in [12]	44
2.23 Feature Extraction Example Results in [12]	45
3.1 Pipeline of Representation Micro-Doppler Signature Generation	49
3.2 SimHumalator Software Working Principle	50
3.3 Range-Time Representations of Dataset-I Classes	51
3.4 Micro-Doppler Representations of Dataset-I Classes	52
3.5 Radar Used in the Dataset-II	53
3.6 Range-Time Representations of Dataset-II Classes	54

3.7	Micro-Doppler Representations of Dataset-II Classes	55
3.8	Radars Used in the Dataset-III	55
3.9	Range-Time Representations of Dataset-III (10 GHz) Classes . . .	56
3.10	Micro-Doppler Representations of Dataset-III (10 GHz) Classes .	57
3.11	Range-Time Representations of Dataset-III (24 GHz) Classes . . .	58
3.12	Micro-Doppler Representations of Dataset-III (24 GHz) Classes .	59
3.13	Range-Time Representations of Dataset-III (77 GHz) Classes . . .	61
3.14	Micro-Doppler Representations of Dataset-III (77 GHz) Classes .	61
3.15	Drones Used in Dataset-IV	62
3.16	Range-Time Representations of Dataset-IV Classes	62
3.17	Micro-Doppler Representations of Dataset-IV) Classes	62
4.1	Frequency Response of the used MTI Filter and Impact of the Filtering	64
4.2	Trainable FrFT vs FFT Accuracy Comparison Table [13]	66
4.3	Effect of the Different Fractional Orders on Micro-Doppler Signatures	68
4.4	Differences between FrFT Order = 1.005 and FrFT Order = 0.995 Respect to FrFT Order = 1.000	69
4.5	Constructed CNN Architecture	72
4.6	Constructed RNN Architectures	76
4.7	GRU-Based Multi Layer Fusion Model	78

4.8	GRU-Based Siamese Model	79
5.1	Model Performance of CNN Architecture vs Initial Order	96
5.2	Model Performance of LSTM Architecture vs Initial Order	103
5.3	Model Performance of GRU Architecture vs Initial Order	110
6.1	Accuracy Comparison of Single-Branch Architectures	120
6.2	Learned FrFT Orders Among 50 Trials of Single-Branch GRU Model Experiment with Second Dataset	121
6.3	Final Mean Fractional Order Among All Experiments in Single- Branch CNN Architecture	122
6.4	Final Mean Fractional Order Among All Experiments in Single- Branch Bi-LSTM Architecture	122
6.5	Final Mean Fractional Order Among All Experiments in Single- Branch Bi-GRU Architecture	123
6.6	Initial FrFT Order Behavior in All Architectures	124
6.7	FT vs FrFT with Initial Different Fractional Orders (Only Testing)	125
6.8	FT vs FrFT Generalization Experiment	126
6.9	Trainable Fractional Order Behavior with Frozen Weights and Ini- tial Value 0.5	127
6.10	Trainable Fractional Order Behavior with Frozen Weights and Ini- tial Value 1.0	127

List of Tables

2.1	LSTM and Bi-LSTM Architectures in [8]	40
3.1	Radar Dataset Summary Table	60
4.1	Descriptions of Symbolic Parameters in the Single-Branch Models	69
4.2	Architecture of the Single-Branch CNN Model with 3 Blocks . . .	71
4.3	Architecture of the Single-Branch Bi-LSTM/Bi-GRU Model with 3 Blocks	75
5.1	Tuned Training Procedure Parameters	83
5.2	CNN Model FT vs. FrFT	88
5.3	CNN Configuration Study: Block Counts	90
5.4	CNN Configuration study: STFT Window and Overlap Sizes . . .	91
5.5	CNN Augmentation Method: Mixup Augmentation	92
5.6	CNN Augmentation Method: Stretching	93
5.7	CNN Model Initial FrFT Order	95

5.8	LSTM Model FT vs. FrFT	97
5.9	LSTM Configuration Study: Block Counts	98
5.10	LSTM Configuration study: STFT Window and Overlap Sizes	99
5.11	LSTM Augmentation Method: Mixup Augmentation	100
5.12	LSTM Augmentation Method: Stretching	101
5.13	LSTM Model Initial FrFT Order	102
5.14	GRU Model FT vs. FrFT	104
5.15	GRU Configuration Study: Block Counts	105
5.16	GRU Configuration study: STFT Window and Overlap Sizes	106
5.17	GRU Augmentation Method: Mixup Augmentation	107
5.18	GRU Augmentation Method: Stretching	108
5.19	GRU Model Initial FrFT Order	109
5.20	CNN Multi-Branch Fusion Model	112
5.21	LSTM Multi-Branch Fusion Model	113
5.22	GRU Multi-Branch Fusion Model	114
5.23	FT vs FRFT Comparison for Siamese Models	116
6.1	CNN Multi-Branch Fusion Model with 5 Blocks Architecture in Dataset 2	129
6.2	Multi-Branch GRU Model with Majority Voting	130

6.3	Siamese GRU Model without Shared Fractional Order	131
6.4	Random Stretching - Extension Only	136
6.5	Comparison of the Proposed Method and Conventional Approach with Well-Known Models	139



Chapter 1

Introduction

1.1 Problem Overview

The first radar systems were designed for military applications such as aircraft detection, missile guidance, and defense; however, with advancements in radar systems, the wide range of application opportunities in civilian areas became possible [14, 15]. The novel radar systems are tailored for various civilian applications, including environmental monitoring, weather forecasting, healthcare monitoring, human activity recognition, and similar fields. In the richness of the usage areas, those novel systems can detect, recognize, and classify the targets autonomously, since the modern industry requires autonomous systems and reliable technology [16–19].

Human activity recognition and radar classification may be an up-and-coming research area among these varying usage fields. Conventional methods focusing on human activity classification are mainly based on camera technologies and wearable sensors [20–24]. The radar-based approach possesses significant advantages over these traditional methods since radars can work robustly in low-lighting, adverse weather conditions, or even with obstructions of objects, while preserving the privacy of the monitored targets.

Human activity classification, by using radars, based on examining the micro-Doppler characteristics of the targets. Chen formally introduced the micro-Doppler concept in the 1990s in the literature, although some earlier studies indicate the underlying physical mechanism of the idea [2, 25–27]. This phenomenon may be explained as the additional frequency modulations caused by non-rigid motions of the targets, such as the helicopter’s rotating blade or the human’s swinging arms during motion [1].

Human activities include movements of the distinct body parts by preserving specific patterns, and micro-Doppler analysis helps to obtain subtle effects of these body parts to obtain particular characteristics, or signatures. Therefore, micro-Doppler signatures of specific patterns leverage the classification performances compared to conventional methods and techniques.

Although the human activity classification’s advantages depend on micro-Doppler signature analysis of the data obtained by radars, various challenges about the subject remain, such as complexities in feature representation, variety of the targets, and different types of radars. Thus, examination of suitable radar signal processing techniques, robust classification methods, and usability with distinct kinds of radars is still essential to indicate the potential of the radar-based human activity classification, which motivates the work presented in this thesis.

1.2 Scope of the Thesis

This thesis aims to provide a framework for radar-based human activity classification with micro-Doppler signatures by utilizing different deep learning architectures and advanced signal processing techniques, especially fractional Fourier transform (FrFT) with trainable orders [13]. Micro-Doppler signatures in the radar data are possible via time-frequency analysis of the signals obtained from the sensor, where the literature mainly focuses on the usage of spectrograms

obtained using the Short Time Fourier Transform [4, 28]. This conventional approach is enhanced in this work by using the Fractional Fourier Transform to obtain more distinguishable time-frequency representations, which brings the potential for an increase in the classification performance. In addition, despite the similar studies working based on FrFT [11, 12], the new methodology is introduced in this work by utilizing the learned transform orders from the distinct types of deep learning models in various scenarios.

Different types of neural network architectures have been examined with the usage of the trainable FrFT approach, and comparative analysis has been made for distinct architectures, including Convolutional Neural Networks (CNN), Long-Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). These well-known deep learning structures have been combined with the trainable FrFT methodology. Each model is evaluated separately with detailed configuration studies, including model complexity analysis with different layer sizes, short-time FrFT parameters with different window/overlap values, and the effect of initial FrFT orders on the model accuracy.

In addition to the conventional CNN/RNN models described in the paragraph above, this study also examines distinct types of multi-branch neural networks using the same architectures. The first multi-branch network type is named multi-alpha fusion structure, which aims to utilize the different fractional orders in the same architecture. This model type is examined with several branch configurations where the branches are constructed by CNN, LSTM, and GRU architectures, each layer having its own and distinct FrFT order. The second type of multi-branch network is a Siamese-based network, with CNN/LSTM/GRU cores, which utilizes the representation of the same data with different radars working in other frequency bands. Therefore, both modality-fusion and transform-level fusion strategies are investigated and reported comprehensively by considering performance and robustness.

Using the trainable FrFT concept in the micro-Doppler signature classification is a key component of this work. Unlike the previous work in the literature, including well-known time-frequency analysis techniques with fixed parameters, this

work aims to construct spectrogram-like images in neural networks by learning the optimal fractional order. In this way, the optimal time-frequency representation of the radar signal can be obtained for classification, which brings the model flexibility and adaptability in different environments, radars, and target types. The behavior of the FrFT order in various deep learning architectures, datasets, and training methods has been examined in this study in terms of convergence, stability, and a comparison with the traditional Fourier transform in different scenarios has been provided.

To enhance the reliability of the work in the thesis, experiments have been completed in various datasets, and a comprehensive analysis of the results among the datasets is also reported. These datasets were chosen by considering different perspectives, including simulated and real-world radar recordings, radars working in various frequency bands, and data collection environments, including controlled and uncontrolled settings. The proposed classification architectures are also evaluated with another separate dataset, including small drones, to widen the usage of the proposed method beyond human activity classification. In addition to assessing the work done with the literature and to assist further studies, all datasets are chosen as publicly accessible, which brings the opportunity to indicate the precise performance analysis of this work.

To increase the statistical quality of the work, all models are evaluated with 50 trials, including different initial conditions and seeds, whose details will be explained in the following chapters. This methodology aids us in ranking the models in terms of performance and stability by considering various metrics, including accuracy, precision, recall, and F1 score. Moreover, experiments are also supported by data augmentation methods, including mix-up augmentation and random stretching, to analyze the effect of the augmented data in different scenarios.

The human activity recognition task might be studied in a more detailed framework, starting with radar design for this specific application, including several

aspects such as radar antenna design/placement, waveform engineering, and circuitry [29–31]. Furthermore, the study can also be enhanced regarding the scenarios, including several targets in one scene, different environmental conditions, obstacle placement in the radar line of sight, and data fusion with other sensors such as cameras and wearable gadgets [32–34]. In the literature, various works focusing on these critical aspects might be found; however, this study isolates itself to investigate how different time-frequency transforms and other deep learning models can be used for classification with micro-Doppler signatures. Through this approach, the main aim is to contribute to the literature by providing a focused and detailed examination of the subject.

1.3 Thesis Outline

This thesis is structured with seven chapters, where the first chapter, with problem overview and scope of the thesis, is already provided.

The second chapter’s primary focus is to build a well-structured information background about the concepts examined in this study. Therefore, the background examination’s starting point is radar signal processing fundamentals with a specific focus on FMCW radars and micro-Doppler concepts, which aims to provide a baseline of the application steps described in the fourth chapter. These explanations are followed by an explanation of the time-frequency representations of the radar signals with different approaches. Since the micro-Doppler signatures are obtained using these techniques, a detailed explanation is necessary for complete understanding. After the signal processing steps overview, deep learning architectures used in the thesis are examined and described theoretically, which will be mentioned again in the discussion chapter. After this step, the augmentation methods used in the thesis are also described theoretically, with possible other augmentation methods and brief descriptions. Since stability and flexibility are required for deep learning models, augmentations are essential and deserve a theoretical explanation. In the last part of this chapter, a comprehensive literature survey is provided, considering chronological order similar to this thesis.

In the previous part of this chapter, after describing the necessary theoretical information and examining the related work, the contribution of this study is given.

The third chapter of the thesis is devoted to the datasets investigated in this study. Although a detailed description is provided in the chapter, brief definitions are given below. The first dataset is constructed with simulated radar data working at 24 GHz, and it includes 11 different classes of activities, which are the basis of the work provided in this thesis [35]. The second dataset is a well-known dataset collected by the University of Glasgow, which includes six different classes of motions with a 24 GHz radar [36]. It should also be noted that this dataset was used in a classification contest by Kaggle. The University of Alabama collects the third dataset with three different radars simultaneously working at 10 GHz, 24 GHz, and 77 GHz, which makes this dataset suitable for Siamese-based architectures [37,38]. The final dataset is provided as an expansion of the primary work proposed by this thesis to different fields [39]. The final dataset involves six different types of drones collected with FMCW radar using a mechanical sweep in the azimuth direction.

The fourth chapter presents the methods and details of the work done in this thesis. It starts with applying the signal processing steps, the theoretical framework examined in the second chapter, and range-time processing of the raw radar data for distinct datasets. Although the basic pipeline is the same for all classes described in the second chapter, it differs slightly for each class due to different data sizes. Also, this range-time processing step includes the basic pre-processing methodology, including the moving target indication (MTI) filtering with details. This step is followed by the spectrogram generation methodology proposed in this work, depending on the Fractional Fourier Transform, with several examples and examples of the different parameters' effect. After this part, the data augmentation methods examined in this work are provided with the usage methodology. In the last section of this part, the deep learning architectures covered in this work are provided with the multi-branch and single-branch networks depending on different core structures, including CNN, LSTM, and GRU structures, where the primary theoretical explanations were provided in the second chapter.

In the fifth chapter, the experiment setups and the results of the distinct experiments are given. The experimental setup is standard primarily for different experiments to increase the consistency of the results, which is examined with three subheadings, including training/testing methodology, performance metrics, and environments where the experiments are conducted. After these descriptions, the results of the experiments are provided by considering the different model structures and the model complexities. In this format, model types are provided. In each model part, results of different experiments are listed, including configuration studies, comparison with traditional Fourier transform and Fractional Fourier transform, influence of the Fractional Fourier Transform, and augmentation methods for each dataset. It should be noted that the single-branch models share the same experiments, whereas the Siamese model can only be used with the third dataset, including the data fusion of the three radars.

The number of examined model architectures and the datasets covered in this study causes an excessive set of experiments, which was reported in the fifth chapter. However, since various metrics should be examined in the scope of the study, a discussion chapter is provided as a separate chapter. The sixth chapter comprehensively analyzes the experiments by considering several aspects. The discussion chapter compares the single-branch architectures among the complete set of experiments instead of giving the model performances among the datasets. Similarly, two different multi-branch architectures are compared in detail by considering their performances among datasets, with systematic comparisons between single-branch and multi-branch architectures. An analysis of the configuration studies, augmentation methods, and overall experimental outcomes follows the systematic architecture evaluation. After we talk about the human activity classification performances, the supplementary research, including the drone target classification experiments, is discussed with systematic consideration of the human activity classification. Since the experimental results are shared in the previous chapter and the detailed discussions, including the datasets and methodology, are integrated in Chapter 6, the overall performances are compared with the baseline studies in the literature utilizing the same datasets to validate the model performances. Before the conclusion chapter, the model limitations and

considerations are provided to criticize the work done in the thesis.

In the last chapter of this study, conclusions and possible further work possibilities are provided. The literature review, which may be combined with this study's proposed fractional Fourier approach, is also in the future work section. Possible improvements and approaches that cannot be accomplished in this study are also shared for those willing to work on this topic.



Chapter 2

Background and Related Works

This chapter aims to provide a general overview of the subjects examined in this work. Fundamental concepts of the radar signal processing, especially for the FMCW radars, are supplied with the micro-Doppler concepts. This basic information is followed by the analysis of micro-Doppler signatures via time-frequency representations covered in this study. The theoretical descriptions of the deep learning architectures and data augmentation methods follow these signal processing-related explanations. In the completion of the theoretical framework of the study, related literature is provided by considering the chronological order with the academic influence. Finally, the motivation behind this thesis is given by considering the provided theoretical framework and the literature.

2.1 Radar Fundamentals

The core idea behind the working principle of the radar may be inferred from the name itself, since radar is an abbreviation of radio detection and ranging. Radar systems use electromagnetic waves to determine the targets' distance, direction, and velocity. The system transmits a known electromagnetic wave into space and collects the echoes of the transmitted wave. Since those echoes are reflected

from the targets, helpful information about the targets might be obtained from the differences between the transmitted and received signals, including time delays, frequency shifts, and amplitude changes [14, 40]. Therefore, the most basic signal chain might be summarized as waveform generation, signal transmission, reception of the transmitted signal, and sampling of the received signal, which is followed by signal processing. Those steps may also be considered as distinct engineering disciplines, such as RF design, antenna theory, and embedded systems; however, in this work, the primary focus will be on the processing of the received signals, or in other words, the digital signal processing of the sampled radar data.

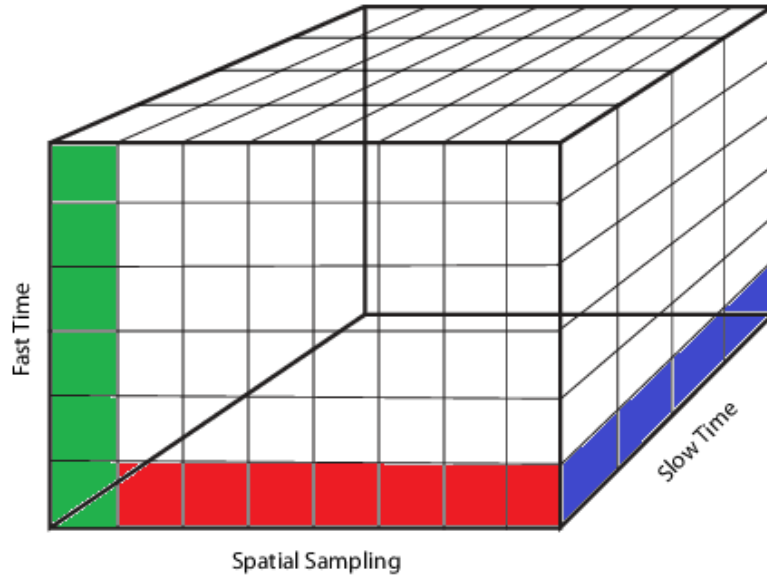


Figure 2.1: Radar Data Cube

In modern radar systems, including the FMCW radars examined in this thesis, whether the radar works with pulsed or continuous waveforms, the received signal is down-converted to a baseband signal with coherent mixers, where this process produces in-phase (I) and quadrature (Q) components that are in quadrature phase. This procedure is also known as IQ demodulation [41, 42]. After this procedure, the calculated IQ components form a complex representation of the received signal, which helps us analyze the received signal's magnitude and phase information. The equation including the original signal, $s(t)$, and the representation of the IQ components, $I(t)$ and $Q(t)$, is provided in the equation 2.1 where all signals share the same center frequency. The magnitude and phase information,

obtained from the IQ decomposition, are vital for the Doppler and micro-Doppler analysis. After the IQ demodulation procedure, the sampled radar data is used to construct a radar data cube [14], a conceptual representation of the radar signals for better space-time analysis, as shown in Figure 2.1 and will be explained below.

$$s(t) = I(t) + jQ(t) \quad (2.1)$$

The description of the axes in the radar data cube is given below in a list form [1, 14]. To provide an appropriate and understandable description of the concept, the dimensions of the radar data cube shown in Figure 2.1 are chosen as K-by-N-by-L with the order of fast-time, spatial, and slow-time directions that are denoted with colors green, red, and blue for better understanding.

- **Fast-Time Samples:** In the radar data matrix, K-by-1 sub-vector taken along the fast-time axis denotes the complex-valued samples acquired during the single pulse repetition interval (PRI), or chirp repetition interval (CRI) from the one antenna element. By utilizing the signal propagation speed, this dimension may be converted to distance; thus, fast-time samples are referred to as range bins.
- **Slow-Time Samples:** In the radar data matrix, if we consider the K-by-L sub-matrix, the vectors having dimension 1-by-L indicate that the same complex samples belong to the same range bins among different pulses for each antenna element. As the name implies, the sampling interval between two adjacent bins is PRI, or CRI. Processing the data along this dimension brings the Doppler spectrum; thus, fast-time samples are referred to as Doppler bins.
- **Spatial Samples:** In the radar applications, antenna arrays having multiple antenna elements are prevalent, and the data cube is constructed by considering each antenna element. If we consider the K-by-N sub-matrix, K denotes the number of fast-time samples in each pulse, or chirp, and N denotes the array elements that sample the same pulse.

After the data is shaped into the radar data cube, the radar signal processing steps may be conducted for different applications. As explained in the previous sections, the micro-Doppler analysis of the radar data is achieved after constructing the radar data cube. It should be noted that various important parameters and further explanation of the concept might be found in multiple studies; however, we only focus on the basic terminology frequently referred to in this thesis study. In addition, in the following parts, the radar data cube is referred to as a radar data matrix since the exploited radars have only one antenna element, degrading the cube's spatial dimension to a matrix.

2.1.1 FMCW Radar Principles

As shown in the previous part, the radar data can be placed into the radar data cube for both pulsed and continuous-wave radars. However, this study only focuses on the frequency modulated continuous wave (FMCW) radars since this type of radar collects the datasets. The FMCW radar is widely used in civil and military applications where short-medium range applications are needed, such as automotive sensing, human motion monitoring, and drone surveillance [16–19]. Unlike pulsed radars, the transmission and reception are continuous processes via chirp signals. A chirp signal may be described as a signal with a linear frequency variation over time [41, 43]. The mathematical representation of the transmitted chirp signal is given in the equation 2.2.

$$s_{tx}(t) = A \cos(2\pi(f_c t + \frac{K}{2} t^2)) \quad (2.2)$$

where f_c is the carrier frequency (Hz), K is the chirp rate (Hz/s), t is the chirp time (s), A is the amplitude. The reflected signal from a target with a known range, say R (in meters), is a time-delayed version of the transmitted signal received. Since the electromagnetic waves travel at the speed of light (c), time delay (in seconds) may be expressed as $\tau = \frac{2R}{c}$. The time-delayed version of the signal can, be written as given in the equation 2.3.

$$s_{rx}(t) = A' \cos(2\pi(f_c(t - \tau) + \frac{K}{2}(t - \tau)^2)) \quad (2.3)$$

In the receiver part of the radar, the aforementioned IQ demodulation technique is applied by mixing the received signal with the copied transmitted signal. After this mixing procedure, a low-frequency intermediate frequency (IF) signal is generated, which has the beat frequency $f_b = \frac{2KR}{c}$. It can be inferred that the beat frequency is proportional to the target range; thus, if we apply fast time Fourier transform (FFT) along the fast-time direction in the radar data matrix, range information about the target can be obtained [14, 40].

Similarly, velocity information about the target can also be obtained from the radar data matrix. If the FFT is applied in the slow-time direction, or to the consecutive chirps, velocity information is obtained thanks to the Doppler effect. Since the moving targets cause a phase shift between chirps, estimation of the radial velocity might be extracted. The Doppler frequency is given as f_D with a target having radial velocity v in the equation 2.4 [14, 40].

$$f_D = \frac{2vf_c}{c} \quad (2.4)$$

Therefore, with the help of the Fourier transform in both dimensions, range and velocity information is obtained and named conventionally as range-Doppler matrix, or map. Since the collected scene is expressed in this form, the joint analysis of the range velocity information about the targets is possible; however, explicit analysis of the scene by considering the range-time map is also plausible, as used in this study.

Compared to pulsed systems, FMCW radars promise various advantages, especially their suitability for cost and power-efficient applications. Many distinct implementations of this type of radar are spread worldwide, with compact embedded platforms like the ones used in the examined datasets. Also, with the deterministic and periodic nature of the chirp signals, the basic signal processing steps are relatively simple.

2.1.2 Micro-Doppler Signatures

The previous section briefly describes the Doppler concept with the proper mathematical background. If the f_D formulation is examined, it might be seen that the velocity and the frequency are inversely proportional. In other words, if the target moves toward the radar, the frequency increases and vice versa. Therefore, standard Doppler analysis might be considered a study of 'bulk translational motion', which refers to the overall movement of a target towards or away from the radar. Due to the assumption of bulk targets, additional movements causing additional frequency shifts created by other possible object parts are not included, which may be helpful for recognition and classification. The effect of these additional frequency shifts is called the micro-Doppler effect and might be used as micro-Doppler signatures since certain cases show certain behaviors, such as limbs and swinging arms of a person [2].

These additional frequency shifts, causing micro-Doppler effects, should be examined using time-frequency analysis methods. The example of the micro-Doppler effect of a helicopter and rotor blades is given in the Figure 2.2 and as shown in the figure, it might be inferred that the main Doppler effect in the spectrogram is distinguishable from the rotors due to the oscillatory motion of the rotors where the bold yellow horizontal line indicates the fuselage of the helicopter. The visualization is taken from the MATLAB official definition, which is a beneficial resource for examining the micro-Doppler subject [1].

Although it is not directly related to this section, the work of Chen should be highlighted here in order to increase the understanding of the concept with a historical remark. For the first time in the literature, Chen describes the micro-Doppler phenomenon as induced frequency modulations in the received radar signal about the center Doppler frequency of the target's body. In this work, vibration and rotation of the structures with an assumption of an ideal reflector are mathematically modeled, and micro-Doppler effects are expressed as a function of position in space. In addition, the micro-Doppler signature of a walking human is shared in this work as shown in Figure 2.3. In this work, like the rotors

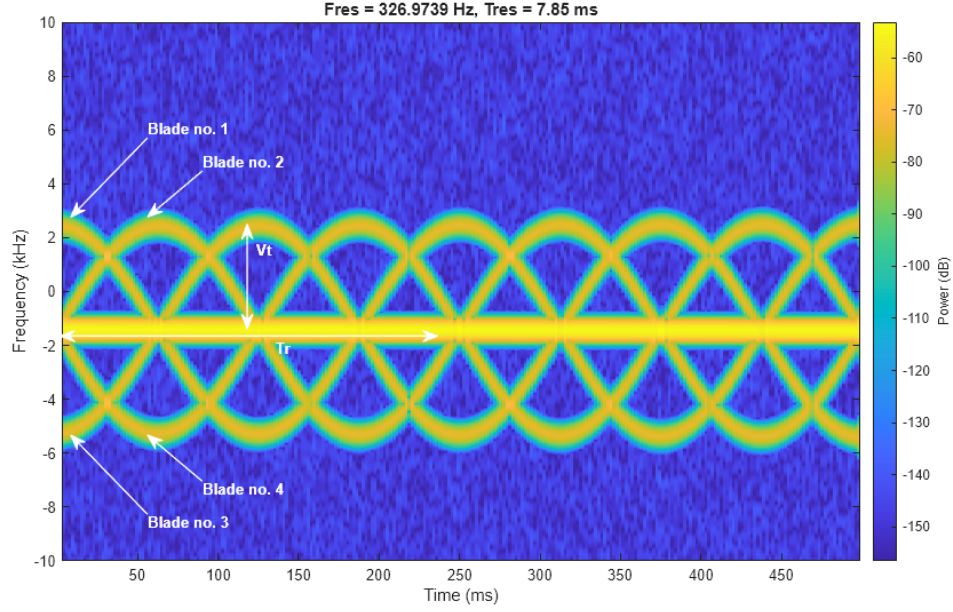


Figure 2.2: Micro-Doppler signature of a helicopter [1]

of a helicopter, he claimed that the swinging arms of a man might be observed with time-frequency analysis of the radar data. Mentioned work and the related work of Chen will be discussed more in the background section, with the various examples from the literature [2].

The brief description of the micro-Doppler concept is provided with the basic explanations about the radar signal processing methodology, which is assumed to be helpful to provide familiarity with the terminology used in this study. The discussion was advanced to the point where time-frequency analysis becomes vital for interpreting the micro-Doppler signatures. Therefore, the following section provides a detailed description of the time-frequency analysis approaches.

2.2 Time-Frequency Representations

In real-world radar applications, received signals have a non-stationary nature in most cases. In other words, the frequency content of the signals alters over

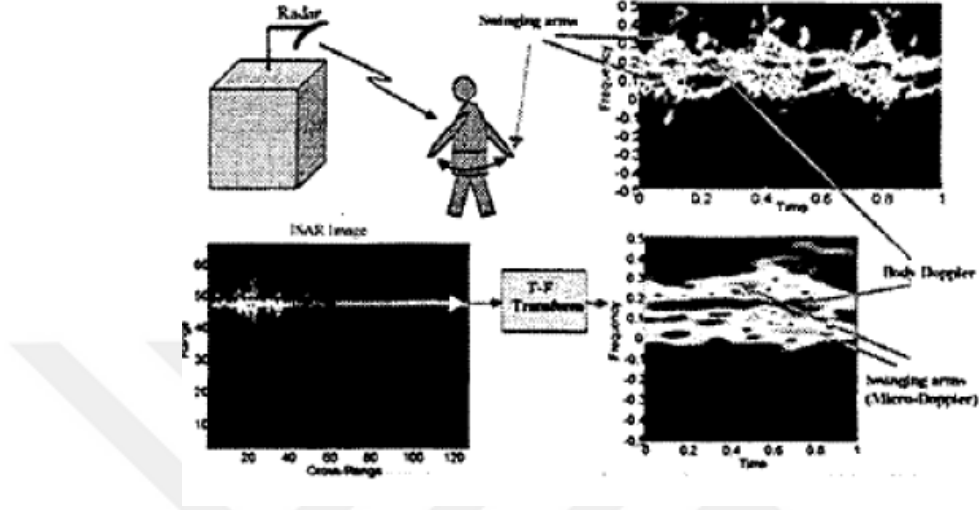


Figure 2.3: Micro-Doppler signature of a walking man [2]

time. Therefore, traditional analysis methods such as time analysis or frequency analysis cannot completely cover the observed signal's behavior. Due to this situation, time-frequency representations are essential [44, 45]. Especially for the micro-Doppler effects that are naturally time-dependent, methods allowing us to analyze the signals in both time and frequency simultaneously are highly needed. Since this thesis mainly aims to provide activity classification via micro-Doppler signatures, time-frequency representations (TFRs) become essential to obtain useful feature representations of the different motion types, having different micro-Doppler signatures inherently.

In the literature, various ways for time-frequency analysis may be found, and in the related work part, some of these are examined; however, the most commonly used one is the short-time frequency transform, STFT [41]. To stay within the scope of the thesis, only the Short-Time Fourier Transform is explained in detail as a traditional approach. This explanation is followed by examining the fractional Fourier transform and short-time fractional Fourier transform, as anticipated.

2.2.1 Short-Time Fourier Transform

A description of the Short-Time Fourier Transform cannot be provided entirely without a proper description of the Fourier Transform itself. Although there are several explanations, the Fourier Transform belongs to the integral transform family in mathematics. Input functions are transformed into output functions, which indicate the various frequencies in the input function. In other words, functions in the time domain can be expressed in the frequency domain with the Fourier Transform and vice versa. Without focusing on mathematical explanations too much, the proper definition of the transform with the complex-valued input $f(x)$ and the complex-valued output $\tilde{f}(\xi)$ might be seen in the equation 2.5 [46].

$$\tilde{f}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi\xi x} dx, \forall \xi \in \mathbb{R} \quad (2.5)$$

The frequency-domain function can be obtained if the Fourier Transform is evaluated for all values of ξ . However, since the application proposed in this thesis includes the discrete data sampled by the radar, the discrete version of the Fourier Transform should also be examined with the Fast Fourier Transform (FFT) algorithm [47]. The Discrete Fourier transform converts the equally-spaced samples of a function in the time domain into equally-spaced samples of the frequency domain. This transformation may be described by considering the complex-valued inputs denoted with $\{x_n\} := x_0, x_1, \dots, x_{N-1}$ into another sequence of complex numbers $\{X_k\} := X_0, X_1, \dots, X_{N-1}$ where the relation is given in the equation 2.6 [41].

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2j\pi \frac{k}{N}n} \quad (2.6)$$

Like the Fourier Transform, the Discrete Fourier Transform also has a considerably rich mathematical background; however, detailed mathematical derivations are avoided again to stay consistent with the topic. In the literature, several

computational algorithms, known as the Fast Fourier Transform (FFT), can be found for calculating the DFT. Still, we will focus on the algorithm proposed by the Cooley-Tukey method [48]. The Cooley-Tukey algorithm is a widely used FFT algorithm due to its efficiency. It is simply a recursive algorithm that recursively divides the original DFT into smaller DFTs, reducing the computational complexity from $O(n^2)$ to $O(n \log n)$. This computation methodology should be highlighted since the thesis's proposed fractional Fourier transform usage has no extra computational cost compared to the traditional method [43, 49].

Since the fundamentals of the Fourier transform are given, usage in the short-time Fourier transform can also be provided in the time-frequency analysis. The STFT is a transform to monitor the frequency and phase information in a signal over a time period. The procedure can be explained as dividing the original signal into shorter segments and applying the Fourier transform to each segment separately. Therefore, frequency, or Fourier, spectra are obtained for each segment, and concatenating these segments helps us to determine the total spectra of the signal over time [50]. The function in the continuous time case is provided in the equation 2.7.

$$STFT\{X(t)\}(\tau, \omega) \equiv X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-j\omega t}dt \quad (2.7)$$

where $w(\tau)$ is the window function, $x(t)$ is the signal which will be transformed, $X(\tau, \omega)$ is the Fourier transform of $x(t)w(t - \tau)$. In other words, $X(\tau, \omega)$ is a complex function including the phase and magnitude information of the input signal over time and frequency domains. As shown in the equation 2.7, it is obtained by multiplying the input signal with a window function; thus, another name for the operation in the literature is windowed Fourier transform [50, 51].

Like in the previous part, an explanation of the operation in the discrete domain is needed since the radar data is discrete. Mathematical description of the discrete-time STFT is provided in the equation 2.8. Like the continuous case, the discrete input data is segmented into sub-frames, the Fourier transform is applied to each frame separately, and the outputs are combined. In this scenario,

the input signal $x[n]$ is discrete, whereas the window function $w[n]$ is continuous; however, since the used methodology includes the usage of the FFT, the window function $w[n]$ is quantized [51].

$$STFT\{x[m]\}(m, \omega) \equiv X(m, \omega) = \sum_{n=0}^{N-1} x[n]w[n-m]e^{-j\omega n} \quad (2.8)$$

To remove the artifacts, frames in the input signal are arranged with overlaps, which are separately examined in this thesis as a configuration study. The window width selection is related to the resolution of the STFT representations with fixed resolution. In the time-frequency representations of the signals, wide windows cause good frequency resolution with poor time resolution. In contrast, the narrow windows provide good time resolution and poor frequency resolution, where this property is related to the uncertainty principle, whose mathematical explanations are not in the scope of this thesis [51–53]. In addition to the length of the window function and the overlap amount, the selection of the window function might be seen as an interesting topic to investigate, since the windowing operation can alter the signal representations. In the literature, commonly used windowing functions are Hann, Hamming, and Gaussian windows, where the Hann window is utilized in this study.

In this study, the micro-Doppler signatures of the samples are primarily given in the form of spectrograms, which might be defined as the magnitude squared, or power spectral density, of the STFT representation as provided in the equation 2.9 [51]. Also, the spectrogram word might be used in the following chapters as the representation of the micro-Doppler in fractional domains; thus, to avoid confusion, the provided equation should be considered as also a representation of the STFrFT operation, which is described in detail in the following section.

$$spectrogram\{x(t)\}(\tau, \omega) \equiv |X(m, \omega)|^2 \quad (2.9)$$

2.2.2 Short-Time Fractional Fourier Transform

Using the classical FT in the time-frequency analysis using STFT may be enhanced by utilizing the Fractional Fourier Transform idea as proposed in this thesis work. Since the key of the study is highly related to the FrFT concept, a proper definition is given in this section by using the same explanation step that was provided for the FT and STFT. Also, the calculation of the discrete-time FrFT is explained briefly. In the experiments conducted in this thesis, the transform is combined with the deep neural network architectures as proposed in [13]; however, the proposed approach is discussed in Chapter 4, Model Architectures. In this chapter, only the mathematical background of the operation is provided and discussed [43].

The Fractional Fourier Transform might be defined by considering several distinct aspects such as linear integral transform, fractional powers of the Fourier transform, rotation in the time-frequency plane, transformation of coordinate multiplication and differentiation operators, differential equation, and hyper-differential operator [43, 54]. Since the ordinary Fourier transform is given in the form of a linear integral transform in the previous part, we will focus on the same form in the fractional case as provided in the equation 2.10, and the other definitions will be discussed briefly by considering the relations to our application.

$$\begin{aligned}
 f_a(u) &\equiv \int_{-\infty}^{\infty} K_a(u, u') f(u') du', \\
 K_a(u, u') &\equiv A_a \exp \left[i\pi (\cot \alpha u^2 - 2 \csc \alpha uu' + \cot \alpha u'^2) \right], \\
 A_a &\equiv \sqrt{1 - \cot \alpha}, \quad \alpha \equiv \frac{a\pi}{2}
 \end{aligned} \tag{2.10}$$

where $f_a(u)$ denotes the a th order fractional Fourier transform when $a \neq 2j$, the $K_a(u, u') \equiv \delta(u - u')$ when $a = 4j$, and $K_a(u, u') \equiv \delta(u + u')$ when $a = 4j \pm 2$ where $j \in \mathbf{Z}$, and $a \in \mathbf{R}$ [43, 49]. Therefore, in the light of the provided definition, the fractional Fourier transform might be expressed as a generalized version of the ordinary Fourier transform with the order parameter denoted as a . Mathematically, a th order of fractional Fourier transform is equivalent to

a th power of the Fourier transform, which means the FrFT with order $a = 1$ is equivalent to the ordinary Fourier transform, and $a = 0$ is equivalent to the function itself.

In other words, the fractional Fourier transform helps to interpolate the function $f(u)$ and its Fourier transform $F(\mu)$, which means the signal is expressed in the fractional domains, related to the transform order, between the frequency-domain and time-domain. Thus, using the fractional Fourier transform operation, more suitable representations of the signals for different applications might be obtained in a continuum of infinitely many signal representations between the time and frequency domains.

Since the radar data is collected discretely, a discrete version of the fractional Fourier transform should also be examined briefly, as completed for the Fourier transform in the previous part. As expected, the discrete fractional Fourier transform is the generalized version of the discrete Fourier transform. Similar to the definition provided in the 2.6, discrete fractional Fourier transform F^a , with the order a , may be defined as given in the equation 2.11.

$$F^a[m, n] = \sum_{k=0}^{N-1} u_k[m] e^{-j \frac{a\pi}{2} k} u_k[n] \quad (2.11)$$

where $k \neq N - 1 + (N)_2$, $(N)_2 \equiv N \bmod 2$, and u_k denotes the k th order discrete Hermite-Gaussian vector. It should be noted that the output dimensions are related to the input dimensions, where the final value of k equals N if the N value is even and $N - 1$ if the N value is odd. As stated before, the trainable version of the operation will be discussed in the following chapters, and these expressions will include the ascriptions to this section.

In the same manner as the Fourier transform part, short-time fractional Fourier transform (STFrFT) can be derived accordingly. As described before, the STFT process helps us analyze the signal in both time and frequency domains by constructing spectrograms. Thanks to the fractional Fourier transform, signals may be analyzed in the fractional domains with time by constructing the spectrograms.

The mathematical definition of the process in the continuous case is provided in the 2.12, which is obtained by multiplying the signals with the corresponding window functions and taking the fractional Fourier transform of the results from the multiplication [49].

$$STFRFT\{X(t)\}(t, t) \equiv X_p(t, u) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)K_p(\tau, u)d\tau \quad (2.12)$$

where p denotes the fractional order and K_p denotes the corresponding transformation kernel. Since the idea is the same in both fractional and ordinary Fourier transforms, there is no need for further explanations, which were completed in the previous part by considering the window types and sizes. The discrete version of the STFrFT operation is also very similar to the ordinary Fourier transform, where the same logic is applied, which was provided in the equation 2.8.

The coverage of the basic mathematical concepts has been provided for both the Fourier and fractional Fourier transform, with the discrete and windowed versions. Although a brief explanation might be represented in a more detailed way, these are not provided to avoid straying from the essence. The deep learning architectures, another key factor in our study, are examined briefly in the following section.

2.3 Deep Learning Architectures

Machine learning may be defined as an academic field related to artificial intelligence, including statistical algorithms that can learn from data and perform without explicit instructions. Machine learning approaches might be used in many fields, including bioinformatics, transportation, physics, and the military [55, 56]. The main approaches of the discipline may be divided into three central paradigms as listed below. It should be noted that there is also a semi-supervised learning

method combining both supervised and unsupervised techniques, which is not provided separately.

- Supervised Learning (SL): The term describes a paradigm where the models are trained with a labeled dataset. In other words, each sample in the input set is associated with a corresponding ground truth output. The primary object is to build a mapping function $f : X \rightarrow Y$ that maps the input space X to the output space Y . In the training process, the main aim is to minimize the loss function, which describes the model's performance, such as mean squared error for regression and cross-entropy loss for classification [57].
- Unsupervised Learning (UL): The term describes the machine learning methods that learn the pattern with only input data, unlike the labeled outputs used in the supervised case. This approach is used in cases where the analysis of the dataset itself is required by uncovering the pattern, distribution, or structure of the dataset. It is used generally for applications such as clustering, anomaly detection, and dimension reduction [58].
- Reinforcement Learning (RL): The term describes the learning methods, including the agents interacting with the environment to learn through trials and receive feedback through rewards and penalties. Therefore, it differs from the other techniques since there is no need for the input-output pairs, and the learning mechanism depends on only the cumulative reward [59].

The models examined in this study are about classification tasks, a sub-branch of supervised learning algorithms. Various architectures might be found in the literature for the classification task, alongside the modern deep learning methods, including conventional methods such as k-NN, SVM, decision trees, etc. [6, 60]. In the related work section, various studies based on the traditional approaches are examined; however, since the models studied in this thesis focus on deep learning architectures, the theoretical framework of the deep learning approach is necessary and shared in this section.

Deep neural networks, or deep learning, have been widely used in the literature since their first appearance. The main inspiration for the application comes from neuroscience by stacking artificial neurons into layers [61]. As mentioned above, there are many usage areas of deep neural networks, such as machine learning; however, we will mainly focus on the classification task to preserve the focus on the work conducted in this thesis. In addition, some key concepts are explained in this section by considering their usage in the models in this work.

2.3.1 Convolutional Neural Networks

A convolutional neural network (CNN) is a specific version of the artificial neural network (ANN) family that learns the features from the data by kernel optimization. These networks have generally been used with image, speech, or audio data and related applications [62,63]. This type of network is constructed mainly using three layers: the convolutional layer, the pooling layer, and the fully connected layer, the explanations of which are provided in this section. The other layers utilized in the architectures supplied in this thesis are not separately examined in this section, such as the flattening layer and the dropout layer, which are discussed in detail in the model description part. The main aim is to explain the most basic concepts related to the subject.

As the name implies, the convolutional layer is the core of the CNN architectures. The data processing step is completed chiefly in the convolutional layers via the convolution operation; thus, a brief description of the operation is required [64]. Convolution might be described as an operation on two functions f and g producing the output $f * g$ as the integral of the two functions after one of them is reflected about the y-axis and shifted. Since we are dealing with the discrete data in the CNN structures, instead of providing the continuous version, the discrete version is represented in a mathematical way in the equation 2.13 for complex-valued functions f and g defined on the set of integers, $\mathbf{Z}$ [65]. It should be noted that the formulation includes the infinite summation; however, if the g has finite support, the summation boundaries might be altered accordingly.

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m] \quad (2.13)$$

The input data is processed with a convolutional operation to obtain certain features in the convolutional layers, and the output is produced. It might be visualized with the 2-D convolutional layers as provided in the Figure 2.4. It should be noted that the convolution procedure shown in the figure is conducted in the 2-D and the provided mathematical description should be enhanced to 2-D space as shown in the equation 2.14 where k_h and k_w parameters denote the height and width of the used kernel. The kernels are small matrices of weights whose values are learned in the network training session. The kernel height k_h and width k_w are hyperparameters that directly alter the network architectures. In other words, in the 2-D case, the kernels are sliding across the input data [64].

$$y[i, j] = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} x[i + m, j + n] \cdot w[m, n] \quad (2.14)$$

where x denotes the 2-D input, y denotes the output, w denotes the 2-D kernel where k_h and k_w denotes the kernel dimensions [65]. In a mathematical sense, the operation might be enhanced to further dimensions. In the convolutional layers examined in this study, to utilize the spatial features of the obtained radar data, 3-D convolutional layers are employed. Although they are not examined in a detailed way, variants of the operator might be found in the literature.

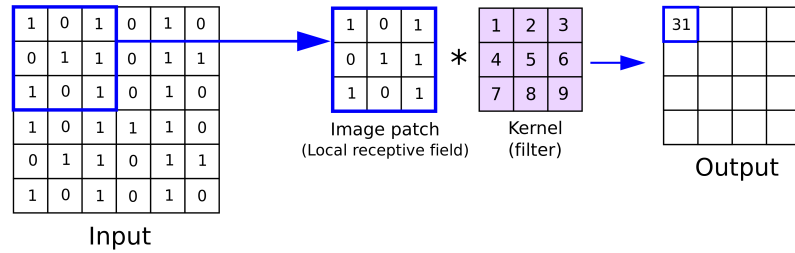


Figure 2.4: Representation of 2-D convolutional layer

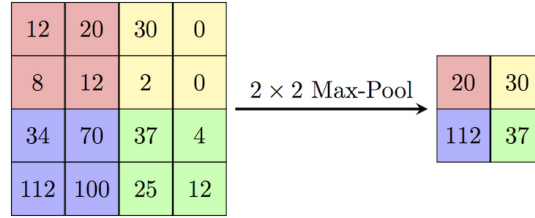


Figure 2.5: Representation of 2-D max-pooling layer

In most CNN architectures, the pooling layer occupies an important role. The idea behind the pooling procedure is down-sampling the original data to a lower dimension, which brings various advantages such as reducing the computational burden and memory usage, removing superfluous information, and increasing the model's robustness [66]. In the literature, multiple types of pooling techniques might be found; however, since only the technique utilized in this study is Max-pooling, only the description of this method is presented.

Max pooling, or MaxPool as used in the literature, is commonly used to reduce the spatial dimensions. Like the convolutional layers, the pooling layers include kernels determining window size, stride determining the window's step size, and padding describing the handling of the borders. The main idea is selecting the maximum value in the considered region, or patch, of the input as demonstrated in the Figure 2.5 [66].

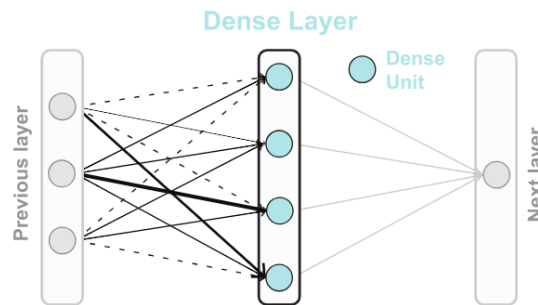


Figure 2.6: Representation of dense layer

The final core structure in the CNN architectures is the fully connected layers, or dense layer, which is demonstrated in Figure 2.6. In these layers, every neuron in the dense layer is connected to every neuron in the previous and subsequent

layers. Although the high computation time is due to dense layers, the combination of the features learned by the network is finalized by the dense layers, which are often placed at the end of the network [67].

2.3.2 Recurrent Neural Networks

Another ANN sub-branch used in this work is a recurrent neural network (RNN) with two specific architectures, including long short-term memory (LSTM) and gated recurrent unit (GRU) units, which are used in our study. The RNNs are devised for applications dealing with sequential data, such as speech, text, and time series. The processing order is essential in this data type and may change the results. Unlike the CNN architectures described above, which process the data once, RNNs use recurrent calculations. This recurrence is created by feeding a neuron with the previous neuron's output. Therefore, the model can acquire the patterns buried in the sequence [68].

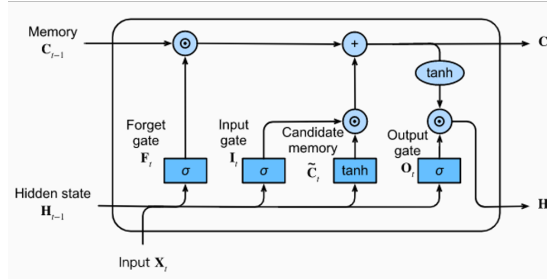


Figure 2.7: Representation of an LSTM unit

The first architecture deserving a separate explanation is the LSTM networks that aim to reduce the vanishing gradient issue, an inherent issue in the traditional RNNs [69]. The single LSTM unit typically consists of a cell with three gates named input, output, and forget gates, whose representation is provided in Figure 2.7. The cell stores the values, and the gating mechanism determines cell inputs and outputs. The mechanism behind the unit is listed below, with the cell update mechanism to provide a better understanding.

- Forget Gate: The main duty of the gate F_t is to determine which parts of

the previous state C_{t-1} should be forgotten. This gate takes the current input X_t and previous hidden state H_{t-1} as the gate inputs and processes those with the sigmoid σ activation function. Equation of the mechanism can be written as $F_t = \sigma(W_f \cdot [x_t, H_{t-1}] + b_f)$ where W_f denotes the weight matrix and b_f denotes the bias term. Since the sigmoid function is used, the output of the cell is located in the interval of $F_t \in [0, 1]$, where values closer to 1 mean keeping the information of the previous state.

- **Input Gate:** In the input gate, which parts of the new input will be added to the cell state is chosen, where the cell inputs are X_t , H_{t-1} , using the sigmoid function σ . Equation of the operation may be written as $I_t = \sigma(W_i \cdot [x_t, H_{t-1}] + b_i)$ and similar to forget gate, W_i and b_i denotes the weight matrix and bias term.
- **Cell Update Mechanism:** Cell state C_t is updated via using element-wise operations including previous state C_{t-1} , and candidate state $\tilde{C}_t$ as follows $C_t = F_t \cdot C_{t-1} + I_t \cdot \tilde{C}_t$. As shown in the 2.7, candidate state $\tilde{C}_t$ is determined with the inputs X_t , H_{t-1} with $\tanh$ function which might be written as $\tilde{C}_t = \tanh(W_c \cdot [x_t, H_{t-1}] + b_c)$ with the weight and bias terms as explained in the previous cells.
- **Output Gate and Cell Output:** Output gate O_t determines how much current cell state will be transferred to the cell output by considering the X_t and H_{t-1} via a sigmoid function, again. The calculation of O_t might be written as $O_t = \sigma(W_o \cdot [X_t, H_{t-1}] + b_o)$ with the related weight and bias values. The final output of the cell H_t is calculated with the element-wise multiplication, which may be written as $H_t = O_t \cdot \tanh(C_t)$ based on the updated cell state.

As mentioned before, other architecture is GRU, which might be seen as a simplified version of the explained LSTM architectures, whose structure might be seen in the 2.8 [70]. In this architecture, the gating mechanism exists; however, the number of gates is reduced to two, update and reset gates, which brings fewer parameters in the architecture compared to LSTM-based applications. The

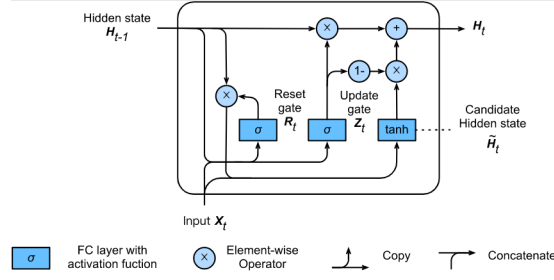


Figure 2.8: Representation of a GRU unit

LSTM model provides the gating mechanism with the output calculation as a cell update mechanism in a list format.

- **Update Gate:** This gate function determines how much the previous state will affect the future, which might be explained as a combination of input and forget gates in the LSTM structures. The calculation of the update gate Z_t with the current input X_t and H_{t-1} may be written as $Z_t = \sigma(W_z \cdot [X_t, H_{t-1}] + b_z)$ with weight and bias values similar to the LSTM application where the Z_t values closer to 1 indicates the keeping more past values.
- **Reset Gate:** To remove the effect of the previous state in the future decision, the reset gate R_t is utilized in GRU blocks. Similar to the other gating mechanisms, the mathematical derivation of the operation is $R_t = \sigma(W_r \cdot [X_t, H_{t-1}] + b_r)$.
- **Cell Update Mechanism:** To determine the candidate hidden state $\tilde{H}_t$, the previous state H_{t-1} is utilized with the reset modification. The proposed algorithm might be written as $\tilde{H}_t = \tanh(W_h \cdot [X_t, R_t \cdot H_{t-1}] + b_h)$ where $R_t \cdot H_{t-1}$ part is used to reset the past state effect and weight bias values are used as usual described in other gate mechanisms.
- **Cell Output:** After the application of the gating mechanisms described above, the final cell output H_t is determined by the equation $H_t = (1 - Z_t) \cdot H_{t-1} + Z_t \cdot \tilde{H}_t$. As one can infer from the equation, Z_t values near one indicate the domination by the new candidate and vice versa.

It should be emphasized that variations might be found in the literature regarding the units' cell architectures and working principles. The only variation discussed is the bidirectional usage of the LSTM and GRU units [71]. In these architectures, to utilize both past and future values in the time series input data, units are placed consecutively where the first one processes the input in the forward direction, whereas the other processes in the backward where the idea is shown in Figure 2.9 via LSTM blocks in unfolded formation. This approach is more beneficial in our study since the micro-Doppler spectrum is spread in both directions as a time series data.

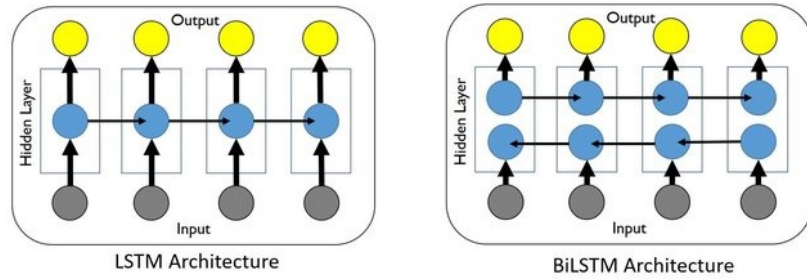


Figure 2.9: Unfolded LSTM vs Bi-LSTM Structures [3]

2.4 Augmentation Methods

The data augmentation means that the usage of the data, which is similar yet different from the data in the training set, in the training session [72], and the idea is formalized by using the Vicinal Risk Minimization (VRM) [73]. In this section, two distinct kinds of augmentation methods examined in this work are explained theoretically. Both methods are constructed by using one parameter. Parameter selection is also described in detail in this section to provide a complete understanding of the application of the methods. In addition, although it is not utilized in the thesis, adaptive methodologies, including parameter selection, are also briefly mentioned.

2.4.1 Mix-Up Augmentation

The Mix-up augmentation method is a simple approach for the data augmentation task that uses the weighted binary combinations of the image data in the training dataset. This process might be expressed mathematically in the 2.15, where $(\hat{x}_i, \hat{y}_i)$ denotes the new augmented data and the corresponding label, (x_i, x_j) denotes the two image data, (y_i, y_j) denotes the labels, and $\lambda \in [0, 1]$ denotes the mix-up constant chosen randomly for each new augmented data.

$$\begin{aligned}\hat{x} &= \lambda x_i + (1 - \lambda)x_j \\ \hat{y} &= \lambda y_i + (1 - \lambda)y_j\end{aligned}\tag{2.15}$$

Using this augmentation method, the original paper examined the well-known image classification datasets, including CIFAR-10, CIFAR-100, and ImageNet-2012, as well as other datasets, including speech signals and six different random classification problems [74]. The results indicate that almost all datasets have been utilized from the proposed augmentation methodology, although the contribution might be seen as minimal, such as %0.2, %0.5, %0.6 [74]. In addition, it should be noted that the λ constant is chosen from the beta distribution $Beta(\alpha, \alpha)$ and different α values have been examined in the study [75]; thus, a brief description of the function is given below.

In statistics, the beta distribution is a member of the probability distribution family which is defined in the interval of $(0, 1)$ or $[0, 1]$ with two positive parameters like $Beta(\alpha, \beta)$ [75]. Those parameters control the shape of the distribution. In the original work, the best improvements are obtained for the values $alpha \in [0.1, 0.4]$, and larger values lead to underfitting if there are no adversarial perturbations [74]. An example obtained from the first dataset utilized in this thesis might be seen in the Figure 2.10 with the fixed $\lambda = 0.2$. Since we are dealing with the inputs in range-time representation, it is provided with the corresponding micro-Doppler signatures.

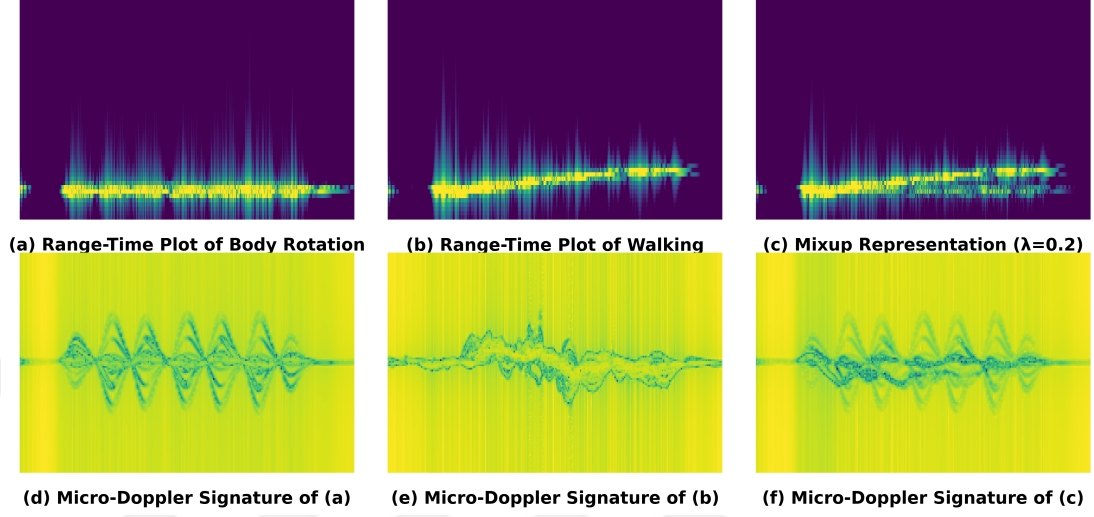


Figure 2.10: Mix-up Augmentation Example with $\lambda = 0.2$

2.4.2 Random Stretching

The second augmentation strategy examined in this thesis randomly stretches the radar data in range and time directions, warping the original input signal. Various studies in spectrogram analysis tasks might be found in the literature, including the same or similar augmentation approach in our research [76–78]. The warping method applied in our study might be explained as scaling in range and time dimensions by using linear interpolation on the original data to get distorted versions. This procedure might be described with a physical background since the scaling in the time axis corresponds to manipulating the target speed, whereas it represents the range hopping in the range direction [76]. In both directions, the mathematical operation is shown as provided in the equation 2.16.

$$x'_{r,t} = x_{r/\beta,t/\alpha}, \alpha \in [\alpha_{min}, \alpha_{max}], \text{ and } \beta \in [\beta_{min}, \beta_{max}] \quad (2.16)$$

Where $x \in \mathbb{C}^{R,T}$, and $\mathbb{C}^{R,T}$ denotes the complex-valued radar signal stored in the radar data matrix, α denotes the stretching factor in the time dimension, and β denotes the stretching factor in the range dimension. These stretching factors alter the original data as compression if the factor value is greater than one,

whereas expansion occurs if the factor is less than one. In our implementation, bilinear interpolation is utilized. It should be highlighted that the output of the operation differs from the original input sizes; therefore, data is cropped or zero-padded after the operation is applied.

As explained in the Mix-up augmentation part, the stretch factors are chosen randomly during training. In our application, the factor values are randomly selected from the normal distribution between the interval of $[0.8, 1.2]$. Also, as provided in the experiments section, separate analyses in both dimensions are provided with the simultaneous usage. The example representation of the application with stretching in time and range dimensions might be seen in the Figure 2.11 with the walking sample in the first dataset in the range-time format. The effect of the operation in terms of the micro-Doppler signatures is also provided in Figure 2.12.

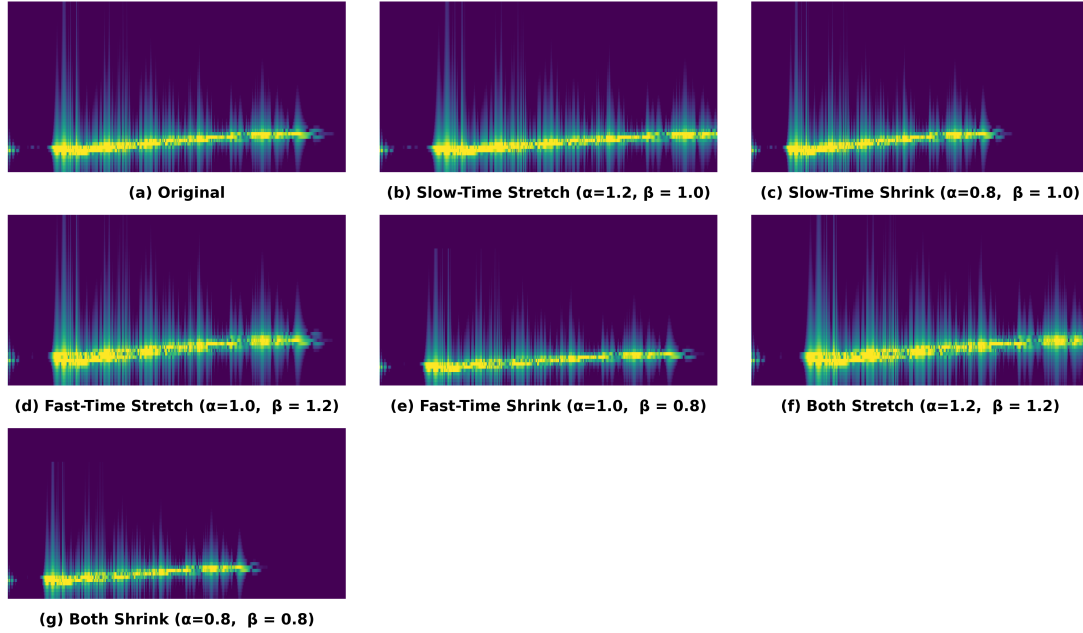


Figure 2.11: Stretching Augmentation Example with Different Constant α and β Values as Range-Time Representation

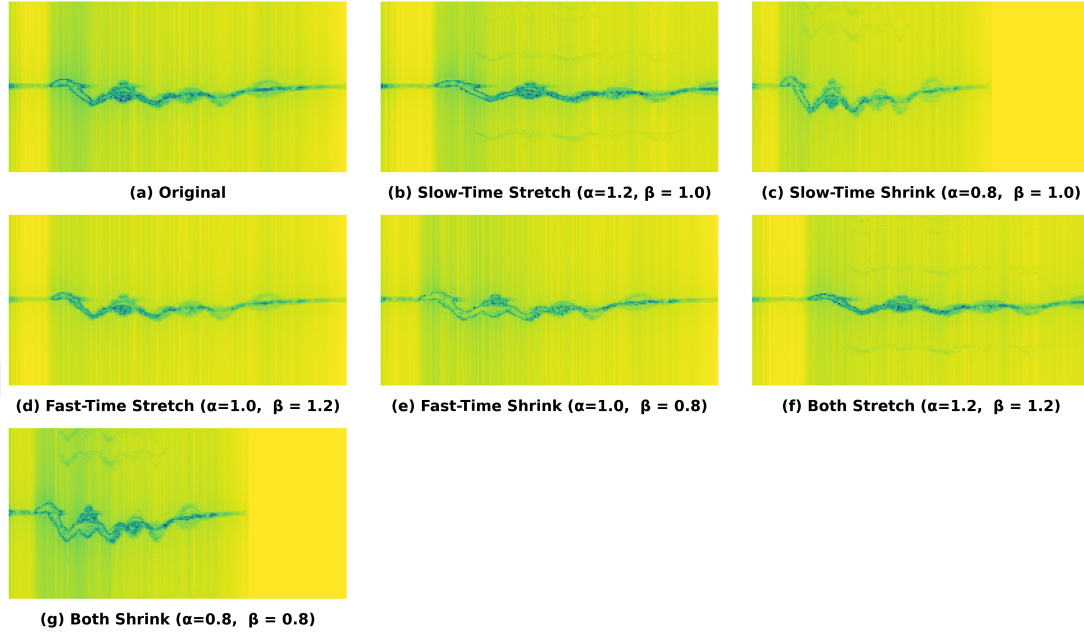


Figure 2.12: Stretching Augmentation Example with Different Constant α and β Values as Micro-Doppler Signatures

2.5 Related Work

As shown in the micro-Doppler explanation part, the micro-Doppler concept was introduced by Chen in the early 2000s with several studies [2, 79] in the radar context; however, as highlighted by Chen, the term was introduced in a patent about laser systems [25]. Also, related studies might be found in the literature before the explicit definition of the micro-Doppler effect. For instance, in the context of the laser systems, scattering in the vibrating systems has been examined in a detailed way by Kleinman & Mack in 1979 [26], and even spectrogram analysis of the vibrating objects has been examined [27]. In the radar context, the micro-Doppler related concepts have been discussed in various studies, especially for including propeller modulation (PM) and jet engine modulation (JEM) contexts [80–82]. PM might be classified as a sub-branch of the radar signal analysis by considering the targets having propellers, such as helicopters and drones, where the JEM introduces a similar analysis considering the rotating structures of the jet engines. Therefore, the analysis of the micro-Doppler has been a popular subject, even though the name of the phenomenon was not known.

In the earlier work of Chen, the concept is analyzed mathematically, and several scenarios were examined, including vibrating targets, rotating targets, tumbling targets, coning targets [79]. In addition, applications including synthetic aperture radar (SAR) imaging and human gait recognition with CW radar were examined using time-frequency analysis methods. In the following works of Chen, applications in several areas were also published by further investigation of the micro-Doppler signatures [83,84]. Even human recognition was examined in these works, which is also studied in this thesis.

After introducing the micro-Doppler phenomenon, classification tasks by utilizing the micro-Doppler signatures have started to be published in the following years, with increasing attention in the studies. In the first work, the classification task was composed by considering the micro-Doppler patterns of different movements simulated using Chen's formulation in a simulation environment. Lei simulated the formulations of the vibration, rotation, coning, and tumbling directly and classified these movements by considering their time-frequency distributions, which are shown in Figure 2.13 [4]. In this work, various conventional classifier architectures have been utilized, such as Bayesian linear and quadratic classifiers, support vector, and k-nearest neighbor classifiers. Also, in another work of Lei and Lu, the same classification problem is examined explicitly by using the Gabor filters for feature extraction with the same classifiers [85].

The very first classification studies by using the real-world data were conducted by Smith et.al., by using military radar data to classify tracked, wheeled armored vehicles and military personnel with the help of the Dynamic Time Warping (DTW) technique [5, 86]. Micro-Doppler signatures of the examined classes in this study might be seen in the Figure 2.13, where the effect of the tracked vehicle may be seen since the motion is spread in the frequency dimension compared to wheeled vehicles that are dominantly occupied by the main torso of the vehicle. Also, the movements of the vehicles differ from those of humans since human movement occurs at a constant speed. In the literature, various studies can also be found using the conventional methods for different classification tasks after this publication; however, we will mainly focus on the human activity classification to stay consistent with the scope.

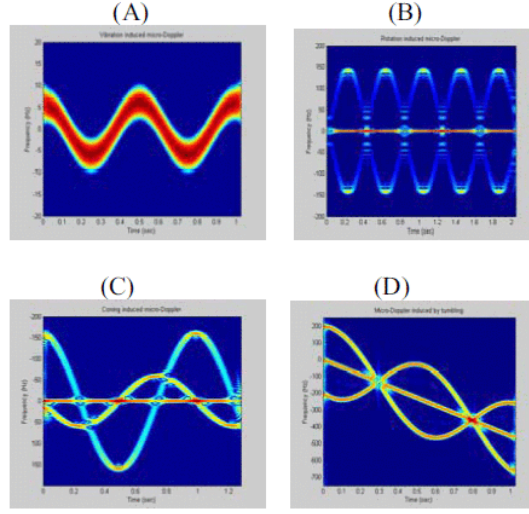


Figure 2.13: Time-Frequency Distribution for Different Dynamics ((A)-Vibration; (B)-Rotation; (C)-Coning; (D)- Tumbling) [4]

Kim and Ling first conduct the human activity classification task using the experimental data collected by these researchers [6, 28]. It should be noted that these and the following work from Youngwook Kim have a considerable influence on the academic literature. The first two studies utilize the same experimental dataset, including seven activities collected by a Doppler radar with 12 human subjects. Like the datasets examined in the scope of this thesis, seven activities were examined, including running, walking, walking with a stick, crawling, boxing while walking, boxing while standing, and sitting still. In the mentioned first work, the main idea is to classify the dataset using support vector machines using six distinct feature extraction techniques to micro-Doppler images of the collected samples [6]. The extracted features might be listed as torso Doppler frequency (1), total BW of the Doppler signal (2), the offset of the total Doppler (3), the BW without micro-Doppler effects (4), the STD of the Doppler strength (5), and period of the limb motion (6), which is shown in the Figure 2.15. It should be noted that the separation of the micro-Doppler signatures of different movements is also examined in this study. In the second work, instead of using SVMs, an artificial neural network (ANN) is proposed by utilizing the same six features instead of directly feeding the micro-Doppler images. In their work, SVM beats the proposed ANN structure; however, if we consider the time when the work is

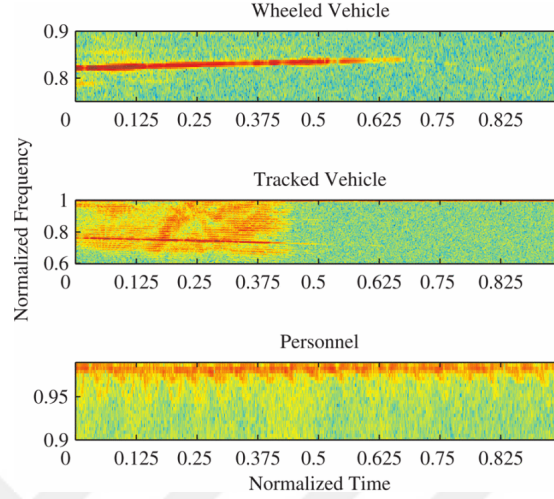


Figure 2.14: Micro-Doppler Signatures of Wheeled Vehicle, Tracked Vehicle, and Walking Human) [5]

published, it might be regarded as an expected result since the investigation of NN structures is very primitive. Several suggestions have been provided in both works, including different radar usage, different radar placement, examination of the window size, and data collection period, which are all explicitly examined in this thesis study.

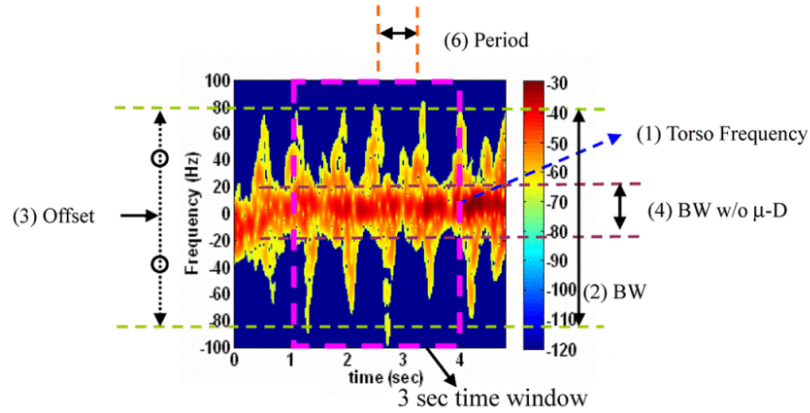


Figure 2.15: Feature Extraction from Micro-Doppler Signature [6]

As described in the introduction, privacy is one of the reasons for using the micro-Doppler analysis of human activities in health monitoring. Especially for elder monitoring, gait recognition is a very attractive subject. In similar dates, gait recognition has started to be examined as a sub-part of the human activity

classification task. For instance, the work of Tahmoush and Silvius examined gait recognition by considering the changes in the torso frequencies in the micro-Doppler images [87]. In their work, they have explicitly filtered the micro-Doppler signature of the torso movement and produced the gait characteristics. It should be noted that their proposed algorithm also includes target detection and tracking, which is not widely examined in the micro-Doppler studies. Two years after this publication, by following similar steps, gender classification based on gait analysis is proposed by Garreau et. al. [60]. It should be considered that these studies are followed by more sophisticated approaches, including detection and recognition of the suspicious activities by utilizing the micro-Doppler signatures of the motions [88–90]. Also, it should be noted that the need for applications in real-time processes also began with this paper, where real-time applications became a completely different examination topic.

Kim has also proposed using deep neural networks in the work "Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks," conducted with the Moon [7]. As explained in the previous paragraphs, Kim occupies a very influential place in the literature. This work is the first time utilizing deep learning approaches in micro-Doppler signature applications, which was published in 2015, similar to the trends related to the usage of deep learning approaches in the literature. Suitability of deep learning in human detection and classification tasks has been introduced with the accuracy values of %97.6 and %90.9. They constructed two different models and datasets for detection and classification tasks, where the detection dataset is a real-world dataset and the activity classification dataset is the same dataset used in the previous work as described in the earlier paragraphs [6]. In the classification task, in the first work, performance was %87.8 with the usage of ANNs with extracted features; therefore, this work paved the way for the novel studies depending on deep learning techniques in the literature. Their proposed deep convolutional neural network structure is shown in Figure 2.16 as a simplified version. In the original work, configuration studies about the architecture might be investigated for further reading.

Since the accomplished accuracy improvement introduced by the basic neural

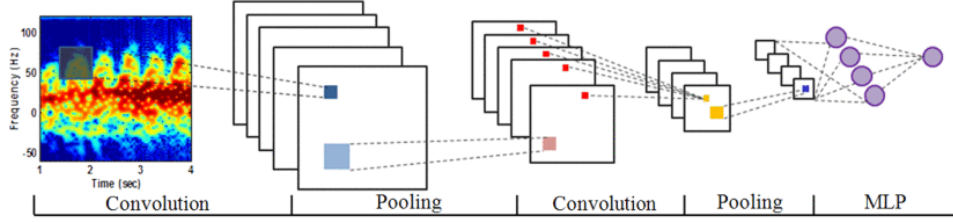


Figure 2.16: Simplified Representation of the First DCNN Architecture in the Literature [7]

network model, including convolutional layers, different neural network structures have been examined, including recurrent neural networks (RNNs). One of the first examples, including usage of RNNs, is the work conducted by Klarenbeek et.al. [91] by using long short-term memory (LSTM) blocks. In this work, an experimental dataset was collected using an FMCW radar for the gait classification task, including three different human subjects. They have compared the usage of the CNN and LSTM blocks with several ablation studies, which indicates the LSTM outperforms the CNN model, where similar studies are conducted in this thesis work. Unfortunately, the complete model structure was not shared in their paper. One of the inspiring studies for our work was undertaken by Shretsa et.al. in 2020 by following the LSTM idea by enhancing the architecture with bi-directional LSTM cores [8]. The idea of the bi-LSTM structures mainly depends on the simultaneous usage of past, present, and future data, which will be explained in detail in the previous sections. Since the radar data is related to time, this approach is significant, as we apply it in our models. Also, they have utilized the same dataset used in our work as a second dataset. Their study evaluated LSTM and bi-LSTM structures with 45 test sequences, whereas we conducted 50 test sequences in our work. Their network structure and results, including LSTM vs biLSTM structures, have been provided in Figure 2.17 and Table 2.1.

Although the idea was not covered in the scope of this work, since we have utilized a dataset provided by the same academicians in our work as dataset three, usage of the deep autoencoder network deserves consideration with the influence on the literature [9]. In the detailed work conducted by Dr. Gurbuz et.al., two different structures, including a CNN structure and a convolutional

Table 2.1: LSTM and Bi-LSTM Architectures in [8]

LSTM Network		
Layer	Size	Properties
Input	240	Based on the frequency bins of the input spectrogram
LSTM	2400	Number selected to store large sequences in memory
LSTM	2400	Number selected to store large sequences in memory
FC	6	Based on the number of possible output classes
Output	1	Single output
Bi-LSTM Network		
Layer	Size	Properties
Input	240	Based on the frequency bins of the input spectrogram
LSTM	2400	Number selected to store large sequences in memory
Bi-LSTM	2400	Number selected to store large sequences in memory
FC	6	Based on the number of possible output classes
Output	1	Single output

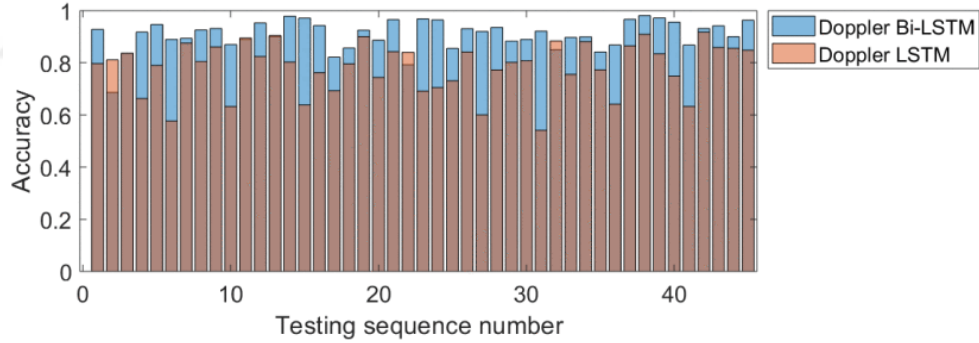


Figure 2.17: LSTM vs Bi-LSTM Experiments Results in [8]

auto-encoder (CAE) structure, have been constructed and analyzed with several ablation studies. The CAE helps to combine the positive effects of supervised CNNs and unsupervised pre-training of auto-encoders (AEs). AE may be defined as a feed-forward neural network aiming at reconstructing the inputs with separate training of the CNNs. After the pre-training, it may be stacked with the other NN structures as proposed in this work. Proposed CAE architectures are provided in the Figure 2.18 with the convolutional and deconvolutional layers. It should be noted that, after the pre-training step, the decoder part is removed, and two fully connected layers with a soft-max classifier are concatenated to the network.

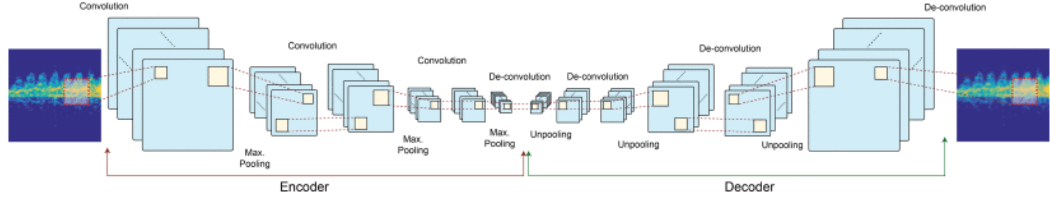


Figure 2.18: Proposed Autoencoder Structure in [9]

As described in the introduction, different techniques for the human activity classification task are covered in the literature by utilizing devices such as wearable sensors and camera-based gadgets. To provide a complete literature review, several studies depending on alternative approaches should be examined to indicate the extent of the research field [20, 21, 92]. As an example for wearable sensor-based human activity classification, the study conducted by Barshan and Koşar might be considered, which proposes a transfer learning approach for the task, which provides significant degradation in the training time without drastically reducing the accuracies [20]. This work might be linked to the proposed approach of this thesis, considering the spectrogram generation methodology and the multi-branch network architectures. One of the popular camera-based human activity classification studies was conducted in the study [93], including the RGB-D camera usage. This study is related to our first dataset, since the simulation environment utilized in the first dataset is constructed with a similar approach.

Until this point, influential academic results have been examined by considering the relationship between the scope of the thesis and the chronological order. In the following parts, as the key improvement of this thesis is the usage of FrFT, studies utilizing the micro-Doppler analysis, including the FrFT, are examined. As the following studies indicate, FrFT may also be used in the micro-Doppler concept successfully; however, since the FrFT with trainable orders is a novel idea, mostly the usage of the FrFT is related to feature extraction from the micro-Doppler radar data.

For instance, Suresh et.al. used the Fourier-Bessel transform and fractional Fourier transform for the micro-Doppler signatures of the rotating targets as a

feature extraction method by utilizing the idea of the turning time-frequency plane, which is calculated by the relation between the rotation angle and the FrFT order as described in the FrFT explanation section [10]. In this work, the sea-clutter in an experimental dataset is removed by utilizing the FrFT as shown in the Figure 2.20 where (a) denotes the STFT of the original signal, (b) and (c) the time-frequency representation of the sea-cluttered removed versions with FB-STFT and FB-WVD representations. In the sea-cluttered signal shown in (a), the appropriate order of the FrFT is used to clear it. The direct removal of the sea clutter with filtering in the frequency domain is also provided in the work as a conventional approach, where it may be seen that the time-frequency representation of the target is distorted due to direct filtering of the clutter. The traditional method examined in the study using frequency domain filtering is also provided in the Figure 2.19. Therefore, it may be claimed that using the FrFT is very beneficial in terms of the signals having a non-stationary nature and chirp behavior like sea clutter.

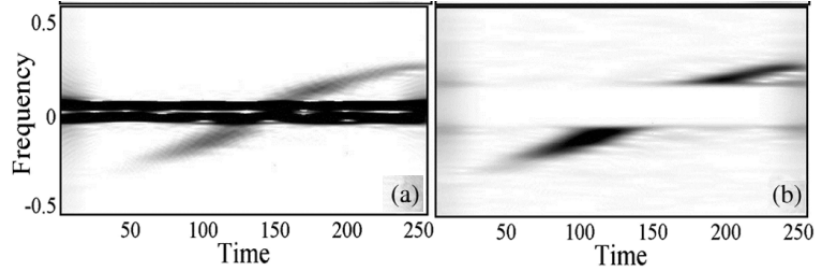


Figure 2.19: Sea Clutter Removal with Frequency-Domain Filtering in [10]

In a similar context to the previous work, Chen et.al. indicate the detection and extraction of motions having micro-motions in the sea-clutter [11]. Unlike the examined work previously, spectrogram data by utilizing STFrFT is reviewed; therefore, the representation of the signals in the fractional domains is constructed, which is similar to the method proposed in this work. The main flowchart of the proposed work is provided in Figure 2.21. Also in this work, although it is not related to our scope, different polarizations of the used antenna are examined in a research topic that might benefit future applications. The FrFT order selection is based on the mathematical time-frequency analysis of the collected data, with a detailed formulation provided in the work. In addition,

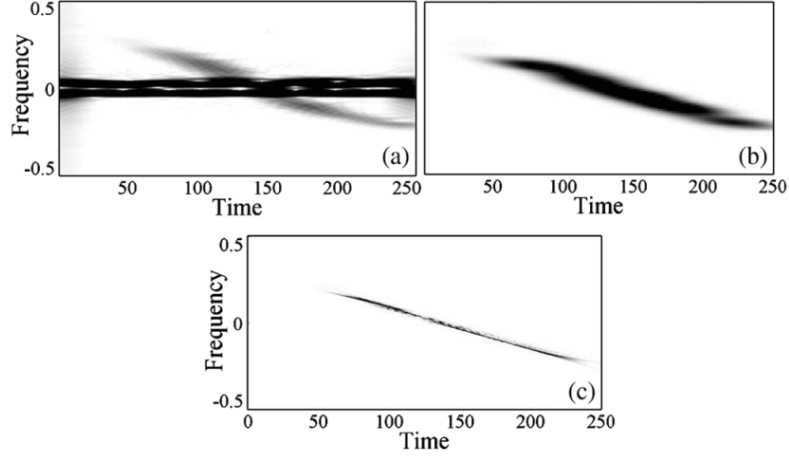


Figure 2.20: Sea Clutter Removal with FrFT in [10]

optimal lengths for the window and overlap parameters are also examined, and a closed-form methodology is proposed. Since the original work is too detailed to describe in this section, it is left to the reader to understand the proposed approach, and only a brief description is provided above.

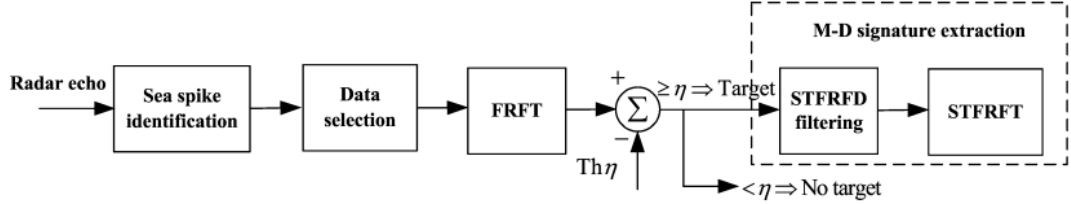


Figure 2.21: Sea Spike Clutter Removal Algorithm in [11]

The usage of the FrFT in the micro-Doppler analysis of human activities is also examined in the literature by Qiao et.al. in 2019 [12]. In this work, the authors used a simulation dataset including a walking human with corrupted simulation data at various noise levels, supported by an experimental dataset. Figure 2.22 provides the proposed algorithm workflow. In their work, the primary focus was the extraction of the torso and limb motions existing as micro-Doppler effects in the time-frequency representation of the signal. The algorithm steps can be summarized as applying STFrFT, filtering to highlight the torso movements in the signal, and applying ISTFrFT to obtain the torso motion extracted signal. The CLEAN algorithm is used to remove the other parts with the inputs of the

original signal and the torso-highlighted signal. In the final part, STFrFT-based morphological component analysis (MCA) is applied to obtain movements of the limbs in the signal. As in the previous work, a more detailed explanation may also be found in the cited paper. It should be noted that the FrFT order selection is based on searching for the optimal value by considering the matched filtering idea with the grid-search methodology, and found as 0.7 and 0.1 for simulation datasets and 0.9 and 0.1 for experimental datasets, respectively, for torso and limbs. Also, they examined the relationship between the window length and the overlap lengths with the different orders of FrFT parameters. Their results are provided in the Figure 2.23; however, they have not provided any classification task, which is left as a future work, as said in their original work.

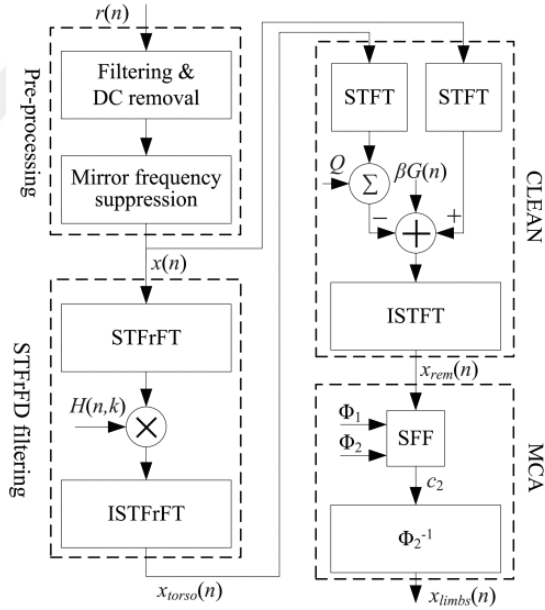


Figure 2.22: Feature Extraction Algorithm Proposed in [12]

In addition to the studies explained in this section, several studies about the subject might be found in the literature. The detailed examination of the various studies might be found in the form of reviews, benchmarking, and analyzing papers [94–97]. In this section, these studies are not explicitly introduced; however, to better understand the subject in the literature, examination of these is beneficial. Especially Table III in the [94] is a very informative summary of the micro-Doppler applications in the literature.

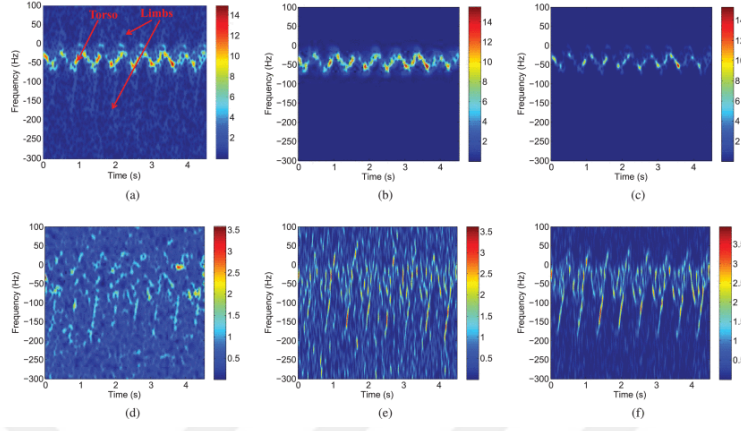


Figure 2.23: Feature Extraction Example Results in [12]

Although the detailed examination of the trainable fractional Fourier transform will be provided in the fourth chapter with related work [13], to complete the literature examination, the confluence of the fractional Fourier transform and deep learning, which constitutes the backbone of this thesis, should be mentioned by considering recent studies. In this scope, two example studies of the proposed framework are examined in this part, including using the fractional Fourier transform in transformer encoders and time-series prediction.

In [98], authors utilize the trainable fractional Fourier transform layers to replace the Fourier transform layer in the well-known FNet architecture. In the FNet architecture, the main idea is to replace the self-attention sublayers with a standard, unparameterized Fourier transform, bringing considerable accuracy improvements with lower training costs [99]. The novel FrFNet approach includes extending the FNet architecture by replacing the FT layer with FrFT layers, and the results indicate that better performance might be obtained via fractional layers [98]. Instead of using trainable layers, the fractional order is determined heuristically in the mentioned work. The FrFT's advantage over FT may be seen in the mentioned work even without a trainable fractional Fourier transform, which is also shown in this thesis for a different application.

The last study should be mentioned to complete the comprehensive literature

analysis, which depends on the trainable fractional Fourier transform. The proposed approach in this thesis is very similar to the approach provided in [100], where the main aim is to use a trainable fractional Fourier transform in the time-series prediction task. The processing of the time-series data is based on short-time fractional Fourier transform, which is utilized for time-series prediction, whereas, in this thesis, the same methodology is implemented on radar data to obtain micro-Doppler signatures and a classification task. They showed that employing the trainable fractional Fourier transform beats the conventional Fourier transform and directly uses the time data with multiple datasets [100].

2.6 Motivation and Contribution of the Thesis

After explaining the theoretical background, followed by the application in this thesis and an informative analysis of the literature, the main contribution of this work may be illustrated in a list format.

- For the time-frequency analysis of the radar signals, the FrFT-based representation is introduced by using the trainable fractional Fourier transform layers. To the best of our knowledge, this is the first application of trainable FrFT layers for radar-based classification in the literature. Experimental results consistently show that FrFT outperforms the conventional Fourier transform-based approach in all test cases.
- Since the FrFT is a generalization of the Fourier transform, different fractional orders correspond to alternative time-frequency representations—some closely related to the Wigner-Ville distribution and the ambiguity function. This study explores how varying the FrFT order impacts the learned spectral features and classification performance. Furthermore, this work is the first to introduce the simultaneous use of multiple representations derived from different FrFT orders within a unified learning framework.

- Various deep learning architectures are constructed to evaluate the suitability of the proposed method. The constructed architectures include single and multi-branch structures having CNN, LSTM, and GRU-based deep learning units. The well-known deep learning architectures are also employed to indicate the contribution of the proposed approach.
- To increase the support of the study, two different augmentation strategies are examined, such as mixup augmentation and random stretching, with the configuration studies that investigate the model complexities by altering block counts and different window-overlap sizes in the spectrogram generation by using grid-search.
- To evaluate the general applicability of the proposed approach, multiple radar datasets are acquired, including synthetic and real-world scenarios. The selected datasets involve radar sensors that are operating at different frequencies, bandwidths, and resolutions, which brings an opportunity for covering a comprehensive assessment through diverse hardware configurations and sensing conditions.
- The scope of the study is extended by applying the proposed method to a drone classification task using a radar sensor with a different working principle compared to other datasets. This investigation highlights the versatility of the approach in handling both diverse target types and varying radar designs.

Chapter 3

Datasets

In this chapter, the datasets that we operate are explained in a detailed way to provide a trustworthy representation of the richness in the study. There are four distinct types of datasets examined in this study, three of which are human activity datasets, and the last one is the drone classification dataset, as mentioned several times in previous chapters. To increase the readability of the text, in each step, the datasets are entitled with distinct names different from the original names and suitable numeration, which might be seen in the sections of this chapter. As the models used in this study take the range-time representations of the samples, the range-time representations are shared with the micro-Doppler representations. The following pipeline of the pre-processing steps is indicated in the Figure 3.1 for representations in this chapter; however, it should be noted that the shared representations, due to numerical differences in the datasets, coloring might differ for each dataset. At the end of the section, Table 3.1 is provided as a summary of the datasets explained in this chapter.

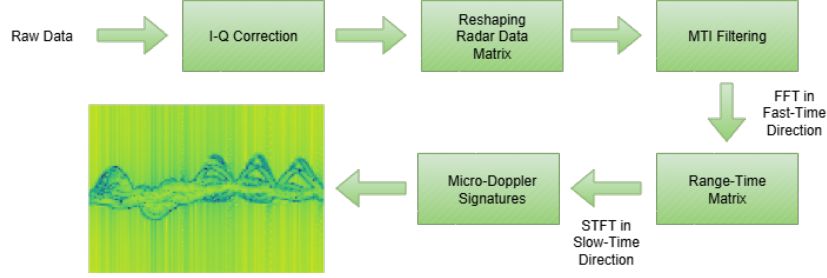


Figure 3.1: Pipeline of Representation Micro-Doppler Signature Generation

3.1 Dataset I, SimHuman Dataset

The first dataset examined in this study is "FMCW Radar Dataset," which was prepared by Binh Nguyen in 2024, and is open to public usage [35]. Also, the provided dataset is utilized in the work in [101] published by the same scholar and his research group. The original work examined the classification problem by using this dataset with deep convolutional neural networks and cross-residual convolutional neural network architectures by adding additional noise to the samples. Although the proposed architecture in their work depends on the conventional approaches using the pre-calculated spectrogram images, in addition to the spectrogram images, the raw data is shared explicitly, which makes this dataset suitable for the methodology claimed in this thesis.

As indicated in the given name of the dataset, the dataset was prepared using a simulation environment created by the SimHumalator software, which was published in 2022 [102]. This software provides an environment for end-to-end human micro-Doppler data generation simulations using FMCW radars. The main idea of the simulation methodology depends on combining two main techniques, including modeling the primary scattering points of the human body and human motion capturing with cameras and related sensors. The captured human motions are structured as a point cloud in space, and the mathematically derived scattering characteristics are applied to the obtained point cloud, as shown in Figure 3.2. The software allows users to place radars, whose basic configuration might be altered as desired, in desired locations to observe 11 distinct human motions.

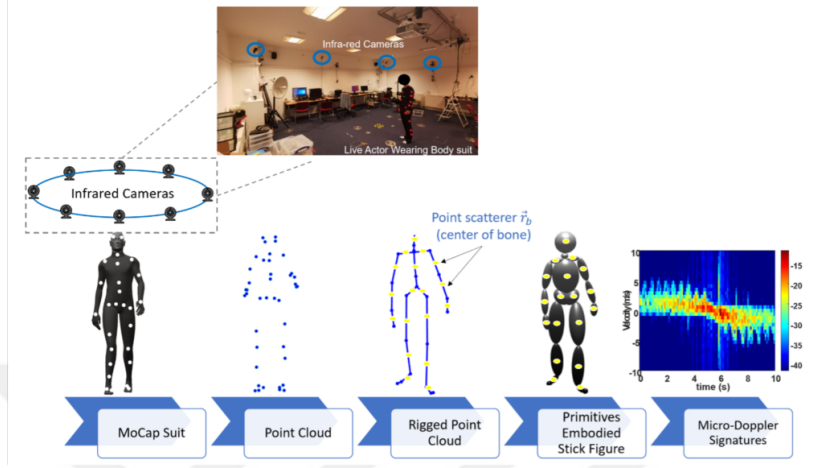


Figure 3.2: SimHumalator Software Working Principle

The examined dataset includes the raw data obtained from the simulation of an FMCW radar with 11 distinct daily human activities, as the maximum number permitted by the software. These activities are walking, body rotation, punching, kicking, grabbing, standing up, sitting down, standing to walk, walking to sit, walking to fall, and falling to walk. All those activities are performed separately and repeated 60 times with various aspect angles, including $0^\circ, -45^\circ, 45^\circ, -90^\circ, 90^\circ$; however, in the shared dataset, only one aspect angle is shared and used in this thesis. It should also be noted that the radar is placed 1m above the ground, and the distance of the targets is fixed at 3m with respect to the radar, and the record duration varies from 8 to 15 seconds.

To provide a complete micro-Doppler analysis, the radar parameters should also be highlighted. As mentioned, data is captured with an FMCW radar whose center frequency is 24 GHz. The chirp duration of the radar is 1 ms, and the chirp bandwidth is 400 MHz, where each chirp is sampled with 128 samples. As mentioned before, execution times vary from 8 to 15 seconds. To create a fixed-length dataset, each execution is clipped to 8 seconds. Therefore, the radar data matrix is constructed with 128 fast-time samples and 8000 slow-time samples.

The range-time plots and micro-Doppler signatures of the different classes are provided in Figure 3.3 and Figure 3.4. It should be emphasized again that the provided spectrograms are obtained via the aforementioned standard

methodology in the Figure 3.1, and range-time plots are provided separately since the architectures proposed in this work accept the inputs shaped as range-time representations.

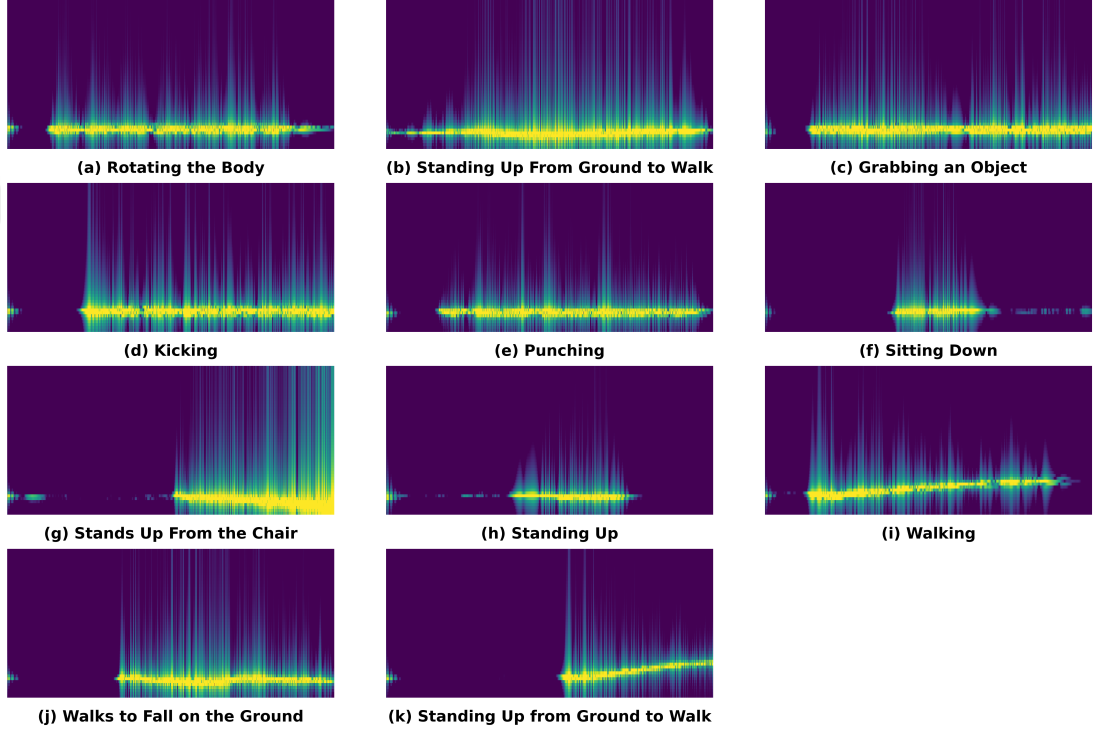


Figure 3.3: Range-Time Representations of Dataset-I Classes

3.2 Dataset II, University of Glasgow Dataset

The second dataset utilized in this study was collected for a research project named "Intelligent RF Sensing for Falls and Health Prediction" conducted by the research group of Dr. Fioranelli in the University of Glasgow [36]. In the scope of their project, the ultimate goal was to construct a system and algorithms that can observe people's activity levels and patterns accurately to decide whether the monitored subject is healthy or not. As we proposed similar motivations in the introduction, their basic motivation relies on the reliability of the radars, contactless measurement ability, and the promise of more privacy compared to conventional methods like cameras.

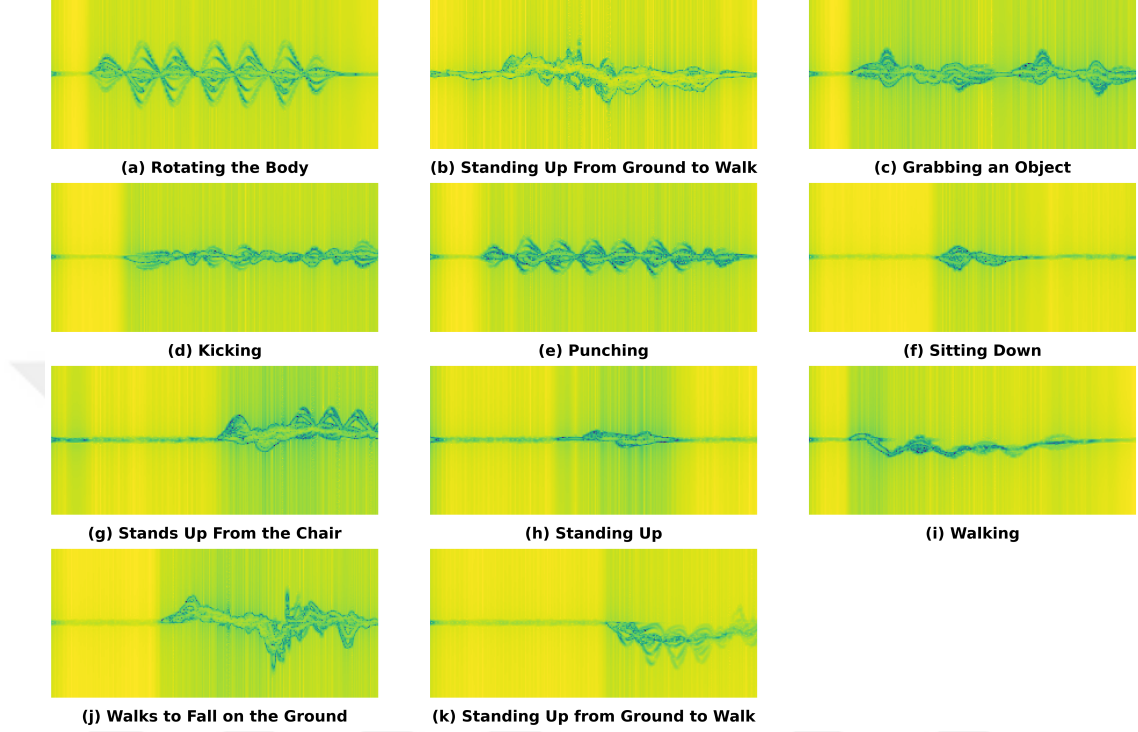


Figure 3.4: Micro-Doppler Representations of Dataset-I Classes

The data was collected on various dates, from 2017 to 2019, and at locations, including controlled and uncontrolled environmental conditions, with different participants whose age, height, gender, and dominant hand usage differ. Six different classes exist in the dataset, which are walking towards and walking away from the radar, sitting down in a chair, standing up, bending to pick an object, falling, and drinking from a cup. Although another type of activity exists, they are not considered in this thesis to avoid class imbalances.

This study uses an FMCW radar operating at a 5.8 GHz center frequency, produced by Ancortek. The frequency bandwidth of the radar is 400 MHz, and the chirp duration is 1 ms, which is the same as in the first dataset. 128 samples are obtained in each chirp, the same as in the first dataset. The duration of the execution varies between 5 and 10 seconds. As done in the first dataset, each execution is clipped to 5 seconds to create a fixed-length dataset, resulting in 128 fast-time samples with 5000 slow-time samples. The appearance of the used radar might be seen in the Figure 3.5.



Figure 3.5: Radar Used in the Dataset-II

The radar range-time matrices and example micro-Doppler signatures are given in Figure 3.6 and Figure 3.7 as provided for the first dataset.

3.3 Dataset III, University of Alabama Dataset

The last dataset examined in this study, including the human activities, was collected by Dr. Sevgi Z. Gurbuz and her research group at the University of Alabama in 2020 [37]. This dataset differs from the previous one since it includes three distinct radars that simultaneously collect the same activity. Therefore, as we proposed in this thesis, it might be considered as three different datasets and a single dataset suitable for multi-input approaches like siamese structures. The radars used in the data collection step are XeThru UWB impulse radar working in 7-10 GHz, a 24 GHz centered FMCW radar by Ancortek with 1500 MHz frequency bandwidth, and a 77 GHz centered FMCW radar by Texas Instruments named IWR1443 with 750 MHz frequency bandwidth. All these radars are placed side-by-side on top of a table whose height is approximately 1 meter from the ground, and the targets conduct their activities in the range of 0.5 to 3 meters with respect

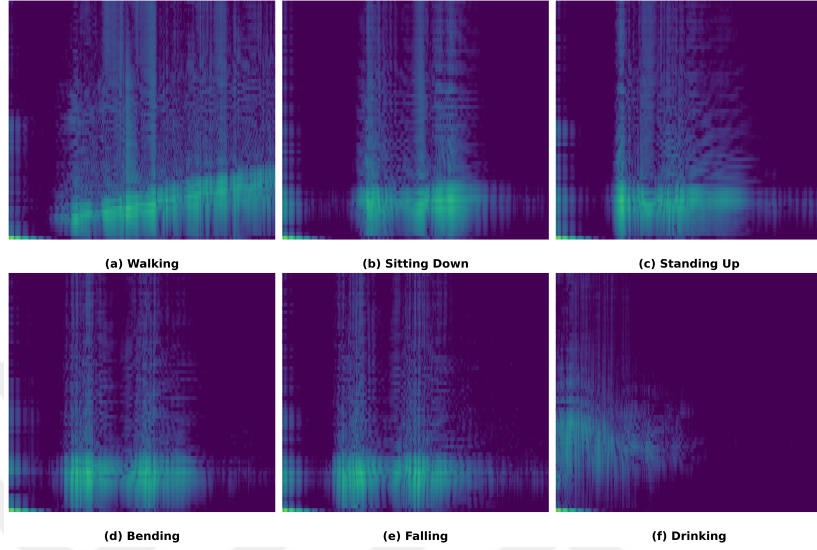


Figure 3.6: Range-Time Representations of Dataset-II Classes

to the radars, which is represented in the Figure 3.8.

In the data collection step, 11 distinct activities are collected with six volunteers of various ages, heights, and weights. The performed activities can be listed as walking towards the radar, walking away from the radar, picking up an object from the ground, bending, sitting on a chair, kneeling, crawling, limping with a stiff leg, walking on toes, walking with short steps, scissor gait. In each activity, 60 samples are collected, which means each volunteer repeated the actions 10 times. The radar data matrices, range-time plots, and micro-Doppler signatures are also given separately for each sensor as provided in the previous datasets. Arrangement of the representation steps is also the same as in the earlier sections, as shown in the Figure 3.1.

As mentioned in the previous paragraph, the first radar used in this study is XeThru, produced by Novelda AS, which operates in the 7-10 GHz as a UWB impulsed radar system on chip (SoC) radar sensor, whose data sheet might be seen in [103]. It should be highlighted that the radar principle is slightly different from other human activity datasets examined in this thesis, since this radar works with a pulsed regime. However, as explained in the second chapter, it is plausible to construct the same radar data matrix with pulsed systems as was done in the

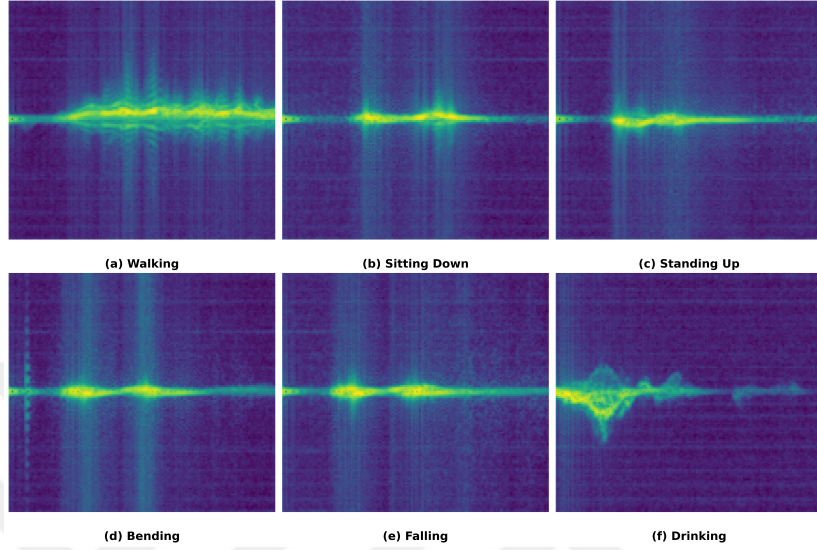


Figure 3.7: Micro-Doppler Representations of Dataset-II Classes

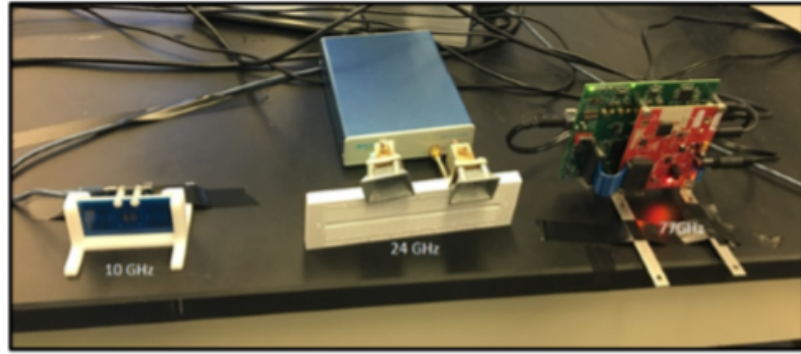


Figure 3.8: Radars Used in the Dataset-III

FMCW radar case. In the collection, the length of the recordings differs from sample to sample. The slow-time samples are fixed to 4240 bins to provide a fixed-length dataset, and longer samples are clipped to this value. Although the operating parameters are not provided in the study, the length of the fast-time dimension of the dataset is found to be 91 in the pre-processing codes provided by the publisher. The radar range-time matrices and example micro-Doppler signatures are given in the Figures 3.9 and Figure 3.10.

The second radar used in this study is very similar to the one used in the previous dataset, where both radars are used by the same manufacturer, Ancortek.

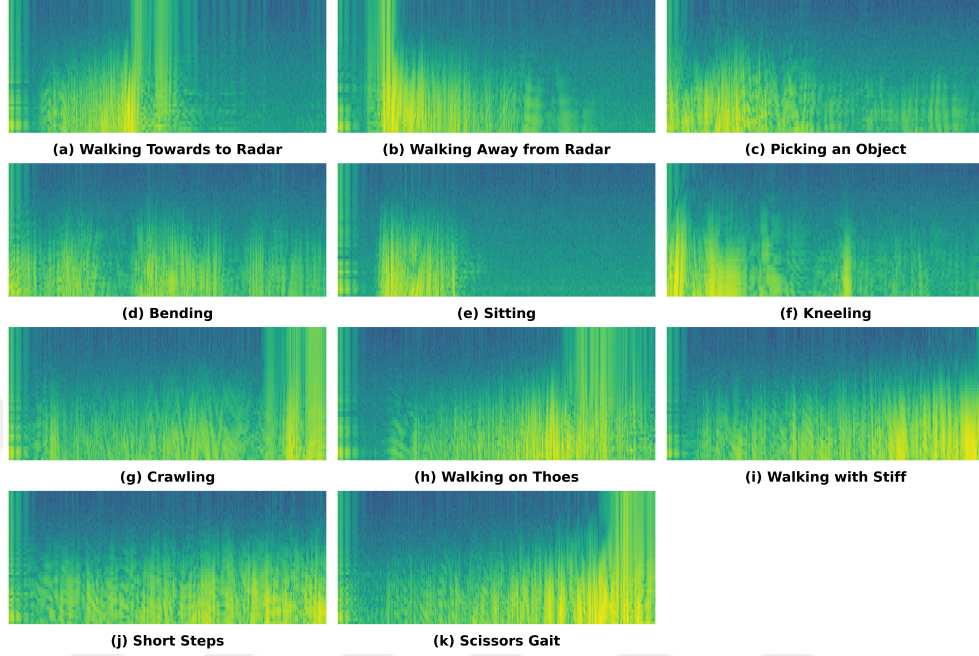


Figure 3.9: Range-Time Representations of Dataset-III (10 GHz) Classes

This radar operates with a 1500 MHz frequency bandwidth, where the center frequency is 24 GHz. Compared to the previous radar, clipping is unnecessary since all the samples are shaped in the same format, 128 bins in the fast-time dimension, whereas 10000 bins in the slow-time dimension. The radar range-time matrices and example micro-Doppler signatures are given in the Figures 3.11 and Figure 3.12.

This dataset's third and final radar is the IWR1443 radar produced by Texas Instruments, whose technical details are provided in [104]. The radar works with a 750 MHz frequency bandwidth and a 77 GHz center frequency in the data collection scenario. This radar also differs from the radars examined in this dataset since four receiving channels produce four slightly different sampled data, and even though it is not applied in this case, it brings an opportunity to use spatial dimensions with the different antenna usage. In our application, we have utilized only one channel among the four provided channels since our initial observations about the multi-channel usage do not provide considerable improvements. Similar to the 24 GHz radar data, there is no need to clip the dataset since the recording time of the samples is identical, and the samples

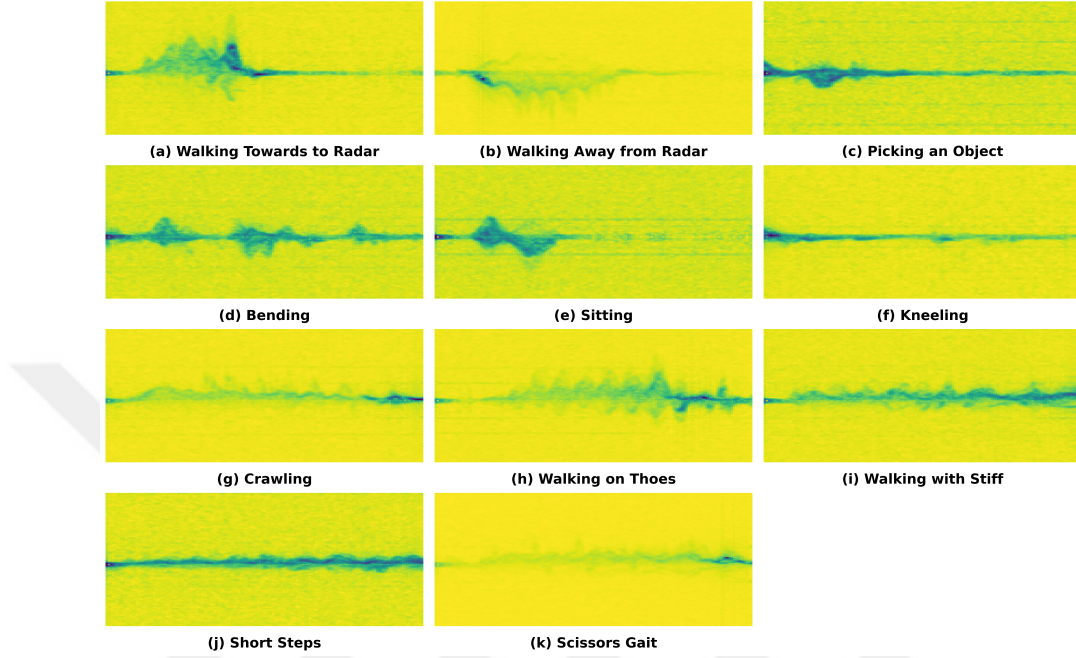


Figure 3.10: Micro-Doppler Representations of Dataset-III (10 GHz) Classes

are shaped as 256 fast-time bins with 32000 slow-time bins. As provided in the previous datasets, the range-time plots and micro-Doppler signatures of the class samples are provided in Figure 3.13 and Figure 3.14.

In the experimental scenarios examined in the scope of this thesis, both stand-alone and simultaneous usage of the datasets of different radars belonging to the third dataset are utilized. In order to highlight the differences of the radars in the following sections, the operation frequency is used. For instance, as shown in the Table 5.3, "Dataset 3 (10 GHz)", "Dataset 3 (24 GHz)", and "Dataset 3 (77 GHz)".

3.4 Dataset IV, Drone Classification Dataset

The fourth and final dataset is very distinguishable from the first three shared datasets in terms of the working principle of the utilized radar and its usage purpose. The used radar technology, SAAB SIRS 1600 FMCW radar operating

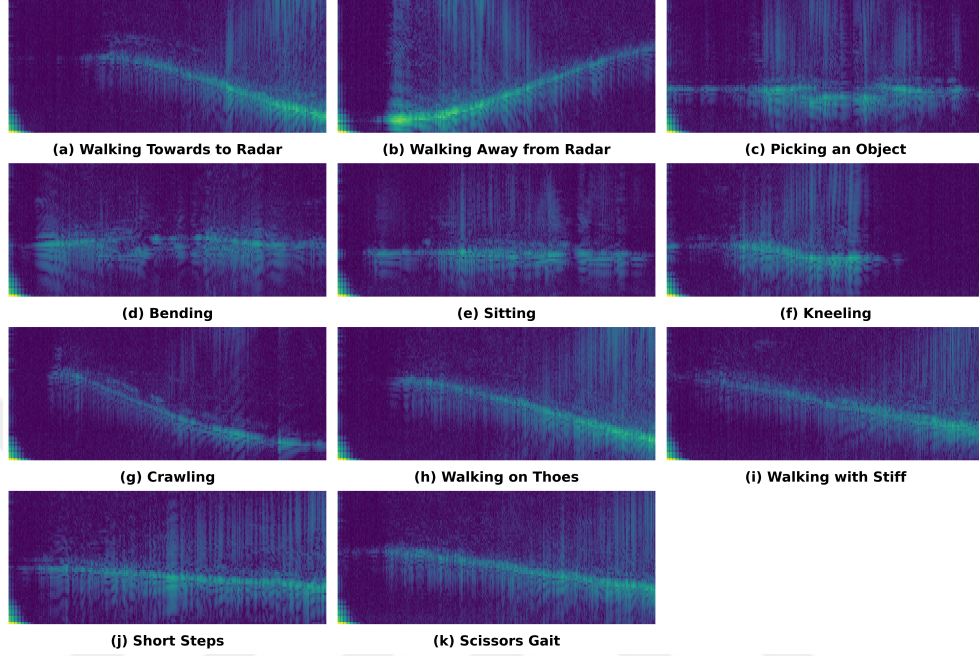


Figure 3.11: Range-Time Representations of Dataset-III (24 GHz) Classes

at 77 GHz with mechanically steering radar as published in the work [39]. The mechanical steering effect occurs in the azimuth direction, and the data processing steps become different due to this steering mechanism. The purpose of this dataset is to classify classes, including six different drones together with humans, and birds, instead of human motion classification. Thus, the dataset is prepared by considering this objective. In this thesis, we focus on the drone classes in the dataset since the original study shares a detailed analysis of the drone classification results that helps us compare our approach with the original findings. The examined drones are presented in Figure 3.15.

Although the main idea is target classification in the original work, the scan-based radar usage for this task is quite rare in the literature. Due to scan-based data formatting, the conventional approaches to micro-Doppler calculations are no longer usable, as we followed in other datasets as a standard pipeline. However, azimuth scanning is completed in a fixed time period and might be interpreted as slow-time as discussed in the works [105, 106]. In light of these studies, we treated the samples of these datasets as ordinary FMCW radar outputs.

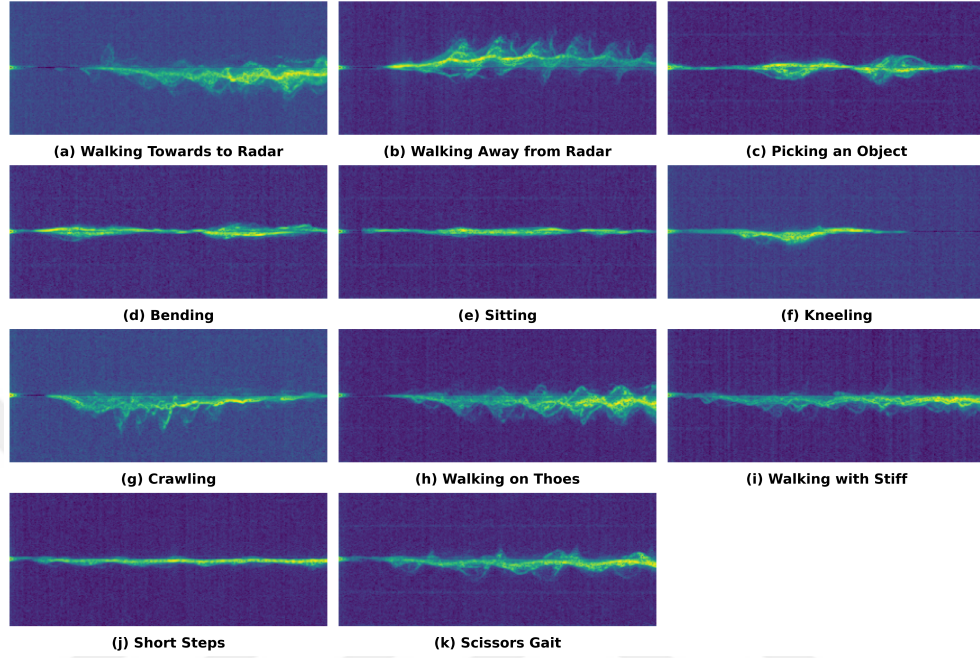


Figure 3.12: Micro-Doppler Representations of Dataset-III (24 GHz) Classes

The samples in the dataset have a range vs azimuth representation. Also, it should be stated that range compression is applied to all range bins, and the authors provide five compressed range bins. In this case, the provided range-azimuth representations are directly interpreted as range-time representations of the class, and the standard pipeline described in the Figure 3.1 is applied without steps, including the range-time calculations, as expected. Considering this approach, the related representations, range-time and micro-Doppler figures, are provided in Figure 3.16 and Figure 3.17. It should also be noted that the number of samples varies between 6921 and 12093 in distinct classes, and to reduce the problems occurring due to class imbalances, we have used the first 6000 samples of each class.

This dataset is explicitly chosen to enhance the scope of the thesis by indicating the performance of the proposed approach in another task, which is considerably distinct in terms of classification type and data structures.

Table 3.1: Radar Dataset Summary Table

Feature	Dataset I	Dataset II	Dataset III – 10 GHz
Frequency Band	24 GHz	5.8 GHz	10 GHz
Environment	Simulation	Real	Real
Number of Classes	11	6	11
Number of Samples in Classes	60 in each class	311, 312, 311, 311, 311, 197	73, 87, 83, 83, 77, 78, 90, 67, 90, 88, 83
Total Samples	660	1,753	899
Sample Size (Fast $\times$ Slow Time)	128×8000	128×5000	91×4220

Feature	Dataset III – 24 GHz	Dataset III – 77 GHz	Dataset IV (Drone)
Frequency Band	24 GHz	77 GHz	77 GHz
Environment	Real	Real	Real
Number of Classes	11	11	6
Number of Samples in Classes	61, 60, 60, 59, 62, 64, 64, 65, 56, 63, 62	60, 59, 62, 58, 56, 60, 61, 60, 58, 58, 60	6921, 9555, 10212, 8732, 12093, 11244
Total Samples	676	652	58,757
Sample Size (Fast $\times$ Slow Time)	128×10000	256×32000	5×256

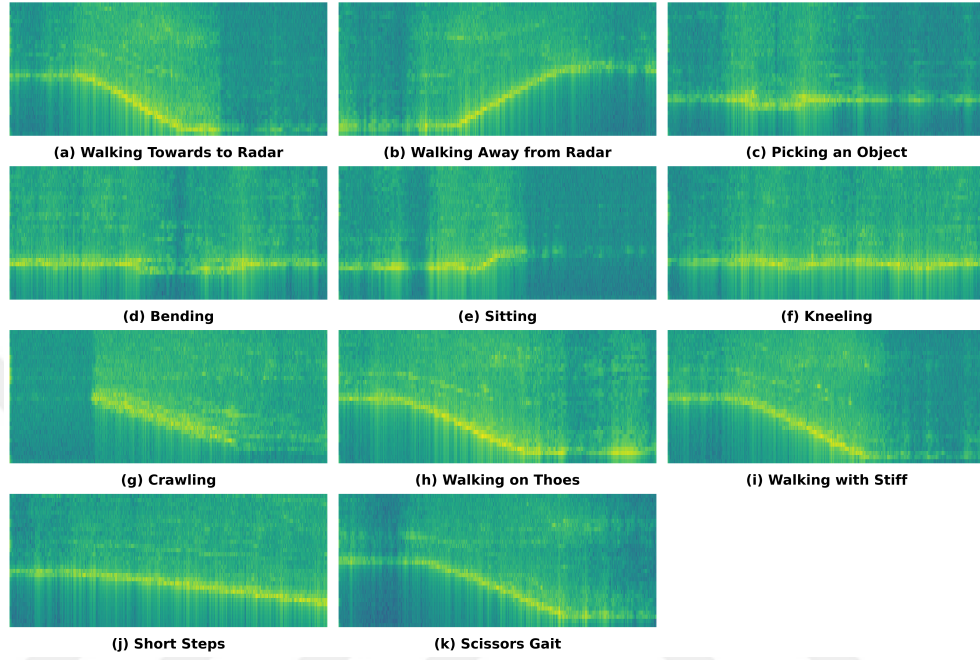


Figure 3.13: Range-Time Representations of Dataset-III (77 GHz) Classes

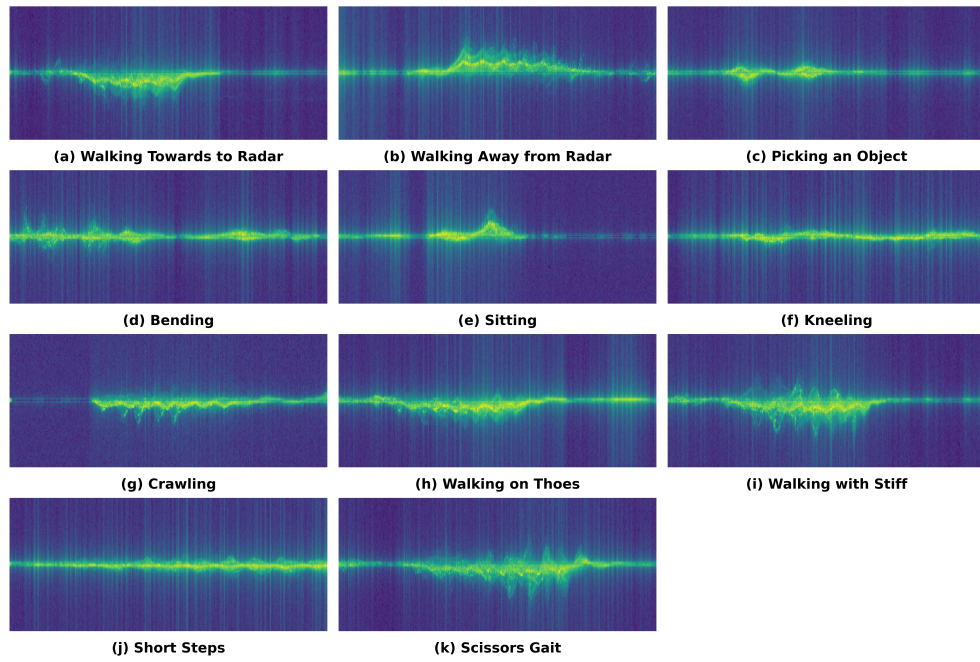


Figure 3.14: Micro-Doppler Representations of Dataset-III (77 GHz) Classes



Figure 3.15: Drones Used in Dataset-IV

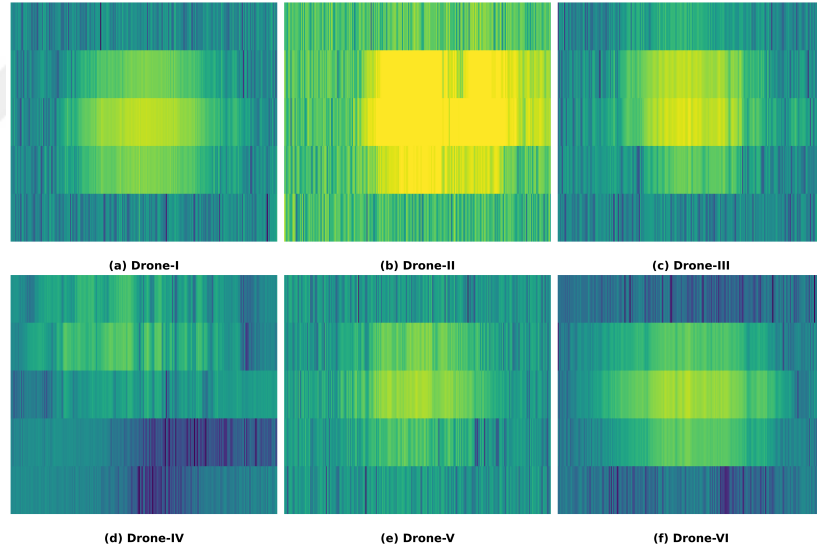


Figure 3.16: Range-Time Representations of Dataset-IV Classes

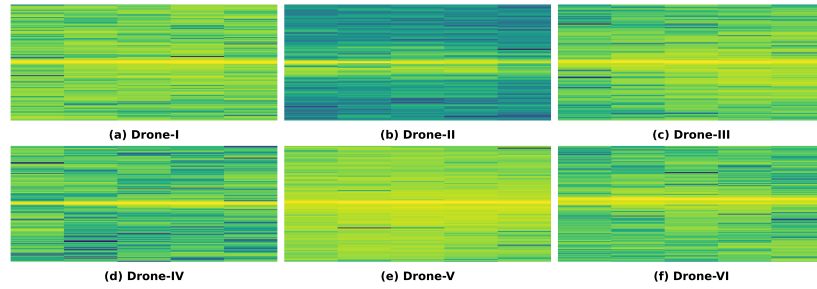


Figure 3.17: Micro-Doppler Representations of Dataset-IV) Classes

Chapter 4

Model Architectures

This chapter aims to explicitly express the different architectures examined in this study and the processing methods by enhancing theoretical descriptions provided in the second chapter with the exact procedures. The chapter starts by explaining the common pre-processing steps followed in the different architectures to obtain range-time plots used to feed networks. As the proposed methodology includes the adaptive generation of the spectrograms within the network architectures, the spectrogram generation methodology is examined with a proper explanation of the trainable fractional Fourier transform layer. After the explanation of adaptive spectrogram generation, the classifier architectures that use spectrograms have been explained in detail, with additional descriptions about the layers and methods, which were not shared in the second chapter.

4.1 Pre-processing Steps

As described in the third chapter, mainly four, separately six, different datasets are utilized in this thesis work. In the classifier models, inputs are expected as range-time plots; thus, the data pre-processing steps mainly include the generation of the range-time plots from the raw data, whose flowchart might be seen in

the Figure 3.1. However, this figure derives the process steps for the dataset representations provided in the third chapter. In contrast, the pre-processing steps used in the constructed algorithms are only applied until the range-time plots are acquired. The details of the procedure steps are provided below.

The first step is the unification of the data representation by considering the aforementioned radar data matrix structure. In the explained datasets, data formats are stored in different file formats such as ".mat", ".txt", ".np", and ".dat"; therefore, uploading scripts are derived accordingly to create the radar data matrix, including fast-time and slow-time samples. Although the datasets are recorded in different formats, all radar data is given in the form of IQ demodulated samples, whose procedure is also examined in the second chapter. Therefore, after uploading, complex-valued radar data matrices are constructed.

The correctly shaped radar data matrices are filtered with an MTI filter [107]. Although it is not necessary to analyze the micro-Doppler signatures of the targets examined by the FMCW radar, it is very beneficial to suppress the background effect in the radar data. The ground clutter is highly influential in the experimental datasets examined in this thesis; thus, the usage is needed to obtain more reliable results. The comparison of the filtered and unfiltered images is provided in Figure 4.1 with the frequency response of the filter constructed via the Butterworth filter [108].

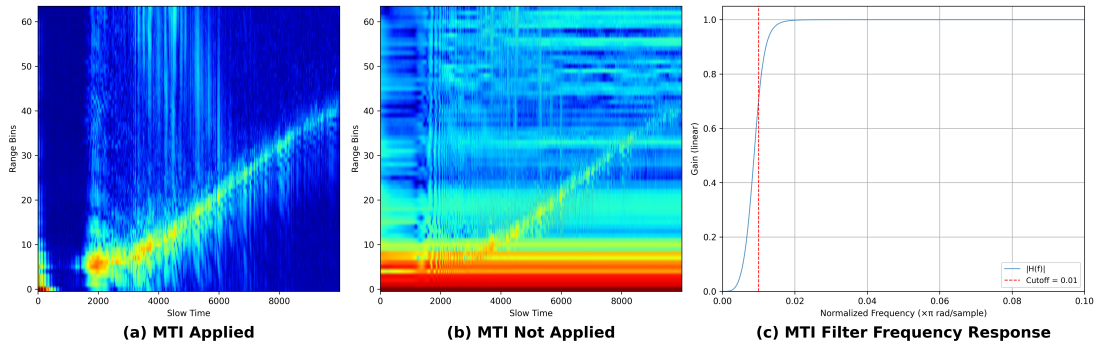


Figure 4.1: Frequency Response of the used MTI Filter and Impact of the Filtering

The generation of the range-time plots from the raw data is theoretically explained in the second chapter as applying a Fourier transform along the fast-time axis in the FMCW radars. Therefore, the FFT is taken along the fast-time direction to obtain the range-time plots. The result of the pre-processing procedure, until this step, is already provided in the previous chapter, including dataset representations. In addition, it should be noted that, since the Dataset-IV is supplied in the range-time format directly, no pre-processing step is applied until this part of the procedure.

The final step of the procedure before feeding samples to the classifier architecture is a well-known normalization process. Although several approaches might be found for the normalization process, the normalization utilized in this study might be described mathematically as provided in the equation 4.1, which is also known as standard score normalization [109, 110]. In deep learning related applications like those proposed in this thesis, normalization helps to stabilize the gradient descent and boosts the training speed by limiting the weights [111].

$$X_{normalized} = \frac{X - X_{mean}}{X_{variance}} \quad (4.1)$$

4.2 Spectrogram Generation

The conventional spectrogram generation and similar time-frequency representations are examined theoretically in the second chapter with examples from the literature. Standard Fourier transform and fractional Fourier transform, which might be seen as a generalized version of the ordinary transform operation, are examined with a detailed description. In this work, one of the key concepts is using the fractional Fourier transform with trainable orders, which deserves a detailed explanation.

The usage of the fractional Fourier transform in deep learning architectures has been examined through the literature, where some examples have already been

introduced in the second chapter about the micro-Doppler applications; however, primarily, the technique is considered with transform order selection as a hyperparameter selection [10, 12]. The adaptive order selection with trainable FrFT layers is proposed in [13]. As proposed in the original work, by using the derivative of the DFrFT matrix, which was provided in the equation 2.11, the update rule of the fractional order α with backpropagation may be obtained. Regardless of the architecture type, the claimed approach is applicable since the alpha order selection process shares the same logic used in model weight updates. In addition, this work shares the experimental results using CNN and RNN structures to support the derived application. They have obtained an increase in the classification performances in well-known datasets including "CUB-200-2011", "CIFAR-100", and "CIFAR-10" with different CNN architectures including "ResNet18", "ResNet34", "ResNet101", and "VGG16". Their analysis table might be seen in the Figure 4.2.

CNN Model	Transform	Dataset		
		CUB-200-2011	CIFAR-100	CIFAR-10
ResNet18	Time	70.95	80.78	95.54
	FrFT	71.56	81.20	95.70
	FFT	71.14	81.09	95.63
ResNet34	Time	74.64	82.89	96.34
	FrFT	74.76	83.91	96.77
	FFT	74.30	83.79	96.65
ResNet101	Time	78.81	85.97	97.64
	FrFT	79.35	86.81	97.75
	FFT	79.00	86.71	96.82
VGG16	Time	77.75	77.13	94.43
	FrFT	78.70	77.70	94.78
	FFT	78.13	77.44	94.42

Figure 4.2: Trainable FrFT vs FFT Accuracy Comparison Table [13]

In this thesis, the proposed approach depends on utilizing the framework proposed by this work using the trainable layer applications shared by the authors on GitHub. As explained before, time-frequency analysis methods are commonly used to obtain the micro-Doppler signatures, and the short-time fractional Fourier transform may also be utilized, whose mathematical background is explained in the second chapter. To integrate the trainable FrFT layer, the network architectures are constructed using pre-processing blocks that produce spectrograms with the trainable orders.

A more beneficial representation of the micro-Doppler signatures in the corresponding fractional domain may be obtained through trainable spectrogram generation, and more accurate classification becomes plausible. The pre-processing layers depend on basically the explanation provided in the second chapter and the equation 2.12. Therefore, three layers in this block take the normalized range-time inputs and produce spectrograms listed below.

- **Segmentation Layer:** The first layer in the spectrogram generation layers is the segmentation layer. As described in the second chapter, the STFrFT may be considered the windowed version of the FrFT operation. In this part, the normalized range-time inputs are segmented into sub-parts, and each sub-part is multiplied by the quantized Hamming window. As described before, the segmentation process is also applied, with some overlaps between sub-parts, to remove artifacts from the result. Since one of the configuration studies investigates window-overlap sizes, this layer is used with parametric inputs of window and overlap sizes. Finally, it produces the segments that will be transformed with FrFT.
- **Trainable FrFT Layer:** In this part, the FrFT is applied to the segments produced by the previous layer, and the results are inserted into the result matrix. As explained above, the FrFT order is used as a trainable parameter within the training process, and complex-valued STFrFT outputs are generated with this approach.
- **Log Scale Layer:** In the final step of the trainable spectrogram generation, the generated outputs of the previous layer are converted to log-scale representation with $20 \cdot \log_{10}(\text{abs}(\cdot))$ operation, and spectrograms are generated. Also, the spectrograms are normalized for better performance, as we did in the range-time data generation.

After those steps, the spectrograms in the learned fractional domain are constructed and fed to the classifier part of the proposed structures. Although several types of architectures are covered in this work, these layers are used in each experimental case. The same layers are also used in the traditional FT and

FrFT comparison experiments; however, the FrFT order is assigned to zero as a constant, non-learnable parameter. Figure 4.3 is provided to demonstrate the fractional order's effect in the spectrogram generation. As shown in this representation, changes in the fractional order may cause observable differences in the micro-Doppler signature representations, where the differences increase with the fractional order values as expected. In addition, although the slight differences cannot be differentiated from the Figure 4.3 in parts (a) and (d), the representation differs as provided explicitly in the Figure 4.4 by taking differences of standard spectrogram and small fractional changed representations.

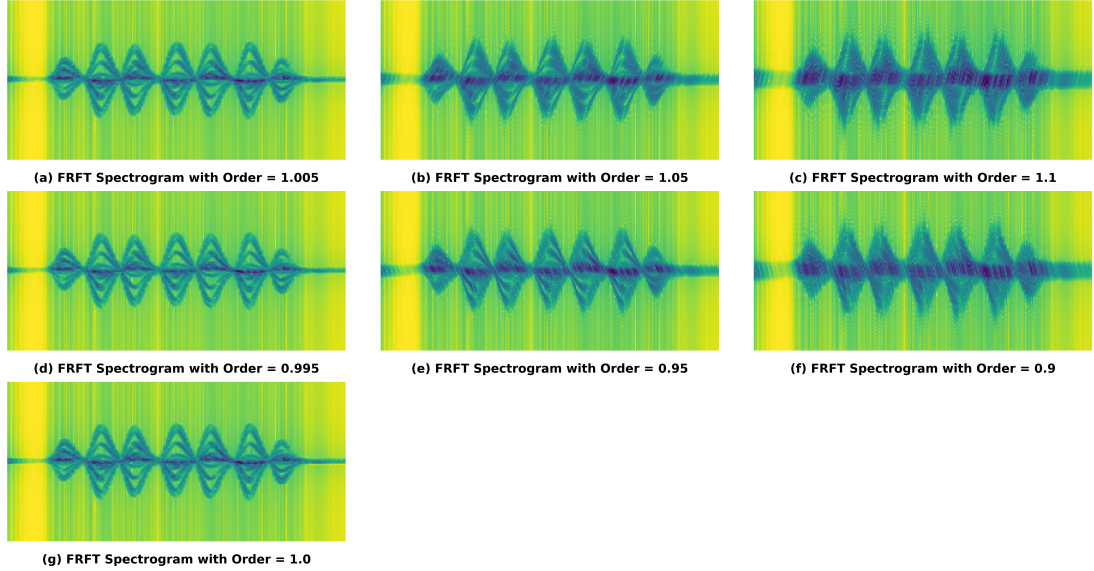


Figure 4.3: Effect of the Different Fractional Orders on Micro-Doppler Signatures

4.3 Single-Branch Models

The details of the single-branch models are established in this section. It should be noted that the model architectures explained in this section differ in the experiments that investigate the influence of block number. Other experiments are conducted using the model architectures explained in this section, where three blocks are implemented. In the block number experiments, the details of the added blocks are highlighted in the corresponding section in the Experiments

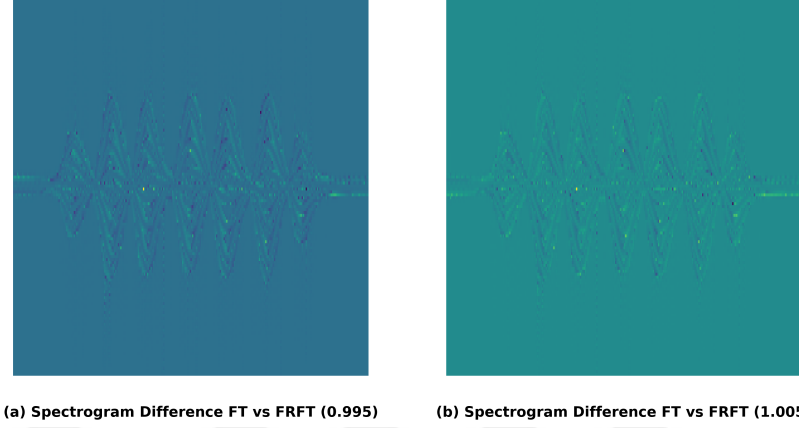


Figure 4.4: Differences between FrFT Order = 1.005 and FrFT Order = 0.995 Respect to FrFT Order = 1.000

and Results chapter.

As described in the third chapter of this thesis, the datasets utilized have samples of different sizes, and the number of classes is not the same for all classes. Therefore, input and output layers have different sizes, which is not shown explicitly in the explanation of the model structures. The parametric structures are provided in the related chapters, and parameter explanations are provided in the Table 4.1. Also, in the single-branch cases, the main difference occurs in the block types whose working principle is explained in the second chapter of this thesis.

Table 4.1: Descriptions of Symbolic Parameters in the Single-Branch Models

Symbol	Description
B	Batch size
R	Number of selected range bins
T	Number of slow-time samples
N	Number of time segments after segmentation
W	Window length used in segmentation (segment size)
H	GRU and LSTM blocks hidden layer size
C	Number of output classes
D	CNN model flattened feature dimension before the first linear layer: $D = 32 \cdot R \cdot \frac{N}{8} \cdot \frac{W}{8}$

4.3.1 CNN Based Model

CNN is the first block type used in the experimental scenarios in the thesis scope. As described in the previous sections, the micro-Doppler effects can be obtained via time-frequency analysis of the collected signals from the radar sensors, where the outputs are constructed as images. Therefore, the task is an image classification task, which is very suitable for the CNN models, as shown in the second chapter by revealing the nature of the CNN blocks.

After constructing the radar data in radar data matrices, as shown in the pre-processing part, time-frequency representations can be obtained for each range bin in the radar data matrix. If the targets move drastically, the target location in the range-time plots spreads to distinct range bins; thus, the analysis of the single range bin cannot be considered a complete analysis of the micro-Doppler effect through the target motion. Therefore, examining the several range bins becomes an obligation to analyze the micro-Doppler effects completely, which means the construction of several time-frequency analyses from different range bins for each sample in the classes is needed.

In the literature, the usage of several spectrograms belonging to the same sample is completed by utilizing different approaches. Most studies construct the time-frequency representations by considering the fixed range bins in their experimental data. For instance, if the range of targets in the dataset is spread in the 5 to 15 range bins in the total number of 128 range bins, the other range bins are ignored, and time-frequency analysis is not conducted on the ignored range bins [37]. Also, several studies have used adaptive techniques for range bin selection, where the range bins are selected using algorithms such as constant false alarm ratio (CFAR) and similar approaches [112, 113].

After selecting the range bins, time-frequency representations of different classes are created. Therefore, one sample of one class is represented by several time-frequency representations, or spectrograms, basically, and several methods

might be found in the literature for obtaining the combinations of these representations [7, 114–116]. These combining approaches include averaging the spectrograms to one representation, concatenation of different spectrograms to one image, and using classifier structures that can handle all spectrograms of one sample properly.

As shown in the proposed CNN structure in the Table 4.2, we have used all spectrograms simultaneously via 3-D convolutional layers, which was explained theoretically in the second chapter, to preserve all information of the micro Doppler signatures and use the advantages of the spatial relation of the simultaneous analysis of the range bins. Since the experiments conducted in this study cover several datasets of different sizes, the networks were fed with all data without range-bin selection approaches. It should be noted that the provided CNN structure is given in the parametric form, as exhibited in the table Table 4.2, and the parameters of the network sizes are explained in the table Table 4.1.

Table 4.2: Architecture of the Single-Branch CNN Model with 3 Blocks

Layer (Type)	Input Shape	Output Shape
Input	$[B, R, T]$	$[B, R, T]$
CreateSegmentsLayer	$[B, R, T]$	$[B, R, N, W]$
FrFTLayerWrapper	$[B, R, N, W]$	$[B, R, N, W]$
ComputeLogScaleLayer	$[B, R, N, W]$	$[B, 1, R, N, W]$
Conv3D (1 $\rightarrow$ 8)	$[B, 1, R, N, W]$	$[B, 8, R, N, W]$
BatchNorm3D (8)	$[B, 8, R, N, W]$	$[B, 8, R, N, W]$
MaxPool3D	$[B, 8, R, N, W]$	$[B, 8, R, N/2, W/2]$
Conv3D (8 $\rightarrow$ 16)	$[B, 8, R, N/2, W/2]$	$[B, 16, R, N/2, W/2]$
BatchNorm3D (16)	$[B, 16, R, N/2, W/2]$	$[B, 16, R, N/2, W/2]$
MaxPool3D	$[B, 16, R, N/2, W/2]$	$[B, 16, R, N/4, W/4]$
Conv3D (16 $\rightarrow$ 32)	$[B, 16, R, N/4, W/4]$	$[B, 32, R, N/4, W/4]$
BatchNorm3D (32)	$[B, 32, R, N/4, W/4]$	$[B, 32, R, N/4, W/4]$
MaxPool3D	$[B, 32, R, N/4, W/4]$	$[B, 32, R, N/8, W/8]$
Flatten	$[B, 32, R, N/8, W/8]$	$[B, D]$
Linear ($D \rightarrow 64$)	$[B, D]$	$[B, 64]$
Dropout ($p = 0.4$)	$[B, 64]$	$[B, 64]$
Linear ($64 \rightarrow C$)	$[B, 64]$	$[B, C]$

The visual representation of the model is provided in Figure 4.5 by following the structure in Table 4.2. Also, although the core idea behind the CNN architectures

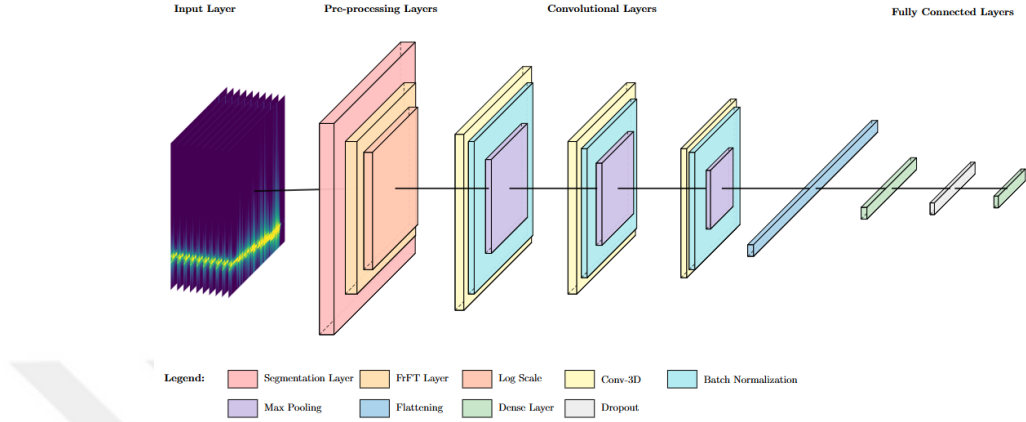


Figure 4.5: Constructed CNN Architecture

was provided in the second chapter, the architecture's additional layers, such as batch normalization, dropout, and flatten layers with the activation function Rectified Linear Unit (ReLU), should be appropriately examined.

The batch normalization layer integrated into the CNN architecture is utilized to stabilize and boost the speed in training sessions by normalizing the input of each layer, as shown in the architecture. The normalization idea is similar to what we used in the range-time matrix construction; however, the normalization is applied to the inputs of the layers across the mini-batches. With the help of the batch normalization layers, the network can avoid internal covariate shifts and produce stable training sessions [117].

The second unexplained layer type is the flatten layer used in the CNN architecture. This layer is used to reshape the multi-dimensional tensors into 1-D vectors since the dense layers expects 1-D inputs. Therefore, the data transmission between the convolutional layers and the dense layers is plausible with the usage of the flatten layers. As commonly used, this layer is placed right after the convolutional blocks and before the dense layers [118].

The final layer in this architecture is a dropout layer, which may also be considered as a regularization method commonly used in neural network applications. The idea behind the layer is randomly setting to zero some of the fractions between the neural network layers, commonly dense layers, to avoid overfitting. In

each training trial, activations are randomly set to zero by considering the probability, which is shown as 0.4. As expected, this layer is enabled only in training sessions, not in testing sessions [119].

A final configuration parameter that should be explained is the activation function used in the architecture. In all possible scenarios, the same activation function, ReLU, is utilized in the model due to the simplicity and efficiency of the function. In a mathematical sense, ReLU might be defined as a piecewise linear function that transmits the inputs to outputs if the value is positive; if the value is negative, the output will be zero, as shown in the equation 4.2 where x denotes the function input. The effectiveness of the model comes from representing the non-linearity without using complicated transformations [120, 121].

$$f(x) = \max(0, x) \quad (4.2)$$

4.3.2 RNN Based Models

As described in the earlier parts, the thesis study also includes the RNN structures in addition to the basic CNN application. The constructed RNN-based architectures share the same layers and configurations except the used RNN blocks, where the first one depends on Bi-LSTM blocks and the second one depends on Bi-GRU blocks. The common architecture with parametric representation is provided in the Table 4.3 with the visual representation demonstrated in the Figure 4.6.

The theoretical framework of the utilized RNN blocks has already been shown in the second chapter with mathematical derivations and corresponding explanations. Similar to the CNN-based architecture, spectrogram generation methods, including the trainable FrFT layer usage, are also used directly in the RNN-based architectures. As a result of the same layer structures, the idea of using several range bins is also applied in the RNN blocks to utilize the spatial relations in the examined samples. However, in addition to the theoretical core of the applications examined in the second chapter, an additional layer of explanations is given

in the CNN model examination, and an attention layer is inserted into the RNN models.

The attention layer, or attention mechanism, is a widely used concept in machine learning applications to emphasize the important parts of the input data in the model architectures. Deep architectures can increase the importance of the specific parts of the input elements dynamically with the help of attention mechanisms. Therefore, important features contribute to the model learning session with the model's focus. In the literature, several attention mechanism types have been proposed, including additive (Bahdanau), multiplicative (Luong), or commonly used self-attention layers with the same idea of focusing on important features [122, 123].

In the RNN structures provided in this thesis, a single-layer attention mechanism is proposed with the additive type, which may be expressed as the score of each RNN output is calculated via the attention score, and important features in the micro-Doppler signatures are highlighted. The mathematical background of the additive attention mechanism might be explained by considering a GRU layer producing a sequence of hidden states $H = [h_1, h_2, \dots, H_T]$, and the attention score e_t is calculated for each time step as shown in the equation 4.3.

$$e_t = v^T \tanh(Wh_t) \quad (4.3)$$

where W denotes the learnable weight matrix, and the v denotes the learnable context vector. The calculated scores via this formula are normalized by utilizing the softmax function, and attention weights a_t are obtained as in the equation 4.4.

$$a_t = \frac{\exp(e_t)}{\sum_{k=1}^T \exp(e_k)} \quad (4.4)$$

The final context vector obtained may be expressed by utilizing the hidden states and the normalized attention weights as given in the equation 4.5. The

context vector is passed to the final classification layer. Therefore, the important parts of the inputs are highlighted with the proposed attention structure.

$$c = \sum_{t=1}^T \alpha_t h_t \quad (4.5)$$

Aforementioned suitability of the RNN structures for the micro-Doppler signature classification task should be stated again. Since the time-frequency analysis of the raw radar data is conducted as a basis of the study, the examined task might be formulated as a time-series classification task in addition to the image classification task. Therefore, RNN structures are very suitable for the task examined by this study.

Table 4.3: Architecture of the Single-Branch Bi-LSTM/Bi-GRU Model with 3 Blocks

Layer (Type)	Input Shape	Output Shape
Input	$[B, R, T]$	$[B, R, T]$
CreateSegmentsLayer	$[B, R, T]$	$[B, R, N, W]$
FrFTLayerWrapper	$[B, R, N, W]$	$[B, R, N, W]$
ComputeLogScaleLayer	$[B, R, N, W]$	$[B, 1, R, N, W]$
LSTM/GRU Layer 1 (Bi)	$[B \cdot R, N, W]$	$[B \cdot R, N, 2H]$
LSTM/GRU Layer 2 (Bi)	$[B \cdot R, N, 2H]$	$[B \cdot R, N, 2H]$
LSTM/GRU Layer 3 (Bi)	$[B \cdot R, N, 2H]$	$[B \cdot R, N, 2H]$
TemporalAttention	$[B \cdot R, N, 2H]$	$[B \cdot R, 2H]$
Linear ($2H \rightarrow 1$)	$[B \cdot R, 2H]$	$[B \cdot R, 1]$
Reshape	$[B \cdot R, 1]$	$[B, R]$
Flatten	$[B, R]$	$[B, R]$
Dropout	$[B, R]$	$[B, R]$
Linear ($R \rightarrow C$)	$[B, R]$	$[B, C]$

4.4 Multi-Branch Models

In the multi-branch models, the main aim is to discover the effects of parallel feeding with two diverse approaches. The classification performances of the single-branch structures, which use single time-frequency representations, might

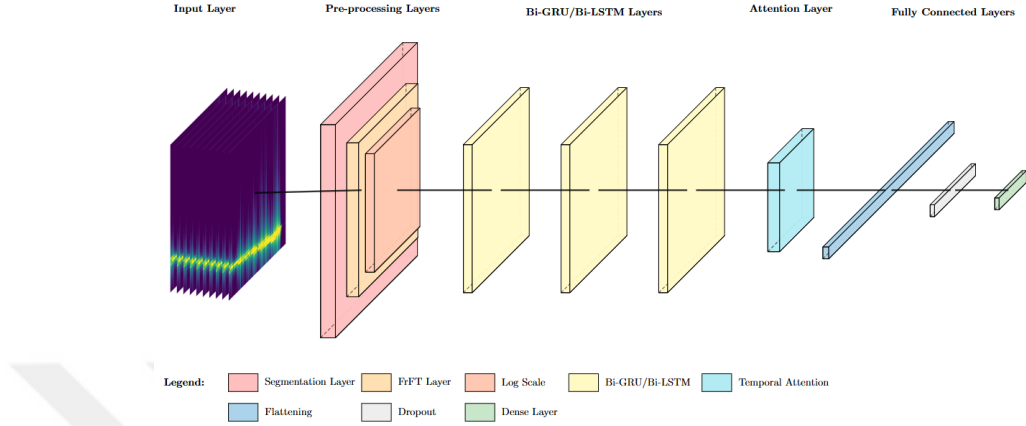


Figure 4.6: Constructed RNN Architectures

be enhanced by combining the different representations in terms of transformation parameters and radar data inputs. This section includes mainly two different techniques with the usage of the single-branch structures in each scenario, that are fusion models and siamese models.

As shown in the next chapter, these models are subjected to the same experimental steps that are used in the single-branch model evaluations, and performances are reported with mean classification performances and the standard deviations. Thus, the contribution of the parallel feeding is examined in terms of the accuracy and stability with the extensive dataset and model structure support.

4.4.1 Fusion Models

In the fusion models, the main aim is to investigate the effect of the simultaneous usage of the different time-frequency representations by using the different fractional orders. The effect of the initial fractional order in the classification models is examined, which might be seen as an investigation of the different time-frequency representations in the classification performances of the different models. However, models only use the one-time-frequency representation for classification, which is determined adaptively by the model with the trainable

fractional orders.

Since the different fractional orders lead to different time-frequency representations, the simultaneous usage of different representations may be beneficial to increase classification performances. In order to achieve this purpose, single-branch models are placed in parallel with different repetitions, including 3, 5, and 7 parallel structures. In each case, one input is fed to these parallel models, and one output is obtained by concatenating the final layers as shown in Figure 4.7, where the three-branch GRU-based architecture is presented. In each parallel branch, the initial fractional order is set to a different point, as given in the list below, to utilize the different time-frequency representation of the same sample simultaneously.

- **Single Branch** $\alpha = 1.0$ (standard Fourier Transform)
- **Three Branch** $\alpha \in \{0.7, 1.0, 1.3\}$
- **Five Branch** $\alpha \in \{0.5, 0.7, 1.0, 1.3, 1.5\}$
- **Seven Branch** $\alpha \in \{0.2, 0.5, 0.7, 1.0, 1.3, 1.5, 1.7\}$

The single-branch model, with an initial starting with the Fourier transform, is chosen as a baseline to analyze the effect of multi-layer structures. The branch numbers and the initial fractional orders are chosen symmetrically with respect to the ordinary Fourier transform to obtain time-dominant and frequency-dominant features simultaneously, as explained in the second chapter. The dominance of the time and frequency components in the FrFT is determined by the transformation order. To increase the reliability of the examination with broader coverage, setups including five and seven branches are also constructed in this manner.

It should be emphasized that the explained multi-branch approach is reiterated by using the three different single-branch structures with all datasets to examine the idea in a more detailed way. The experimental results and the discussions will be shared in the next chapters in a detailed way.

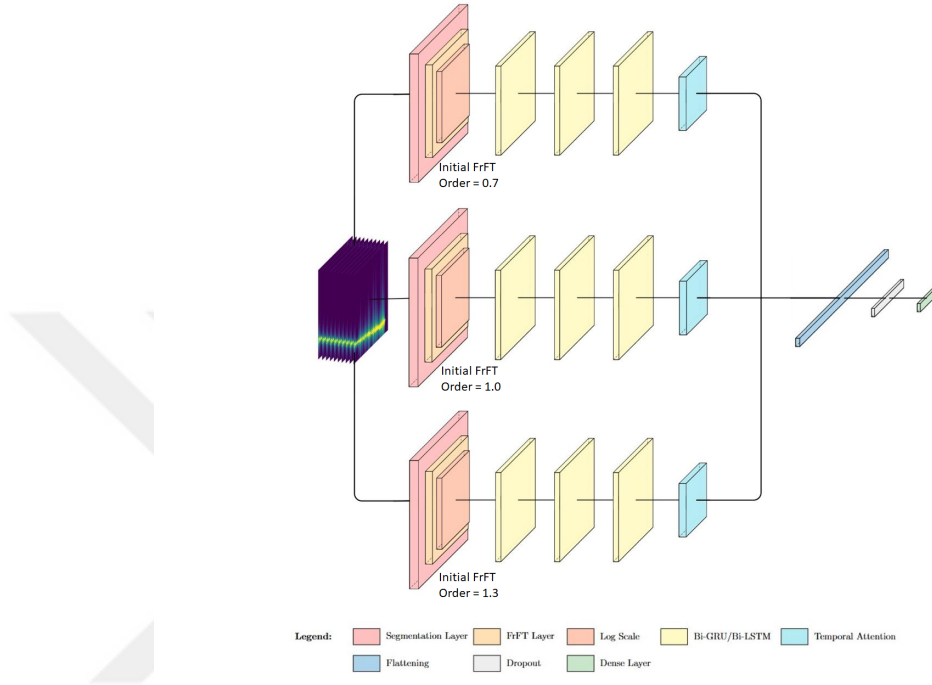


Figure 4.7: GRU-Based Multi Layer Fusion Model

4.4.2 Siamase Models

The other multi-branch structure investigated in this thesis is the Siamase multi-branch structure. siamese neural networks include branches sharing the same weights in the branches where each branch is fed with different input. The main idea behind this approach is to process distinct inputs with an identical process with the help of the shared network weights. Although the idea is introduced by the studies including similarity learning, it might be very beneficial for tasks that involve structurally similar inputs collected by different sensors. In our case, it might be considered as the different radars collecting the same scene with different frequency bands, such as the third dataset. Also, in the literature, various results utilizing this type of architecture might be found, even for the human activity classification task with FMCW radars, as examined in the scope of this thesis [38, 124, 125].

Since the third dataset examined in this study includes data from three distinct

radars collecting the same scene, it is very suitable for the siamase model application. The idea introduced in the single-branch structures is preserved; however, three inputs are fed to three branches, sharing the same weights, simultaneously, and results are concatenated in the final fully connected layer as provided in Figure 4.8. In this case, the siamese models are constructed three times for different backbones introduced in the single-branch models, including CNN, LSTM, and GRU blocks, as we conduct for the fusion model examination. The experimental results and the related discussion are also provided in the upcoming chapters.

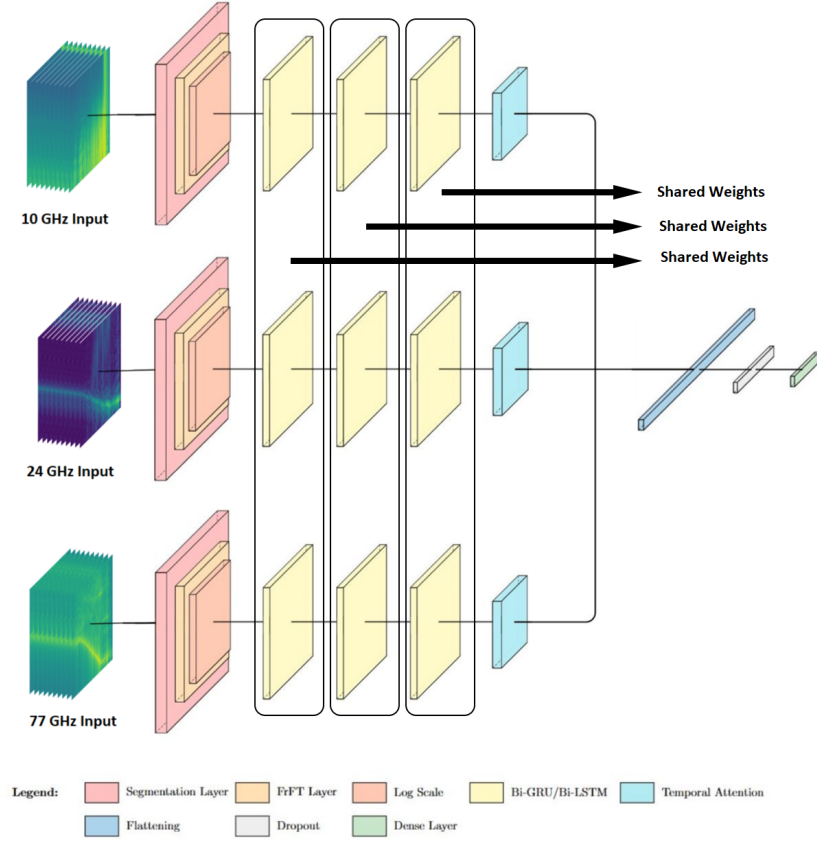


Figure 4.8: GRU-Based Siamese Model

Chapter 5

Experiments and Results

As stated several times in the previous chapters, to provide an in-depth examination of the usage of the trainable FrFT operation, distinct datasets are evaluated with different types of deep learning architectures. This chapter presents the experimental setup and the results of the different sets of experiments. Since experiments cover different architectures and datasets, this chapter might be seen as repetitive; however, to avoid confusion, the results are given in order, considering the architecture type and performance in different datasets. It should be stated that the results are discussed briefly in this chapter by considering separate experiments; overall analysis and discussion are shared in the next chapter since the experiment set is considerably broad.

5.1 Experimental Setup

An excessive set of experiments is conducted, including different model structures and datasets explained in the previous chapters. In this section, the training and testing methodology is presented first. It should be noted that the experimental procedure is derived for the first dataset, similar to the hyperparameter tuning in the model construction, and applied to all experimental scenarios, which will also

be discussed briefly. In the following part, the performance metrics are reported by considering the proper explanations of the well-known metrics in classification tasks. The last part reports details of the hardware and software environments.

5.1.1 Training and Testing Procedure

As stated several times, the study’s primary aim may be supervised learning for a multi-class classification task in machine learning terminology. To achieve this task, the categorical cross-entropy loss minimization problem is considered, which is commonly used in the literature. The mathematical expression of the categorical cross-entropy loss is given in the equation 5.1. This loss function is used to penalize incorrect predictions and reward correct predictions.

$$\mathcal{L}_{CE} = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (5.1)$$

where C is the number of the output classes, $y_i \in \{0, 1\}$ is the one-hot encoded ground truth label, and $\hat{y}_i$ is the softmax-normalized probability of class i predicted by the model.

Before the training session, the datasets are shuffled and split into training (%80), validation (%10), and test (%10) sets. It should also be noted that the shuffle and split steps are completed using fixed seeds. This is a key step since the main aim is to observe the effects of the operations, ensuring the same splits in different scenarios become necessary. Also, in the experiments, including comparing scenarios like FT and FrFT, models are initialized with the same weights to compare only the effect of the examined subject, concerning the control experiment used in each experiment.

In the training session, the Adam optimizer updates the model parameters. The Adam (Adaptive Moment Estimation) optimization method combines the advantages of both Momentum and RMSprop techniques to manipulate the learning

rates robustly [126–128]. Therefore, learning rates in the training period change dynamically with faster convergence thanks to the Adam optimizer procedure. Equations related to the Adam optimization procedure are given in the equation 5.2. The first line of the equation indicates the First moment estimate, and the second line indicates the second moment estimate, or mean and variance estimates. The third line indicates the bias correction to avoid the tendency to zero due to the m_t and v_t values. In the last step, the weight update is provided where the θ value indicates the weights in the deep learning model [126].

$$\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla \mathcal{L}_t \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2) (\nabla \mathcal{L}_t)^2 \\
\hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\
\theta_{t+1} &= \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
\end{aligned} \tag{5.2}$$

where the θ denotes the model parameters updated during training, α denotes the learning rate which is set to 0.001 in all experiments. The β_1 and β_2 are assigned the default values of 0.9 and 0.999, where those parameters are used to control decay rates for moments, and $\epsilon = 10^{-8}$ is a default constant to avoid division by zero. It should also be highlighted that the ridge regression, or L2 regularization, term is applied with 10^{-4} to avoid overfitting by penalizing large weights [129]. As explained in the previous sections, dropout layers are also used in the architectures.

A learning rate scheduler is also utilized to improve robustness and efficiency. The scheduler used in the experiments is ReduceLROnPlateau, which follows the validation accuracy and reduces the learning rate to half of the current value if no improvements occur in three consecutive epochs [130]. This method brings the opportunity to obtain more stable models.

In addition to the explained settings, early stopping is utilized in the training to consider improvements in validation accuracy. If there is no improvement in the validation accuracy in the eight epochs, the model training is stopped, and

the best model is stored. Also, the maximum epoch number is set to 50 epochs, which is not reached in any experiments; thus, it may be considered a suitable limit.

After completion of the training session, the best model concerning validation accuracy is evaluated with the test set, and performances are analyzed by considering different metrics as explained in the following part. This procedure is repeated 50 times with different seeds, meaning that different data shuffles and network initializations are utilized. If there is no need for further investigations, the mean and standard deviations among 50 trials are reported in the experimental scenarios. Also, a summary of the training parameters is provided in Table 5.1. In this table, the general methodology is summarized. The scenarios, including the parameter changes in this procedure, are stated in the related section, and the architecture-related parameters are explained in the previous chapter.

Table 5.1: Tuned Training Procedure Parameters

Parameter	Value
Batch Size	16
Segment Window Length	128
Segment Overlap	100
FrFT Order	Trainable (with initial value $\alpha = 1.0$)
Loss Function	Categorical Cross-Entropy
Optimizer	Adam
Learning Rate	1e-3
Early Stopping	Patience = 3
Number of Trials	50

5.1.2 Performance Metrics

A wide set of performance metrics is used to compare the results of the different experiments. It should be noted that, since some metrics indicate a high correlation, all metric outputs are not shared in the experiment sections separately to repeat the same information. The basic metrics obtained from the test sets can be listed as follows [131].

- Training Accuracy and Loss: The proportion of correctly predicted labels in the last epoch of the training session. Also, the value of the loss function described in the previous part is used in the same training session.
- Validation Accuracy and Loss: The best accuracy value and corresponding loss value on the validation set. This metric is also used to determine the model checkpoint.
- Testing Accuracy and Loss: The proportion of correctly predicted labels on the unseen test data using the best model obtained from training.

Additionally, classification metrics calculated from those values are also provided with the results. Those metrics and computation methods are provided in this section, including Precision, Recall, and F1-Score. A brief description of the terminology is needed to provide a suitable definition for these metrics, which is listed below [132]. It should also be noted that the order alpha is also stored to follow the relation between the alpha order and the model performances.

- True Positive (TP): The number of instances where the model correctly predicted the positive class.
- True Negative (TN): The number of instances where the model correctly predicted the negative class.
- False Positive (FP): The number of instances where the model incorrectly predicted the positive class.
- False Negative (FN): The number of instances where the model incorrectly predicted the negative class.

Considering the brief description in the list above, the precision can be explained as a proportion of TP value to the total positives in predictions, as given in the equation 5.3. In other words, this metric indicates how many positive predictions are true compared to all positive predictions.

It should be highlighted that the given definitions are given for the binary classification task, whereas multiclass classification is applied in the scope of this work. The provided metrics are calculated with a one-vs-rest strategy in the multiclass case. This strategy assumes the target class is the positive class, where the others are considered as negative, as consistent with the literature [132]. After the calculation of the metrics for each class, results are averaged, which is known as macro-averaging, to obtain final metrics such as precision, recall, and F1-score. This approach ensures a fair evaluation strategy since the macro-averaging solves the problems caused by the class imbalances.

$$precision = \frac{TP}{TP + FP} \quad (5.3)$$

The recall metric, or true positive rate, measures the model performance by considering all positives. In other words, the proportion of true positives to data should be predicted as true, which is firmly related to the precision metric and calculated with the equation 5.4.

$$recall = \frac{TP}{TP + FN} \quad (5.4)$$

The final metric, F1-score, combines precision and recall metrics by taking the harmonic mean of these metrics. Thanks to the F1-score, model performance may be computed effectively by eliminating extreme values from precision or recall. Mathematical representation of the metric calculation can be seen in the equation 5.5.

$$F1\text{-Score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (5.5)$$

Although the accuracy and presented metrics yield similar numerical results in balanced datasets such as our first dataset, F1-score and related metrics provide a more informative evaluation by considering both false positives and false negatives. In the imbalanced class distribution scenarios contribution of the used

methods becomes more visible, such as in our second dataset, since the F1-score and related metrics offer a more nuanced and reliable assessment compared to accuracy alone.

Therefore, these metrics are stored in each experiment to assess the model performance. As mentioned before, all experiments are repeated 50 times with random seeds. In the experiments, testing accuracy and F1-score, which include both precision and sensitivity, values are shared instead of sharing all stored values to increase the readability. Although both metrics indicate similar behavior, in the occurrence of imbalanced classes, like the last class of the second dataset, the F1-score is more reliable since the problems due to imbalances are eliminated [133]. Also, instead of sharing the results of all 50 trials, the mean and standard deviations among the 50 experiments are provided to highlight the model performance and stability with the aim of a statistically trustworthy analysis.

5.1.3 Hardware and Software Setups

All experiments in this thesis are conducted in the Jupyter Lab environment with Python base. Since the experiment set is wide, three hardware environments are used simultaneously to complete all possible options, including datasets and deep learning architectures. The setups used in the experiments are given in the list below. The main features of the distinct setups are also provided accordingly.

- Primary Setup: A local workstation includes an NVIDIA RTX 4090 GPU with 24 GB VRAM.
- Secondary Setup: A local computer includes an Intel Xe Graphics DPU with 8 GB VRAM.
- Tertiary Setup: A remote workstation provided by Google Colab Pro with NVIDIA A100 GPU comes with 40 GB VRAM.

The software environment, which is shared by all hardware options listed above, includes the same packages and versions. Software environment includes Python 3.10, PyTorch 2.7, CUDA Toolkit 11.8, and JupyterLab 4.4.2 with the suitable packages including numpy, pandas, matplotlib, scikit-learn, and torch-frft.

5.2 Single Branch Models

To increase the readability and consistency of the results representations, the examined model architectures are separated by considering the number of branches. In this section, outcomes of the single-branch architectures are discussed. These architectures are given in the order of CNN-based, LSTM-based model, and GRU-based models, as mentioned before in this thesis. Firstly, a comparison of ordinary FT and FrFT operations and outcomes in terms of performance metrics is presented. In each architecture, configuration studies are shared, including block number and the STFT parameters, i.e., window size and overlap size. After that, experiments about augmentation methods are shared, including Mix-up augmentation and slow-time stretching. Finally, the effect of the initial value of the FrFT order is examined separately in each architecture.

5.2.1 CNN-Based Model

The structure of the CNN model is provided in the third chapter by sharing the 3-block architecture in both visual representation and a table, including the parametric representation of the input and output sizes. The working principle behind the blocks is explained in the related parts. In this section, all experiments were conducted with a three-block representation, except for the configuration study about the effect of the number of blocks.

5.2.1.1 FT vs FrFT

In this section, the key claim of the thesis is covered with the CNN architectures among different datasets with the same training methodology, with three block counts covered in the previous sections to stay consistent. The primary purpose of this section is to demonstrate the positive effect of the fractional orders in the different datasets with the same CNN architecture. The strategy presented in the previous sections about the trainable fractional Fourier transform layers that produce spectrograms in fractional domains is implemented as was done in the earlier experiments.

To demonstrate the difference between the proposed approach and the classical approach based on the Fourier transform, the fractional order value is set to 1 in the fractional Fourier transform layer with no update. Therefore, the spectrogram generation is completed with the traditional Fourier transform. It should be noted that, instead of using the Fourier transform function code provided with the Python package numpy, the same code utilized in the fractional Fourier transform is employed to avoid numerical differences in the application procedure. Similarly, the commonly used STFT and spectrogram functions are not used. Therefore, directly comparing the methods provided in the literature becomes a challenging task, which will be discussed in the next chapter with a systematic analysis of the results. The overall results, including the two approaches, are provided in the Table 5.2.

Table 5.2: CNN Model FT vs. FrFT

Dataset	Accuracy (%)		F1 Score (%)		Mean Alpha Order	
	FT	FrFT	FT	FrFT	FT (Fixed=1.0)	FrFT (Trainable)
Dataset 1	89.7 $\pm$ 4.6	90.2 $\pm$ 3.6	88.6 $\pm$ 4.9	89.2 $\pm$ 4.0	1.0	1.002
Dataset 2	85.0 $\pm$ 2.9	85.3 $\pm$ 2.8	85.6 $\pm$ 2.6	85.9 $\pm$ 2.7	1.0	1.002
Dataset 3 (10 GHz)	69.2 $\pm$ 5.7	70.3 $\pm$ 5.7	67.2 $\pm$ 5.8	68.2 $\pm$ 6.1	1.0	1.004
Dataset 3 (24 GHz)	58.1 $\pm$ 2.9	59.1 $\pm$ 3.0	55.5 $\pm$ 3.1	56.6 $\pm$ 3.2	1.0	0.998
Dataset 3 (77 GHz)	68.9 $\pm$ 1.6	70.2 $\pm$ 1.5	66.6 $\pm$ 1.7	67.6 $\pm$ 1.8	1.0	0.994
Dataset 4	97.3 $\pm$ 1.8	97.6 $\pm$ 1.5	97.2 $\pm$ 1.8	97.5 $\pm$ 1.6	1.0	1.001

Considering the overall results, although the optimal CNN block counts are not implemented in this case, the trainable fractional Fourier transform approach beats the traditional Fourier transform method in each dataset. The mean accuracy and F1-score performances obtained from the trainable FrFT

layers are higher than the conventional Fourier transform by amounts from [%0.5, %0.3, %1.1, %1.0, %1.3] in terms of both test accuracy and F1-score. However, it should be highlighted that the standard deviation values in all experiments are considerably high, which will be discussed in the discussion chapter by considering similar works.

5.2.1.2 Configuration Study

The first configuration study for this architecture is searching for the effect of the block numbers. The general architecture of the model is provided in the 4.5, where a CNN block includes a "Conv-3D" layer, a "BatchNorm3D" layer, and a "MaxPool-3D" layer. In this experiment, the complexity of the model is altered with block numbers to investigate the impact on the classification success in terms of F1 scores and test accuracy values. The training methodology was provided in the previous chapter. The overall results might be seen in the Table 5.3. Also, the mean of the final FrFT order values among the 50 trials is reported.

When block numbers increase, hyperparameters related to the convolutional layers, such as kernel sizes and padding values, are fixed. However, the feature maps created by the convolutional layers are increased by a factor of two. For instance, the first block produces 8, the second block produces 16, the third block produces 32, and so on.

The overall results indicate a positive correlation between the model performances and the number of layers until the accuracy does not improve, as expected. The optimum block numbers are highlighted with bold values where the model accuracies are maximum among the examined options. The overall results indicate that, with the used structure, the optimum architecture should include 5-6 CNN blocks to classify the data accurately. Architectures including more blocks are not covered since the plateau is obtained in all cases.

As mentioned, the model hyperparameter tuning was performed using the first dataset, including the simulation data. It might be inferred that the best

Table 5.3: CNN Configuration Study: Block Counts

CNN Blocks	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
1	64.2 $\pm$ 7.2	64.0 $\pm$ 6.5	0.996	76.4 $\pm$ 3.8	75.5 $\pm$ 3.9	1.004	41.5 $\pm$ 6.2	44.0 $\pm$ 6.7	1.003
2	72.9 $\pm$ 7.3	74.6 $\pm$ 5.9	1.001	82.1 $\pm$ 3.1	81.6 $\pm$ 3.0	0.998	45.6 $\pm$ 5.3	48.1 $\pm$ 6.0	0.999
3	88.8 $\pm$ 5.3	90.0 $\pm$ 4.6	1.003	86.3 $\pm$ 2.4	85.7 $\pm$ 2.5	0.998	66.1 $\pm$ 5.7	68.5 $\pm$ 6.3	1.001
4	92.7 $\pm$ 3.7	93.2 $\pm$ 3.4	0.998	89.2 $\pm$ 2.3	88.7 $\pm$ 2.4	0.998	75.0 $\pm$ 5.0	79.7 $\pm$ 4.6	0.998
5	95.2 $\pm$ 3.5	95.6 $\pm$ 3.2	1.002	90.6 $\pm$ 2.2	90.2 $\pm$ 2.1	0.999	83.7 $\pm$ 4.5	84.8 $\pm$ 4.1	0.998
6	96.5 $\pm$ 3.3	97.0 $\pm$ 2.6	0.997	90.3 $\pm$ 1.9	89.9 $\pm$ 1.9	0.999	84.3 $\pm$ 4.7	85.3 $\pm$ 4.5	1.001
7	96.2 $\pm$ 3.3	96.5 $\pm$ 3.4	1.002	90.1 $\pm$ 2.1	89.6 $\pm$ 2.2	0.999	82.2 $\pm$ 5.5	83.3 $\pm$ 5.0	1.001

CNN Blocks	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
1	42.6 $\pm$ 6.2	46.6 $\pm$ 6.3	0.999	43.7 $\pm$ 7.0	46.9 $\pm$ 7.4	1.000	77.6 $\pm$ 4.2	78.1 $\pm$ 4.2	0.997
2	50.5 $\pm$ 6.8	54.3 $\pm$ 6.7	1.006	54.2 $\pm$ 6.6	57.0 $\pm$ 6.5	1.002	91.3 $\pm$ 2.6	91.6 $\pm$ 2.6	0.999
3	57.6 $\pm$ 5.8	60.4 $\pm$ 5.8	1.006	60.7 $\pm$ 6.4	63.3 $\pm$ 6.4	0.999	96.0 $\pm$ 2.2	96.1 $\pm$ 2.1	1.000
4	68.9 $\pm$ 6.0	70.2 $\pm$ 6.2	1.029	72.8 $\pm$ 6.6	75.3 $\pm$ 6.0	1.000	97.5 $\pm$ 1.6	97.5 $\pm$ 1.6	0.999
5	79.1 $\pm$ 6.1	80.4 $\pm$ 6.2	1.007	81.5 $\pm$ 5.7	83.2 $\pm$ 4.9	1.000	97.9 $\pm$ 1.5	98.0 $\pm$ 1.4	1.001
6	83.8 $\pm$ 4.8	84.7 $\pm$ 5.2	1.012	83.6 $\pm$ 5.4	85.6 $\pm$ 4.4	0.999	97.5 $\pm$ 1.7	97.5 $\pm$ 1.6	1.000
7	82.5 $\pm$ 5.7	83.8 $\pm$ 5.6	0.999	84.2 $\pm$ 6.5	85.9 $\pm$ 5.7	0.998	97.4 $\pm$ 2.1	97.5 $\pm$ 2.0	1.000

performance is achieved in the first dataset, naturally; however, the model can work with other datasets that are examined in a detailed way in Chapter 3. Also, if we consider the behavior of the FrFT order with different block counts, no significant changes are observed in any dataset. However, the tendency of the datasets should be highlighted where the second dataset tends to have FrFT order values less than one, and the 24 GHz version of the third dataset tends to have order values higher than one in each block number option.

The second configuration study includes searching for the CNN architecture's window and overlap sizes. As mentioned in the second chapter, in the time-frequency analysis approaches, there is a trade-off between the time and frequency contributions, which is determined by the window sizes. The effect of the window size is also examined in the literature, even in the micro-Doppler representations. Among these studies, optimum window selection is also discussed in detail. However, this work focuses on examining the effects of following the grid search methodology. Although the experiments obtain the optimum values with different block counts, a CNN structure with three blocks is employed. Different window overlap sizes are implemented in the segmentation layer, and the models are started with the same initialization to observe only the effect of the time-frequency representations. It should be highlighted that where the order values 1.000 indicates not a direct Fourier transform, which slightly differs from 1, but is

too close to the Fourier transform since the effect is very little after three digits.

Table 5.4: CNN Configuration study: STFT Window and Overlap Sizes

Window / Overlap	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
(128, 64)	89.4 $\pm$ 4.1	90.3 $\pm$ 3.7	1.001	84.2 $\pm$ 2.7	83.7 $\pm$ 2.9	1.000	65.7 $\pm$ 5.5	68.2 $\pm$ 5.2	1.001
(128, 100)	85.9 $\pm$ 5.5	86.9 $\pm$ 4.7	1.000	81.8 $\pm$ 2.7	81.3 $\pm$ 2.8	0.999	60.9 $\pm$ 5.6	63.5 $\pm$ 5.7	1.000
(128, 112)	85.6 $\pm$ 4.8	86.2 $\pm$ 4.7	1.000	81.4 $\pm$ 2.7	80.8 $\pm$ 2.7	0.999	60.0 $\pm$ 6.1	62.3 $\pm$ 5.4	1.000
(160, 120)	87.3 $\pm$ 5.1	88.1 $\pm$ 4.6	1.000	82.7 $\pm$ 3.3	82.2 $\pm$ 3.5	0.999	62.0 $\pm$ 5.1	64.8 $\pm$ 5.1	0.998
(160, 128)	86.0 $\pm$ 4.8	87.2 $\pm$ 4.4	1.000	82.3 $\pm$ 3.1	81.7 $\pm$ 3.2	1.001	62.8 $\pm$ 5.6	65.1 $\pm$ 5.9	1.000
(192, 128)	88.0 $\pm$ 4.7	89.0 $\pm$ 4.3	1.000	84.0 $\pm$ 3.1	83.5 $\pm$ 3.1	0.998	66.2 $\pm$ 5.8	68.5 $\pm$ 6.3	1.001
(192, 160)	87.3 $\pm$ 5.5	88.0 $\pm$ 5.1	0.999	82.0 $\pm$ 2.5	81.4 $\pm$ 2.7	0.999	62.0 $\pm$ 4.9	64.5 $\pm$ 5.4	1.000
(256, 128)	92.4 $\pm$ 4.0	93.0 $\pm$ 3.4	1.000	86.4 $\pm$ 2.0	85.9 $\pm$ 2.1	1.001	69.0 $\pm$ 5.4	61.2 $\pm$ 5.8	1.000
(256, 200)	87.7 $\pm$ 5.6	88.8 $\pm$ 5.1	0.998	83.9 $\pm$ 2.7	83.3 $\pm$ 2.6	1.000	64.8 $\pm$ 5.6	67.2 $\pm$ 5.5	1.000
(256, 224)	86.0 $\pm$ 4.6	87.0 $\pm$ 4.3	0.999	82.6 $\pm$ 2.6	82.0 $\pm$ 2.7	0.999	63.3 $\pm$ 5.3	65.8 $\pm$ 5.1	1.001

Window / Overlap	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
(128, 64)	64.8 $\pm$ 5.5	66.8 $\pm$ 5.4	0.999	71.0 $\pm$ 9.0	72.5 $\pm$ 9.0	0.998	95.1 $\pm$ 2.4	95.2 $\pm$ 2.3	0.998
(128, 100)	62.9 $\pm$ 6.9	64.7 $\pm$ 7.1	1.001	62.5 $\pm$ 11.	64.4 $\pm$ 11.	0.997	95.0 $\pm$ 2.3	95.1 $\pm$ 2.2	0.996
(128, 112)	60.5 $\pm$ 5.6	62.6 $\pm$ 6.1	0.999	62.4 $\pm$ 10.	64.5 $\pm$ 9.0	0.998	93.5 $\pm$ 3.9	93.6 $\pm$ 3.8	0.999
(160, 120)	63.2 $\pm$ 5.0	65.4 $\pm$ 5.1	1.000	63.6 $\pm$ 11.	65.5 $\pm$ 11.	1.002	94.7 $\pm$ 2.5	94.8 $\pm$ 2.5	0.998
(160, 128)	62.0 $\pm$ 6.2	63.7 $\pm$ 5.7	0.999	63.4 $\pm$ 10.	65.0 $\pm$ 10.	0.999	94.4 $\pm$ 2.4	94.5 $\pm$ 2.3	0.999
(192, 128)	64.0 $\pm$ 6.0	65.9 $\pm$ 6.5	1.001	62.8 $\pm$ 9.0	65.4 $\pm$ 8.0	0.999	94.0 $\pm$ 2.7	94.2 $\pm$ 2.7	0.999
(192, 160)	60.6 $\pm$ 6.1	63.1 $\pm$ 5.9	1.001	63.2 $\pm$ 9.0	65.8 $\pm$ 9.0	0.999	94.0 $\pm$ 2.4	94.2 $\pm$ 2.3	0.997
(256, 128)	69.4 $\pm$ 5.8	71.0 $\pm$ 5.7	0.999	64.8 $\pm$ 9.0	67.3 $\pm$ 9.0	0.997	85.0 $\pm$ 4.4	85.4 $\pm$ 4.3	0.996
(256, 200)	63.5 $\pm$ 5.8	65.8 $\pm$ 6.0	1.001	64.3 $\pm$ 8.0	66.5 $\pm$ 8.0	0.999	84.4 $\pm$ 4.6	84.7 $\pm$ 4.6	0.997
(256, 224)	61.8 $\pm$ 6.1	64.0 $\pm$ 6.1	1.001	59.1 $\pm$ 12.	62.1 $\pm$ 11.	0.999	85.4 $\pm$ 3.7	85.8 $\pm$ 3.7	0.997

Considering the overall results as indicated in the Table 5.4, in each scenario, it might be claimed that the overall behavior is the same for all datasets. The model performance increases with the window size, and the overlap length negatively influences the model performance. Also, the FrFT orders are too close to one, and the Fourier transform does not fluctuate in different scenarios. Therefore, in the CNN architecture, no clear relation exists between the window size and the fractional representation in any dataset.

5.2.1.3 Augmentation Methods

Similar to the configuration studies, two augmentation strategies are implemented in the scope of the thesis, the theoretical background of which is explained in the second chapter. The first augmentation method explained is Mix-up augmentation, and the second is warping augmentation. In each scenario, the CNN architectures, including three blocks, are examined with the window-overlap sizes equal to 128 and 100.

The first augmentation strategy is Mix-up augmentation, whose results are shared in the Table 5.5. As mentioned in the theoretical explanation section, the beta distribution is employed with different parameters in the original work. In light of the original study, we have also examined the effect of the different beta distributions where $Beta(\beta, \beta)$, the boundaries are selected as symmetric values. The impact of the different boundary β values is examined.

Table 5.5: CNN Augmentation Method: Mixup Augmentation

Mixup β	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.0	89.3 $\pm$ 5.0	90.5 $\pm$ 4.2	1.000	86.2 $\pm$ 3.3	85.5 $\pm$ 3.5	1.000	67.3 $\pm$ 7.	69.2 $\pm$ 6.	1.000
0.2	86.3 $\pm$ 6.5	87.6 $\pm$ 5.8	0.999	86.0 $\pm$ 2.6	85.5 $\pm$ 2.8	0.999	66.6 $\pm$ 11.	68.6 $\pm$ 10.	0.997
0.4	86.4 $\pm$ 5.8	87.8 $\pm$ 4.8	1.001	85.8 $\pm$ 3.2	85.3 $\pm$ 3.2	0.999	63.8 $\pm$ 12.	65.8 $\pm$ 11.	0.999
0.6	85.9 $\pm$ 5.8	86.8 $\pm$ 5.0	1.000	85.1 $\pm$ 2.5	84.6 $\pm$ 2.4	0.999	63.3 $\pm$ 11.	65.4 $\pm$ 10.	0.997
0.8	84.8 $\pm$ 6.4	85.9 $\pm$ 6.0	1.000	84.0 $\pm$ 3.6	83.6 $\pm$ 3.5	0.999	64.6 $\pm$ 9.0	66.5 $\pm$ 8.0	1.002
1.0	84.0 $\pm$ 6.6	85.0 $\pm$ 6.4	0.998	84.1 $\pm$ 3.4	83.7 $\pm$ 3.6	1.000	60.7 $\pm$ 11.	63.5 $\pm$ 10.	0.999
1.2	84.4 $\pm$ 5.2	85.7 $\pm$ 4.9	1.000	84.1 $\pm$ 3.1	83.9 $\pm$ 3.2	0.999	59.8 $\pm$ 13.	62.4 $\pm$ 12.	0.998
1.4	83.4 $\pm$ 6.9	84.7 $\pm$ 6.1	0.997	83.1 $\pm$ 3.1	82.9 $\pm$ 3.2	0.999	58.7 $\pm$ 15.	61.3 $\pm$ 13.	1.001
1.6	83.1 $\pm$ 6.5	84.7 $\pm$ 6.2	1.000	83.8 $\pm$ 3.6	83.5 $\pm$ 3.6	0.998	57.9 $\pm$ 17.	60.5 $\pm$ 16.	0.999
1.8	82.4 $\pm$ 6.6	83.7 $\pm$ 5.9	1.000	83.1 $\pm$ 3.5	82.9 $\pm$ 3.3	0.999	56.4 $\pm$ 17.	59.2 $\pm$ 15.	1.002
2.0	82.0 $\pm$ 6.9	83.6 $\pm$ 6.1	0.997	83.3 $\pm$ 3.3	83.0 $\pm$ 3.3	0.998	58.2 $\pm$ 14.	60.9 $\pm$ 13.	0.997

Mixup β	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.0	55.5 $\pm$ 3.2	58.0 $\pm$ 3.0	1.000	65.6 $\pm$ 19.	68.5 $\pm$ 1.7	1.000	96.9 $\pm$ 2.2	97.0 $\pm$ 2.1	1.001
0.2	50.0 $\pm$ 3.3	53.1 $\pm$ 3.1	1.000	25.0 $\pm$ 2.9	30.9 $\pm$ 2.7	1.000	97.2 $\pm$ 1.8	97.3 $\pm$ 1.7	0.997
0.4	48.4 $\pm$ 3.3	51.7 $\pm$ 3.1	0.998	26.7 $\pm$ 2.9	33.1 $\pm$ 2.7	0.997	97.1 $\pm$ 1.8	97.2 $\pm$ 1.8	1.000
0.6	47.6 $\pm$ 3.1	51.2 $\pm$ 3.0	0.999	26.3 $\pm$ 2.9	32.3 $\pm$ 2.7	0.998	97.4 $\pm$ 1.8	97.5 $\pm$ 1.7	1.000
0.8	46.4 $\pm$ 3.2	49.9 $\pm$ 3.0	0.996	26.4 $\pm$ 2.9	33.0 $\pm$ 2.7	0.998	97.1 $\pm$ 1.6	97.2 $\pm$ 1.6	0.997
1.0	41.5 $\pm$ 3.4	45.8 $\pm$ 3.1	1.000	29.2 $\pm$ 3.0	35.0 $\pm$ 2.8	0.999	97.0 $\pm$ 1.8	97.1 $\pm$ 1.8	0.997
1.2	38.7 $\pm$ 3.4	42.8 $\pm$ 3.2	1.001	25.5 $\pm$ 2.9	31.6 $\pm$ 2.8	0.996	96.7 $\pm$ 1.8	96.8 $\pm$ 1.8	0.999
1.4	33.4 $\pm$ 3.3	38.3 $\pm$ 3.1	1.001	26.6 $\pm$ 2.8	32.8 $\pm$ 2.7	0.997	96.9 $\pm$ 1.7	96.9 $\pm$ 1.7	0.997
1.6	30.1 $\pm$ 3.5	34.6 $\pm$ 3.3	0.998	28.7 $\pm$ 2.9	34.2 $\pm$ 2.8	0.997	96.8 $\pm$ 1.8	96.9 $\pm$ 1.8	0.997
1.8	30.6 $\pm$ 3.1	36.2 $\pm$ 3.2	0.996	23.5 $\pm$ 2.7	29.8 $\pm$ 2.6	0.998	96.7 $\pm$ 2.3	96.8 $\pm$ 2.2	0.997
2.0	28.7 $\pm$ 3.4	33.2 $\pm$ 3.4	1.002	24.2 $\pm$ 2.6	31.0 $\pm$ 2.6	0.996	96.3 $\pm$ 1.9	96.4 $\pm$ 1.9	0.997

In the overall results, the Mixup augmentation strategy does not provide any improvement in any cases. Although the improvement is not acquired in the original study in any cases, mostly the values between the $\beta \in [0.2, 0.4]$ cause improvements. In our case, no progress has been achieved with the CNN model architecture. However, the adverse effect of the larger bounds, which cause underfitting, is obtained clearly in all scenarios examined, including all datasets. Also, it is not observed that a clear relation exists between the fractional order and the augmentation strategy. The only thing that might be highlighted about the fractional orders is that the tendency of the second dataset for lower orders than one is still valid in this experiment.

The different training methodologies, which were conducted in the original work, might be acquired in this augmentation method; however, since the main aim is to compare the model performances and datasets in various architectures, the same training strategy is followed in all experiments, which was discussed in a detailed way in the previous section.

The second augmentation method is also explained, and the warping methodology includes warping in the range-time images in both range-time dimensions, including random shrinking and stretching. The mathematical representation is provided in the second chapter, like the previous augmentation strategy. As explained in the related section, the scaling factor is chosen in the $[0.8, 1.2]$ range and applied in range and time directions separately with the combined version. The aforementioned physical interpretation of the operation might be expressed as manipulating the targets' speed and the movement through complete movement. Also, the training methodology and model architecture are constructed as provided in the previous augmentation method.

Table 5.6: CNN Augmentation Method: Stretching

Stretch Dimension	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
Fast-Time	84.7 $\pm$ 6.5	85.9 $\pm$ 5.7	1.001	87.2 $\pm$ 3.1	86.5 $\pm$ 3.2	0.997	68.6 $\pm$ 5.4	72.0 $\pm$ 5.8	0.996
Slow-Time	84.7 $\pm$ 6.5	88.3 $\pm$ 4.1	1.003	88.3 $\pm$ 2.9	87.7 $\pm$ 2.9	0.999	69.4 $\pm$ 6.5	73.0 $\pm$ 6.5	1.001
Both	80.6 $\pm$ 7.3	82.0 $\pm$ 6.5	0.998	87.9 $\pm$ 2.3	87.4 $\pm$ 2.5	0.999	65.2 $\pm$ 4.5	69.7 $\pm$ 5.2	0.998
None	89.5 $\pm$ 4.7	90.5 $\pm$ 4.8	1.000	85.9 $\pm$ 2.9	85.1 $\pm$ 3.0	1.001	68.3 $\pm$ 5.8	70.8 $\pm$ 6.4	0.996

Stretch Dimension	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
Fast-Time	54.3 $\pm$ 6.9	56.9 $\pm$ 7.0	0.998	48.9 $\pm$ 3.0	53.1 $\pm$ 2.8	0.997	95.4 $\pm$ 2.2	95.6 $\pm$ 2.1	0.997
Slow-Time	54.4 $\pm$ 6.8	57.5 $\pm$ 6.7	1.000	52.9 $\pm$ 2.8	57.0 $\pm$ 2.7	0.998	97.0 $\pm$ 1.8	97.1 $\pm$ 1.8	0.995
Both	51.0 $\pm$ 7.3	54.5 $\pm$ 7.0	1.001	33.3 $\pm$ 3.2	39.0 $\pm$ 3.0	0.994	93.3 $\pm$ 3.3	93.4 $\pm$ 3.3	0.998
None	56.7 $\pm$ 5.5	59.0 $\pm$ 5.4	1.003	67.8 $\pm$ 1.8	70.4 $\pm$ 1.7	0.999	97.5 $\pm$ 1.9	97.6 $\pm$ 1.8	1.000

In all cases, as shown in the examined literature, some increments are obtained by applying the same augmentation task, but no improvement is obtained in the second dataset. Therefore, the proposed augmentation method might be helpful in some cases. The main difference of the second dataset is having more samples in the class sets, which might be interpreted similarly.

5.2.1.4 Initial FrFT Order Effect

The fractional Fourier transform and its relation to other time-frequency transform methods are explained in the second chapter with a theoretical background. For example, the ambiguity function, Wigner-Ville distribution, and Fourier transform might be stated as functions related to the fractional Fourier transform examined in the literature. Different representations are also offered since the FrFT presents a continuum analysis of the entire time-frequency plane.

In the other experiments, including the CNN architecture, the initial value of the fractional order is set to 1, to simulate starting from the conventional Fourier transform. Since the employed operation allows us to analyze different transformations, this experiment is conducted by changing the initial fractional order from 0.1 to 2.0 by 0.1 steps, and the results are reported. The values higher than 2.0 are not covered since the FrFT operation is periodic with order 2.0 as demonstrated in the second chapter. Therefore, it is aimed to analyze models learning the order starting from different transformations. The results are reported in the Table 5.7.

The results indicate that, in all cases, the initial FrFT order with the value of 1 leads to higher performance in the CNN architecture. In addition, as shown in the other experiments of CNN architecture, the fractional order tends to fluctuate around 1, demonstrating that better fractional representations are obtained near the conventional Fourier transform. In addition, a positive correlation exists between the fractional orders near one and the model performance. In other words, the model's performance degrades as the initial value moves away from 1. Even the shape of the degradation might be visualized as a simple triangular shape for the first 2 datasets concerning the different fractional orders, as given in Figure 5.1, as F1-score values. Therefore, the conventional Fourier transform is preferable to other transforms in this architecture.

It should also be highlighted that the difference between the mean final alpha order value and the initial value is higher when the initial value is away from 1. Therefore, the models learn alpha values different from the initial points,

Table 5.7: CNN Model Initial FrFT Order

Initial α	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.1	78.8 $\pm$ 5.8	80.4 $\pm$ 5.2	0.112	74.6 $\pm$ 3.7	73.9 $\pm$ 3.7	0.158	50.4 $\pm$ 7.0	53.0 $\pm$ 6.2	0.106
0.2	80.4 $\pm$ 6.4	81.5 $\pm$ 6.5	0.230	77.6 $\pm$ 3.9	76.9 $\pm$ 4.1	0.262	54.3 $\pm$ 6.9	56.4 $\pm$ 6.6	0.219
0.4	84.0 $\pm$ 7.0	85.3 $\pm$ 5.9	0.424	81.3 $\pm$ 3.4	80.6 $\pm$ 3.5	0.426	57.8 $\pm$ 10.	60.0 $\pm$ 9.6	0.414
0.6	83.9 $\pm$ 7.6	84.9 $\pm$ 7.5	0.606	82.2 $\pm$ 6.5	81.8 $\pm$ 5.5	0.594	59.8 $\pm$ 8.7	62.7 $\pm$ 7.4	0.602
0.8	84.2 $\pm$ 12.	85.9 $\pm$ 12.	0.808	84.4 $\pm$ 3.0	83.9 $\pm$ 3.2	0.789	63.9 $\pm$ 8.6	65.7 $\pm$ 7.6	0.793
1.0	88.9 $\pm$ 5.6	89.9 $\pm$ 4.9	1.005	85.0 $\pm$ 3.6	84.5 $\pm$ 3.8	0.995	67.5 $\pm$ 9.2	69.8 $\pm$ 8.0	1.001
1.2	84.8 $\pm$ 12.	86.0 $\pm$ 12.	1.196	83.6 $\pm$ 3.1	83.0 $\pm$ 3.3	1.209	63.1 $\pm$ 9.0	65.5 $\pm$ 8.1	1.204
1.4	86.7 $\pm$ 4.5	87.5 $\pm$ 4.2	1.393	83.3 $\pm$ 3.9	82.7 $\pm$ 4.1	1.409	60.1 $\pm$ 10.	62.2 $\pm$ 9.4	1.402
1.6	83.0 $\pm$ 7.3	84.3 $\pm$ 6.5	1.581	80.6 $\pm$ 3.3	80.0 $\pm$ 3.4	1.574	57.0 $\pm$ 7.9	59.7 $\pm$ 7.1	1.586
1.8	80.5 $\pm$ 6.1	82.0 $\pm$ 5.5	1.773	77.4 $\pm$ 3.5	76.7 $\pm$ 3.9	1.733	51.4 $\pm$ 6.6	54.3 $\pm$ 5.9	1.787
2.0	78.0 $\pm$ 7.7	79.8 $\pm$ 6.7	1.986	71.5 $\pm$ 3.8	70.1 $\pm$ 3.8	1.978	48.9 $\pm$ 6.7	51.7 $\pm$ 6.0	1.994

Initial α	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.1	58.3 $\pm$ 2.5	60.4 $\pm$ 2.4	0.100	58.7 $\pm$ 2.2	62.2 $\pm$ 1.9	0.112	76.7 $\pm$ 4.7	77.1 $\pm$ 4.8	0.111
0.2	61.2 $\pm$ 2.1	63.1 $\pm$ 2.0	0.201	55.6 $\pm$ 2.3	58.9 $\pm$ 2.1	0.212	82.8 $\pm$ 4.5	83.3 $\pm$ 3.4	0.226
0.4	62.6 $\pm$ 2.5	64.9 $\pm$ 2.3	0.399	57.9 $\pm$ 2.7	60.9 $\pm$ 2.4	0.407	88.0 $\pm$ 3.4	88.4 $\pm$ 2.9	0.404
0.6	61.5 $\pm$ 2.7	63.9 $\pm$ 2.4	0.598	60.4 $\pm$ 2.4	63.2 $\pm$ 2.3	0.602	92.4 $\pm$ 2.9	92.7 $\pm$ 2.2	0.625
0.8	62.8 $\pm$ 2.6	64.9 $\pm$ 2.5	0.793	65.0 $\pm$ 1.9	67.6 $\pm$ 1.8	0.800	95.3 $\pm$ 2.4	95.5 $\pm$ 1.8	0.826
1.0	61.9 $\pm$ 2.9	64.0 $\pm$ 2.7	0.998	67.6 $\pm$ 1.8	70.1 $\pm$ 1.6	0.999	97.5 $\pm$ 1.9	97.6 $\pm$ 2.2	1.002
1.2	61.3 $\pm$ 2.5	63.2 $\pm$ 2.4	1.195	62.5 $\pm$ 2.5	65.4 $\pm$ 2.4	1.201	95.7 $\pm$ 2.2	95.7 $\pm$ 2.1	1.162
1.4	60.6 $\pm$ 2.4	62.5 $\pm$ 2.3	1.394	59.9 $\pm$ 2.6	62.7 $\pm$ 2.4	1.398	93.0 $\pm$ 2.2	93.2 $\pm$ 3.7	1.378
1.6	60.2 $\pm$ 2.8	62.1 $\pm$ 2.6	1.580	63.6 $\pm$ 2.1	65.8 $\pm$ 2.0	1.590	88.1 $\pm$ 3.8	88.4 $\pm$ 3.8	1.583
1.8	59.4 $\pm$ 2.3	61.5 $\pm$ 2.1	1.785	54.7 $\pm$ 2.6	58.5 $\pm$ 2.3	1.791	83.3 $\pm$ 3.6	83.8 $\pm$ 3.5	1.768
2.0	55.2 $\pm$ 2.6	57.4 $\pm$ 2.3	1.992	52.1 $\pm$ 2.7	55.8 $\pm$ 2.5	2.001	75.6 $\pm$ 5.0	76.2 $\pm$ 5.0	1.990

especially in the values close to boundaries in the examined initial point set. These differences become lower as the initial order approaches 1. As described in the second chapter, the fractional order determines the contribution of the time and frequency components in the transform. Therefore, the transforms tend to be more suitable as they approach the conventional Fourier transform.

Another key observation about the final fractional orders should be highlighted, although some results violate the pattern. The mean of the final fractional order values tends to 1 in most cases. For example, in the first dataset, the final fractional order values are closer to 1 compared to their initial points, which also occurs in other datasets. However, in any case, the final order value cannot reach 1, which may be considered the existence of local minimums in the examined set, if we consider the final fractional orders and their performances together.

The results presented and discussed in this section are stated again in the following chapter by considering the key observations across other NN architectures

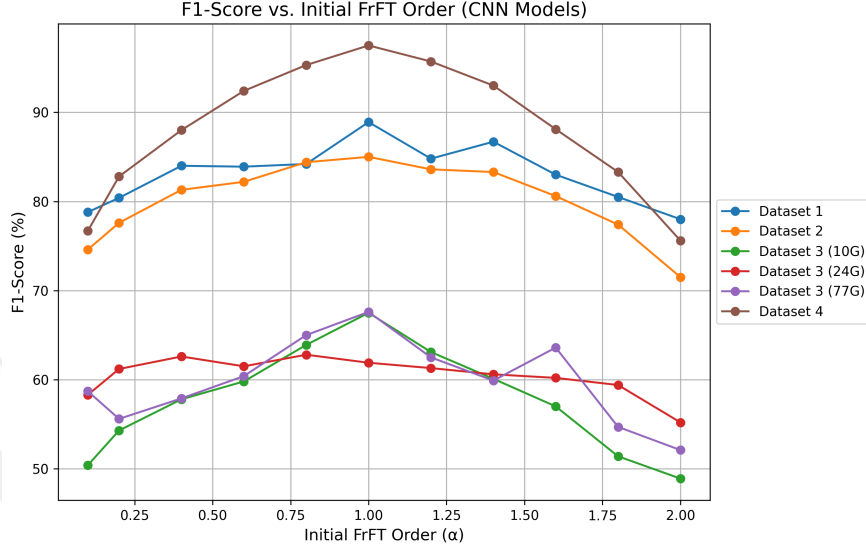


Figure 5.1: Model Performance of CNN Architecture vs Initial Order

in a more detailed way. The key observations common across the architectures and specific to this architecture are analyzed thoroughly.

5.2.2 LSTM-Based Model

In the fourth chapter, RNN models are described and explained under the same title since the only difference in these models is the selection of the core block, which is theoretically explained in the second chapter. However, experimental results are examined in different sub-sections since their performances differ. As the title highlights, the first RNN architecture investigated in this thesis is LSTM-based. The same experiments conducted in the CNN architecture are performed with this architecture.

5.2.2.1 FT vs FRFT

The key of this thesis is also examined with the LSTM-based architecture as conducted in the CNN-based structure. As described several times before, the

three-block architecture is utilized again. The adaptive generation of the spectrogram images based on the fractional Fourier transform is used with the same training and testing approach.

As described in the CNN-based architecture experiment about the conventional Fourier transform and fractional Fourier transform, the traditional Fourier transform method is applied to evaluate the model with a constant fractional order, where the trainable case has a trainable fractional order whose initial point is chosen as 1. As given in the Table 5.8, the proposed approach achieves higher performances in each scenario than traditional Fourier transform usage. The explanations given in the CNN-based approach may be considered and used for this architecture to avoid repetitions.

Table 5.8: LSTM Model FT vs. FrFT

Dataset	Accuracy (%)		F1 Score (%)		Mean Alpha Order	
	FT	FrFT	FT	FrFT	FT (Fixed=1.0)	FrFT (Trainable)
Dataset 1	93.7 $\pm$ 3.9	94.7 $\pm$ 2.3	93.4 $\pm$ 4.1	94.3 $\pm$ 2.7	1.0	0.985
Dataset 2	86.7 $\pm$ 2.4	86.9 $\pm$ 2.2	87.3 $\pm$ 2.2	87.5 $\pm$ 2.1	1.0	0.994
Dataset 3 (10 GHz)	72.9 $\pm$ 6.7	74.2 $\pm$ 5.7	71.3 $\pm$ 7.2	72.0 $\pm$ 6.0	1.0	0.996
Dataset 3 (24 GHz)	81.3 $\pm$ 5.9	81.5 $\pm$ 5.7	80.5 $\pm$ 6.4	80.6 $\pm$ 6.5	1.0	0.993
Dataset 3 (77 GHz)	75.5 $\pm$ 6.2	76.9 $\pm$ 6.7	73.1 $\pm$ 7.0	74.4 $\pm$ 7.1	1.0	0.997
Dataset 4	92.4 $\pm$ 1.7	93.7 $\pm$ 1.9	92.3 $\pm$ 1.7	93.6 $\pm$ 2.0	1.0	0.998

Similar to the CNN case, the improvements provided by the proposed approach deserve to be highlighted separately. The improvement amounts obtained in this architecture might be listed as [1.0, 0.3, 1.3, 0.2, 1.4, 1.3] as test accuracy values and [0.9, 0.2, 0.7, 0.1, 1.3, 1.3] as F1-score values. The correlation between the accuracy and F1-score continues, where the total improvements are similar to those obtained in the CNN-based architectures. In addition, the fractional order tendency is still preserved in this experiment, as observed in other experiments that are presented in the following sections of the LSTM-based architecture.

5.2.2.2 Configuration Study

Similar to the CNN architecture, the first configuration study for this architecture includes investigating the block counts in the LSTM architecture. As stated in the fourth chapter, the employed LSTM blocks are in the form of Bi-LSTM blocks.

The overall results among the six datasets are provided in the Table 5.9, where the block numbers change from 1 to 5. The block parameters stay the same for all blocks added to the architecture. Therefore, the visual representation provided in the Figure 4.6 with the three blocks is reconstructed with a different number of blocks without changing any other parameter. Also, it should be stated again that the training and testing procedure is employed.

Table 5.9: LSTM Configuration Study: Block Counts

LSTM Blocks	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
1	94.5 $\pm$ 3.7	95.0 $\pm$ 3.5	0.993	87.5 $\pm$ 2.6	87.0 $\pm$ 2.7	0.995	76.6 $\pm$ 4.8	78.0 $\pm$ 4.8	1.000
2	95.2 $\pm$ 2.8	95.6 $\pm$ 2.6	0.995	87.4 $\pm$ 2.4	86.9 $\pm$ 2.4	0.995	73.6 $\pm$ 4.7	75.5 $\pm$ 4.7	1.000
3	94.3 $\pm$ 2.7	94.7 $\pm$ 2.3	0.985	87.0 $\pm$ 2.3	86.4 $\pm$ 2.5	0.995	71.0 $\pm$ 6.7	73.2 $\pm$ 6.1	0.997
4	93.3 $\pm$ 3.5	93.9 $\pm$ 3.1	0.988	86.7 $\pm$ 2.5	86.2 $\pm$ 2.8	0.993	68.4 $\pm$ 6.9	70.4 $\pm$ 7.3	1.001
5	91.8 $\pm$ 5.3	92.5 $\pm$ 5.0	0.991	85.8 $\pm$ 2.6	85.2 $\pm$ 2.7	0.995	66.8 $\pm$ 6.5	69.0 $\pm$ 6.8	0.998

LSTM Blocks	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
1	82.3 $\pm$ 4.8	83.7 $\pm$ 4.3	0.998	79.7 $\pm$ 5.6	81.8 $\pm$ 5.2	1.005	96.1 $\pm$ 2.1	96.3 $\pm$ 2.0	0.998
2	81.9 $\pm$ 5.8	82.8 $\pm$ 5.8	0.995	77.4 $\pm$ 5.6	79.7 $\pm$ 5.5	1.002	95.4 $\pm$ 1.9	95.6 $\pm$ 1.8	0.997
3	79.8 $\pm$ 5.5	80.8 $\pm$ 4.9	0.997	74.8 $\pm$ 6.5	77.4 $\pm$ 6.1	0.999	93.4 $\pm$ 3.0	93.6 $\pm$ 2.9	1.002
4	77.9 $\pm$ 5.4	79.1 $\pm$ 5.4	0.995	73.6 $\pm$ 7.6	75.8 $\pm$ 7.3	1.006	92.1 $\pm$ 2.8	92.3 $\pm$ 2.7	0.998
5	73.1 $\pm$ 5.8	75.0 $\pm$ 5.5	0.994	70.6 $\pm$ 7.5	72.9 $\pm$ 7.6	1.001	89.2 $\pm$ 4.0	89.6 $\pm$ 3.7	0.996

The overall results indicate that the best performances are obtained mainly in the configurations including 1 or 2 blocks. In addition, the increasing model complexity brings a degradation in performance. This situation might be explained by considering the number of samples among the classes, where the explained datasets are relatively small compared to other well-known image datasets in the literature. In addition, the mean of the final fractional orders tends to values lower than 1, which differs from the CNN architecture. The only exception is the 77 GHz radar data of the third dataset, which is higher than 1. Therefore, there is no correlation between the learned fractional order and the model complexity in the LSTM-based architecture.

As presented for the CNN-based architecture, the effect of the window-overlap sizes is also examined for this architecture to investigate the impact of the trade-off between the time-frequency contributions in the windowed transforms, whose theoretical background is provided in the second chapter. This structure is also subjected to the same grid search methodology employed in the CNN architecture; thus, theoretical explanations provided before are not given to avoid repetition.

It should be stated that the experiments in this one and the others in this section are conducted using an architecture that has three blocks.

Table 5.10: LSTM Configuration study: STFT Window and Overlap Sizes

Window / Overlap	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
(128, 64)	93.4 $\pm$ 3.9	93.9 $\pm$ 3.4	0.990	86.5 $\pm$ 2.5	86.0 $\pm$ 2.7	0.996	71.8 $\pm$ 4.9	73.5 $\pm$ 5.2	0.996
(128, 100)	94.3 $\pm$ 2.7	94.7 $\pm$ 2.3	0.985	87.0 $\pm$ 2.4	86.3 $\pm$ 2.5	0.995	70.9 $\pm$ 6.6	73.1 $\pm$ 6.1	0.997
(128, 112)	94.1 $\pm$ 3.5	94.7 $\pm$ 3.0	0.987	86.3 $\pm$ 2.3	85.7 $\pm$ 2.4	0.992	70.0 $\pm$ 5.9	72.0 $\pm$ 5.5	0.995
(160, 120)	93.4 $\pm$ 5.0	94.0 $\pm$ 4.2	0.988	87.1 $\pm$ 1.9	86.6 $\pm$ 1.9	0.991	69.5 $\pm$ 6.1	71.7 $\pm$ 6.5	1.003
(160, 128)	93.0 $\pm$ 4.5	93.5 $\pm$ 4.0	0.985	87.3 $\pm$ 2.0	86.7 $\pm$ 2.1	0.995	70.8 $\pm$ 6.4	73.1 $\pm$ 6.5	0.997
(192, 128)	93.2 $\pm$ 4.7	93.8 $\pm$ 4.3	0.995	87.0 $\pm$ 2.6	86.3 $\pm$ 2.8	0.994	71.6 $\pm$ 6.3	73.3 $\pm$ 6.7	0.994
(192, 160)	94.6 $\pm$ 3.8	95.1 $\pm$ 3.4	0.988	86.8 $\pm$ 2.3	86.3 $\pm$ 2.5	0.994	71.9 $\pm$ 6.1	73.6 $\pm$ 5.9	1.003
(256, 128)	93.5 $\pm$ 4.1	94.1 $\pm$ 3.4	0.985	86.4 $\pm$ 2.9	85.8 $\pm$ 2.9	1.002	70.8 $\pm$ 5.9	72.8 $\pm$ 5.9	1.004
(256, 200)	93.9 $\pm$ 4.6	94.5 $\pm$ 3.9	0.990	86.6 $\pm$ 2.7	86.2 $\pm$ 2.6	0.998	73.4 $\pm$ 5.3	74.9 $\pm$ 5.4	1.002
(256, 224)	94.1 $\pm$ 4.3	94.6 $\pm$ 3.9	0.985	87.3 $\pm$ 2.2	86.7 $\pm$ 2.3	0.994	72.4 $\pm$ 5.5	74.0 $\pm$ 5.8	0.998

Window / Overlap	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
(128, 64)	80.5 $\pm$ 6.1	81.6 $\pm$ 6.1	1.000	75.1 $\pm$ 6.4	77.7 $\pm$ 5.6	1.005	92.9 $\pm$ 3.1	93.0 $\pm$ 3.0	1.000
(128, 100)	79.8 $\pm$ 5.5	80.8 $\pm$ 4.9	0.997	74.8 $\pm$ 6.5	77.3 $\pm$ 6.1	0.999	93.4 $\pm$ 3.0	93.6 $\pm$ 2.9	1.002
(128, 112)	78.2 $\pm$ 6.2	79.5 $\pm$ 5.4	0.993	73.2 $\pm$ 8.5	76.1 $\pm$ 7.8	1.007	94.2 $\pm$ 2.6	94.3 $\pm$ 2.6	1.001
(160, 120)	80.1 $\pm$ 5.1	81.3 $\pm$ 4.9	0.999	74.9 $\pm$ 6.2	77.3 $\pm$ 6.0	1.012	92.1 $\pm$ 2.8	92.3 $\pm$ 2.8	0.999
(160, 128)	79.0 $\pm$ 6.0	80.3 $\pm$ 5.9	1.007	74.2 $\pm$ 5.6	76.6 $\pm$ 5.7	1.004	93.0 $\pm$ 3.0	93.1 $\pm$ 2.8	0.995
(192, 128)	81.5 $\pm$ 6.7	82.5 $\pm$ 6.0	1.000	75.5 $\pm$ 6.1	77.5 $\pm$ 5.9	1.004	91.8 $\pm$ 3.1	92.0 $\pm$ 3.0	0.999
(192, 160)	80.2 $\pm$ 5.7	81.1 $\pm$ 5.2	1.002	75.2 $\pm$ 5.7	77.6 $\pm$ 5.6	1.005	91.2 $\pm$ 3.5	91.4 $\pm$ 3.4	1.005
(256, 128)	80.4 $\pm$ 5.1	81.5 $\pm$ 4.9	0.998	75.7 $\pm$ 4.3	78.1 $\pm$ 4.1	0.996	87.9 $\pm$ 3.7	88.2 $\pm$ 3.9	1.000
(256, 200)	81.0 $\pm$ 5.6	82.0 $\pm$ 5.5	0.996	74.2 $\pm$ 6.9	76.8 $\pm$ 6.5	0.999	87.9 $\pm$ 3.7	88.2 $\pm$ 3.9	1.000
(256, 224)	81.0 $\pm$ 5.8	82.2 $\pm$ 4.7	0.992	73.8 $\pm$ 5.6	76.2 $\pm$ 5.2	0.996	87.9 $\pm$ 3.7	88.2 $\pm$ 3.9	1.000

In contrast to the experiment conducted in the CNN architecture, window-overlap sizes do not demonstrate any convergence to one configuration. This situation is discussed in the next chapter.

5.2.2.3 Augmentation Methods

This section also applies the Mix-up augmentation method to the LSTM-based architecture, as completed in the CNN architecture. To stay consistent with the approaches examined in the scope of this study, the same parametric analysis is applied; thus, no further explanation is provided in this part. The overall results are presented in the Table 5.11.

As shown in the second chapter, the higher accuracy improvements are mainly obtained with the beta distribution values in the interval of [0.2,0.4] with the well-known datasets. However, in the configuration studies shared in the original

Table 5.11: LSTM Augmentation Method: Mixup Augmentation

Mixup β	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.0	94.3 $\pm$ 2.7	94.7 $\pm$ 2.3	0.985	87.3 $\pm$ 2.5	86.8 $\pm$ 2.5	0.995	71.8 $\pm$ 6.4	73.7 $\pm$ 6.5	0.999
0.2	93.6 $\pm$ 5.4	94.2 $\pm$ 4.6	0.987	88.1 $\pm$ 1.9	87.6 $\pm$ 2.0	0.997	58.2 $\pm$ 8.2	61.4 $\pm$ 8.1	1.006
0.4	93.7 $\pm$ 4.5	94.1 $\pm$ 4.2	0.985	88.5 $\pm$ 2.4	88.0 $\pm$ 2.5	0.997	57.2 $\pm$ 8.2	60.2 $\pm$ 8.7	1.007
0.6	91.8 $\pm$ 5.9	92.4 $\pm$ 5.4	0.991	87.9 $\pm$ 2.5	87.4 $\pm$ 2.6	1.006	56.6 $\pm$ 8.0	59.8 $\pm$ 8.4	1.009
0.8	92.0 $\pm$ 4.3	92.5 $\pm$ 4.0	0.978	87.7 $\pm$ 2.6	87.2 $\pm$ 2.7	1.005	58.0 $\pm$ 6.9	61.1 $\pm$ 6.8	1.007
1.0	89.9 $\pm$ 6.3	90.9 $\pm$ 5.4	0.980	88.1 $\pm$ 2.4	87.6 $\pm$ 2.5	0.998	57.1 $\pm$ 7.5	59.9 $\pm$ 7.6	1.007
1.2	89.7 $\pm$ 5.5	90.5 $\pm$ 4.6	0.985	87.6 $\pm$ 2.6	87.0 $\pm$ 2.7	0.999	56.7 $\pm$ 6.4	60.0 $\pm$ 6.6	1.008
1.4	88.6 $\pm$ 4.9	89.5 $\pm$ 4.1	0.981	87.5 $\pm$ 2.5	87.0 $\pm$ 2.6	0.999	56.5 $\pm$ 7.4	59.4 $\pm$ 7.5	1.005
1.6	87.2 $\pm$ 5.9	88.5 $\pm$ 5.1	0.978	86.8 $\pm$ 2.6	86.3 $\pm$ 2.7	0.996	57.6 $\pm$ 6.8	60.8 $\pm$ 6.7	1.009
1.8	86.7 $\pm$ 5.9	88.0 $\pm$ 5.3	0.987	87.3 $\pm$ 2.8	86.8 $\pm$ 2.9	0.997	55.5 $\pm$ 6.4	58.8 $\pm$ 6.6	1.007
2.0	85.2 $\pm$ 8.5	86.6 $\pm$ 7.0	0.983	87.6 $\pm$ 2.5	87.2 $\pm$ 2.6	0.997	57.0 $\pm$ 7.4	59.8 $\pm$ 7.3	1.010

Mixup β	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.0	80.6 $\pm$ 6.5	81.5 $\pm$ 5.7	0.993	74.3 $\pm$ 6.9	76.8 $\pm$ 6.7	1.005	89.9 $\pm$ 3.3	90.3 $\pm$ 3.2	0.999
0.2	78.0 $\pm$ 6.7	79.4 $\pm$ 5.9	0.998	65.7 $\pm$ 6.5	69.0 $\pm$ 5.9	1.007	91.0 $\pm$ 2.4	91.3 $\pm$ 2.2	0.997
0.4	77.9 $\pm$ 6.8	79.2 $\pm$ 6.1	1.001	65.8 $\pm$ 7.1	69.2 $\pm$ 6.4	1.012	91.8 $\pm$ 2.4	92.1 $\pm$ 2.3	0.997
0.6	76.2 $\pm$ 7.3	77.5 $\pm$ 7.2	1.002	66.2 $\pm$ 6.0	69.7 $\pm$ 5.9	1.006	91.5 $\pm$ 2.6	91.8 $\pm$ 2.5	0.995
0.8	76.5 $\pm$ 7.0	77.7 $\pm$ 6.6	1.002	66.9 $\pm$ 6.3	70.5 $\pm$ 5.6	1.005	91.3 $\pm$ 2.8	91.5 $\pm$ 2.7	0.998
1.0	75.1 $\pm$ 6.9	76.5 $\pm$ 7.0	1.000	65.6 $\pm$ 6.2	69.3 $\pm$ 5.3	1.009	90.8 $\pm$ 2.9	91.2 $\pm$ 2.8	0.998
1.2	75.6 $\pm$ 6.9	76.9 $\pm$ 6.4	0.995	65.8 $\pm$ 6.8	69.3 $\pm$ 6.0	1.005	90.8 $\pm$ 3.1	91.1 $\pm$ 2.9	0.997
1.4	74.7 $\pm$ 8.0	76.0 $\pm$ 7.8	1.003	67.0 $\pm$ 6.5	70.5 $\pm$ 5.8	1.009	90.9 $\pm$ 2.7	91.2 $\pm$ 2.6	1.000
1.6	75.4 $\pm$ 7.4	76.8 $\pm$ 6.6	0.997	67.1 $\pm$ 7.0	70.7 $\pm$ 6.3	1.001	90.7 $\pm$ 3.0	91.0 $\pm$ 2.9	1.001
1.8	74.3 $\pm$ 8.3	76.2 $\pm$ 7.0	0.999	66.6 $\pm$ 6.2	70.3 $\pm$ 5.7	1.007	91.0 $\pm$ 2.6	91.4 $\pm$ 2.4	0.996
2.0	73.6 $\pm$ 7.4	75.2 $\pm$ 7.2	0.996	67.2 $\pm$ 5.9	71.0 $\pm$ 5.4	1.006	90.8 $\pm$ 2.4	91.1 $\pm$ 2.2	1.001

work, improvements may also be obtained by using other beta distribution parameters higher than 0.4. The overall results indicate that the method helps increase accuracy values in the second and fourth datasets. The accuracy improvements in these datasets indicate the same behavior that was obtained in the original work. However, in other datasets, the method diminishes the performance. Especially in the third dataset with all radar configurations, the performance degrades considerably. This situation will be discussed in detail in the next chapter.

Also, as results indicate, there is no clear relation between the augmentation technique and the beta distribution parameters. However, the model tendencies about the final alpha order are still preserved in this application, where the final values are generally less than 1. Also, since the method has not been applied in the CNN architecture in any dataset, the interpretations should be completed accordingly, as given in the next chapter.

The second stretching augmentation methodology is also applied to this architecture to continue with the comprehensive analysis, as explained in the previous section. The results are shared with the same style as in Table 5.12.

Table 5.12: LSTM Augmentation Method: Stretching

Stretch Dimension	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
Fast-Time	93.8 $\pm$ 4.4	94.1 $\pm$ 4.3	0.988	87.0 $\pm$ 2.3	86.6 $\pm$ 2.4	0.992	64.5 $\pm$ 6.3	67.0 $\pm$ 6.0	1.005
Slow-Time	93.2 $\pm$ 4.7	93.9 $\pm$ 3.9	0.989	86.4 $\pm$ 2.6	85.8 $\pm$ 2.8	0.997	61.8 $\pm$ 5.6	64.8 $\pm$ 5.7	1.002
Both	92.0 $\pm$ 5.1	93.0 $\pm$ 4.4	0.992	86.3 $\pm$ 2.4	85.8 $\pm$ 2.5	0.993	55.4 $\pm$ 6.4	58.5 $\pm$ 6.6	0.994
None	94.2 $\pm$ 2.7	94.7 $\pm$ 2.3	0.985	87.5 $\pm$ 2.3	87.0 $\pm$ 2.3	0.995	71.3 $\pm$ 7.0	73.5 $\pm$ 6.8	1.002

Stretch Dimension	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
Fast-Time	75.5 $\pm$ 6.2	77.3 $\pm$ 5.9	0.997	68.2 $\pm$ 7.3	72.1 $\pm$ 6.4	1.006	91.1 $\pm$ 3.4	91.3 $\pm$ 3.4	1.000
Slow-Time	66.0 $\pm$ 9.5	68.8 $\pm$ 8.5	0.999	64.3 $\pm$ 6.9	68.9 $\pm$ 6.3	1.001	91.1 $\pm$ 3.3	91.4 $\pm$ 3.2	1.004
Both	65.9 $\pm$ 8.0	68.7 $\pm$ 7.3	0.995	60.8 $\pm$ 6.7	66.1 $\pm$ 5.8	1.006	82.8 $\pm$ 4.6	83.2 $\pm$ 4.7	1.001
None	80.6 $\pm$ 6.5	81.5 $\pm$ 5.7	0.993	74.8 $\pm$ 6.5	77.3 $\pm$ 6.1	0.999	92.2 $\pm$ 2.7	92.4 $\pm$ 2.6	1.001

The results indicate that the stretching is not working in any case in the LSTM-based architecture, unlike the CNN architecture. Although the detailed discussion will be covered in the next chapter, this situation might be explained considering the related work in the literature and the theoretical background of the block used in these architectures. The LSTM structure utilizes the relation between the time-series data, which is corrupted with the stretching functions, although the main aim is to increase the model's generalization. In the explained stretching approach, padding and cropping are applied to create fixed-size input data, which might be considered the corruption of the time-related features within the dataset. Thus, the usability of the proposed method with the RNN structures might be problematic. A more detailed analysis of other configurations is provided in the next chapter. In addition, the studies that are considered to be utilizing this type of augmentation are mainly based on CNN-based architectures.

5.2.2.4 Initial FrFT Order Effect

The fractional order and the time frequency representations relations are provided in the second chapter, and a detailed description of the experiment is provided in the CNN-based architecture. Similar to the other steps covered for the LSTM-based architecture, the same reasoning shared for the CNN-based architecture

might be employed again. The overall results, including the initial fractional order effect in the examined architecture with all datasets, are provided in the Table 5.13.

Table 5.13: LSTM Model Initial FrFT Order

Initial α	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.1	69.8 $\pm$ 9.5	71.8 $\pm$ 8.5	0.160	85.3 $\pm$ 2.9	84.8 $\pm$ 2.8	0.238	53.9 $\pm$ 6.5	56.4 $\pm$ 6.4	0.148
0.2	83.7 $\pm$ 8.1	85.0 $\pm$ 7.9	0.287	86.6 $\pm$ 2.6	86.0 $\pm$ 2.8	0.299	61.5 $\pm$ 7.0	63.9 $\pm$ 6.6	0.256
0.4	91.3 $\pm$ 5.0	92.0 $\pm$ 4.4	0.444	87.0 $\pm$ 2.0	86.5 $\pm$ 2.1	0.432	63.9 $\pm$ 6.8	66.1 $\pm$ 7.2	0.441
0.6	94.2 $\pm$ 3.9	94.6 $\pm$ 3.8	0.608	87.1 $\pm$ 2.7	86.6 $\pm$ 2.4	0.609	66.4 $\pm$ 6.7	68.3 $\pm$ 7.0	0.620
0.8	94.7 $\pm$ 3.2	95.4 $\pm$ 2.7	0.798	86.8 $\pm$ 2.3	86.3 $\pm$ 2.2	0.790	69.3 $\pm$ 5.9	71.5 $\pm$ 5.8	0.806
1.0	94.2 $\pm$ 4.8	94.8 $\pm$ 4.0	0.986	87.0 $\pm$ 2.4	86.4 $\pm$ 2.5	0.995	71.0 $\pm$ 6.7	73.2 $\pm$ 6.1	0.997
1.2	94.4 $\pm$ 4.1	95.1 $\pm$ 3.5	1.199	87.0 $\pm$ 2.1	86.5 $\pm$ 2.2	1.205	68.0 $\pm$ 5.8	70.5 $\pm$ 5.8	1.191
1.4	93.2 $\pm$ 3.6	93.5 $\pm$ 3.3	1.398	86.7 $\pm$ 2.8	86.1 $\pm$ 2.9	1.394	65.6 $\pm$ 6.8	67.7 $\pm$ 6.8	1.382
1.6	89.5 $\pm$ 6.5	90.6 $\pm$ 6.0	1.560	86.9 $\pm$ 2.5	86.4 $\pm$ 2.5	1.567	63.5 $\pm$ 7.3	66.0 $\pm$ 7.1	1.564
1.8	81.2 $\pm$ 10.	83.2 $\pm$ 9.2	1.717	85.7 $\pm$ 2.4	85.1 $\pm$ 2.5	1.702	60.5 $\pm$ 6.0	63.0 $\pm$ 6.3	1.740
2.0	67.2 $\pm$ 7.3	69.2 $\pm$ 7.0	1.925	75.7 $\pm$ 4.7	75.0 $\pm$ 4.7	1.967	49.9 $\pm$ 5.9	52.6 $\pm$ 5.6	1.976

Initial α	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.1	72.8 $\pm$ 6.0	74.2 $\pm$ 6.2	0.143	70.5 $\pm$ 5.3	72.8 $\pm$ 5.5	0.173	58.6 $\pm$ 5.0	59.4 $\pm$ 5.0	0.105
0.2	74.2 $\pm$ 6.5	75.8 $\pm$ 6.4	0.238	72.0 $\pm$ 5.6	74.5 $\pm$ 5.8	0.267	63.9 $\pm$ 5.7	65.0 $\pm$ 5.9	0.207
0.4	75.1 $\pm$ 6.6	76.5 $\pm$ 5.6	0.438	72.5 $\pm$ 6.7	75.3 $\pm$ 5.9	0.460	71.6 $\pm$ 5.2	72.4 $\pm$ 5.2	0.403
0.6	75.5 $\pm$ 5.3	76.9 $\pm$ 4.8	0.617	74.6 $\pm$ 5.1	77.1 $\pm$ 4.7	0.634	81.0 $\pm$ 4.4	81.4 $\pm$ 4.5	0.611
0.8	76.7 $\pm$ 5.3	78.0 $\pm$ 5.5	0.808	73.8 $\pm$ 5.4	76.5 $\pm$ 5.3	0.814	87.5 $\pm$ 3.4	87.8 $\pm$ 3.4	0.803
1.0	79.8 $\pm$ 5.5	80.8 $\pm$ 5.0	0.997	74.9 $\pm$ 6.6	77.4 $\pm$ 6.2	0.999	92.6 $\pm$ 2.8	92.8 $\pm$ 2.7	0.997
1.2	78.2 $\pm$ 7.5	79.2 $\pm$ 6.9	1.191	74.2 $\pm$ 7.1	77.2 $\pm$ 5.7	1.183	85.9 $\pm$ 3.3	86.2 $\pm$ 3.3	1.193
1.4	74.7 $\pm$ 7.1	76.0 $\pm$ 6.6	1.376	73.8 $\pm$ 6.3	76.5 $\pm$ 5.3	1.371	80.7 $\pm$ 3.6	81.2 $\pm$ 3.6	1.394
1.6	73.2 $\pm$ 6.9	74.7 $\pm$ 6.2	1.564	73.4 $\pm$ 5.8	76.0 $\pm$ 5.3	1.542	72.7 $\pm$ 5.1	73.6 $\pm$ 5.0	1.594
1.8	73.5 $\pm$ 6.3	75.3 $\pm$ 6.2	1.759	72.4 $\pm$ 7.6	74.6 $\pm$ 7.2	1.738	66.1 $\pm$ 4.9	66.9 $\pm$ 4.8	1.790
2.0	70.5 $\pm$ 6.7	72.3 $\pm$ 5.9	1.970	69.2 $\pm$ 6.6	71.6 $\pm$ 6.2	1.951	59.6 $\pm$ 4.8	60.4 $\pm$ 4.9	2.000

Considering the overall results, the first key observation might be the consistency between the experiment conducted with the CNN-based architecture. The initial values closer to 1 lead to higher model performances than those in the examined initial point set. However, the degradation in the accuracy values concerning fractional orders slightly differs from the CNN-based architecture. In this experiment, the visualization of overall degradation is closer to a rectangular shape instead of a triangular shape, which is observed in the previous experiment in the human classification problem. This representation might be seen in Figure 5.2. Thus, the LSTM-based architecture is more robust in terms of the different fractional representations of the same dataset. In addition to alterations in the overall characteristics between the structures, the LSTM-based architecture performs best in different initial fractional orders other than 1 in the two datasets. Dataset-I achieves the best performance with the initial order $\alpha = 0.8$, and the

second dataset achieves the best performance with the order $\alpha = 0.6$.

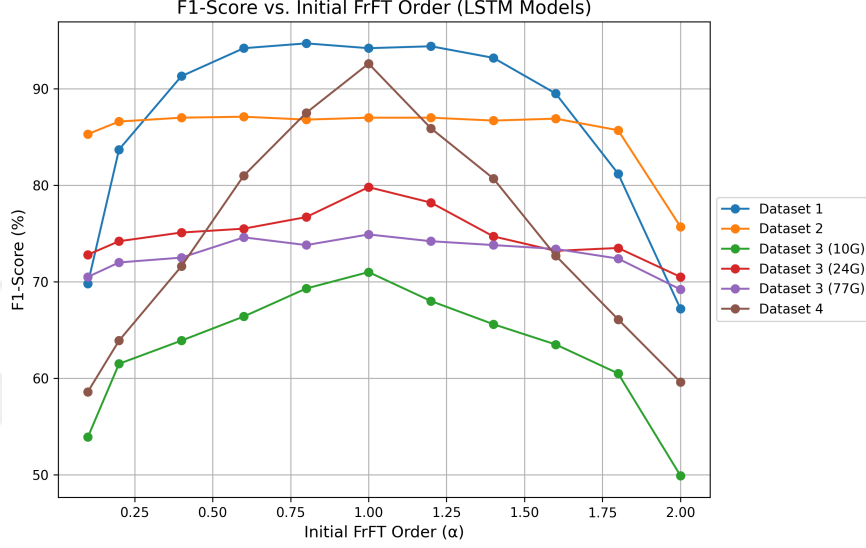


Figure 5.2: Model Performance of LSTM Architecture vs Initial Order

In addition to the robustness in the LSTM-based architecture, the characteristics obtained in the CNN-based architecture, which may be explained as the difference between the final and initial orders concerning different initial values, are also observed again. The initial values close to 1 have the final mean order values closer to the initial point.

In the other experiments, including the LSTM architecture usage, the tendency of the final fractional orders is mentioned several times, where the final mean orders have values less than 1. This situation is observed again in this experiment. Also, as observed in the CNN-based architecture, the final mean alpha orders try to converge values closer to 1 in each dataset and initial order. For instance, if we consider the first dataset, initial order $\alpha = 0.1$ converges to $\bar{\alpha} = 0.16$, initial order $\alpha = 0.2$ converges to $\bar{\alpha} = 0.287$, and so on. Thus, the tendency to the Fourier transform in the models is also observed in the LSTM-based architecture, similarly to the CNN-based architecture. However, the local minimum problems observed in the CNN-based architecture are also observed in this case.

In this section, the results obtained from experiments conducted with the LSTM-based architecture are shared and discussed briefly, considering the results

and explanations in the CNN-architecture section. A detailed and comprehensive analysis of the overall experiments with systematic comparisons will be provided in the next chapter; the mentioned CNN comparisons in this section are provided to increase the readability of the thesis and familiarity of the reader with the experiments.

5.2.3 GRU-Based Model

As explained in the LSTM-based method section, the GRU-based approach is provided in a different chapter since comparing experiments conducted with different RNN blocks is needed. The steps and sections covered for the LSTM-based architecture are repeated with the GRU-based architecture without changing any other parameter except the block structure. Therefore, instead of focusing on the theoretical explanations briefly shared in the evaluation of other architectures, the main focus of this section is on the comparison with the LSTM-based architecture.

5.2.3.1 FT vs FrFT

The comparison of the proposed methodology and the conventional approaches is completed with the GRU-based architecture, with the three-block architecture, as used in the other single-branch models examined in this chapter. The complete results are provided in the Table 5.14, and since the only difference is the usage of the GRU units instead of LSTM units, no further theoretical explanation is provided.

Table 5.14: GRU Model FT vs. FrFT

Dataset	Accuracy (%)		F1 Score (%)		Mean Alpha Order	
	FT	FrFT	FT	FrFT	FT (Fixed=1.0)	FrFT (Trainable)
Dataset 1	96.0 $\pm$ 2.9	96.1 $\pm$ 3.0	95.4 $\pm$ 3.3	95.5 $\pm$ 3.5	1.0	0.993
Dataset 2	87.1 $\pm$ 2.5	87.4 $\pm$ 2.7	87.7 $\pm$ 2.3	87.9 $\pm$ 2.7	1.0	0.998
Dataset 3 (10 GHz)	73.9 $\pm$ 5.7	74.5 $\pm$ 5.5	72.3 $\pm$ 6.0	72.9 $\pm$ 5.9	1.0	1.001
Dataset 3 (24 GHz)	83.4 $\pm$ 4.9	83.6 $\pm$ 5.4	82.4 $\pm$ 5.4	82.8 $\pm$ 5.4	1.0	0.999
Dataset 3 (77 GHz)	77.6 $\pm$ 6.0	78.6 $\pm$ 5.8	74.8 $\pm$ 6.6	75.9 $\pm$ 6.9	1.0	1.013
Dataset 4	93.2 $\pm$ 1.5	94.2 $\pm$ 1.6	93.1 $\pm$ 1.5	94.2 $\pm$ 1.6	1.0	0.998

The results indicate that improvements are obtained in all datasets using the proposed methodology, including the trainable FrFT transform. The observed improvement amounts in terms of accuracy and F1-score are $[0.1, 0.3, 0.6, 0.2, 1.2, 1.1]$ and $[0.1, 0.2, 0.6, 0.4, 1.1, 1.1]$. Also, the tendency of the final FrFT orders is still preserved in this scenario, similar to the LSTM-based architectures and the experiments conducted with the GRU-based architecture.

5.2.3.2 Configuration Study

The configuration study focusing on the effect of the number of blocks is conducted using a GRU-based architecture. As stated several times before, the same structure is used in this architecture as in the LSTM-based architecture. The result of the configuration study is given in the same form used in the previous section, as demonstrated in the Table 5.15.

Table 5.15: GRU Configuration Study: Block Counts

GRU Blocks	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
1	94.2 ± 4.3	94.7 ± 4.0	0.988	87.3 ± 2.6	86.6 ± 2.9	0.998	74.6 ± 5.8	76.0 ± 5.4	0.998
2	95.8 ± 3.4	96.1 ± 3.2	0.994	87.9 ± 2.7	87.4 ± 2.8	0.998	75.3 ± 5.2	76.6 ± 5.3	0.999
3	95.5 ± 3.5	96.0 ± 3.0	0.993	87.7 ± 2.4	87.1 ± 2.4	0.998	73.9 ± 5.3	75.4 ± 5.3	1.001
4	94.4 ± 4.4	94.8 ± 4.2	0.989	87.5 ± 2.6	87.0 ± 2.6	0.997	71.2 ± 6.2	72.3 ± 6.5	1.000
5	94.6 ± 3.8	95.1 ± 3.2	0.991	87.7 ± 2.2	87.2 ± 2.4	0.997	70.0 ± 6.5	72.2 ± 6.0	1.002

GRU Blocks	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
1	80.8 ± 5.4	82.0 ± 5.4	1.000	75.1 ± 7.1	77.6 ± 6.5	1.009	95.8 ± 2.0	96.0 ± 1.9	0.998
2	81.8 ± 5.7	82.7 ± 5.2	0.999	75.6 ± 5.6	78.0 ± 5.5	1.010	95.4 ± 2.3	95.6 ± 2.2	0.999
3	80.6 ± 6.2	81.8 ± 6.0	0.998	75.8 ± 6.8	78.3 ± 5.8	1.014	94.3 ± 1.9	94.5 ± 1.9	1.001
4	80.7 ± 6.5	81.8 ± 6.1	1.000	74.2 ± 6.1	76.6 ± 6.4	1.011	92.7 ± 2.5	93.0 ± 2.4	1.000
5	78.6 ± 6.3	80.3 ± 5.6	0.999	74.5 ± 7.3	76.6 ± 7.2	1.012	91.8 ± 2.9	92.1 ± 2.9	1.001

The results obtained in the investigation of the block counts indicate the high similarity between the RNN block types. Like the LSTM-based case, the maximum performances are obtained by employing one or two blocks, and the high complexity causes degradation in the model performance. Using two blocks for the human motion classification task provides the best performance among all datasets, whereas the drone classification task works well with one block. Also, the final mean fractional orders in the first two datasets indicate the characteristic

convergence for the fractional orders less than one. However, the final fractional orders in the third dataset do not demonstrate clear behavior in this experiment utilizing the GRU-based architecture, unlike the LSTM-based approach.

Table 5.16: GRU Configuration study: STFT Window and Overlap Sizes

Window / Overlap	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
(128, 64)	94.5 $\pm$ 4.0	95.0 $\pm$ 3.6	0.989	87.9 $\pm$ 2.6	87.3 $\pm$ 2.7	0.996	74.8 $\pm$ 5.3	76.2 $\pm$ 5.3	1.000
(128, 100)	95.5 $\pm$ 3.5	96.0 $\pm$ 3.0	0.993	87.7 $\pm$ 2.4	87.1 $\pm$ 2.5	0.998	73.9 $\pm$ 5.3	75.4 $\pm$ 5.3	1.000
(128, 112)	94.8 $\pm$ 3.5	95.3 $\pm$ 3.2	0.993	88.3 $\pm$ 2.5	87.7 $\pm$ 2.6	0.997	74.1 $\pm$ 5.3	75.7 $\pm$ 5.2	1.000
(160, 120)	95.1 $\pm$ 3.7	95.5 $\pm$ 3.1	0.990	88.3 $\pm$ 2.3	87.8 $\pm$ 2.3	0.996	74.1 $\pm$ 6.0	75.8 $\pm$ 5.6	0.998
(160, 128)	95.4 $\pm$ 3.9	95.8 $\pm$ 3.8	0.990	88.2 $\pm$ 2.2	87.7 $\pm$ 2.3	0.997	74.6 $\pm$ 6.2	76.1 $\pm$ 6.3	0.999
(192, 128)	95.3 $\pm$ 3.4	95.7 $\pm$ 3.3	0.986	87.6 $\pm$ 2.6	87.1 $\pm$ 2.5	0.996	73.6 $\pm$ 6.0	75.3 $\pm$ 6.1	0.999
(192, 160)	95.0 $\pm$ 3.3	95.5 $\pm$ 2.9	0.988	88.0 $\pm$ 2.2	87.5 $\pm$ 2.2	0.998	72.1 $\pm$ 5.1	74.3 $\pm$ 4.9	0.998
(256, 128)	93.7 $\pm$ 4.2	93.9 $\pm$ 4.2	0.988	87.6 $\pm$ 2.0	87.0 $\pm$ 2.4	0.995	73.2 $\pm$ 4.5	74.6 $\pm$ 4.6	1.001
(256, 200)	95.3 $\pm$ 3.6	95.7 $\pm$ 3.0	0.990	87.2 $\pm$ 2.8	86.7 $\pm$ 3.0	0.997	73.5 $\pm$ 5.2	75.1 $\pm$ 5.2	1.000
(256, 224)	95.2 $\pm$ 3.3	95.8 $\pm$ 2.7	0.989	87.8 $\pm$ 2.6	87.3 $\pm$ 2.6	0.995	73.9 $\pm$ 4.8	75.3 $\pm$ 4.9	0.999

Window / Overlap	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
(128, 64)	80.4 $\pm$ 5.6	81.7 $\pm$ 5.3	0.999	78.0 $\pm$ 7.2	80.4 $\pm$ 6.8	1.007	94.0 $\pm$ 2.2	94.2 $\pm$ 2.2	0.999
(128, 100)	80.6 $\pm$ 6.2	81.8 $\pm$ 6.0	0.998	75.7 $\pm$ 6.8	78.3 $\pm$ 5.8	1.013	94.3 $\pm$ 1.9	94.5 $\pm$ 1.9	1.001
(128, 112)	79.5 $\pm$ 6.9	80.7 $\pm$ 6.4	1.001	75.4 $\pm$ 6.5	78.1 $\pm$ 5.9	1.014	94.2 $\pm$ 2.0	94.4 $\pm$ 2.0	1.000
(160, 120)	81.0 $\pm$ 6.0	82.3 $\pm$ 5.3	1.001	74.8 $\pm$ 6.2	77.4 $\pm$ 5.8	1.013	93.3 $\pm$ 2.9	93.5 $\pm$ 2.7	1.001
(160, 128)	80.8 $\pm$ 6.5	81.9 $\pm$ 5.9	1.000	75.6 $\pm$ 6.0	78.1 $\pm$ 5.5	1.009	93.6 $\pm$ 2.6	93.8 $\pm$ 2.5	0.996
(192, 128)	82.9 $\pm$ 5.4	84.1 $\pm$ 4.7	0.999	75.5 $\pm$ 5.3	78.1 $\pm$ 4.9	1.002	92.0 $\pm$ 2.9	92.2 $\pm$ 2.8	0.999
(192, 160)	81.6 $\pm$ 6.3	83.0 $\pm$ 5.5	1.000	76.8 $\pm$ 5.9	78.9 $\pm$ 5.7	1.005	92.2 $\pm$ 2.3	92.5 $\pm$ 2.3	1.000
(256, 128)	83.1 $\pm$ 5.3	84.1 $\pm$ 4.9	0.999	75.8 $\pm$ 5.7	78.1 $\pm$ 5.4	0.998	90.3 $\pm$ 2.8	90.6 $\pm$ 2.8	0.999
(256, 200)	82.8 $\pm$ 4.8	84.1 $\pm$ 4.3	1.001	75.5 $\pm$ 5.7	77.8 $\pm$ 5.7	1.000	90.3 $\pm$ 2.8	90.6 $\pm$ 2.8	0.999
(256, 224)	82.8 $\pm$ 5.8	83.8 $\pm$ 5.7	0.998	75.6 $\pm$ 5.9	77.8 $\pm$ 5.5	0.998	90.3 $\pm$ 2.8	90.6 $\pm$ 2.8	0.999

In the second configuration study, window-overlap sizes are examined using the same application procedure with the GRU-based architecture. The results are presented in the Table 5.16 as provided in the previous experiments. Unlike the RNN architecture, the model performances are not directly correlated with the examined set in both CNN-based structures. Since both GRU and LSTM-based architectures indicate different behavior than the CNN-based architecture, this situation might be explained by the RNN structures, which will be discussed in the next chapter.

5.2.3.3 Augmentation Methods

As a common technique for single-branch architectures, the Mix-up augmentation is applied to the GRU-based architecture to investigate the possible other outcomes. The method did not improve the CNN-based architecture, whereas the

LSTM-based architecture utilizes the technique in the second and fourth datasets. The overall results are shared in the Table 5.17.

Table 5.17: GRU Augmentation Method: Mixup Augmentation

Mixup β	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.0	95.5 $\pm$ 3.5	96.0 $\pm$ 3.0	0.993	88.2 $\pm$ 2.6	87.7 $\pm$ 2.6	0.997	73.3 $\pm$ 6.0	75.2 $\pm$ 5.6	0.999
0.2	95.0 $\pm$ 3.6	95.4 $\pm$ 3.2	0.991	88.5 $\pm$ 2.4	88.0 $\pm$ 2.2	0.998	58.8 $\pm$ 6.8	61.4 $\pm$ 6.9	1.001
0.4	94.0 $\pm$ 3.8	94.7 $\pm$ 3.2	0.992	88.3 $\pm$ 2.6	87.7 $\pm$ 2.6	0.998	56.9 $\pm$ 7.7	59.8 $\pm$ 7.1	1.001
0.6	93.5 $\pm$ 4.0	94.2 $\pm$ 3.6	0.988	88.3 $\pm$ 2.7	87.7 $\pm$ 2.8	0.998	56.5 $\pm$ 7.3	59.5 $\pm$ 6.7	1.000
0.8	92.5 $\pm$ 3.9	93.4 $\pm$ 3.3	0.988	87.8 $\pm$ 2.0	87.3 $\pm$ 1.9	0.998	55.2 $\pm$ 8.2	58.3 $\pm$ 8.2	1.002
1.0	91.9 $\pm$ 4.5	92.5 $\pm$ 3.9	0.991	87.4 $\pm$ 2.5	86.8 $\pm$ 2.6	0.999	57.1 $\pm$ 7.3	59.9 $\pm$ 6.8	1.001
1.2	89.8 $\pm$ 5.0	90.6 $\pm$ 4.6	0.987	87.7 $\pm$ 2.6	87.1 $\pm$ 2.7	0.998	56.6 $\pm$ 7.3	59.6 $\pm$ 7.8	1.000
1.4	89.4 $\pm$ 5.6	90.5 $\pm$ 4.8	0.988	87.6 $\pm$ 2.8	87.0 $\pm$ 3.0	0.998	56.5 $\pm$ 7.5	59.4 $\pm$ 7.0	1.000
1.6	89.8 $\pm$ 4.9	90.6 $\pm$ 4.5	0.991	87.4 $\pm$ 2.3	86.8 $\pm$ 2.4	0.998	56.3 $\pm$ 9.3	59.0 $\pm$ 9.3	0.999
1.8	88.0 $\pm$ 5.4	89.2 $\pm$ 4.5	0.986	87.0 $\pm$ 2.8	86.5 $\pm$ 2.8	0.999	56.8 $\pm$ 6.9	60.1 $\pm$ 6.6	0.999
2.0	87.9 $\pm$ 5.2	89.0 $\pm$ 5.0	0.989	87.5 $\pm$ 2.7	87.0 $\pm$ 2.6	0.999	56.5 $\pm$ 7.4	59.5 $\pm$ 7.5	0.999

Mixup β	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.0	82.8 $\pm$ 5.4	83.6 $\pm$ 5.4	0.999	75.8 $\pm$ 7.2	78.3 $\pm$ 6.5	1.014	94.3 $\pm$ 1.9	94.5 $\pm$ 1.9	1.001
0.2	80.8 $\pm$ 6.8	82.0 $\pm$ 6.4	0.999	66.7 $\pm$ 5.9	69.6 $\pm$ 6.0	1.005	95.2 $\pm$ 2.0	95.4 $\pm$ 1.9	0.998
0.4	78.8 $\pm$ 6.9	80.3 $\pm$ 6.5	0.998	67.5 $\pm$ 5.4	70.4 $\pm$ 5.1	1.005	94.8 $\pm$ 2.4	95.0 $\pm$ 2.4	0.999
0.6	78.3 $\pm$ 6.8	79.6 $\pm$ 6.7	0.999	65.7 $\pm$ 6.0	69.0 $\pm$ 6.2	1.005	94.8 $\pm$ 1.9	95.0 $\pm$ 1.9	1.001
0.8	77.0 $\pm$ 7.0	78.6 $\pm$ 6.0	1.001	66.7 $\pm$ 5.1	70.1 $\pm$ 5.0	1.006	94.8 $\pm$ 2.1	95.0 $\pm$ 2.0	0.997
1.0	75.8 $\pm$ 8.2	77.6 $\pm$ 6.9	0.999	65.9 $\pm$ 6.4	69.6 $\pm$ 6.1	1.004	94.5 $\pm$ 2.4	94.8 $\pm$ 2.3	0.999
1.2	75.2 $\pm$ 7.6	77.4 $\pm$ 7.1	0.999	66.9 $\pm$ 6.0	70.2 $\pm$ 5.4	1.005	94.9 $\pm$ 2.4	95.0 $\pm$ 2.4	0.999
1.4	75.0 $\pm$ 8.3	77.1 $\pm$ 7.3	0.999	64.3 $\pm$ 7.1	68.3 $\pm$ 5.8	1.006	94.7 $\pm$ 2.1	94.9 $\pm$ 2.1	1.000
1.6	76.0 $\pm$ 9.0	77.7 $\pm$ 7.9	0.999	66.0 $\pm$ 5.6	69.3 $\pm$ 5.6	1.005	94.6 $\pm$ 2.5	94.8 $\pm$ 2.5	1.000
1.8	75.0 $\pm$ 9.4	76.8 $\pm$ 8.4	0.998	66.2 $\pm$ 6.2	69.9 $\pm$ 5.5	1.006	95.0 $\pm$ 2.0	95.2 $\pm$ 1.9	0.998
2.0	74.6 $\pm$ 8.4	76.4 $\pm$ 7.6	0.999	64.9 $\pm$ 6.5	68.4 $\pm$ 6.2	1.007	94.8 $\pm$ 2.0	95.0 $\pm$ 1.9	1.000

The overall results indicate similar behavior observed in the LSTM-based case, where the improvements are obtained for the same datasets, and the best performances are acquired in the same value, $\beta = 0.2$. Also, as stated before, the best performance is observed in the similar β values as provided in the original work, and the LSTM-based architecture. Therefore, the Mix-up augmentation method is claimed to be suitable for these cases. However, the third dataset cannot benefit from the applied technique in any radar data belonging to this dataset, especially for 10 GHz radar data, where the performance decreases considerably. Therefore, the nature of the datasets and comparing the CNN and RNN-based structures should be examined since the RNN-based models produce similar outcomes within the same datasets. This discussion is left to the next chapter, as was the case with the identical discussion topics briefly mentioned in this section.

Table 5.18: GRU Augmentation Method: Stretching

Stretch Dimension	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
Fast-Time	95.1 $\pm$ 4.0	95.4 $\pm$ 3.9	0.991	87.3 $\pm$ 2.5	86.7 $\pm$ 2.6	0.996	68.0 $\pm$ 8.5	70.3 $\pm$ 8.0	0.999
Slow-Time	94.6 $\pm$ 3.5	95.3 $\pm$ 3.2	0.988	87.6 $\pm$ 2.9	87.0 $\pm$ 3.0	0.993	65.5 $\pm$ 6.8	67.7 $\pm$ 6.7	1.007
Both	93.2 $\pm$ 4.7	94.3 $\pm$ 3.4	0.991	86.3 $\pm$ 3.0	85.7 $\pm$ 3.2	0.994	59.5 $\pm$ 7.2	62.3 $\pm$ 7.2	1.003
None	95.5 $\pm$ 3.5	96.0 $\pm$ 3.0	0.993	88.0 $\pm$ 2.2	87.6 $\pm$ 2.3	0.998	73.6 $\pm$ 6.0	75.3 $\pm$ 5.9	1.000

Stretch Dimension	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
Fast-Time	76.2 $\pm$ 7.9	77.9 $\pm$ 6.8	0.998	67.8 $\pm$ 7.2	72.1 $\pm$ 6.4	1.018	90.7 $\pm$ 3.4	91.1 $\pm$ 3.2	1.002
Slow-Time	67.5 $\pm$ 9.3	69.9 $\pm$ 7.9	0.999	67.2 $\pm$ 7.7	70.9 $\pm$ 6.7	1.007	92.3 $\pm$ 3.1	92.5 $\pm$ 2.9	0.997
Both	63.3 $\pm$ 9.5	66.4 $\pm$ 8.0	0.998	63.9 $\pm$ 7.6	68.7 $\pm$ 7.2	1.009	82.4 $\pm$ 5.4	82.9 $\pm$ 5.1	0.997
None	82.8 $\pm$ 5.4	83.6 $\pm$ 5.4	0.999	75.7 $\pm$ 6.8	78.3 $\pm$ 5.8	1.013	93.5 $\pm$ 2.7	93.7 $\pm$ 2.6	0.998

In the second augmentation method, the stretching was utilized as provided in the previous architectures. The last results indicate the performance increments in the CNN-based architecture and degradations in the LSTM architecture. Like the LSTM-based case, the stretching drastically negatively influences the GRU-based architecture. Therefore, the RNN-based structures cannot utilize the stretching that deserves a separate examination provided in the next chapter.

5.2.3.4 Initial FrFT Order Effect

The different representations of the signals in the different fractional domains are also examined using the same methodology applied in the LSTM-based and GRU-based architectures. In the previous sections, some common characteristics have been obtained, and a comparison of those characteristics should be completed in this case, which is completed under this heading. The overall results might be seen in the Table 5.19.

Similar to the outcomes of the previous architectures, triangular to rectangular changes in the performance are also obtained in this section in all datasets. This situation is also presented in Figure 5.3 to highlight the common characteristics. Similar to the LSTM-based case, the highest accuracies are obtained in the second dataset with the initial FrFT orders different than 1. However, initial order values of [0.6, 1.4, 1.6] indicate this performance increment, which was only one case in the LSTM-based architecture. Thus, although similar characteristics

Table 5.19: GRU Model Initial FrFT Order

Initial α	Dataset 1			Dataset 2			Dataset 3 (10 GHz)		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.1	71.4 $\pm$ 10.	73.6 $\pm$ 9.4	0.160	86.1 $\pm$ 3.1	85.4 $\pm$ 3.1	0.252	56.7 $\pm$ 7.3	59.1 $\pm$ 7.5	0.144
0.2	87.7 $\pm$ 7.9	88.4 $\pm$ 7.1	0.320	86.7 $\pm$ 3.0	86.2 $\pm$ 3.2	0.302	63.0 $\pm$ 5.2	64.7 $\pm$ 5.8	0.246
0.4	93.2 $\pm$ 4.4	93.9 $\pm$ 4.0	0.448	87.6 $\pm$ 2.1	87.0 $\pm$ 2.1	0.438	66.5 $\pm$ 5.9	68.7 $\pm$ 5.7	0.428
0.6	94.9 $\pm$ 4.0	95.2 $\pm$ 3.9	0.604	88.4 $\pm$ 2.0	87.9 $\pm$ 2.2	0.607	67.3 $\pm$ 6.2	69.5 $\pm$ 6.1	0.612
0.8	94.2 $\pm$ 3.8	94.6 $\pm$ 3.7	0.804	87.3 $\pm$ 2.4	86.8 $\pm$ 2.6	0.789	71.7 $\pm$ 6.7	73.4 $\pm$ 6.6	0.803
1.0	95.5 $\pm$ 3.2	96.0 $\pm$ 2.8	0.992	87.7 $\pm$ 2.4	87.1 $\pm$ 2.4	0.998	73.9 $\pm$ 5.3	75.4 $\pm$ 5.3	1.000
1.2	95.3 $\pm$ 2.8	95.7 $\pm$ 2.5	1.194	87.4 $\pm$ 2.6	86.9 $\pm$ 2.7	1.208	70.7 $\pm$ 5.8	72.6 $\pm$ 5.5	1.196
1.4	94.9 $\pm$ 3.5	95.3 $\pm$ 3.2	1.393	88.3 $\pm$ 2.3	87.8 $\pm$ 2.4	1.394	67.6 $\pm$ 5.4	69.7 $\pm$ 5.5	1.389
1.6	92.9 $\pm$ 4.0	93.6 $\pm$ 3.7	1.558	88.0 $\pm$ 2.2	87.4 $\pm$ 2.3	1.562	65.4 $\pm$ 6.0	67.7 $\pm$ 6.2	1.573
1.8	88.1 $\pm$ 7.5	89.0 $\pm$ 7.3	1.692	86.8 $\pm$ 2.9	86.2 $\pm$ 2.9	1.698	62.7 $\pm$ 6.9	64.8 $\pm$ 7.3	1.749
2.0	69.9 $\pm$ 7.6	72.3 $\pm$ 7.0	1.967	77.0 $\pm$ 4.3	76.5 $\pm$ 4.2	1.965	52.0 $\pm$ 5.8	54.6 $\pm$ 5.9	1.993

Initial α	Dataset 3 (24 GHz)			Dataset 3 (77 GHz)			Dataset 4		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
0.1	73.4 $\pm$ 7.5	74.9 $\pm$ 7.3	0.161	72.0 $\pm$ 5.8	74.3 $\pm$ 5.8	0.179	59.1 $\pm$ 5.2	59.9 $\pm$ 5.3	0.101
0.2	75.9 $\pm$ 6.0	77.2 $\pm$ 6.0	0.250	72.9 $\pm$ 6.7	75.2 $\pm$ 6.5	0.265	64.7 $\pm$ 4.8	65.7 $\pm$ 4.9	0.206
0.4	74.7 $\pm$ 6.3	76.1 $\pm$ 6.5	0.437	74.0 $\pm$ 6.4	76.4 $\pm$ 6.2	0.458	72.5 $\pm$ 4.4	73.4 $\pm$ 4.4	0.400
0.6	74.5 $\pm$ 6.2	76.0 $\pm$ 6.0	0.621	73.8 $\pm$ 7.3	76.6 $\pm$ 6.3	0.634	81.2 $\pm$ 4.2	81.7 $\pm$ 4.3	0.605
0.8	75.4 $\pm$ 9.5	77.0 $\pm$ 8.8	0.816	73.1 $\pm$ 6.9	76.2 $\pm$ 6.5	0.811	88.3 $\pm$ 3.2	88.7 $\pm$ 3.0	0.808
1.0	80.6 $\pm$ 6.2	81.9 $\pm$ 6.0	0.998	75.8 $\pm$ 6.8	78.4 $\pm$ 5.8	1.013	93.0 $\pm$ 2.4	93.3 $\pm$ 2.3	0.999
1.2	74.9 $\pm$ 8.6	76.6 $\pm$ 8.0	1.182	73.0 $\pm$ 7.5	75.8 $\pm$ 6.8	1.187	86.6 $\pm$ 3.6	87.0 $\pm$ 3.4	1.191
1.4	74.0 $\pm$ 7.5	75.9 $\pm$ 7.1	1.380	74.1 $\pm$ 7.0	76.9 $\pm$ 6.2	1.361	81.0 $\pm$ 3.6	81.6 $\pm$ 3.6	1.395
1.6	74.0 $\pm$ 6.3	75.6 $\pm$ 6.8	1.560	74.0 $\pm$ 6.4	76.7 $\pm$ 6.1	1.540	73.3 $\pm$ 5.2	74.2 $\pm$ 5.1	1.595
1.8	75.2 $\pm$ 7.3	76.6 $\pm$ 6.7	1.745	73.3 $\pm$ 7.2	75.4 $\pm$ 6.8	1.735	65.9 $\pm$ 3.5	66.7 $\pm$ 3.9	1.792
2.0	72.4 $\pm$ 6.2	73.8 $\pm$ 5.8	1.998	69.4 $\pm$ 6.8	72.1 $\pm$ 7.2	1.977	59.7 $\pm$ 5.0	60.4 $\pm$ 5.1	1.996

are preserved in all architectures, some architecture-specific outcomes are also obtained.

Another typical behavior that should be highlighted is the tendency of fractional values. In all cases, different from the initial point 1, the final fractional orders try to approach 1, regardless of the initial value.

With this experiment, the experiments conducted with the single-branch architectures are completed. The results are reported systematically, and experiments are analyzed by highlighting the critical outcomes with comparisons. However, more systematic comparisons are provided in the discussion chapter; this chapter should be considered as reporting the experiments with basic explanations. The chapter continues with the multi-branch models that were introduced in Chapter 4.

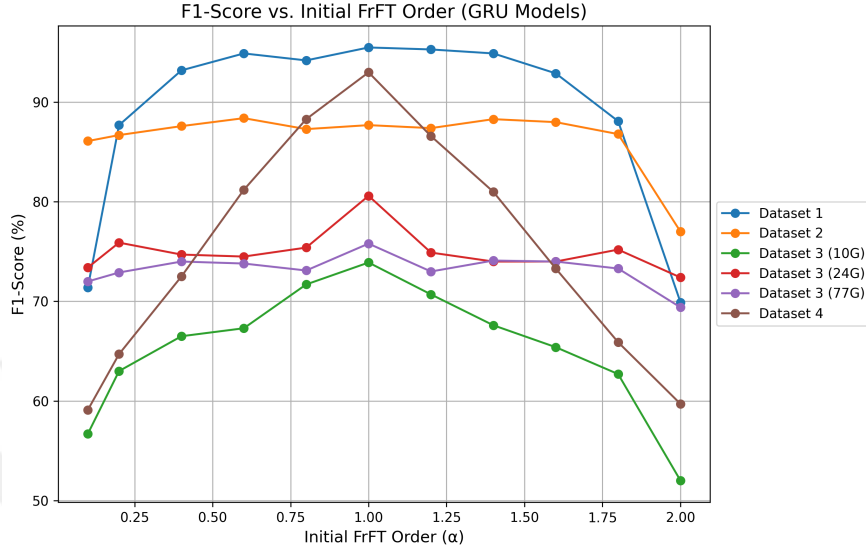


Figure 5.3: Model Performance of GRU Architecture vs Initial Order

5.3 Multi-Branch Models

In this section, outcomes of the multi-branch architectures are examined. The multi-branch models include two types of multi-branch architectures: fusion and Siamase. In the multi-branch fusion model, the main idea is to use a single input to the model, including repeated neural network structures several times, as shown in Figure 4.7. The only difference between the branches in the fusion architectures is the initial FrFT order value. The aim is to examine the possible improvements from using different time-frequency transforms simultaneously via different FrFT orders. The second type of multi-branch model is the Siamase model structure. In this type, different inputs are fed into different branches by bounded weights across the branches, as shown in Figure 4.8. It should be highlighted that this type of approach is possible for only the third dataset examined in this study, since it includes the records of the same samples collected by three different types of radar simultaneously.

5.3.1 Fusion Based Models

The architecture of the fusion-based models is provided in the fourth chapter. The primary motivation behind the employment of these types of models is also provided in the previous chapters. The experiments conducted in single-branch models, which were reported in a detailed order, can be utilized to explain the motivation behind the results. In the experiments investigating the effect of the initial order, meaningful, or even better, in some cases, results are obtained in the cases with the initial orders different than 1. Thus, the fusion models are utilized to take advantage of the various representations of the same signal.

All single-branch architectures are employed as multi-branch fusion models, including CNN, LSTM, and GRU blocks. However, the set of experiments related to augmentation strategies and configuration studies is not completed with this model, since the main aim is to observe only the contribution of the different fractional representations. It should also be highlighted that the employed multi-branch structures include the repetitive construction of the 3-block variants of the single-branch models. To avoid confusion, the control experiment is also conducted in each scenario, shown with one branch count in every experimental scenario.

In the training and testing procedure, no difference is applied, since one of the purposes is to compare these multi-branch architectures with the single-branch architectures. It should be noted that no incompatibility is observed in the application step or the application procedure. The section is constructed in the same order as the single-branch structure, which starts with the CNN-based architecture and ends with the GRU-based architecture.

5.3.1.1 CNN-Fusion Model

The first multi-layer fusion model is constructed with a CNN-based architecture, which was described in the single-branch architecture section. The branch counts are examined as described in the previous chapter, including from 1 to 7

branches, by considering the symmetry with respect to the initial order 1. The main motivation of these multi-branch architectures is to utilize different time-frequency representations obtained via FrFT simultaneously. As observed in all experimental cases, high variance is also observed in this case; therefore, similar to the experiments completed for the single-branch cases, one control experiment is added by using only one branch.

Table 5.20: CNN Multi-Branch Fusion Model

Branch Count	Dataset 1		Dataset 2		Dataset 3 (10 GHz)	
	F1	Acc	F1	Acc	F1	Acc
1	89.3 $\pm$ 5.3	90.8 $\pm$ 4.4	84.3 $\pm$ 3.2	83.6 $\pm$ 3.3	69.7 $\pm$ 5.6	72.3 $\pm$ 5.5
3	89.9 $\pm$ 4.6	91.1 $\pm$ 4.1	82.7 $\pm$ 2.9	81.9 $\pm$ 3.1	69.7 $\pm$ 5.8	72.4 $\pm$ 5.5
5	90.3 $\pm$ 5.0	91.4 $\pm$ 4.5	83.2 $\pm$ 2.4	82.5 $\pm$ 2.5	68.6 $\pm$ 5.9	71.8 $\pm$ 5.5
7	91.1 $\pm$ 4.7	91.9 $\pm$ 4.4	83.5 $\pm$ 2.5	82.8 $\pm$ 2.6	69.5 $\pm$ 6.2	72.4 $\pm$ 6.5

Branch Count	Dataset 3 (24 GHz)		Dataset 3 (77 GHz)		Dataset 4	
	F1	Acc	F1	Acc	F1	Acc
1	60.3 $\pm$ 5.6	62.1 $\pm$ 5.3	61.6 $\pm$ 5.2	64.0 $\pm$ 5.6	95.9 $\pm$ 1.9	95.9 $\pm$ 1.8
3	58.8 $\pm$ 6.1	61.1 $\pm$ 5.8	62.7 $\pm$ 5.2	65.8 $\pm$ 6.4	95.4 $\pm$ 2.3	95.6 $\pm$ 2.3
5	59.5 $\pm$ 6.2	61.0 $\pm$ 5.8	63.1 $\pm$ 5.4	66.9 $\pm$ 6.1	94.9 $\pm$ 2.5	95.1 $\pm$ 2.4
7	59.9 $\pm$ 5.5	61.8 $\pm$ 5.9	62.0 $\pm$ 5.0	65.0 $\pm$ 5.9	95.5 $\pm$ 2.1	95.6 $\pm$ 2.0

Unlike the findings obtained in the RNN cases, which are explained below, adding branches does not provide improvements in each dataset. For instance, the performance of the first dataset increases with the addition of branches, even though it is not covered in the scope of this experiment; adding more branches may provide additional improvements, since a gradual improvement is observed with more branches. However, in other datasets, adding branches indicates a negative impact on the model performance. Since the multi-branch scenarios demonstrate different influences on the CNN and RNN-based models, observation might be linked to the working principle of the utilized units. The detailed comparison and effects will be discussed in the next chapter since similar difference is observed in the single-branch cases, which are also discussed in the next chapter.

5.3.1.2 LSTM-Fusion Model

The first RNN unit examined in this section is an LSTM-based architecture, similar to the single-branch architecture discussed in the previous section. The overall results, including the analysis of 1 to 7 branches, are provided in the Table 5.21. In the single-branch case, especially for the second dataset, it has been observed that the fractional orders different than one improve the model's performance. Thus, with the fusion of the different fractional representations, improvement is expected, naturally.

In all experimental scenarios, high deviation is observed; thus, to represent a suitable comparison, the control experiment is also conducted with a single-branch model, as stated before.

Table 5.21: LSTM Multi-Branch Fusion Model

Branch Count	Dataset 1		Dataset 2		Dataset 3 (10 GHz)	
	F1	Acc	F1	Acc	F1	Acc
1	93.2 $\pm$ 4.2	93.8 $\pm$ 4.1	87.3 $\pm$ 2.4	86.8 $\pm$ 2.5	71.6 $\pm$ 5.7	73.8 $\pm$ 5.5
3	95.9 $\pm$ 2.8	96.1 $\pm$ 2.6	87.8 $\pm$ 2.5	87.2 $\pm$ 2.5	71.8 $\pm$ 5.3	74.0 $\pm$ 4.7
5	95.6 $\pm$ 3.3	95.8 $\pm$ 3.0	88.3 $\pm$ 2.2	87.8 $\pm$ 2.2	73.8 $\pm$ 5.7	75.7 $\pm$ 4.6
7	95.2 $\pm$ 3.2	95.5 $\pm$ 3.1	88.0 $\pm$ 2.2	87.5 $\pm$ 2.3	73.9 $\pm$ 5.8	76.0 $\pm$ 5.5

Branch Count	Dataset 3 (24 GHz)		Dataset 3 (77 GHz)		Dataset 4	
	F1	Acc	F1	Acc	F1	Acc
1	80.1 $\pm$ 6.8	81.3 $\pm$ 5.9	73.1 $\pm$ 7.4	75.8 $\pm$ 6.8	92.7 $\pm$ 2.7	92.9 $\pm$ 2.6
3	82.0 $\pm$ 6.1	82.8 $\pm$ 5.8	77.5 $\pm$ 6.2	79.7 $\pm$ 5.6	92.7 $\pm$ 3.5	93.0 $\pm$ 3.4
5	84.1 $\pm$ 5.0	84.6 $\pm$ 5.6	76.8 $\pm$ 6.7	79.2 $\pm$ 5.9	93.6 $\pm$ 2.6	93.7 $\pm$ 2.5
7	83.5 $\pm$ 4.9	83.9 $\pm$ 4.7	78.7 $\pm$ 6.7	81.9 $\pm$ 6.1	94.2 $\pm$ 2.4	94.3 $\pm$ 2.4

If we consider the overall results, as expected, the improvements are obtained with the usage of the different branches. In all datasets, performance is increased with the employment of multiple branches compared to the single-branch case; however, the improvement amounts differ. For instance, the first dataset shows the best performance with three branches, whereas the second dataset shows the best performance with five branches. The branch numbers causing the best performance can be listed as [3, 5, 7, 5, 7, 7] with the dataset order in the table. The differences between the best and single-branch performance can be shown as percentages of [2.7, 1.0, 2.3, 4.0, 5.6, 1.5]. Therefore, meaningful improvements are

obtained, although the branch counts differ for the datasets. Also, it should be highlighted that the standard deviations of the accuracies decrease in all multi-branch cases, representing the increasing model stability.

5.3.1.3 GRU-Fusion Model

The second RNN unit examined with the multi-branch case is GRU, which was described in the fourth chapter. It should be highlighted that the same procedure followed in the LSTM case is applied here; therefore, no additional explanations are provided for this section. The overall results might be seen in the Table 5.22.

Table 5.22: GRU Multi-Branch Fusion Model

Branch Count	Dataset 1		Dataset 2		Dataset 3 (10 GHz)	
	F1	Acc	F1	Acc	F1	Acc
1	95.1 $\pm$ 4.2	95.7 $\pm$ 3.4	87.7 $\pm$ 2.5	87.2 $\pm$ 2.5	73.9 $\pm$ 5.7	75.4 $\pm$ 5.5
3	95.9 $\pm$ 3.1	96.4 $\pm$ 2.7	88.4 $\pm$ 2.6	87.9 $\pm$ 2.7	74.4 $\pm$ 5.6	75.9 $\pm$ 5.4
5	95.6 $\pm$ 3.2	96.0 $\pm$ 3.0	88.8 $\pm$ 2.4	88.3 $\pm$ 2.4	74.8 $\pm$ 5.8	76.3 $\pm$ 5.4
7	95.9 $\pm$ 3.1	96.3 $\pm$ 3.0	88.6 $\pm$ 2.6	88.1 $\pm$ 2.7	73.4 $\pm$ 5.4	75.2 $\pm$ 5.1

Branch Count	Dataset 3 (24 GHz)		Dataset 3 (77 GHz)		Dataset 4	
	F1	Acc	F1	Acc	F1	Acc
1	82.6 $\pm$ 6.3	83.4 $\pm$ 5.9	76.1 $\pm$ 7.4	74.7 $\pm$ 7.0	93.2 $\pm$ 2.4	93.4 $\pm$ 2.3
3	82.7 $\pm$ 6.7	83.4 $\pm$ 6.5	77.6 $\pm$ 6.7	76.1 $\pm$ 6.3	94.7 $\pm$ 2.6	94.9 $\pm$ 2.6
5	82.5 $\pm$ 6.2	83.4 $\pm$ 5.7	77.7 $\pm$ 6.8	76.2 $\pm$ 6.3	94.0 $\pm$ 2.6	94.3 $\pm$ 2.5
7	84.2 $\pm$ 6.4	85.0 $\pm$ 5.8	75.1 $\pm$ 6.2	73.4 $\pm$ 6.3	94.4 $\pm$ 2.0	94.6 $\pm$ 1.9

Similar to the previous LSTM-based architecture, in all datasets, significant improvements are obtained by utilizing the multi-branch fusion approach compared to the base, including the one-branch architecture. The branch counts where the best performance is received may be given as, with the dataset order, [3, 5, 5, 7, 5, 3], different from the obtained branch counts in the LSTM-based architecture. The difference between the RNN units in these architectures might explain this situation. Also, the improvements obtained with these branches in the GRU architecture might be listed as follows: [0.8, 1.1, 0.9, 1.6, 1.5, 1.5]. Although the improvement amounts are lower compared to the LSTM-based architecture, it should be considered that the single-branch performance of the GRU-based

architecture is already higher than that of the LSTM-based architecture. In addition, the obtained stability improvement in the LSTM-based architecture is still valid for the GRU-based architecture.

In light of the outcomes, the contribution of the multi-branch architectures is observable in all base units covered in the scope of this thesis. The positive effect provided by this approach is highlighted by the systematic experiments, including all six classes. The final systematic results, including the Siamase models, a multi-branch structure, are provided in the next section.

5.3.2 Siamase Based Models

The main idea behind Siamese architecture is provided in the previous chapters. With this approach, models utilizing different inputs may be obtained without increasing the model parameters since the weights are shared across the branches. This type of usage is possible with datasets that include the representation of the same scene with different sensors, as explained in the second chapter, and the examined third dataset is very suitable for this type of approach.

Until this point, three different types of units have been utilized: CNN, LSTM, and GRU. This idea is still followed in this section, and the Siamese architecture is constructed with all units examined in the scope of this thesis. The visualization of the model is provided in the fourth chapter as Figure 4.8. It can be described as a repetitive structure including the single-branch models whose weights are bonded and updated simultaneously. Similar to the multi-branch fusion structures, no set of experiments is conducted, including the configuration and augmentation strategies, to emphasize only the outcomes of the architectural technique. The overall results are provided in the Table 5.23. In the results, as conducted for the FT vs FrFT experiments with single-branch models, the Fourier transform is applied with fixed order. Unlike the previous cases, only the third dataset is used here since the usability of the architecture is only plausible with this dataset.

Table 5.23: FT vs FRFT Comparison for Siamese Models

Model	Transform	F1 Score (%)	Accuracy (%)	$\bar{\alpha}$
Siamese CNN	FT	85.6 ± 5.5	87.0 ± 4.8	1.000
	FRFT	85.8 ± 5.8	87.4 ± 5.5	0.997
Siamese LSTM	FT	85.1 ± 4.3	86.6 ± 4.2	1.000
	FRFT	86.0 ± 5.0	87.5 ± 4.3	0.999
Siamese GRU	FT	85.8 ± 4.9	87.1 ± 5.0	1.000
	FRFT	86.9 ± 5.9	88.3 ± 4.2	1.001

The results indicate that the proposed trainable fractional orders beat the ordinary Fourier transform in each scenario, similar to the previous cases with the accuracy improvements [0.4, 1.1, 1.2], and F1-scores [0.2, 0.9, 1.1] as percentages, which are provided with the order of the used units shown in the results table. Thus, the suitability of the proposed approach is still valid in this scenario.

Although a detailed analysis of the unit's performance is provided in the next chapter, it might be stated here since the single table shows the overall characteristics. The GRU-based architecture is more successful than the LSTM-based architecture, as observed in the previous experiments. However, unlike the earlier cases, the CNN architecture with three blocks beat the LSTM-based architecture in the Siamese case, which is not observed in the single-branch experiments. Therefore, it is likely claimed that CNN-based architecture benefits more than RNN-based architecture.

As a final observation, the Siamese architecture outperforms all single-branch models applied separately to the radar data in the previous scenarios in both the conventional and proposed approaches. To indicate that clearly, results shared in the Tables 5.2, 5.8, 5.14 might be considered. In the CNN single-branch case, the different radars indicate the performance of [68.2, 56.6, 67.6], which is increased to 85.8 in terms of the F1-score with the proposed usage of the FrFT layers, and a similar improvement is acquired with the conventional Fourier transform. In the LSTM-based architecture, the F1-scores are increased to 86.0 from [72.0, 80.6, 74.4] with the same radar order. If the last GRU unit is considered,

the F1-scores are increased to 88.3 from [72.9, 82.8, 75.9]. Therefore, the performance is increased significantly in all scenarios, especially for the CNN-based architecture.



Chapter 6

Discussion

This chapter of the thesis aims to provide a comprehensive analysis of the experimental results shared in the previous chapter, including results by considering the experimental setups and scenarios. This analysis includes the overall model performance, including single-branch and multi-branch architectures. The effect of the completed augmentation and configuration studies is also interpreted by adding additional experiments about the classification performances without using all classes in the dataset to indicate the consistency of the work with the previous literature. The examination of the detailed interpretation of the experiments, including the drone classification dataset, follows these examinations. Finally, the chapter is completed by comparing the results with the base studies and a criticism of the work conducted within the scope of the thesis.

6.1 Single-Branch Models

In the single-branch models, three types of deep learning units and methods are examined systematically using the same training strategy across all six datasets. In this section, instead of focusing on experiments separately, which were conducted in the previous part, common tendencies are shared with the cooperative

analysis.

6.1.1 Performance Comparison

Firstly, it should be stated that the proposed methodology, including the trainable fractional Fourier transform, in all single-branch architectures, causes performance increments. As described in the second chapter, the fractional Fourier transform might be considered the generalized version of the ordinary Fourier transform, or it might be claimed that the proposed approach includes the ordinary Fourier transform. Therefore, the improvement caused by the fractional Fourier transform may be appraised as an expected outcome. All the results might be seen clearly in the Tables 5.2, 5.8, and 5.14.

In the experiments, as stated several times in the previous chapter, the three-block representation is utilized among all experimental scenarios. However, the configuration study about the investigation of the block numbers indicates that the best performances are obtained via block numbers different from three. The primary motivation behind using the three-block architecture is to work with models that are not ideal and leave opportunities for improvement. Although it cannot be interpreted as completely fair, all architectures are implemented with the same number of blocks for comparison. Let's consider the results of these three block representations, which were given in the tables mentioned in the previous paragraphs. The GRU-based model is better than the LSTM-based model, and the worst model is the CNN-based model in terms of accuracy. This phenomenon is observed in all six examined datasets. The aforementioned table is reconstructed into Figure 6.1 for a more visible demonstration by considering the F1-scores.

As shown in the figure, the performance differences indicate similar behavior among all the datasets, especially for the RNN-based architectures. However, if we consider the best performances obtained in the block count investigation, provided in the Tables 5.3, 5.15, and 5.15, the CNN model outperforms the other RNN architectures. Therefore, for our task, including the well-known deep

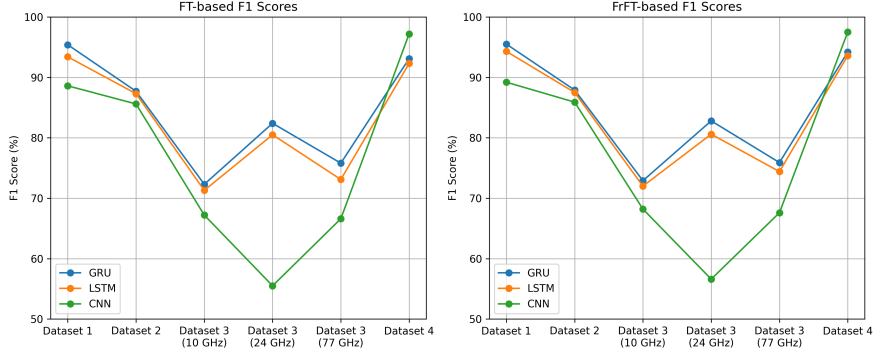


Figure 6.1: Accuracy Comparison of Single-Branch Architectures

learning units, CNN is preferable if the computation time and complex models are required. However, if the model complexity is a limitation, RNN-based structures are more preferable. If we compare the performances of the RNN-units, it can be seen that the GRU-based application beats the LSTM-based application in all scenarios presented in this thesis.

6.1.2 Behavior of the Fractional Orders

In addition to the performance comparison, the investigation of the fractional orders might be conducted from the experimental set. In the previous chapter, the behavior of the fractional orders is stated in specific experiments. As a reminder, in all experiments, the fractional order is set to 1 as an initial value, except for the analysis of the initial order effect to emphasize the positive impact of the proposed approach compared to the conventional Fourier transform. This section comprehensively examines the fractional order's behavior in different architectures and datasets in experimental scenarios.

As a first observation, in all conducted experiments with the initial value set to 1, fractional orders tend to fluctuate around 1, and there is no apparent convergence to a different fractional order. In the reported examples, the mean of the final fractional orders is reported among the completed 50 trials; the changes in those 50 trials should be examined and highlighted to indicate the fluctuation. In the experiments reported in the Tables 5.2, 5.8, and 5.14, the alpha order

values of each trial are given in the Figure 6.2 by considering the second dataset. Although the mean of the final fractional orders might be considered a tendency, there is no specific convergence to another fractional order, and the convergence depends on the parameters related to the seed, such as initialization weights and dataset splitting into training, validation, and test sets.

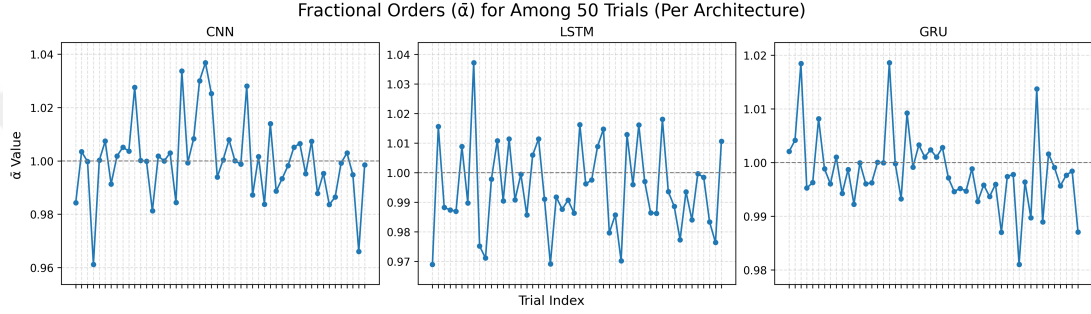


Figure 6.2: Learned FrFT Orders Among 50 Trials of Single-Branch GRU Model Experiment with Second Dataset

Also, in addition to the experiments about comparing FT and FrFT approaches, the fluctuations around one are observed in all experimental sets, including configuration and augmentation strategies, if the final mean fractional orders are considered among the experiments. There are no drastic changes between the initial and final fractional orders. Also, in the same dataset, the final mean fractional order may be lower or higher than the initial values, which indicates the dependency on the seed, similar to the mentioned characteristic in the previous paragraph. However, some characteristics might be obtained as provided in the following explanations.

In a more detailed way, if we consider the single-branch CNN-based experiments, the following results may be obtained by considering all the experiment sets reported in the previous chapter, including the final mean fractional orders, to investigate a possible tendency of the datasets in the CNN-based architecture compared to the Fourier transform. The total number of reported experiments is 33, without the initial FrFT order experiment that cannot be integrated into this part. 7 experiments from block counts, 10 experiments from window-overlap sizes investigation, 11 experiments from Mix-up augmentation, 4 from stretching augmentation, and 1 from the FT vs FrFT experiment for all datasets, which are

concatenated in the figures described in this paragraph. Therefore, the overall final fractional orders might be shown in Figure 6.3. In a similar fashion, the same analysis might be carried out for the LSTM-based and GRU-based architectures as provided in Figures 6.4 and 6.5; however, it should be remembered that the block count experiments in RNN-based architectures are completed with the five blocks; thus, the total number of experiments is 31 in these cases.

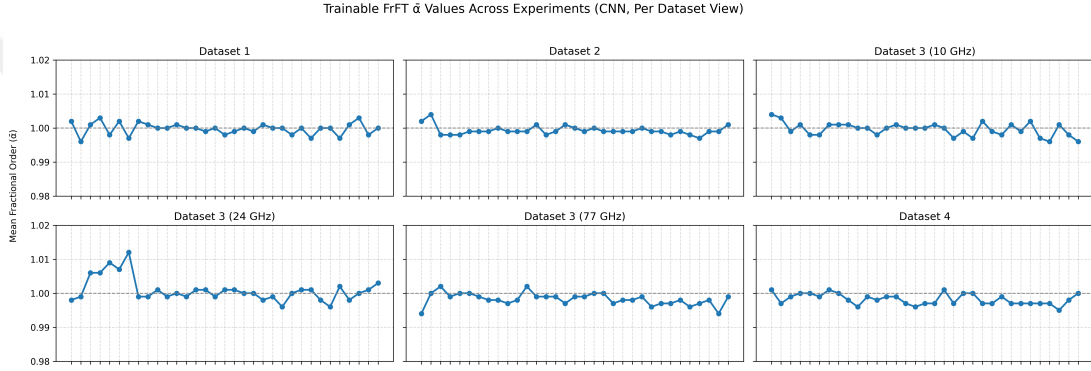


Figure 6.3: Final Mean Fractional Order Among All Experiments in Single-Branch CNN Architecture

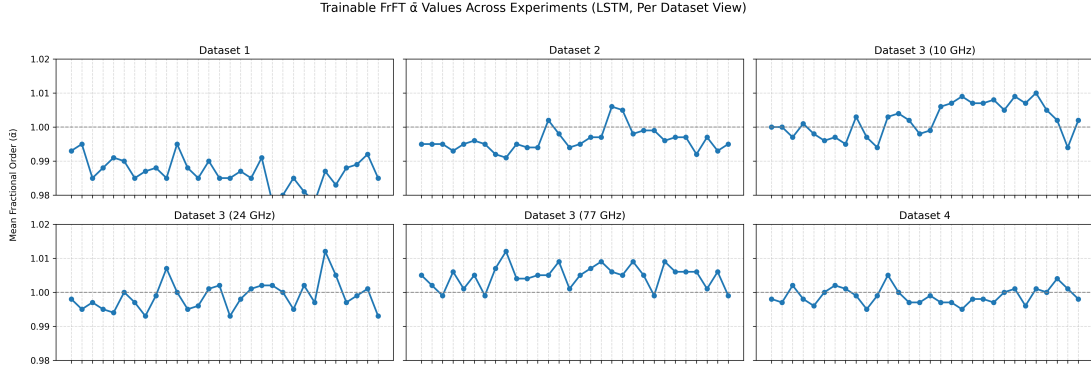


Figure 6.4: Final Mean Fractional Order Among All Experiments in Single-Branch Bi-LSTM Architecture

Therefore, although the learned fractional orders are highly dependent on the seed, as shown in Figure 6.2, the overall examination of the learned orders across whole experimental sets indicates that there is a tendency for the datasets for fractional orders compared to conventional Fourier transform, represented by the fractional order 1. Since these tendencies of the classes are preserved in all architecture types, it may be concluded that a better representation of the classes

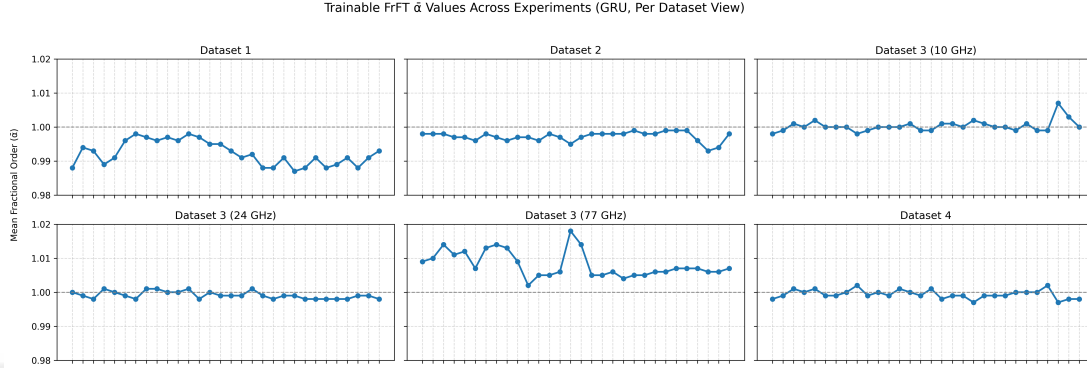


Figure 6.5: Final Mean Fractional Order Among All Experiments in Single-Branch Bi-GRU Architecture

in time-frequency representations exists compared to the traditional representation also, since the examined datasets include different types of radars and collection methods, the relation between the radar types and the most suitable time-frequency representation might be considered as a promising research topic.

Another key observation about the different time-frequency representations might be obtained from the experiments that analyzed the different initial order effects on the model performances. As briefly mentioned in the previous chapter, the initial fractional orders considerably influence the model's performance, and the initial values closer to 1 lead to higher accuracies, which were described as triangular to rectangular shapes with respect to the initial order in the related sections. The aforementioned behavior might be summarized by considering the Tables 5.7, 5.13, and 5.19, using the Figures 5.1, 5.2, and 5.3. In these representations, although some outliers might be observed, it might be seen that the best accuracies are obtained with the initial value closer to 1. Therefore, our results indicate that although there exist better time-frequency representations for micro-Doppler signature classifications other than the conventional approaches, these candidate representations obtained from the fractional Fourier transform are close to the conventional Fourier transform, as we consider the relation between the FT and FrFT as described in the second chapter.

Additionally, in the experiments related to the initial fractional order values, there is a convergence to 1, which is stated several times. To emphasize this

effect, the differences between the learned and initial orders are provided for all architectures and datasets in Figure 6.6, similar to the accuracy analysis provided above. It might be observed that the common characteristic across all scenarios is the tendency of the initial fractional orders—being far from 1—to approach 1 gradually. However, the convergence rate slows down as they get closer to 1. Therefore, it might be inferred that the models suffer from local minima situations in the initial orders closer to 1, since the distinctiveness of the fractional orders is adequate, and models do not try to learn other fractional orders.

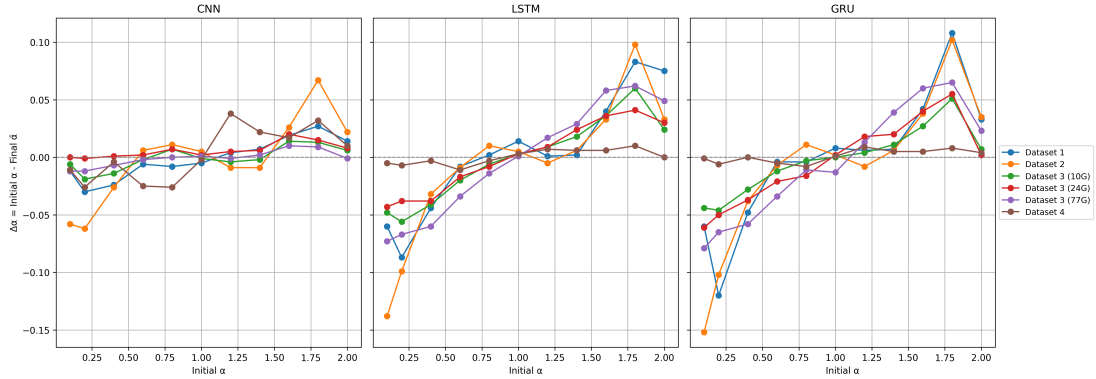


Figure 6.6: Initial FrFT Order Behavior in All Architectures

In order to investigate the observed characteristics by eliminating the explained local minima problem, several additional experiments are conducted with different datasets by increasing the learning rates for the fractional order and altering the loss functions to penalize the models if the learned order is closer to the initial order, but no meaningful results are obtained. Therefore, the results of these experiments are not reported to avoid confusion. However, in light of our findings, the more detailed comparisons, including different time-frequency representations obtained with the fractional Fourier transform, are an interesting topic.

6.1.3 Verification of the Learned Orders

Although better performances are obtained in the comprehensive experiments, an additional experiment using the second dataset with the GRU model is conducted to validate the results. As mentioned in the previous section, datasets

indicate a slight tendency to fractional orders different than the conventional Fourier transforms; thus, the experiment, including the investigation of the initial fractional order, is conducted. The models are trained with the trainable fractional Fourier transform blocks and tested with both the learned and initial orders, with the learned network weights obtained from the training session. The results demonstrate that the proposed approach provides improvement compared to fixed time-frequency transforms. However, the high variances and the dependency of the fractional orders on the model initialization steps were examined in the previous section. The comparison is provided in Figure 6.7, instead of a table representation, to increase the readability. Therefore, there is a difference in model performance, since the model weights are trained to work with learned fractional orders instead of initial fractional orders.

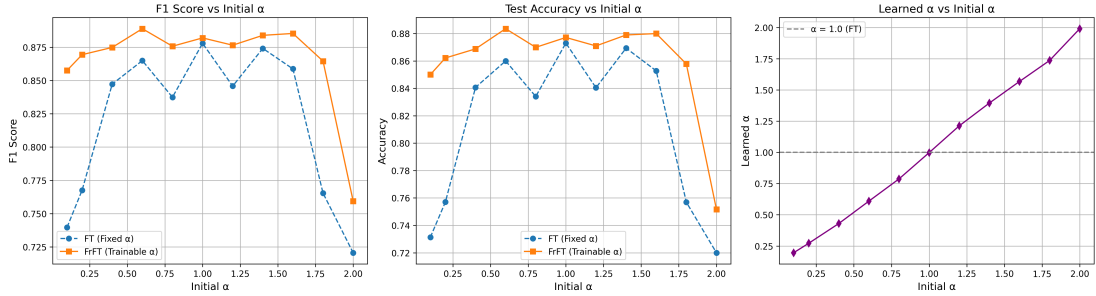


Figure 6.7: FT vs FrFT with Initial Different Fractional Orders (Only Testing)

Also, by considering the same methodology followed in the FT vs FrFT experiments, by using the GRU-based architecture with the second dataset again, another additional experiment is conducted to verify the outcomes of comparing the fixed time-frequency transforms and the trainable version proposed by this thesis. In this additional experiment, instead of using trainable FrFT blocks, the fixed FrFT orders are used in both training and testing. Therefore, this additional experiment may be seen as a generalization of the FT vs FrFT experiment, including comparison of the trainable and constant blocks in addition to the 1. The result of the experiment is provided in Figure 6.8. It might be seen that, as observed in the FT vs FrFT experimental cases, the improvement provided by our approach is valid for all initial fractional-order cases. Therefore, as stated several times in the previous chapters, the proposed approach is superior to other time-frequency representations, such as WVD, which is related to the fractional

Fourier transform.

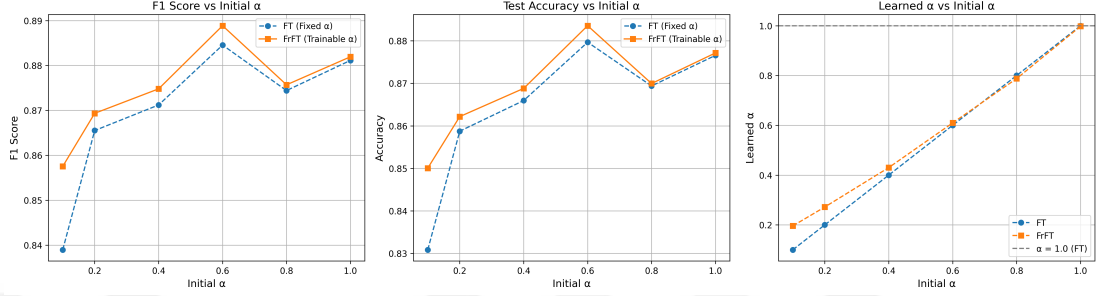


Figure 6.8: FT vs FrFT Generalization Experiment

One final additional experiment is obtained by utilizing the second dataset and a single-branch GRU architecture to highlight the fractional order effect. In this scenario, models are trained with an initial fractional order value 0.5 and 1.0, and tested as conducted in the previous experiments. The used models are saved, and the model weights are frozen except the trainable fractional order layers. The training and testing procedures are repeated, only updating the fractional order with different initial values. The results are shared in Figure 6.9, and Figure 6.10 includes three sub-figures: (a) denotes the fractional order behavior in the first training step, and the final learned fractional order, (b) denotes the fractional order behaviors in the second step, which includes only training the fractional order with frozen weights, and (c) denotes the F1-score values of the different fractional orders. In both scenarios, the fractional orders converge to the learned order values if the initial value is not too far away from the learned order. It may be seen as an indication where the alpha orders are updated with the model weights, and they are bound to each other. Therefore, as accuracy values indicate, a network trained with a fractional order cannot work with another fractional order properly.

6.2 Multi-Branch Models

In the multi-branch configurations, their performance analysis is provided in more detail than in the single-branch experiments in the previous chapter, since the

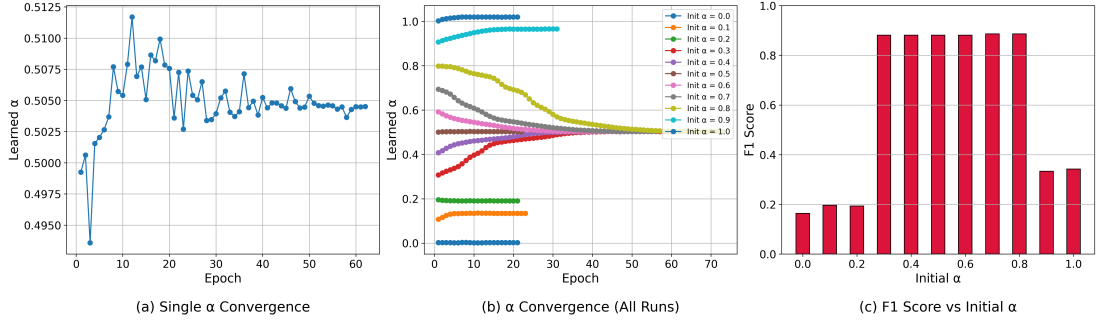


Figure 6.9: Trainable Fractional Order Behavior with Frozen Weights and Initial Value 0.5

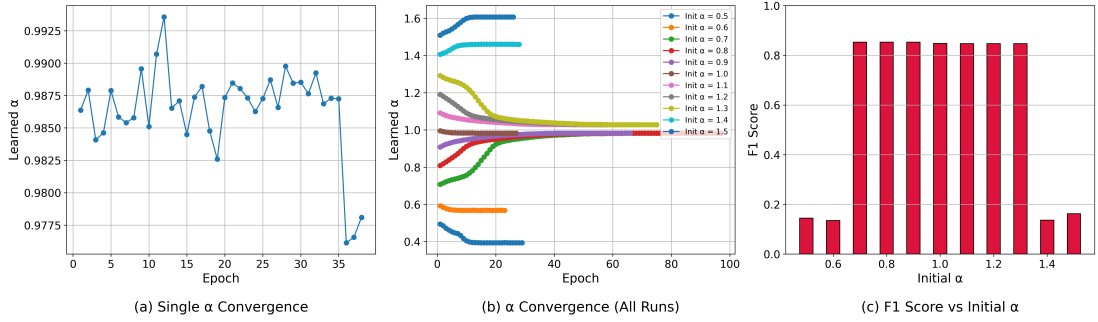


Figure 6.10: Trainable Fractional Order Behavior with Frozen Weights and Initial Value 1.0

primary purpose of constructing these architectures is to demonstrate plausible improvements by utilizing alternative contributions. As examined in the single-branch architecture, comparative analysis of multi-branch structures should also be discussed in this chapter, in addition to the observations made in the previous chapter.

As mentioned in the second chapter, the fractional Fourier transform allows the analysis of different fractional domains by changing the fractional order parameter in the transform. The effect of different fractional orders, or relatively different time-frequency representations, on performance is analyzed with single-branch architectures by changing the initial fractional orders as provided in the earlier chapters. These experiments demonstrate that the model performance may be better in different initial fractional order values rather than 1.0. However, the best performance is generally acquired in 1.0 as given in the Figures 5.1, 5.2, and 5.3. Thus, simultaneous usage of different fractional usage is aimed to increase

improvements in the performances, which is observed generally in our experiments as given in the Tables 5.20, 5.21, and 5.22.

In the literature, multi-branch structures may be found, such as parallel feeding of range-Doppler and micro-Doppler spectrograms, radar data matrix and micro-Doppler spectrograms, WVD and micro-Doppler spectrograms [38, 134]. However, this thesis first introduces the use of different time-frequency representations simultaneously obtained from the fractional Fourier transform. In addition, instead of using fractional orders directly for different time-frequency representations, the trainable version is applied, another novelty proposed by this study. As explained earlier, the proposed approach is constructed with different deep learning units, including CNN, LSTM, and GRU types. The fractional orders are chosen symmetrically around 1.0 to utilize both time and frequency components of the signal simultaneously with balanced influence, and only odd numbers of branches are examined due to the same symmetrical reasoning. The branch count and the initial order selections are interesting subjects for further studies.

The overall results indicate a discrepancy between the CNN-based and RNN-based architectures. In the CNN-based case, adding more branches only improves the performance on the first and 10 GHz third datasets, whereas in the RNN-based architectures, adding more branches significantly improves each case. However, the optimal branch count differs for the RNN unit and the dataset type. Similar to the findings of the single-branch architectures, this discrepancy may be explained by considering the working principle of the units. Since the CNN-based architecture tries to learn spatial features with the convolutional processes, the simultaneous employment of different time-frequency distributions may require more complex CNN-based architectures. The different fractional order effects are shown in the 4.3, since the spectrograms constructed with ordinary Fourier transform are highly disrupted by the fractional representations, the need for more complex CNN architectures might also be claimed. Unlike the CNN-based architectures, the RNN-based architectures focus on the variation in the time-series representation; thus, even with the lighter models, the positive effect of using different time-frequency representations might be observed, as shown in

the overall results. Also, the performance differences of utilized units are similar to the single-branch case, where the GRU-based architecture indicates the best performance, and the CNN-based architecture indicates the worst. However, identical to the analysis of the single-branch experiments, the CNN architecture is constructed with three blocks, and direct comparison might be seen as unfair since the CNN-based architecture cannot achieve its best performance with three blocks.

In light of the explanations given in the previous paragraph, to evaluate the CNN architectures with more complex settings, an additional experiment only using the second dataset with five blocks is conducted, and results are reported in Table 6.1 with a maximum of five blocks. The block count is chosen from the single-branch configuration study where the best performance is obtained. Compared to the results obtained in the three-block representation case provided in Table 5.20, there is a clear improvement, consistent with the results in the CNN architecture block count experiment. Also, a slight improvement may be obtained with the combination of the branches as shown in the table; however, it may be considered as insufficient to indicate that the multi-branch fusion approach is valid for all CNN cases, unlike the RNN-based architectures.

Table 6.1: CNN Multi-Branch Fusion Model with 5 Blocks Architecture in Dataset 2

Branch Count	F1	Acc
1	89.1 ± 2.1	88.6 ± 2.1
3	89.2 ± 2.4	88.6 ± 2.4
5	89.0 ± 1.8	88.5 ± 1.8

As presented earlier, the second multi-branch architecture examined in this thesis is Siamese-based. The primary motivation behind the architecture is provided in the second chapter, which uses more information without increasing the model parameters. To use this type of architecture, different sensors collecting the same scene are required; thus, only the third dataset might be helpful for this type of architecture. In the literature, siamese-based architecture usage with the same dataset can be found, and will be considered in the "Comparison with Work in the Literature" section [10,38,135,136]. Several characteristics might be

observed if we consider the experiment outputs provided in Table 5.23. First, the unit-based comparison supplied for the previous architecture might be conducted since the performances might be sorted as GRU-LSTM-CNN in decreasing order, a common phenomenon whose interpretation has already been provided. Secondly, as given in the fifth chapter, the Siamese-based architecture improves each case compared to the single-branch architecture.

However, a direct comparison of single-branch architectures with the siamese model can be perceived as unfair; therefore, an additional experiment is conducted to determine the architecture’s performance more suitably. In this scenario, the GRU-based architecture is utilized with the second dataset, where the three single-branch architectures are operated simultaneously with majority voting instead of comparing them separately with the siamese-based structure. The results are provided in Table 6.2, with the mean final fractional orders of different branches from 10 GHz radar to 77 GHz radar datasets. Compared to the results of the siamese model, provided in Table 5.23 as 86.9 F1-score and 88.3 accuracy value, the majority voting provides worse performance. Therefore, it may be claimed that the siamese architecture improves the performance of single-branch architectures, even when the majority using method is employed.

Table 6.2: Multi-Branch GRU Model with Majority Voting

Model	F1 (%)	Acc (%)	$\bar{\alpha}_1$	$\bar{\alpha}_2$	$\bar{\alpha}_3$
Siamese GRU	86.3 ± 4.8	87.8 ± 4.4	1.000	1.003	1.004

In addition, as shown in the 5.23, the fractional order is also considered a shared parameter in the training, and the final mean fractional order is presented. To observe the effect of the only shared weights, the fractional orders are left as an unshared parameter across the branches; therefore, we left the model to use its fractional orders to investigate the possible differences. Similar to the experiment explained above, only the GRU-based architecture is considered in this scenario. As presented in the results given in the Table 6.3, where the $\bar{\alpha}_1$ denotes the fractional order of 10 GHz, $\bar{\alpha}_2$ denotes the fractional order of 24 GHz, and $\bar{\alpha}_3$ denotes the fractional order of 77 GHz radar data. Although the F1-scores are the same, there is a degradation in the standard deviation, which may be considered

a sign of an increase in stability. However, the test accuracy values are worse in this scenario in both the mean and standard deviation metrics. Therefore, it may be claimed that the shared fractional order benefits the model, like the other model weights. Also, the fractional order tendencies demonstrated in the Figure 6.5 are preserved in this scenario, where the final fractional order of the 77 GHz radar is slightly higher than 1.0, whereas the other two radars tend to lower values than 1.0.

Table 6.3: Siamese GRU Model without Shared Fractional Order

Model	F1 (%)	Acc (%)	$\bar{\alpha}_1$	$\bar{\alpha}_2$	$\bar{\alpha}_3$
Siamese GRU	86.9 ± 4.8	88.1 ± 4.6	0.999	0.999	1.003

6.3 Configuration and Augmentation Effects

In the previous section, a comprehensive analysis of the model performances and class-based characteristics was provided, mainly considering the outcomes of the main contribution of this thesis, including the trainable fractional Fourier transform usage. Since the proposed approach is analyzed by utilizing several additional techniques, which are provided as configuration and augmentation strategies, overall observations obtained from these should be appropriately examined, as given in this section.

6.3.1 Analysis of Configuration Studies

This study’s configuration study mainly includes two investigations: the effect of the model complexities on block counts and the window-overlap lengths. In the ”Experiments and Results” chapter, a brief analysis of these methods is conducted, primarily considering the experiment itself; however, a more comprehensive study is performed by considering the overall characteristics observed through all covered architectures.

The first configuration study includes the effect of the model numbers in different single-branch architectures. As stated in the relevant sections, the block numbers are investigated until no additional improvements can be achieved. Therefore, the CNN-based architecture is analyzed until seven blocks, whereas the RNN-based architectures are analyzed until five. As shown in the Tables 5.3, 5.9, 5.15, the behavior of the models is similar across the models; therefore, the observed characteristics of the optimum block numbers might be associated with the theoretical backgrounds of the utilized deep learning units.

The CNN-based architecture reaches the peak performance around the 5-6 blocks, whereas the RNN-based architectures reach their peak around 2-3 blocks. This phenomenon is related to the learning principle of the learning methodology of the CNN blocks, known as hierarchical feature learning in the literature. In each convolutional block, the features of the input data are extracted in a hierarchical order, such that the first layer learns the basic features like edges and textures, and the following layers learn the complex patterns like shapes or relationships within the data sections. Since the spectrograms might be considered time-series representations, the CNN-based architecture needs deeper layers to reveal the time-related features, unlike the RNN-based models. As explained in the second chapter, RNN-based architectures are designed for time-series analysis, and their working principle is more suitable for this data type since the time-related features are stored in the units themselves. In the developed models, the time-related features are preserved and even further enhanced thanks to bidirectional architectures, which allow the network to capture both past and future dependencies in the temporal domain. Therefore, considerably successful performances might be obtained even with a single block.

The comparison of the peak performance is mentioned in the first section of this chapter, where CNN-based models can beat the RNN-based models with an adequate number of blocks. This situation indicates that CNN-based architectures with enough complexity can conduct more successful spectral calculations compared to RNN-based models. If we compare the RNN-based architectures, although they are very close, the LSTM-based models reach the peak value with fewer block counts, which can be seen in the second dataset and all radars in the

third dataset, as shown in the aforementioned tables. As described in the second chapter, the GRU units are a simplified version of the LSTM units. Therefore, LSTM-based architectures are inherently more complex than the GRU-based architectures, which leads to architectures reaching peak performance with fewer blocks.

The other phenomenon obtained in this study is degradation in the performance after some block counts in all architectures among all datasets. This situation occurs since the more complex models try to learn features that are not beneficial for the classification task, which leads to overfitting. To overcome overfitting, several techniques such as changes in the training session or usage of dropout layers might be utilized; however, to provide a comprehensive analysis, we have used the same methodology with the same training procedure. Also, the vanishing gradient problem occurs in RNN-based architectures with deeper architectures, which is caused by the recursive nature of information propagation. As the results indicate, the LSTM-based architecture is more fragile, and the increment in the block numbers causes more drastic degradation in the performance compared to the GRU-based architecture.

Another configuration study in this thesis investigates the window and overlap sizes used in the segmentation procedure for the short-time fractional Fourier transform calculation. As mentioned in the second chapter, selecting the window and the overlap length is crucial since this selection determines the balance between the temporal and spectral resolution. Although it is not directly related to the work covered in this thesis, a comprehensive analysis seems necessary for a better understanding of the mechanism of the time-frequency representations. Since the spectrograms are created using these values, In the literature, various studies might be found focusing on the determination of the optimum window length concerning the specific signal classes or representations; however, to preserve the focus of the study, we have only applied a grid-search-based methodology to analyze the effect of the selection on the model performance by considering the some studies working on the micro-Doppler signature classifications. Additionally, in the work representing the trainable FrFT approach, the window length selection is also highlighted [13].

The overall results, including single-branch architectures, are presented in the Tables 5.4, 5.10, and 5.16. The results indicate some common patterns related to the used architecture type, which is unrelated to the dataset-specific behaviors. If we compare the CNN-based and RNN-based architectures, it might be concluded that the CNN-based architecture is more sensitive to the window-overlap sizes. However, since the slow-time direction has 256 samples in the drone dataset, the effect of the window-overlap sizes might be obtained directly with the drastic changes in all architectures.

In all datasets, common behavior in the CNN-based architecture is the increment with the window size and the decrement with the overlap length. As the larger windows bring better frequency resolution, the produced micro-Doppler signatures include more distinctive features captured by the convolutional blocks. On the other hand, while temporal overlapping is theoretically beneficial for reducing spectral artifacts—as discussed in Chapter 2—it introduces redundant and highly correlated spectrogram segments. In CNN-based models, this leads to oversampling in the spatial domain, where the network overemphasizes repeated, less informative content rather than learning class-specific discriminative features. In other words, in CNN-based architecture, which cannot inherently benefit from the sequential context, unlike RNN-based architectures, the construction of the micro-Doppler segments is vital for model performance.

In the RNN-based architecture, there is no clear evidence that the window-overlap sizes affect the model performance characteristically. As provided in the previous paragraph, since the RNN-based architectures are able to cover sequential data, the selection of the window-overlap sizes cannot affect the performance directly. In addition to the lack of common behavior among the datasets, even within a class itself, it does not indicate specific behavior. In these experimental cases, RNN-based architectures indicate apparent robustness against the parameters of the windowing transform procedure. Analyzing the micro-Doppler signatures with the RNN-based architectures might also provide a characteristic similar to the CNN-based architecture; therefore, it may be considered a possible research topic. Also, it should be highlighted that, in the experiments, no correlation between the final mean fractional orders and the window-overlap settings

is observed.

6.3.2 Analysis of Augmentation Methods

As shown in the previous chapters, two different augmentation strategies are examined in this thesis's scope: Mix-up augmentation and image stretching. In this section, the outcomes of these applications are discussed and supported with additional experimental sets by considering the previous studies utilizing the same strategy.

The first examined augmentation method is Mix-Up augmentation, which is introduced and explained in the second chapter in a detailed way by considering the original work [74]. In light of the original study, instead of examining the proposed augmentation strategy with the fixed distribution, different beta distributions are also considered, as stated several times before this chapter. In the experimental set, to avoid confusion, no Mix-up applied control experiment is also conducted, which is denoted with the 0.0 in the Tables 5.5, 5.11, and 5.17 with the different beta distributions whose parameters are given in the table.

The overall results indicate that the augmentation strategy positively influences the fourth dataset in all architectures, whereas the second dataset's performance only increases in the RNN-based architectures. However, in other datasets, using the Mix-up augmentation considerably decreases the model performance. As stated in the original work and the previous chapters, the positive effect of the augmentation strategy might be changed with the beta distribution parameters, which occurs in our case. For instance, if we consider the second dataset in the Table 5.17, model performance decreases after some beta distribution parameters, indicating consistent behavior with the original work [74].

The second augmentation strategy followed in this thesis study is stretching, which may be seen as a subsection of the warping augmentations. The application procedure is explained in the second chapter, and the results are provided in the previous chapter in the same form as the other experimental sets in the

Tables 5.6, 5.12, and 5.18. The overall results demonstrate discriminative behavior in the CNN-based and RNN-based architectures. The method improves the second and third datasets with 10 GHz radar data in the CNN-based architecture. In contrast, no improvement is obtained in the RNN-based architectures. The working principles of the units and the method used to augment them may explain this situation. In the augmentation method, if the original data is shrinking, zero-padding is applied to provide a fixed-size input to the model. In the CNN-based architecture, the spatial features are learned, and zero-padding does not influence the model performance negatively; whereas the RNN-based models learned the features by considering the sequential orders; thus, zero-padding creates disruptions in the learning mechanism.

To overcome this issue, an additional experiment is conducted with the second dataset with the RNN architectures by only applying the stretching in the related dimensions. The results are provided in Table 6.4. Therefore, no zero-padding is applied to the input data; only stretching is used, and inputs are cropped to a fixed size as an augmentation strategy. Results indicate that the augmentation strategy with this approach might be used with the RNN-based architectures, and the zero-padding strategy is not suitable for the RNN-based architectures. Also, since the destructive effect of the zero-padding is removed, the fluctuations in the accuracy metrics are solved successfully.

Table 6.4: Random Stretching - Extension Only

Stretch Dimension	LSTM-based Architecture			GRU-based Architecture		
	F1	Acc	$\bar{\alpha}$	F1	Acc	$\bar{\alpha}$
Fast-Time	87.7 ± 2.5	87.1 ± 2.7	0.996	88.2 ± 2.1	87.7 ± 2.2	0.996
Slow-Time	86.9 ± 2.2	86.4 ± 2.3	0.997	87.5 ± 2.4	87.0 ± 2.5	0.998
Both	85.9 ± 2.7	85.5 ± 2.8	0.995	86.3 ± 3.2	85.8 ± 3.2	0.997
None	87.4 ± 2.2	86.9 ± 2.2	0.995	88.0 ± 2.2	87.5 ± 2.3	0.998

6.4 Drone Target Classification

In the previous chapter, the drone dataset was examined with the other dataset covered in this thesis without any changes specific to the dataset. However, since the dataset differs from the other classes in terms of the working principle of the

radar and the class types, analyzing the results in different sections is required.

The data collection differs from the other examined datasets, and each sample in the dataset is elementary compared to the other datasets in terms of size, which is shown in detail in the third chapter. The classification performance of this dataset with the constructed models is considerably higher than that of the other classes in all examined cases, as presented in the previous chapter. Therefore, the proposed models may exceed the requirements imposed by this dataset and may be considered incompetent since there is no need for complex models to classify this dataset.

Although the classification performances are high in all cases compared to the human classification task, the experimental scenarios demonstrate that the classification performance might be influenced by the approaches followed in the augmentation and configuration studies. Since these are examined with the other datasets in the previous sections, they are not stated in this chapter again. In light of the findings, future studies will likely use the proposed methodology in different tasks as we conducted in this dataset.

6.5 Additional Experiments with Well-Known Models

Before comparing the work completed in this thesis with the literature, the additional experiments should be shared to highlight the effectiveness of the proposed approach [13]. In the original work, where the trainable fractional Fourier transform concept is presented, experimental results were shared with the well-known models and datasets, as shown in Figure 4.2. In light of this approach, the second dataset examined in this thesis is combined with the well-known models instead of the well-known datasets.

Since this experiment is conducted to highlight the suitability of the proposed

method in the micro-Doppler signature classification, it is not covered in the experimental results chapter, which mainly includes the different architectures and methodologies for all datasets examined in the thesis. Therefore, this experiment might be seen as the expansion of the presented work with the well-known architectures that are not explicitly constructed for this work. For this purpose, the proposed training spectrogram generation method, including three three-layer configurations that include segmentation, fractional Fourier, and log-scaler layers, is combined with the ResNet-18, ResNet-34, ResNet-50, and MobileNetV2 models.

Although the aforementioned models deserve specific descriptions and examination, their complete examination is omitted to avoid diverting from the main scope. However, it is worth emphasizing that all models are used for image classification tasks in the literature, which is incompatible with the direct usage of the method proposed in this thesis. In the proposed approach, by utilizing all range-bins, the output of the trainable spectrogram block is three-dimensional, including the spectrograms created by all range-bins; however, the mentioned models are constructed for inputs having two dimensions, like images. Therefore, this experiment’s methodology is slightly changed by averaging the spectrograms created by the spectrogram generation block with the 2-dimensional images. As the related work section shared, the averaged spectrograms obtained from different range bins may be used in the literature. Thus, although comparing the proposed architecture with these models might be uneven, comparing the traditional Fourier transform and the proposed methodology is beneficial. The overall results are shared in Table 6.5.

The results in this section indicate the performance improvements that can be made by utilizing the proposed approach. As shown in the Figure 4.2, the improvements obtained by the original study are consistent with the improvements in our experiment [13]. However, a significant difference that should be highlighted here is that we have used the trainable fractional Fourier transform in the segmentation part. In contrast, the original study applied this method in the magnitude pooling layers. Also, it should be stated that the training methodology is different from the original work, where we used the same training procedure

Table 6.5: Comparison of the Proposed Method and Conventional Approach with Well-Known Models

Model	Used Method	F1-Score (%)	Accuracy (%)	$\bar{\alpha}$
ResNet-18	Traditional FT	88.45 ± 2.18	88.45 ± 2.31	1.000
	Trainable FrFT	88.57 ± 2.45	88.12 ± 2.40	0.999
ResNet-34	Traditional FT	87.95 ± 2.43	87.41 ± 2.52	1.000
	Trainable FrFT	88.89 ± 2.18	88.36 ± 2.23	0.998
ResNet-50	Traditional FT	87.57 ± 3.10	87.12 ± 3.13	1.000
	Trainable FrFT	87.77 ± 3.11	87.38 ± 2.99	0.998
MobileNetV2	Traditional FT	88.38 ± 2.01	87.88 ± 2.32	1.000
	Trainable FrFT	88.89 ± 1.33	88.40 ± 1.59	0.997

followed in this thesis, which was described in the fourth chapter.

6.6 Comparison with Work in the Literature

This section compares the results obtained with those in the literature. Since the examined datasets might be used for various studies, the comparison is mainly conducted by considering the original works that published the datasets. However, especially for the second and third datasets, comparisons are enhanced by considering the studies that provide improvements compared to the original works we can find. The provided dataset order will be regarded from first to fourth for a readable comparison.

In the study [101], utilizing the first dataset, the classification performance is considered by considering several aspects, including additive noise, radar position, and target distances, with a novel denoising approach. However, their shared dataset does not include the aspect angle variation and different noise levels. Thus, in the comparisons, their experiments are analyzed by considering the results obtained by experiments without additional noise. Also, similar to our work, some configuration studies are provided; they are appropriately stated

in the comparisons. In light of these, the original work’s Tables 2, 3, 4, and 8 are considered. Table 2 of the considered study presented the model accuracies regarding the model complexity, which is similar to the configuration study conducted in the thesis, including the effect of block numbers. The best performance they provided is %89.82 in this experiment, where our CNN model, which is constructed with five blocks, achieved %97.0 as shown in the Table 5.3—also, constructed RNN blocks whose accuracies are shared in Tables 5.8, 5.14 reach %94.7, and %96.1. If we consider Table 3, including a configuration study focusing on the kernel sizes of the convolutional block with different noise levels, which is not covered in our research, their best average performance is %89.11. The best performance is %98.33, which our architectures outperform if we consider the average performance. Since we have covered the denoising approach, the reported average value might be better. In Table 4, they shared a comparative study comparing their model with the well-known architectures, which was conducted similarly to our work. In this experiment, their reported best performance is %89.1, which our models still beat. In their last table, Table 8, they shared their comparative results, again with the well-known models, with their denoising method. They indicated that their model achieves %91.61 with the noisy dataset, and %99.55 with the denoising algorithm. The combination of denoising and their model may be seen to outperform the architectures constructed in this thesis. However, since we have not covered the additional noise effect or denoising approaches, the mentioned comparisons might be seen as unfair.

In the second dataset, the collectors did not conduct a classification task with the dataset directly [36]. However, as stated earlier, the dataset is utilized in a machine learning challenge with 188 contributors, where 12 teams were chosen as finalists [97]. Therefore, the comparison will be conducted on the overall results reported in the benchmark study related to this challenge, instead of focusing on the finalist models separately. In Table IV of the benchmark study, the overall results of the finalists, including unseen data or test set performances, are shared. It should be highlighted that instead of following the evaluation methodology of the challenge, the models are trained with the shared dataset and evaluated with another dataset labeled as unseen. The overall accuracy performance of the

finalist models varies between %84 to %100. Among our experiments, the best accuracies are obtained with the five-block CNN, as shown in the 5.3, and the multi-branch GRU model with five branches, as shown in Table 5.22, as %90.2, %88.3. Thus, it may be seen that our models might also be finalists in the challenge with both FT and FrFT approaches. However, most finalist models for this challenge outperform the models constructed in this thesis. Although our main aim is to propose a trainable FrFT-based approach, instead of superior deep learning architectures for the micro-Doppler classification task, the constructed models may compete with the models in the literature.

The third dataset includes three radars, and the original work consists of a classification performance analysis based on these radars [37]. Their methodology and data usage should be appropriately examined since they did not use the simultaneous usage of all datasets, as we did in this study. Instead, they have analyzed the performances of their proposed method with CAE architecture, and cross-performances are evaluated, like training with the first radar and testing with other radars. They have also utilized some synthetic data construction, which should be considered. In their separate examination of the radar datasets with the VGG16 model, they have reported the achieved accuracies as %83.01, %85.57, and %89.23 for the 10 GHz, 24 GHz, and 77 GHz radars. In our experimental scenarios, %85.3 is achieved with the six block with CNN in Table 5.3 for 10 GHz radar, and %85.0 with the multi-branch GRU model in Table 5.22; however 77 GHz radar dataset, our accuracies cannot compete with the reported values of the original work. Table 2 of the original work shows that their proposed CAE architecture reaches maximum performances of %89.1, %86.9, and %88.5, which is again better than our results. Their best accuracy value is achieved for the 24 GHz radar as %91.5 with their CNN model among their detailed work. Our study's best performance is acquired with the siamese GRU network as %88.3 as shown in Table 5.23. Thus, it may be inferred that more sophisticated architectures can be constructed for this dataset compared to models proposed in this thesis, which are mainly built to indicate the effectiveness of the trainable fractional Fourier transform approach.

In addition, to the best of our knowledge, the most successful studies using

this dataset are considered for better comparison. Also, since we have examined siamese structures, similar methods in the literature are discussed for better analytical comparisons for the third dataset. The SiamaseHAR is proposed in the work [137] by utilizing the third dataset, with a Siamase-based architecture similar to the one constructed in this thesis. In this work, the original work [37] outperforms the siamese-based architecture by achieving %94.5. Therefore, the improvements in this thesis with the siamese-based architecture are consistent with the previous studies, where the constructed architectures required more sophisticated designs since the achieved performance is worse than both of the mentioned studies.

The fourth dataset, similar to the first dataset, only utilized in only one work; therefore, our main focus will be the outcomes of the original work [39]. As mentioned before, the original study includes other classes in the dataset, such as birds and humans. Also, they have conducted the study as an image classification task with their CNN-based architecture by using the aforementioned data structure of the samples without any additional calculations, such as spectrogram generation, as we did. In their work, they have not directly shared the performance since they have conducted several investigations about the SNR levels and class-based performances. In their conclusion, they have stated that the final accuracy values are obtained as %88.0 for drones, and %93.0 for non-drones, which may be increased to %100.0 by utilizing all datasets having high SNR values. Therefore, it may be claimed that the overall outcomes obtained in this study are similar to the original work, which is almost above %95.0 in the experimental scenarios.

6.7 Limitations and Considerations

Before drawing definitive conclusions about the presented study, some prominent shortcomings were encountered in different experimental scenarios, and the limitations should be highlighted properly. In this section, observed caveats are highlighted by considering the amount of influence on the results obtained in this thesis.

The first phenomenon in the experimental scenarios that might be assessed as a drawback is the high standard deviation in the results. The performance of the constructed models depends highly on the training data and the model initializations. To overcome this issue, the number of experiments is set to 50 for each experimental trial as described before. In the analysis provided in previous and current chapters, the focus mainly is on the average model performances regarding the F1-score and testing accuracy values; however, due to high variances, the superiority of the models cannot be determined by using statistical test calculations such as the Wilcoxon Signed-Rank Test among 50 trials. Therefore, the performance improvement proposed by the method, for instance, in the FT vs FrFT experiments, cannot directly mean that it is a superior model performance statistically.

However, in the literature, various studies suffering from the same high deviation issue might be found, especially for the studies focusing on directly spectrograms without additional techniques [8, 138, 139]. In our case, this situation might be linked to the usage of all spectrograms constructed with all range bins, and no noise removal technique is applied. Thus, although our findings indicate a general improvement in each case, techniques to reduce variances should be appropriately examined to provide more reliable outcomes, which were not achieved correctly in this study.

The second phenomenon, encountered in the experimental cases, is the assessment of the model's performance by comparing the model with those in the literature. In the related section, the problems for the comparison task are discussed briefly. However, most of the studies in the literature only focus on the preprocessed micro-Doppler signatures, like an ordinary image classification task. Since the proposed architectures construct the micro-Doppler signatures dynamically by learning the appropriate fractional orders, direct comparison cannot be assessed as completely reliable. To overcome this issue, experiments, including well-known models, were conducted; however, the constructed models in this work cannot be clearly distinguished from the literature.

Also, it should be stated that the datasets related to micro-Doppler analysis,

like the utilized datasets in this study, are considerably limited regarding the sample sizes compared to well-known image classification datasets or time-series classification datasets. Therefore, a more detailed analysis might be conducted by constructing new datasets or using different synthetic data generation methodologies. In the considered work, several approaches are provided; however, they are not covered in the scope of this thesis.

Finally, as stated several times, the spectrograms are created by considering all possible range bins, which causes a computational burden to the classification task. The above-mentioned range bin analysis should be appropriately regarded. In most of the examined work in the literature, similar methodologies are followed. Also, the window type, window-overlap size investigations could be conducted for better analysis which is only conducted by considering a grid of values with a fixed window type.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis investigates the influence of the trainable fractional Fourier transform with various model architectures and different datasets on the human activity classification task. Therefore, unlike the previous studies in the literature working with the determined time-frequency representation of the radar signals, an approach is proposed that helps to construct the best time-frequency representation of the signals with learnable fractional orders within the deep neural networks. The model architectures are separated into single-branch and multi-branch configurations to provide a trackable layout.

In the single-branch architectures, CNN-based, LSTM-based, and GRU-based architectures are constructed, and the proposed approach is widened with the two different augmentation methods, Mix-Up and Random Stretching augmentation. Also, to investigate the model parameters in detail, two configuration studies are conducted for each configuration to examine the effect of the block counts in the architectures with a different range of window-overlap size configurations. Thus, the validity of the proposed approach is demonstrated with different experimental settings.

The usage of the trainable fractional Fourier transform layers in the single-branch architectures beats the ordinary Fourier transform with %0.3 to %1.3 F1-score improvements in the CNN-based architecture, %0.1 to %1.1 in the LSTM-based architecture %0.1 to %1.1 F1-score improvements in the GRU-based architecture in different datasets whose detailed results are presented in Tables 5.2, 5.8, and 5.14. The overall results of those single-branch architectures in the aforementioned tables demonstrate that the GRU-based architecture provides better results than the LSTM-based architecture. In contrast, the CNN-based architecture offers the worst performance with the 3-block construction of the models. On the contrary, as discussed in the configuration experiments investigating the effect of the block numbers in single-branch architectures, the CNN-based architecture may beat the other RNN-based architectures with an adequate number of blocks, which is provided in the Tables 5.3, 5.9, 5.15. These results indicate that CNN-based architecture may be preferable to RNN-based architectures if the computation cost is not considered a critical component.

The investigation about the impact of the window-overlap sizes in the examined model architectures indicates that the CNN-based architecture generally performs better with the larger window sizes and narrower overlap sizes, whereas the RNN-based architectures do not indicate a clear correlation. The results are provided in Tables 5.4 The method analyzing the different configurations is conducted with a fixed grid of values to observe the effect of the parameters; however, as mentioned several times, the optimum selection of parameters might be considered as a possible research subject by considering the meaningful micro-Doppler signatures of different body parts in the spectrograms [140].

Two different augmentation strategies, mix-up augmentation and random stretching, are also employed with single-branch architectures, which are discussed in detail. The overall results of both approaches indicate no clear correlation between the architectures and datasets, where both may be beneficial in some datasets with different deep learning architectures.

The final experimental scenario with the single-branch architectures is also employed by analyzing the effect of the initial fractional order value. Although a

detailed examination is provided in the discussion chapter, two key findings may be summarized. Firstly, models generally tend to perform better with the initial orders near 1.0, and there is a tendency for the fractional orders to approach 1.0, regardless of the initial value. However, models cannot reach 1.0 directly in our experimental cases due to local minima, which indicates the situation where all fractional orders may be employed as an alternative, since meaningful results can be obtained. The second finding, by considering all experimental cases and final mean fractional order values, some classes tend to values lower or higher than 1.0, or in other words, the different datasets indicate a plausibility of better time-frequency representations compared to the ordinary Fourier transform.

In multi-branch architectures, two distinct types of networks are mainly constructed by simultaneously using all single-branch architectures, fusing different initial fractional orders and Siamese-based models. In light of the findings, the positive contributions based on the proposed approach are shown with the extension of the single-branch architectures. Especially, with the fusion of the single-branch architectures having different initial fractional orders, a novel approach is presented based on the simultaneous usage of different frequency-time representations.

In the fusion-based architecture, the simultaneous usage of different fractional orders contributes to the model performance to a considerable extent, especially in the RNN-based architectures, which were demonstrated in Tables 5.20, 5.21, and 5.22. In the LSTM-based architecture, adding additional branches with different initial orders increases the model F1-scores in the %1.0 to %5.6. In the GRU-based architecture, improvements are observed as %0.8 to %1.6 in different datasets. In the siamese-based architectures, compared to the performances of single-branch architectures, considerable increments are obtained in all deep learning units, which are provided in Table 5.23. Additionally, to give a fair comparison, the single-branch architectures are constructed by using majority voting, which was presented in Table 6.2, where the siamese-based model performs better with %0.6 higher F1-score.

To increase the reliability of the proposed approach, the mentioned experiments and models are evaluated with four different datasets, where the first dataset consists of three different radar datasets. Therefore, all constructed architectures are examined with six different datasets, but the siamese-based architecture utilizes only the third dataset across all scenarios. In addition to the human activity classification, the drone dataset is used in the fourth dataset to enhance the scope of the thesis.

The overall results of the experimental set are provided in a systematic order, and overall results and observations are provided with a comprehensive analysis. Additional experiments are also conducted based on observations of the outcomes of the examined experimental set. To increase the credibility of the proposed approach, further experiments are performed based on well-known deep learning architectures.

In all scenarios, the proposed approach beats the ordinary Fourier transform in terms of accuracy and F1-scores with variable amounts depending on the architecture and datasets. Also, the simultaneous use of the different fractional orders brings a considerable increment compared to single-branch architectures. These are reported systematically in the corresponding chapters.

7.2 Future Work

Results from a distinct set of experimental configurations demonstrate promising improvements, which are shared and discussed in detail. However, as given in the discussion chapter, some outcomes might be assessed as deficiencies, leading to several improvement directions. A list of possible improvements is provided in this section, considering the issues encountered in the experiments with the approaches in the literature. The "Limitations and Considerations" section has already presented and discussed several of these, which were not stated here again to avoid repetition.

- **Extension to Real-Time Applications:** As mentioned in the introduction, monitoring humans with radar sensors is vital for the autonomous applications covered in this thesis. The main task conducted in this work is to analyze a proposed approach with detailed experiments; however, the real-time applications should be covered in light of the findings presented in this work to adapt the findings to real-world applications by combining the radars and processors for this task, similar to the studies in literature [141, 142].
- **Data Fusion with Other Sensors:** The contribution of the several sensors is presented in this work by considering only the radars; however, it is possible to enhance the performance by combining the outputs with other sensors like cameras or wearable sensors, even with cell phones as demonstrated in the work [143]. Combining the proposed approach with similar works may increase the performance.
- **Adaptive Pre-processing Steps:** In earlier chapters, it is mentioned that several parameters are chosen as constant in this thesis; however, adaptive selection of the parameters, such as window type, size, and overlap amounts, range-bin selection, or deep learning hyperparameters, may lead to better performance.
- **Multi-Target Scenarios:** In the considered datasets, the scenes only include a single target performing a single activity; however, to stay consistent with the real-world scenarios, different experimental cases should be constructed, including more than one human or other subjects performing various activities in the same scene. Furthermore, these experimental cases might be enhanced to different distances or environmental conditions. Similar to the studies in the literature, the proposed approach might be evaluated with these conditions [91]
- **Multi-Radar Scenarios:** In this thesis, three radar scenarios are examined; however, any configuration of the multi-radar scenarios is not covered. Even in the literature, this subject remains an unexplored field. The parameters related to different experimental scenarios, such as the count of the radars,

their frequency bands, line of sight features, or even their power consumption, can be considered as a valuable research topic as an optimization problem. In addition, multi-static radars, or even quantum radars, may be used for monitoring human activity. Therefore, the novel radar technologies might be employed for this task.



Bibliography

- [1] MathWorks, “Introduction to micro-doppler effects.” <https://www.mathworks.com/help/radar/ug/introduction-to-micro-doppler-effects.html>, 2025. Accessed: Jun. 2025.
- [2] V. Chen, “Analysis of radar micro-doppler with time-frequency transform,” in *Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing (Cat. No.00TH8496)*, pp. 463–466, 2000.
- [3] A. Mohan and D. Gaitonde, “A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks,” *arXiv preprint arXiv:1804.09269*, Apr. 2018.
- [4] J. Lei, “Pattern recognition based on time-frequency distributions of radar micro-doppler dynamics,” in *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*, pp. 14–18, 2005.
- [5] G. E. Smith, K. Woodbridge, and C. J. Baker, “Template based micro-doppler signature classification,” in *2006 European Radar Conference*, pp. 158–161, 2006.
- [6] Y. Kim and H. Ling, “Human activity classification based on micro-doppler signatures using a support vector machine,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.

- [7] Y. Kim and T. Moon, “Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, 2016.
- [8] A. Shrestha, H. Li, J. Le Kernec, and F. Fioranelli, “Continuous human activity classification from fmcw radar with bi-lstm networks,” *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13607–13619, 2020.
- [9] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, “Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, 2018.
- [10] P. Suresh, T. Thayaparan, T. Obulesu, and K. Venkataramanah, “Extracting micro-doppler radar signatures from rotating targets using fourier–bessel transform and time–frequency analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3204–3210, 2014.
- [11] X. Chen, J. Guan, Z. Bao, and Y. He, “Detection and extraction of target with micromotion in spiky sea clutter via short-time fractional fourier transform,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 2, pp. 1002–1018, 2014.
- [12] X. Qiao, T. Shan, R. Tao, X. Bai, and J. Zhao, “Separation of human micro-doppler signals based on short-time fractional fourier transform,” *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12205–12216, 2019.
- [13] E. Koç, T. Alikasıfoğlu, A. C. Aras, and A. Koç, “Trainable fractional fourier transform,” *IEEE Signal Processing Letters*, vol. 31, pp. 751–755, 2024.
- [14] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar: Basic principles*. The Institution of Engineering and Technology, 2010.
- [15] M. S. Greco, J. Li, T. Long, and A. Zoubir, “Advances in radar systems for modern civilian and commercial applications: Part 1 [from the guest editors],” *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 13–15, 2019.

- [16] D. Amitrano, G. Di Martino, R. Guida, P. Iervolino, A. Iodice, M. N. Papa, D. Riccio, and G. Ruello, “Earth environmental monitoring using multi-temporal synthetic aperture radar: A critical review of selected applications,” *Remote Sensing*, vol. 13, no. 4, 2021.
- [17] Z. Gu, X. He, G. Fang, C. Xu, F. Xia, and W. Jia, “Millimeter wave radar-based human activity recognition for healthcare monitoring robot,” *arXiv preprint arXiv:2405.01882*, May 2024. Accessed: Jul. 26, 2025.
- [18] R. El Hail, P. Mehrjouresht, D. M. M.-P. Schreurs, and P. Karsmakers, “Radar-based human activity recognition: A study on cross-environment robustness,” *Electronics*, vol. 14, no. 5, 2025.
- [19] S. Z. Gurbuz and M. G. Amin, “Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring,” *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16–28, 2019.
- [20] B. Barshan and E. Koşar, “Bidirectional transfer learning between activity and user identity recognition via 2d cnn-lstm model for wearables,” *IEEE Internet of Things Journal*, pp. 1–1, 2025.
- [21] A. Yurtman, B. Barshan, and S. Redif, “Position invariance for wearables: Interchangeability and single-unit usage via machine learning,” *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8328–8342, 2021.
- [22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: A large video database for human motion recognition,” in *2011 International Conference on Computer Vision*, pp. 2556–2563, 2011.
- [23] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.
- [24] M. Ergeneci, E. Bayram, A. Yildiz, D. Carter, and P. Kosmas, “semg motion classification via few-shot learning with applications to sports science,” *TechRxiv*, 2023. Preprint.

- [25] M. S. Zediker, R. R. Rice, and J. H. Hollister, “Method for extending range and sensitivity of a fiber optic micro-doppler ladar system and apparatus therefor,” Dec. 8 1998. US Patent 5,847,817.
- [26] R. Kleinman and R. Mack, “Scattering by linearly vibrating objects,” *IEEE Transactions on antennas and propagation*, vol. 27, no. 3, pp. 344–352, 1979.
- [27] A. L. Kachelmyer and K. I. Schultz, “Spectrogram processing of laser vibration data,” in *Applied Laser Radar Technology*, vol. 1936, pp. 78–89, SPIE, 1993.
- [28] Y. Kim and H. Ling, “Human activity classification based on micro-doppler signatures using an artificial neural network,” in *2008 IEEE Antennas and Propagation Society International Symposium*, pp. 1–4, 2008.
- [29] F. Fioranelli, M. Ritchie, and H. Griffiths, “Bistatic human micro-doppler signatures for classification of indoor activities,” in *2017 IEEE Radar Conference (RadarConf)*, pp. 0610–0615, 2017.
- [30] Y. Kim, I. Alnujaim, and D. Oh, “Human activity classification based on point clouds measured by millimeter wave mimo radar with deep recurrent neural networks,” *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13522–13529, 2021.
- [31] Z. Wang, Z. Yu, X. Lou, B. Guo, and L. Chen, “Gesture-radar: A dual doppler radar based system for robust recognition and quantitative profiling of human gestures,” *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 1, pp. 32–43, 2021.
- [32] R. J. de Jong, M. J. C. Heiligers, J. J. de Wit, and F. Uysal, “Radar and video multimodal learning for human activity classification,” in *2019 International Radar Conference (RADAR)*, pp. 1–6, 2019.
- [33] T. Zhang, X. Qiao, X. Li, and C. Bai, “Radar feature analysis of human activity recognition under multiview scenes,” *IEEE Sensors Journal*, vol. 24, no. 14, pp. 21997–22010, 2024.

- [34] H. Li, A. Shrestha, F. Fioranelli, J. Le Kernec, H. Heidari, M. Pepa, E. Cipitelli, E. Gambi, and S. Spinsante, “Multisensor data fusion for human activities classification and fall detection,” in *2017 IEEE SENSORS*, pp. 1–3, 2017.
- [35] B. Nguyen, “Fmcw radar dataset,” may 2024. Available: https://figshare.com/articles/dataset/FMCW_radar_dataset/25874515.
- [36] F. Fioranelli, S. A. Shah, H. Li, A. Shrestha, S. Yang, and J. Le Kernec, “Radar sensing for healthcare,” *Electronics Letters*, vol. 55, pp. 1022–1024, Sept. 2019.
- [37] S. Z. Gurbuz, M. M. Rahman, E. Kurtoglu, T. Macks, and F. Fioranelli, “Cross-frequency training with adversarial learning for radar micro-doppler signature classification (rising researcher),” in *Radar Sensor Technology XXIV*, vol. 11408, p. 114080A, International Society for Optics and Photonics, 2020.
- [38] M. Ege, “Human activity classification with deep learning using fmcw radar,” Master’s thesis, Bilkent University, 2022. Accessed: July 2025.
- [39] A. Karlsson, M. Jansson, and M. Hämäläinen, “Model-aided drone classification using convolutional neural networks,” in *2022 IEEE Radar Conference (RadarConf22)*, pp. 1–6, 2022.
- [40] M. I. Skolnik, *Radar Handbook*. Boston: McGraw–Hill, 3rd ed., 1990.
- [41] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & systems (2nd ed.)*. USA: Prentice-Hall, Inc., 1996.
- [42] G. Wade, *Signal Coding and Processing*, vol. 1. Cambridge, UK: Cambridge University Press, 2 ed., Sept. 1994.
- [43] H. M. Ozaktas, M. A. Kutay, and Z. Zalevsky, *The Fractional Fourier Transform: With Applications in Optics and Signal Processing*, vol. 39 of *Wiley Series in Pure and Applied Optics*. New York: Wiley, 2001.
- [44] L. Cohen, *Time–Frequency Analysis*. New York: Prentice Hall, 1995.

- [45] P. Flandrin, *Time–Frequency/Time–Scale Analysis*, vol. 10 of *Wavelet Analysis and its Applications*. San Diego: Academic Press (Elsevier), Oct. 1998.
- [46] M. A. Pinsky, *Introduction to Fourier Analysis and Wavelets*, vol. 102 of *Brooks/Cole Series in Advanced Mathematics*. Pacific Grove, CA: Brooks/Cole (Cengage Learning), 2002.
- [47] J. W. Cooley, P. A. Lewis, and P. D. Welch, “The finite fourier transform,” *IEEE Transactions on Audio and Electroacoustics*, vol. 17, no. 2, pp. 77–85, 1969.
- [48] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [49] C. Candan, M. Kutay, and H. Ozaktas, “The discrete fractional fourier transform,” *IEEE Transactions on Signal Processing*, vol. 48, no. 5, pp. 1329–1337, 2000.
- [50] J. B. Allen and L. R. Rabiner, “A unified approach to short-time fourier analysis and synthesis,” *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.
- [51] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Pearson Education, 3 ed., 2010.
- [52] D. L. Donoho and P. B. Stark, “Uncertainty principles and signal recovery,” *SIAM Journal on Applied Mathematics*, vol. 49, no. 3, pp. 906–931, 1989.
- [53] W. Heisenberg, “Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik,” *Zeitschrift für Physik*, vol. 43, no. 3–4, pp. 172–198, 1927.
- [54] H. M. Ozaktas and M. A. Kutay, “The fractional fourier transform,” in *2001 European Control Conference (ECC)*, pp. 1477–1483, 2001.
- [55] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

- [56] E. Alpaydm, *Introduction to Machine Learning*. Cambridge, MA: MIT Press, 4 ed., 2020.
- [57] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Adaptive Computation and Machine Learning, Boston, MA: MIT Press, 2012.
- [58] W. Wu, “Unsupervised learning.” Lecture notes (PDF), may 2022. Archived PDF (final version), retrieved Apr 26, 2024.
- [59] R. Sutton and A. Barto, “Reinforcement learning: An introduction,” *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998.
- [60] G. Garreau, C. M. Andreou, A. G. Andreou, J. Georgiou, S. Dura-Bernal, T. Wennekers, and S. Denham, “Gait-based person and gender recognition using micro-doppler signatures,” in *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 444–447, 2011.
- [61] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [62] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [63] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [64] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [65] S. Damelin and W. Miller, *The Mathematics of Signal Processing*. Cambridge, UK: Cambridge University Press, 2011.
- [66] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Cambridge; New York; Port Melbourne; New Delhi; Singapore: Cambridge University Press, 2024. Chapter 7.5: Pooling.

- [67] CS231n course authors, “Convolutional neural networks (cs231n).” <https://cs231n.github.io/convolutional-networks/>, 2025. Accessed July 2025.
- [68] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [69] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [70] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [71] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [72] P.-Y. Simard, Y. LeCun, J. S. Denker, and B. Victorri, “Transformation invariance in pattern recognition: Tangent distance and tangent propagation,” in *Neural Networks: Tricks of the Trade*, pp. 239–274, Springer, 1998.
- [73] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, “Vicinal risk minimization,” in *Advances in Neural Information Processing Systems*, vol. 13, pp. 416–422, MIT Press, 2000.
- [74] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [75] C. Rose and M. D. Smith, “mathstatica: Mathematical statistics with mathematica,” in *Compstat* (W. Härdle and B. Rönz, eds.), (Heidelberg), pp. 437–442, Physica-Verlag HD, 2002.
- [76] H.-T. Tran, E. Heading, and B. W.-H. Ng, “On the slow-time k-space and its augmentation in doppler radar tomography,” *Sensors*, vol. 20, no. 2, 2020.

- [77] N. Kern and C. Waldschmidt, “Data augmentation in time and doppler frequency domain for radar-based gesture recognition,” in *2021 18th European Radar Conference (EuRAD)*, pp. 33–36, 2022.
- [78] S. Wei, S. Zou, F. Liao, and W. Lang, “A comparison on data augmentation methods based on deep learning for audio classification,” *Journal of Physics: Conference Series*, vol. 1453, p. 012085, 01 2020.
- [79] V. Chen, F. Li, S.-S. Ho, and H. Wechsler, “Micro-doppler effect in radar: phenomenon, model, and simulation study,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, 2006.
- [80] B. D. Bullard and P. C. Dowdy, “Pulse doppler signature of a rotary-wing aircraft,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 6, no. 5, pp. 28–30, 2002.
- [81] G. Fliss and D. Mensa, “Instrumentation for rcs measurements of modulation spectra of aircraft blades,” in *National Radar Conference, Los Angeles, CA*, pp. 95–99, 1986.
- [82] M. R. Bell and R. A. Grubbs, “Jem modeling and measurement for radar target identification,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 1, pp. 73–87, 1993.
- [83] V. Chen, “Spatial and temporal independent component analysis of micro-doppler features,” in *IEEE International Radar Conference, 2005.*, pp. 348–353, 2005.
- [84] V. C. Chen, “Detection and analysis of human motion by radar,” in *2008 IEEE Radar Conference*, pp. 1–4, 2008.
- [85] J. Lei and C. Lu, “Target classification based on micro-doppler signatures,” in *IEEE International Radar Conference, 2005.*, pp. 179–183, 2005.
- [86] G. E. Smith, K. Woodbridge, and C. J. Baker, “Micro-doppler signature classification,” in *2006 CIE International Conference on Radar*, pp. 1–4, 2006.

- [87] D. Tahmoush and J. Silvius, “Radar micro-doppler for long range front-view gait recognition,” in *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, pp. 1–6, 2009.
- [88] Y. Li, Z. Peng, R. Pal, and C. Li, “Potential active shooter detection based on radar micro-doppler and range-doppler analysis using artificial neural network,” *IEEE Sensors Journal*, vol. 19, no. 3, pp. 1052–1063, 2019.
- [89] Y. Li, Z. Peng, and C. Li, “Potential active shooter detection using a portable radar sensor with micro-doppler and range-doppler analysis,” in *2017 International Applied Computational Electromagnetics Society Symposium (ACES)*, pp. 1–2, 2017.
- [90] M. Chakraborty, H. C. Kumawat, S. V. Dhavale, and A. A. B. Raj, “Diat-u radhar (micro-doppler signature dataset) & u radnet (a lightweight dcnn)—for human suspicious activity recognition,” *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6851–6858, 2022.
- [91] G. Klarenbeek, R. I. A. Harmanny, and L. Cifola, “Multi-target human gait classification using lstm recurrent neural networks applied to micro-doppler,” in *2017 European Radar Conference (EURAD)*, pp. 167–170, 2017.
- [92] M. Ergeneci, E. Bayram, D. Binningsley, D. Carter, and P. Kosmas, “Attention-enhanced frequency-split convolution block for semg motion classification: Experiments on premier league and ninapro datasets,” *IEEE Sensors Journal*, vol. 24, no. 4, pp. 4821–4830, 2024.
- [93] S. Gaglio, G. L. Re, and M. Morana, “Human activity recognition process using 3-d posture data,” *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 5, pp. 586–597, 2015.
- [94] A. Hanif, M. Muaz, A. Hasan, and M. Adeel, “Micro-doppler based target recognition with radars: A review,” *IEEE Sensors Journal*, vol. 22, no. 4, pp. 2948–2961, 2022.
- [95] D. Tahmoush, “Review of micro-doppler signatures,” *IET Radar, Sonar & Navigation*, vol. 9, no. 9, pp. 1140–1146, 2015.

- [96] V. C. Chen, “Advances in applications of radar micro-doppler signatures,” in *2014 IEEE Conference on Antenna Measurements & Applications (CAMA)*, pp. 1–4, 2014.
- [97] S. Yang, J. L. Kernec, O. Romain, F. Fioranelli, P. Cadart, J. Fix, C. Ren, G. Manfredi, T. Letertre, I. D. H. Sáenz, J. Zhang, H. Liang, X. Wang, G. Li, Z. Chen, K. Liu, X. Chen, J. Li, X. Wu, Y. Chen, and T. Jin, “The human activity radar challenge: Benchmarking based on the ‘radar signatures of human activities’ dataset from glasgow university,” *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 4, pp. 1813–1824, 2023.
- [98] F. Şahinuç and A. Koç, “Fractional fourier transform meets transformer encoder,” *IEEE Signal Processing Letters*, vol. 29, pp. 2258–2262, 2022.
- [99] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, “FNet: Mixing tokens with Fourier transforms,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, eds.), (Seattle, United States), pp. 4296–4313, Association for Computational Linguistics, July 2022.
- [100] E. Koç and A. Koç, “Fractional fourier transform in time series prediction,” *IEEE Signal Processing Letters*, vol. 29, pp. 2542–2546, 2022.
- [101] N. Nguyen, M. Pham, V.-S. Doan, and V. Le, “Improving human activity classification based on micro-doppler signatures of fmcw radar with the effect of noise,” *PLOS ONE*, vol. 19, no. 8, p. e0308045, 2024.
- [102] S. Vishwakarma, W. Li, C. Tang, K. Woodbridge, R. Adve, and K. Chetty, “Simhumalator: An open-source end-to-end radar simulator for human activity recognition,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 3, pp. 6–22, 2022.
- [103] Novelda AS, “X4 – datasheet, reve,” technical report, Novelda AS / Radar-Tutorial.eu, 2021. Impulse radar transceiver SoC (7.29/8.748GHz).

- [104] Texas Instruments, “Iwr1443 single-chip 76- to 81-ghz mmwave sensor datasheet (rev. c),” Technical Report SLAA781C, Texas Instruments, Aug. 2018. Published 15August2018; mmWave FMCW radar 76–81GHz, 3TX/4RX, 4GHz chirp.
- [105] E. Kaya and G. B. Kaplan, “Neural network based drone recognition techniques with non-coherent s-band radar,” in *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6, 2021.
- [106] S. Björklund, “Target detection and classification of small drones by boosting on radar micro-doppler,” in *2018 15th European Radar Conference (EuRAD)*, pp. 182–185, 2018.
- [107] J. C. Whitaker, *The Electronics Handbook*. CRC Press, 2005.
- [108] S. Butterworth, “On the theory of filter amplifiers,” *Experimental Wireless and the Wireless Engineer*, vol. 7, pp. 536–541, Oct. 1930.
- [109] Y. Dodge, ed., *The Oxford Dictionary of Statistical Terms*. Oxford: Oxford University Press, 2003. Entry: ”Normalization of scores”.
- [110] E. Kreyszig, *Advanced Engineering Mathematics*. New York: Wiley, 4 ed., 1979. See equation 5 on page 880.
- [111] L. Huang, *Normalization Techniques in Deep Learning*. Synthesis Lectures on Computer Vision, Cham: Springer International Publishing, 2022.
- [112] Z. Zhao, H. Wang, L. Cao, D. Wang, and C. Fu, “Doppler-spread target summation variability index cfar detector for fmcw radar,” *IEEE Sensors Journal*, vol. 24, no. 20, pp. 32519–32532, 2024.
- [113] S. M. D. Rizvi, S. Ahmad, K. Khan, A. Hasan, and A. Masood, “Deep learning approach for fixed and rotary-wing target detection and classification in radars,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 3, pp. 32–42, 2022.
- [114] Z. A. Cammenga, G. E. Smith, and C. J. Baker, “Combined high range resolution and micro-doppler analysis of human gait,” in *2015 IEEE Radar Conference (RadarCon)*, pp. 1038–1043, 2015.

- [115] L. Vignaud, A. Ghaleb, J. L. Kerneć, and J.-M. Nicolas, “Radar high resolution range & micro-doppler analysis of human motions,” in *2009 International Radar Conference "Surveillance for a Safer World" (RADAR 2009)*, pp. 1–6, 2009.
- [116] A. A. Mardiev and V. D. Kuptsov, “Convolutional neural networks for processing micro-doppler signatures and range-azimuth radar maps of frequency modulated continuous wave radars,” in *2022 International Conference on Electrical Engineering and Photonics (EExPolytech)*, pp. 75–78, 2022.
- [117] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 448–456, PMLR, 2015.
- [118] Keras Team, “Flatten layer — keras api documentation.” https://keras.io/api/layers/reshaping_layers/flatten/, 2025. Accessed: 2025-07-16.
- [119] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, “Regularization of neural networks using dropconnect,” in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, vol. 28, pp. 1058–1066, PMLR, 2013.
- [120] A. S. Householder, “A theory of steady-state activity in nerve-fiber networks: I. definitions and preliminary lemmas,” *The Bulletin of Mathematical Biophysics*, vol. 3, pp. 63–69, jun 1941.
- [121] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” *arXiv preprint arXiv:1710.05941*, oct 2017.
- [122] Z. Niu, G. Zhong, and H. Yu, “A review on the attention mechanism of deep learning,” *Neurocomputing*, vol. 452, pp. 48–62, sep 2021.
- [123] D. Soydaner, “Attention mechanism in neural networks: Where it comes and where it goes,” *Neural Computing and Applications*, vol. 34, pp. 13371–13385, aug 2022.

- [124] S. Özer, M. Ege, and M. A. Özkanoglu, “Siamesefuse: A computationally efficient and a not-so-deep network to fuse visible and infrared images,” *Pattern Recognition*, vol. 129, p. 108712, 2022.
- [125] J. Chen, Z. Yuan, J. Peng, L. Chen, H. Huang, J. Zhu, Y. Liu, and H. Li, “Dasnet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1194–1206, 2021.
- [126] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [127] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, oct 1986.
- [128] G. Hinton, “Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude.” https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2012. Accessed: 2020-03-19, Slide 26.
- [129] D. E. Hilt and D. W. Seegrist, “Ridge: a computer program for calculating ridge regression estimates,” research note ne-236, U.S. Department of Agriculture, Forest Service, Northeastern Forest Experiment Station, Upper Darby, PA, 1977.
- [130] PyTorch Contributors, “torch.optim.lr_scheduler.reduceLronplateau — pytorch documentation.” https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html, 2025. Accessed: 2025-07-16.
- [131] M. Ibrahim, “A deep dive into learning curves in machine learning.” <https://wandb.ai/mostafaibrahim17/ml-articles/reports/A-Deep-Dive-Into-Learning-Curves-in-Machine-Learning-Vmldzo0NjA1ODY0>, 2023. Accessed: 2025-07-16.

- [132] R. Parikh, A. Mathai, S. Parikh, G. Chandra Sekhar, and R. Thomas, "Understanding and using sensitivity, specificity and predictive values," *Indian Journal of Ophthalmology*, vol. 56, no. 1, pp. 45–50, 2008.
- [133] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," in *International Symposium on Intelligence Computation and Applications*, pp. 461–471, Springer, 2009.
- [134] O. Gunes, M. Ege, and Ö. Morgül, "Ground target classification using mmWave radar with bidirectional long short-term memory networks," *Journal of Applied Remote Sensing*, vol. 16, no. 4, p. 044527, 2022.
- [135] R. Tan, H. S. Lim, A. B. Smits, R. I. A. Harmanny, and L. Cifola, "Improved micro-doppler features extraction using smoothed-pseudo wigner-ville distribution," in *2016 IEEE Region 10 Conference (TENCON)*, pp. 730–733, 2016.
- [136] W. Li, B. Xiong, and G. Kuang, "Micro-doppler parameter estimation based on improved time-frequency analysis," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–4, 2017.
- [137] M. Ege and Ö. Morgül, "Siamesehar: Siamese-based model for human activity classification with fmcw radars," in *Intelligent Systems Design and Applications* (A. Abraham, S. Pllana, G. Casalino, K. Ma, and A. Bajaj, eds.), (Cham), pp. 291–302, Springer Nature Switzerland, 2023.
- [138] Y. Zhao, A. Yarovoy, and F. Fioranelli, "Angle-insensitive human motion and posture recognition based on 4d imaging radar and deep learning classifiers," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 12173–12182, 2022.
- [139] X. Yang, R. G. Guendel, A. Yarovoy, and F. Fioranelli, "Radar-based human activities classification with complex-valued neural networks," in *2022 IEEE Radar Conference (RadarConf22)*, pp. 1–6, 2022.
- [140] M. O. Padar, A. E. Ertan, and C. Candan, "Classification of human motion using radar micro-doppler signatures with hidden markov models," in *2016 IEEE Radar Conference (RadarConf)*, pp. 1–6, 2016.

- [141] Z. Shen, J. Nunez-Yanez, and N. Dahnoun, “Advanced millimeter-wave radar system for real-time multiple-human tracking and fall detection,” *Sensors*, vol. 24, no. 11, 2024.
- [142] A. Sengupta, F. Jin, R. Zhang, and S. Cao, “mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns,” *IEEE Sensors Journal*, vol. 20, p. 10032–10044, Sept. 2020.
- [143] E. Gruzewska and P. Rao, “Measuring heart rate with consumer ultra-wideband radar.” Google Research Blog, July 2025. Accessed: 2025-07-20.